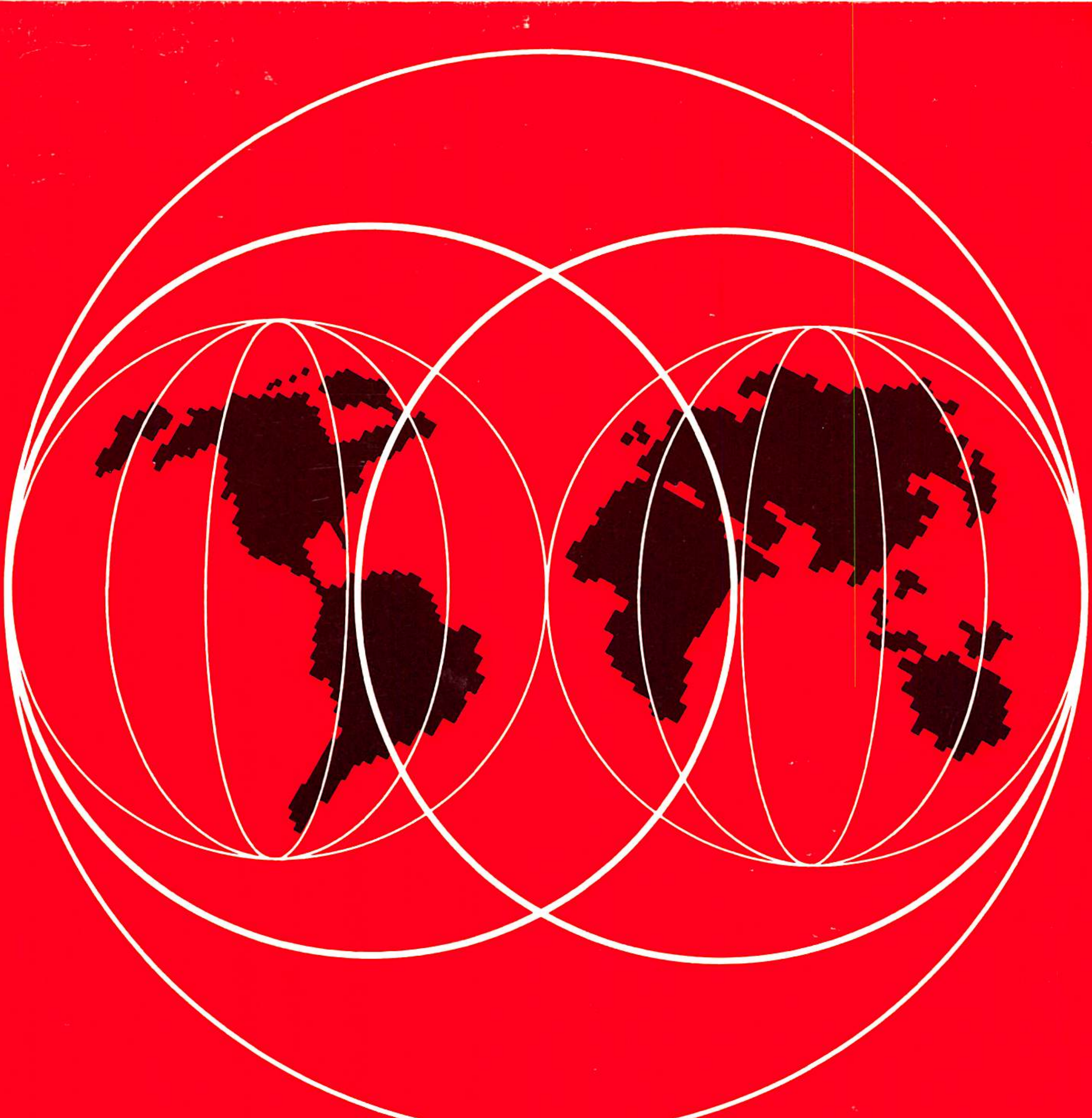


IBM

International Technical Support Centers

OS/2 VERSION 1.3
VOLUME 1: NEW FEATURES

GG24-3630-00

OS/2 Version 1.3
Volume 1: New Features

Document Number GG24-3630-00

February 1991

International Technical Support Center
Boca Raton

Take Note

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

First Edition (February 1991)

This edition applies to OS/2 Version 1.3.

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

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, International Technical Support Center
Dept. 91J, Building 235-2 Internal Zip 4423
P. O. Box 901 NW 51st Street
Boca Raton, Florida 33432 USA

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.

Abstract

This document describes the new features of OS/2 Version 1.3, including the Extended Edition.

The purpose of this bulletin is to explain and exploit the most important enhancements. It is not a programmer's guide nor can it replace the standard reference manuals of OS/2.

The contents of this bulletin can be used as course reference material for instructors and students of an OS/2 class.

Also, the picture segments of the original Presentation Manager screens might be useful for developing additional material.

PSYS (232 pages)

Acknowledgements

The project leader and editor was:

Hans J. Goetz
International Technical Support Center, IBM Boca Raton.

The authors of this document are:

Pascal Brossard
IBM France

Alfredo Gutiérrez
EMEA Education Support Center, IBM Boca Raton

Dave Pettigrew
IBM Canada

Jean Paul Simoen
IBM France

Jacques Soukin
ISM South Africa

This publication is the result of a residency conducted at the International Technical Support Center, Boca Raton, which covered the Standard Edition part.

Another residency was conducted in the International Technical Support Center, Austin, which covered the Extended Edition part.

Thanks to the following people for the invaluable advice, guidance and contributions provided in the production of this document:

David Kerr
Programming Center, IBM Boca Raton

Ian McCallion
Presentation Management Systems Planning, IBM UK Hursley

Martin McElroy
Workstations Technical Marketing, IBM UK Basingstoke

Michael Lukens
Programming Center, IBM USA, Boca Raton

Franck Minssieux
International Technical Support Center, IBM Austin

Peter Schönhofer
International Technical Support Center, IBM Boca Raton.

Contents

Figures

Tables

Special Notices

This publication is intended to help customers and systems engineers to understand and utilize the new features in version 1.3 of OS/2. The information in this publication is not intended as the specification of the programming interfaces that are provided by the Programming Tools and Information package for use by customers in writing programs to request or receive its services. See the PUBLICATIONS SECTION of the IBM Programming Announcement for OS/2 Version 1.3.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms, which are used in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
AIXWindows
Application System/400
AS/400
AT
Basic Compiler/2
Big Blue
C/2
COBOL/2
DB2
DisplayWrite
ETHERAND
FORTRAN/2
IBM
Macro Assembler/2
Micro Channel
MVS/ESA
MVS/SP
MVS/XA
Operating System/2
Operating System/400
OS/2
OS/400
PASCAL Compiler/2
PC/XT
Personal Computer AT
Personal Computer XT
Personal System/2
Presentation Manager
PROFS
Proprinter
PS/2
Quickwriter
Quietwriter
RISC System/6000
SAA
System/370
System/390
Systems Application Architecture
Virtual Machine/Extended Architecture
VM/XA
VTAM
Wheelwriter
XT
3090
400.

The following terms, which are also used in this publication, are trademarks of other companies.

Adobe is a trademark of Adobe Systems Inc.
Adobe Garamond is a trademark of Adobe Systems Inc.
Adobe Type Set is a trademark of Adobe Systems Inc.
Apple is a trademark of Apple Computer Inc.
Helvetica is a trademark of Linotype Company.
HP and Hewlett-Packard are trademarks of Hewlett Packard Corporation.
Intel, 386 and 486 are a trademark of Intel Corporation.
ITC Avant Garde Gothic is a trademark of International Typeface Corporation.
ITC Bookman is a trademark of International Typeface Corporation.
ITC Zapf Chancery is a trademark of International Typeface Corporation.
ITC Zapf Dingbats is a trademark of International Typeface Corporation.
LaserJet is a trademark of Hewlett-Packard Corporation.
Lotus is a trademark of the Lotus Development Corporation.
Macintosh is a trademark of Apple Computer Inc.
Microsoft is a trademark of Microsoft Corporation.
Palantino is a trademark of Linotype Company.
PostScript is a trademark of Adobe Systems, Inc.
Times New Roman is a registered trademark of Monotype Corporation Limited.
UNIX is a trademark of UNIX System Laboratories, Inc.

All other references to computer systems, software, and peripherals
use trademarks owned by their respective manufacturers.

Preface

This document describes the new features in OS/2 Version 1.3 Standard Edition. All these new functions and features in the Standard Edition apply also to the Extended Edition. If there are any differences then they are pointed out throughout this document.

Extended Edition explicit details are discussed in Chapter 14, "New Features in Extended Edition" on page 131 and do not apply to the Standard Edition. Though there are no changes in the Database Manager part since version 1.2 of OS/2, this chapter will focus mainly on the Communications Manager part and cover some of the LAN related topics.

A much more detailed discussion in regards to the print subsystem, including the LAN and Communications Manager, can be found in *OS/2 V1.3 Volume 2: Print Subsystem* reference number *GG24-3631*.

Audience

This document is intended for IBM system engineers, marketing representatives and customers.

It is assumed that they fulfill certain prerequisites:

- Have a thorough understanding of OS/2 V1.2
- Have some OS/2 application programming knowledge
- Have some understanding of the PC and PS/2.

Structure

This document is divided into the following major sections:

- Chapter 1, "Overview" on page 1, describes the new features of OS/2 SE V1.3

  After completing this chapter, the reader should know the new features of the Operating System/2 Standard Edition Version 1.3.

- Chapter 2, "Installation and Configuration" on page 5, describes the installation and the configuration of OS/2 SE V1.3.

  After completing this chapter, the reader should understand how to install OS/2 SE V1.3.

- Chapter 3, "Migration Considerations" on page 11, describes the pre-installation consideration.

  After completing this chapter, the reader should be aware of the system requirement and how to upgrade his current system with the new version of OS/2. The reader should also understand the requirements and implications of upgrading the operating system.

- Chapter 4, "Performance Considerations" on page 19, describes the performance of OS/2 SE V1.3.

After completing this chapter, the reader should understand how the performance enhancements and DASD and memory reductions have been achieved under OS/2 V1.3.

- Chapter 5, "Procedures Language 2/REXX" on page 25, describes this procedures language in general terms.

  After completing this chapter, the reader should understand how and to what extend REXX is implemented under OS/2 V1.3.

- Chapter 6, "Presentation Manager Shell" on page 35, describes some of the new features under the Presentation Manager shell.

- Chapter 7, "File Manager" on page 41, describes some of the specific enhancements of the File Manager.

- Chapter 8, "Print Manager" on page 45, describes some of the enhancements in the print subsystem.

  After completing this chapter, the reader should know how to use the new functions of Print Manager.

  A much more detailed discussion in regards to the print subsystem can be found in *OS/2 V1.3 Volume 2: Print Subsystem* reference number *GG24-3631*

- Chapter 9, "Fonts Usage in OS/2 V1.3" on page 61, provides a font overview.

  After completing this chapter, the reader should understand what kind of different fonts are available under OS/2 and where to find them.

- Chapter 10, "More Fonts for OS/2 V1.3 VIO Windows" on page 67, describes the new VIO fonts.

  After completing this chapter, the reader should understand how to use the new fonts for the VIO windows under PM.

- Chapter 11, "Adobe Type Manager (ATM)" on page 71, describes implementation of ATM.

  After completing this chapter, the reader should be able to understand how the Adobe Type Manager works and how PM applications can use these new fonts. A background lesson on outline fonts in general is also provided.

- Chapter 12, "Antialiased Fonts" on page 113, describes the new XGA fonts.

  After completing this chapter, the reader should understand how they can and cannot be used under PM.

- Chapter 13, "New and Revised APIs" on page 119, describes some of the new APIs.

  After completing this chapter, the reader should get a better idea how to use these new system functions for his/her own programs.

- Chapter 14, "New Features in Extended Edition" on page 131, focuses purely on the Extended Edition.

  After completing this chapter, the reader should understand which new features and functions can be found in OS/2 V1.3 EE. It is mainly focusing on Communications Manager issues and some LAN relevant items. There is no discussion about Database Manager since there are no changes to the previous version.

- Chapter 15, "Support for New Hardware" on page 157, describes how some of the new PS/2 hardware is supported by OS/2 V1.3.

- The appendix contains the following chapters:

  1. Appendix A, "National Language Support" on page 163, lists the national languages supported by OS/2 V1.3.
  2. Appendix B, "CONFIG.SYS Configuration File" on page 165, lists some sample CONFIG.SYS files.
  3. Appendix C, "XGA Installation under OS/2 V1.2" on page 171, describes the XGA installation under OS/2 V1.2.
  4. Appendix D, "Fonts Under OS/2 V1.3" on page 175, summarizes the fonts available under OS/2 V1.3.
  5. Appendix E, "Install Log Files" on page 177, lists two sample log files.
  6. Appendix F, "OS/2 V1.3 Extended Edition Memory and Fixed Disk Requirements" on page 183, provides some useful tables in regards to the new features of Extended Edition.
  7. Appendix G, "GATEWAY.DAT File" on page 187, shows another sample listing from an Extended Edition system.

Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document.

Prerequisite Publications

- *OS/2 V1.3 Standard Edition Getting Started* reference number *84F7700*
- *OS/2 V1.3 Standard Edition Using Advanced Features* reference number *84F7701*
- *Using Type 1 Fonts*, order number *85F1573.*
- *OS/2 Programming Tools and Information V1.3* reference number *85F1671*.

Additional Publications

- *OS/2 V1.3 Volume 2: Print Subsystem* reference number *GG24-3631*
- *OS/2 V1.2 Standard Edition Internals and Evaluation* reference number *GG24-3466*
- *IBM OS/2 V1.2 Dialog Manager* reference number *GG24-3467*
- *IBM CUA 1989 Evaluation* reference number *GG24-3456*
- *IBM OS/2 SE Kernel and Presentation Manager* reference number *GG24-3370*
- *OS/2 Information and Planning Guide* reference number *G360-2650*
- *Printing PostScript Language* reference number *GG24-3529*
- *PS/2 Models 90 XP 486, 95 XP 486, 55 SX LX and P75 486 Fundamentals* reference number *GG24-3616*
- Ross Smith, *Learning PostScript, A Visual Approach*, Peachpit Press, *ISBN 0-938-151-12-6*
- Holzgang David, *Understanding PostScript Programming*, SYBEX, *ISBN 0-89588-566-2*
- Adobe Systems Inc., *Adobe Type 1 Font Format*, Addison-Wesley, *ISBN 0-201-57044-0*
- IBM, *Personal Systems Developer*, reference number *G362-0001*
- IBM, *Personal Systems*, reference number *G325-5009.*

Chapter 1. Overview

IBM[1] Operating System/2[2] , Standard Edition,Version 1.3 called OS/2 SE V1.3 includes many improvements that make OS/2 a durable and high-function platform for the workstation applications of today and tomorrow.

Operating System/2 is the advanced operating system for IBM Personal System/2[3] , (PS/2) Models 30 286, 50, 50 Z, 55 SX, 60, 65, 70 386, P70 386, P75 486, 80 386, 90 XP 486 and 95 XP 486; IBM Personal Computer XT[4] Model 286; and the IBM Personal Computer AT[5] ..

In other words, it is designed to run on IBM PCs and PS/2s with an INTEL[6] , 80286, 80386[7] or 80486[8] Processor.

OS/2 SE Version 1.3 will run on a 2 MB machine with a 20 MB hard disk, thus expanding the range of "OS/2 capable" machines, and maintaining all the functions delivered with OS/2 V1.2.

OS/2 SE 1.3 also features:

- Significant improvements in performance and memory management.
- New font capabilities with Adobe font-rendering technology.
- Selective install for savings of disk space.
- Printing improvements, including a new spooler and device drivers. These facilities are detailed in *OS/2 V1.3 Volume 2: Print Subsystem.* .
- Extra facilities for improved usability.
- New program properties that enable simplified application of auto start at boot time.
- Procedures Language 2/REXX.
- Support for more SCSI hard disks.
- Several enhancements for the Extended Edition parts.

[1] IBM is a registered trademark of International Business Machine Corporation.

[2] Operating System/2 and OS/2 are trademarks of International Business Machines Corporation.

[3] Personal System/2 and PS/2 are registered trademark of International Business Machines Corporation.

[4] Personal Computer XT and PC XT are registered trademark of International Business Machines Corporation.

[5] Personal Computer AT and AT are registered trademark of International Business Machines Corporation.

[6] INTEL is a registered trademark of INTEL Corporation.

[7] 386 is a registered trademark of INTEL Corporation.

[8] 486 is a registered trademark of INTEL Corporation.

1.1 New Features

The refresh installation configuration panel provides two choices:

- **Reinstalling OS/2**

  With this option, the OS/2 installation does not format any hard disk partition. It only adds operating system files to the files that already exist on the hard disk. All current files remain intact.

  > **Note**
  >
  > There is no option to format the hard disk.
  >
  > The option to format the hard disk is available only when you install OS/2 V1.3 for the first time.

- **Adding selectable OS options**

  - Country information
  - Online documentation
  - Fonts
  - High Performance File System
  - System utilities
  - OS/2 DOS Compatibility session
  - Picture utilities
  - Serial device support
  - Services and diagnostic aids.

1.1.1 Printer Installer

Printer Installer is a program that can be used to install the printer device drivers and define a new printer and queue. It is now possible to set up a printer and queue that use device drivers containing single or multiple files, all in only one step.

See Chapter 8, "Print Manager" on page 45, for more details.

1.1.2 DOS Timeout

The print manager also allows a printer timeout to be set for all DOS Compatibility Session applications. This will allow printing to start from DOS quicker than before, and start printing without the need to exit a DOS application. See Chapter 8, "Print Manager" on page 45 and section 8.4, "Printer Install" on page 52 for more details.

1.2 Enhanced Print Subsystem

The new Print Manager requires less memory than the previous version, and provides faster spooling.

The new Print Manager also simplifies printer device driver installation including associating printers with specific queues. This puts printer install and setup in one place now. The Control Panel may still be used to install new printer device drivers and to install queue drivers. However, it is recommended to use the Print Manager for installing any printer device drivers from now on.

Note

To add or delete fonts, delete printer device drivers or printer queue processors, you still have to use the Control Panel. These functions are not available in Print Manager.

There are a number of updated printer device drivers with increased functionality:

- IBM Proprinter[9] (includes support for resident h/w fonts)
- IBM Quietwriter[10]
- IBM 4019 LaserPrinter (includes PostScript[11] support)
- HP LaserJet[12] (support selectable and downloadable fonts)
- Epson[13] (all nine pin and 24 pin models supported)
- PostScript[14] (40 different named PostScript devices) with downloadable fonts
- IBM 4216-020 PostScript printer (requires an extra device monitor).

1.3 New Print Queue Processor

The PMPRINT queue processor is enhanced and allows for smoother print job handling and modification. The Plot Q Processor (PMPLOT) is especially designed for improved plotter support. It improves the appearance of pictures spooled to a plotter by reverse (or vector) clipping the data. PMPLOT removes hidden lines and patterns from solid objects that overlap in the picture, so that underlaying objects are clipped by the overlaying ones. The output page created in this way is closer to the screen image.

For a much more detailed discussion of the entire OS/2 print subsystem including the LAN and Communications Manager, see the ITSC bulletin *OS/2 V1.3 Volume 2: Print Subsystem.*

1.4 Enhanced Features

The following features, which were available in OS/2 Version 1.2, have been improved:

- The OS/2 loader considerably improves performance in a LAN and local environment.

- Reductions in working sets and improved memory management require less system memory to run the same applications as IBM Operating System Version 1.2, with equal to better levels of performance. See Chapter 4, "Performance Considerations" on page 19 for more information.

[9] Proprinter is a trademark of International Business Machine Corporation.

[10] Quietwriter is a trademark of International Business Machine Corporation.

[11] PostScript is registered trademarks of Adobe Systems, Inc.

[12] LaserJet is a registered trademark of Hewlett-Packard Corporation.

[13] Epson is registered trademarks of Epson Corporation.

[14] PostScript is registered trademarks of Adobe Systems, Inc.

- SWAPDOS is now the default in the CONFIG.SYS file and allows the system to use real memory (640 KB) for both, DOS and OS/2 sessions.

Chapter 2. Installation and Configuration

During installation, the user is prompted to select either the default predefined system configuration (the complete system), or to define a specific system configuration. If users are short of disk space, they might want to consider using the **Select System Configuration** choice in the installation program. **Select System Configuration** allows users to tailor the system to meet their specific needs, thus further reducing the amount of memory required.

2.1 Reinstall and Separate Install

After OS/2 is installed, it is possible to reinstall or to add selectable options. The Refresh Installation Configuration panel provides two choices:

- **Reinstalling OS/2**

  With this option, the OS/2 installation does not format any hard disk partition. It only adds operating system files to the files that already exist on the hard disk. All current files remain intact.

  > **Note**
  >
  > There is no option to format the hard disk.
  >
  > The option to format the hard disk is available only when you install OS/2 V1.3 for the first time.

- **Adding selectable OS options**
  - Country information
  - Online documentation
  - Fonts
  - High Performance File System
  - System utilities
  - OS/2 DOS Compatibility session
  - Picture utilities
  - Serial device support
  - Services and diagnostic aids.

2.1.1 Add Selectable Options

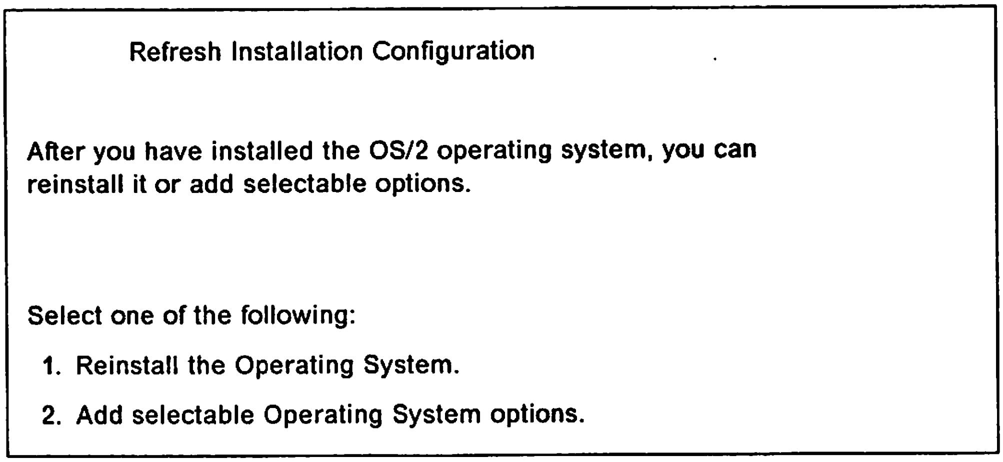
You might want to use this option to limit the amount of hard disk space required to install the operating system. Or, you might use this option to install options that were not previously installed.

How to add selectable options:

1. Insert the installation diskette into drive A:.
2. Turn on the computer.

   If the computer is already on, press and hold the Control (CTRL) and Alternate (ALT) keys, and then press the Delete (DEL) key.
3. The system will restart and load OS/2 from the installation diskette. After a while you will see the system logo displayed on the screen.
4. Follow the instructions on the screen.

5. After inserting diskette #1 of OS/2, you will see the following screen:



Refresh Installation Configuration

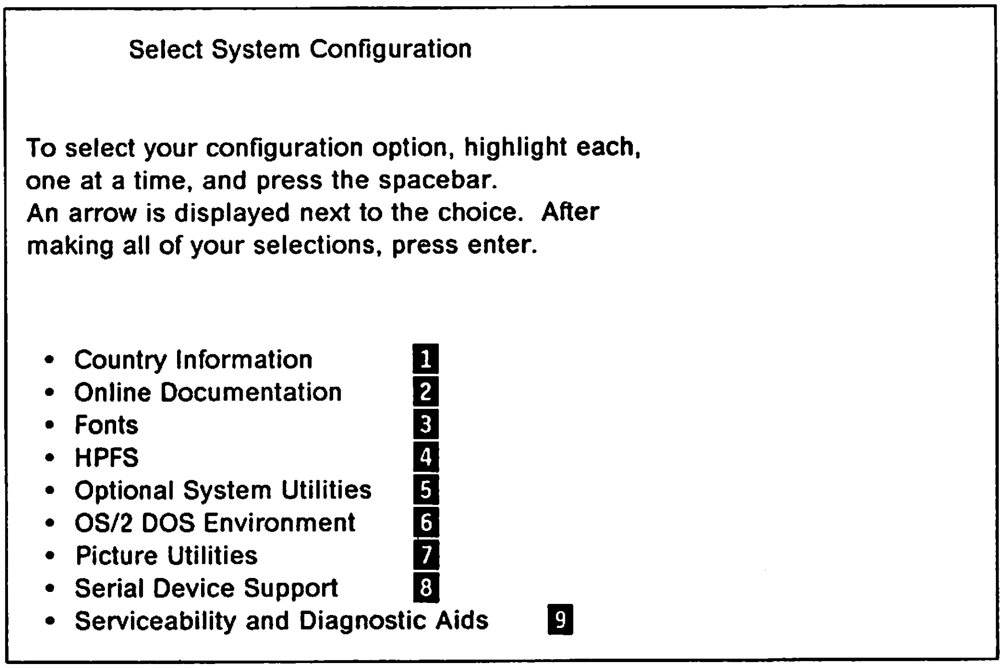
After you have installed the OS/2 operating system, you can reinstall it or add selectable options.

Select one of the following:

1. Reinstall the Operating System.
2. Add selectable Operating System options.

Figure 1. OS/2 V1.3 SE Installation Panel

6. Select option 2, and the following screen will be displayed:



Select System Configuration

To select your configuration option, highlight each, one at a time, and press the spacebar.
An arrow is displayed next to the choice. After making all of your selections, press enter.

- Country Information 1
- Online Documentation 2
- Fonts 3
- HPFS 4
- Optional System Utilities 5
- OS/2 DOS Environment 6
- Picture Utilities 7
- Serial Device Support 8
- Serviceability and Diagnostic Aids 9

Figure 2. OS/2 V1.3 SE Selective Install

1 **Country Information.** Determines the code pages, sort orders and symbols the computer uses to print or display information.

2 **Online Documentation.** Gives you the opportunity to decide later in the installation process if you want to install the *Introducing OS/2* tutorial and the *OS/2 Command Reference* online reference manual.

3 **Fonts.** Determines which Presentation Manager[15] fonts will be installed on your system. When no fonts are selected, the system default font and the mono

[15] Presentation Manager is a registered trademark of International Business Machine Corporation.

space font are installed on your system. For more details, see section Chapter 9, "Fonts Usage in OS/2 V1.3" on page 61, and section 11.4, "Adobe Type Manager: Installation" on page 86 in this volume.

4 **High Performance File System (HPFS).** Provides extremely fast access to very large disk volumes. Features of HPFS include:

- File names up to 254 characters in length
- Large file support
- Strategic allocation of directory structures
- Extended attribute support
- Caching of directories, data, and file system structures
- Processing of metacharacters that are generally used for displaying and printing graphics.

5 **Optional System Utilities.** These additional utilities provide a full set of system utilities such as BACKUP, FDISKPM, LABEL, LINK, RECOVER, RESTORE, SORT, and TREE.

6 **OS/2 DOS Environment.** Allows DOS applications to run on OS/2 (DOS Compatibility Session).

7 **Picture Utilities.** Allows the user to convert, display, and print OS/2 PM metafiles and IBM Picture Interchange Format (PIF) files.

8 **Serial Device Support.** System support for attached serial devices such as a modem, a serial plotter, or a serial printer assigned to a communication port. This option is not required for mouse support.

9 **Serviceability and Diagnostic Aids.** Provides information for problem determination, primarily for your service coordinator, to isolate and correct system problems.

For further information on installing options after initial installation, see the section on "Customized Installation" in the *Using Advanced Features* manual.

2.2 New Directory Structure

Information on a computer's hard disk is organized into files and directories. A **file** is a piece of information, such as a document. It can contain text, graphics, or even a complete program. Program file names usually have extensions of .EXE or .COM. The extension .CMD is used for OS/2 batch or REXX procedures.

A **directory** is a group of related files, like a folder containing related documents. A directory that is part of another directory is called a **subdirectory**.

File and directory names must conform to either the High Performance File System (HPFS) or File Allocation Table (FAT) naming conventions. See the *OS/2 Standard Edition Version 1.3 Getting Started* manual for more details. See Figure 3 on page 8 for a sample directory structure of an OS/2 V1.3 Standard Edition system after a normal installation.

```
C:\ 1
 │
 ├── OS2 2
 │    │
 │    ├── DDFONTS  3 *
 │    ├── BOOK  4
 │    ├── DLL  5
 │    │    │
 │    │    ├── IBM5152  6 *
 │    │    ├── PSCRIPT  6 *
 │    │    └── HP  7 *
 │    │         │
 │    │         └── PLC  8 *
 │    │              │
 │    │              └── LASERJET  9 *
 │    ├── HELP  10
 │    ├── INSTALL  11
 │    ├── INTRO  12
 │    └── SYSTEM  13
 ├── PSFONTS  14 *
 ├── SPOOL  15
 │    │
 │    ├── 515201    16 *
 │    ├── IBM40192  16 *
 │    └── PSCRIPT3  16 *
 └── 4019FONT  17 *

 * New or Different in OS/2 Standard Edition Version 1.3.
```

Figure 3. OS/2 V1.3 SE Directory Structure

1 The root directory contains AUTOEXEC.BAT, CONFIG.SYS, STARTUP.CMD, COMMAND.COM, and ABIOS.SYS on a PS/2. Also the hidden files IBMBIO, IBMDOS, OS2KRNL, OS2LDR, EA_DATA._SF on a FAT file system and OS2BOOT on a HPFS boot partition.

2 This subdirectory contains all the OS/2 external commands, for example, CHKDSK, BACKUP, E, etc. Also the OS2.INI, and OS2SYS.INI PM and system profiles reside here.

3 This subdirectory contains all XGA Antialiased (system) fonts. See Chapter 12, "Antialiased Fonts" on page 113, section Chapter 9, "Fonts Usage in OS/2 V1.3" on page 61, and section 11.4, "Adobe Type Manager: Installation" on page 86 in this volume.

4 This subdirectory contains all online reference manuals. They have the filename extension of .INF and can be browsed with the VIEW.EXE utility, coming with the Information Presentation Facility (IPF). When seeking help, IPF will search in all directories specified by the "SET BOOKSHELF" statement in the CONFIG.SYS file.

5 This subdirectory contains all dynamic link libraries .DLL, the fonts files .FNT, all the installed PM fonts, and the printer device driver files .DRV. The .DLL files can be relocated into a different subdirectory, depending on the LIBPATH statement in the CONFIG.SYS file.

6 These subdirectories are created when you install the printer device drivers. In this case, IBM5152, PSCRIPT, and HP[16]

7 This subdirectory contains the .DLL files for the HP LaserJet printer. The subdirectory is created during the printer device driver installation.

8 This subdirectory contains the printer fonts for the HP LaserJet. The subdirectory is created during the printer device driver installation.

9 This subdirectory contains the actual printer device driver for the HP LaserJet. The subdirectory is created during the printer device driver installation.

10 This subdirectory contains all online help information. They have the filename extension of .HLP and are used by the Information Presentation Facility (IPF), whenever help is requested. When requesting help (F1 Key), IPF will search in all directories, specified by the "SET HELP" statement in the CONFIG.SYS file.

11 This subdirectory contains the installation dependent files. System level information, installation and maintenance log files.

12 This subdirectory contains the online tutorial. It can be helpful for the OS/2 beginner to learn how to use the operating system. It is not required for the normal operation of OS/2 and therefore can save the user some DASD space if not needed.

13 This subdirectory contains system dependent files. By default the swapper file will be placed here. This can be modified through the SWAPPATH statement in the CONFIG.SYS file. The standard message files are stored in this area. As are all files used by the Dual Boot facility. When switching the operating system from DOS to OS/2, or OS/2 to DOS, the following files will be used and copied:

Table 1. Switching from OS/2 to DOS

	Before issuing the BOOT command	After issuing the BOOT command
OS/2 system files	C:\CONFIG.SYS C:\AUTOEXEC.BAT	C:\OS2\SYSTEM\CONFIG.OS2 C:\OS2\SYSTEM\AUTOEXEC.OS2
DOS system files	C:\OS2\SYSTEM\CONFIG.DOS C:\OS2\SYSTEM\AUTOEXEC.DOS	C:\CONFIG.SYS C:\AUTOEXEC.BAT
Boot system info.	C:\OS2\SYSTEM\BOOT.DOS	C:\OS2\SYSTEM\BOOT.OS2

[16] HP and Hewlett-Packard are registered trademarks of Hewlett-Packard Corporation.

Table 2. Switching from DOS to OS/2

	Before issuing the BOOT command	**After issuing the BOOT command**
DOS system files	C:\CONFIG.SYS C:\AUTOEXEC.BAT	C:\OS2\SYSTEM\CONFIG.DOS C:\OS2\SYSTEM\AUTOEXEC.DOS
OS/2 system files	C:\OS2\SYSTEM\CONFIG.OS2 C:\OS2\SYSTEM\AUTOEXEC.OS2	C:\CONFIG.SYS C:\AUTOEXEC.BAT
Boot system info.	C:\OS2\SYSTEM\BOOT.OS2	C:\OS2\SYSTEM\BOOT.DOS

14 This subdirectory contains any Type 1 fonts other than the IBM core fonts. They can be used by the Adobe Type Manager. If Necessary, they are converted into PM display fonts or downloaded to any PostScript printer. They are installed through the Control Panel. You can also choose any other name for this subdirectory, if you wish.

15 The SPOOL subdirectory is created during OS/2 installation. It has to hold all spooled printer data until it gets printed. By default, the SPOOL directory will be placed here. This can be modified through the spooler setup option in Print Manager.

16 The subdirectories are created during the OS/2 installation or when you define a printer queue with Print Manager. These subdirectories contain the jobs and their data to be printed on this particular queue (printer).

17 This subdirectory contains any additional IBM 4019 printer fonts. These fonts will be downloaded to the printer by the Print Manager whenever necessary.

Chapter 3. Migration Considerations

During installation, the system prompts the user to decide whether to accept the predefined system configuration (the complete system), or to define a custom system configuration. This is a new feature of OS/2 Version 1.3.

WARNING: If a previous version of OS/2 is installed and the system configuration is selected, only the features selected are installed. Existing system files are either copied (overwritten) or erased, based on the selections.

Many programs, designed to run under DOS, may be started from the DOS Compatibility Session, which is provided under OS/2. There are, however, limitations. The DOS operating system can be installed on the same machine as OS/2. This feature is called **Dual Boot**. It allows you to start either DOS or OS/2 from the same boot partition on the hard disk (drive C). Dual Boot provides an alternative method for using those applications which do not run in the DOS compatibility session.

Note

Dual Boot cannot be used if the High Performance File System is installed on drive C.

Also, DOS will not be able to access any HPFS partitions for reading or writing any data.

Dual Boot requires DOS Version 3.20 (or later). However, it is recommended that DOS 4.01 with the latest CSD for greater compatibility with OS/2 V1.3 be used.

Starting with CSD UR31300, applied to DOS 4.01 (US), DOS can recognize a HPFS partition in a limited way. DOS' FDISK command will recognize HPFS partitions and display their parameters.

Watch Out

Because the HPFS partitions will be recognized, the DOS FORMAT command will format them under DOS, therefore destroying any (OS/2) data on them!

Earlier DOS versions have not even been able to recognize any HPFS partition. This "misbehavior", in turn, provided at least some limited security for such OS/2 partitions.

On the other hand, the drive letters for all partitions stay the same now under DOS as under OS/2. Because all partitions are recognized now, HPFS partitions cannot be accessed any further than that.

By the way, this DOS CSD is also required if more than two SCSI hard disks are installed in a PS/2.

It is recommended that you install DOS on the same system with OS/2 if there are programs that :

- Are compiled with the IBM BASIC Compiler 1.0
- Use direct hardware control
- Use the following keys for program functions:
  - Alt, Ctrl, Ctrl and Esc, Ctrl and Alt, Alt and Esc, or PrtSc
- Use block device drivers, such as random access memory (RAM) disk and tape systems
- Use extended or expanded memory
- Determine storage sizes by checking hardware RAM
- Use certain features of special DOS communications programs.

Also, existing print spoolers do not function in the DOS Compatibility Session. Instead, you have to set up print spooling under OS/2's Print Manager, or use the OS/2 SPOOL command. If the DOS program can print without interaction, make sure that all its printing has stopped before ending a DOS program. See section 8.9, "DOS Timeout" on page 59, for more details.

3.1 Pre-Installation Tasks

Before installing OS/2 V1.3, the user should prepare the system environment to meet the requirements. Several items that the user needs to know before proceeding with the installation:

- Fixed disk requirements for OS/2 installation
- Backing up programs and data
- Using the Dual Boot feature or a DOS Startup Diskette
- Using the High Performance File System (HPFS) or the File Allocation Table (FAT) file system or both.

3.2 Fixed Disk Requirements

OS/2 V1.3 SE requires between 8 MB to 11 MB of hard disk space, depending on the option selected during the installation. If you do not have enough space, you can increase the hard disk space in many ways.

- Relocate the existing files from drive C to another drive.
- Back up the files from drive C to any other logical drives that may be available on the hard disk. Then increase the partition size to a size larger than the previous partition by using the FDISK command.

Table 3. OS/2 SE V1.3 Memory Estimation

Function	Memory	Comments
Base Operating System (required)	1.5 MB	
DOS Compatibility Session	0.5 MB	Automatically installed but removable by changing the PROTECTONLY=NO in CONFIG.SYS to PROTECTONLY=YES. This memory is also available to OS/2 whenever the DOS session is in the background, if SWAPDOS is specified in CONFIG.SYS.
High Performance File System (HPFS)	0.3 MB	This does not include the HPFS cache, which can be specified during the HPFS CACHE initialization in the CONFIG.SYS.
Active Spooling (while printing)	0.3 MB	This Memory is given back to OS/2 while the spooler is idle.
System Performance Buffer	0.5 MB	This additional system memory allows better performance for transient conditions such as program loading, program termination and print spooling.

Note: This is only a rough estimate of memory, recommended for these tasks.

Table 4. OS/2 V1.3 Standard Edition Hard Disk Space

Function	Disk Space	Comments
Base Operating System (required)	7.4 MB	Standard Edition only.
Base Operating System (optional)	3.5 MB	• Documentation = 0.5 MB • Type 1 fonts = 0.4 MB * • PM fonts = 0.5 MB • HPFS = 0.3 MB • Country support = 0.1 MB • System utilities = 0.5 MB • DOS utilities = 0.2 MB • Picture utilities = 0.3 MB • Diagnostic utilities = 0.2 MB
Swap Data Set	2.5 MB	A 2MB to 4MB swap area is adequate for many environments. However, the swap area is dynamically allocated as applications require more memory than is available. A swap data set larger or smaller than shown may be needed. Can be switched off through the MEMMAN statement in CONFIG.SYS.

Note: This is only a rough estimate of hard disk space, requirement for OS/2.

* See also Chapter 9, "Fonts Usage in OS/2 V1.3" on page 61, and Chapter 11, "Adobe Type Manager (ATM)" on page 71, for more details.

3.3 LOG File

After installing OS/2, it is possible to browse the INSTALL.LOG file, which contains all messages from the OS/2 installation process. A sample of an INSTALL.LOG file is shown in Appendix E, "Install Log Files" on page 177.

Every time OS/2 is reinstalled, the INSTALL.LOG file is renamed to INSTALL.BAK or INSTALL.00X (where X represents the sequence number).

3.4 Upgrading from DOS to OS/2 V1.3

OS/2 Version 1.3 supports partition sizes that are larger than 32 MB. During installation, the system will detect if there is enough disk space to install the operating system. The size of the partition can be increased before installing OS/2. It is also recommended to move all DOS relevant modules to another subdirectory or partition.

3.4.1 Without Dual Boot

If the DOS operating system is already installed on the system, make sure that the DOS commands and utilities are copied to a DOS subdirectory, in case they are still needed. DOS applications and their data files are not replaced or deleted if they are kept in separate subdirectories.

3.4.2 With Dual Boot

Before installing Dual Boot on an OS/2 V1.3 system, refer to the following guidelines:

1. A bootable DOS system must be installed before installing OS/2 V1.3
2. The DOS CONFIG.SYS and AUTOEXEC.BAT files must be created, or modified before OS/2 V1.3 is installed. During OS/2 installation, OS/2 CONFIG.SYS and AUTOEXEC.BAT files are created. OS/2 places any existing DOS CONFIG.SYS and AUTOEXEC.BAT files in the C:\OS2\SYSTEM subdirectory with a .DOS extension. This is to make sure that OS/2 uses the correct version of CONFIG.SYS and AUTOEXEC.BAT during any OS/2 boot process.
3. The system installation program installs the Dual Boot feature on drive C if the user has a bootable DOS operating system installed.
4. Only the File Allocation Table (FAT) file system is supported by the Dual Boot feature on the boot partition. Native DOS will not recognize files created with the High Performance File System (HPFS) because the formats are totally different.
5. The BOOT command is used for switching from OS/2 to DOS or vice versa. This command is actually a family API program and therefore can be used in OS/2 mode, DOS compatibility mode, and native DOS mode.
6. All DOS operating system files must be located in a separate subdirectory, for example C:\DOS.
7. Files that have the same names as OS/2 files, such as COMMAND.COM and COUNTRY.SYS should be copied to that DOS subdirectory.
8. DOS 4.0 or above must be used when any of the OS/2 disk partitions are larger than 32 MB.

9. DOS and OS/2 do not operate at the same time.
10. It is recommended that, prior to issuing the BOOT command, all data is saved by the user. Otherwise, data could be lost. A warning message is issued by the system as a reminder.

Dual Boot can be used any time after OS/2 is installed.

Warning

Do not select the FORMAT option while installing OS/2 V1.3

3.5 Upgrading from OS/2 V1.0 to OS/2 V1.3

OS/2 Standard Edition V1.0 allows multiple applications to run at the same time, but only one is displayed at a time. With the Presentation Manager, multiple applications can now appear on the screen simultaneously.

The applications defined in the Program Selector of OS/2 V1.0 will not be included in the Desktop Manager or Main Group of OS/2 V1.3. This must be done manually after installing OS/2. Therefore, the whole process is more like a new install action rather than an upgrade.

After OS/2 V1.3 is installed, you have to update the Desktop Manager by creating a new Group for the applications defined in the "old" Program Selector of V1.0 or by installing new Programs in the Main Group.

With OS/2 Version 1.3, you have new features, for example:

- Presentation Manager
- OS/2 System Editor
- Dual Boot facility
- OS/2 Dialog Manager
- REXX
- High Performance File System (HPFS)
- Print Manager
- File Manager
- Information Presentation Facility with online documentation and help
- etc.

With OS/2 V1.3, you have the option to install the Dual Boot facility, which allows you to start your system with OS/2 or DOS; therefore, there are two installation procedures:

- Without Dual Boot
- With Dual Boot.

Note

If you decide to install OS/2 V1.3 without Dual Boot, you cannot install Dual Boot afterwards without reformatting the boot partition.

3.5.1 Without Dual Boot

When upgrading from OS/2 V1.0 to V1.3, the previous version of OS/2 is replaced. All other data stays unchanged.

3.5.2 With Dual Boot

You must install a bootable version of DOS before installing OS/2 V1.3. DOS should be installed into a subdirectory called C:\DOS. The CONFIG.SYS and AUTOEXEC.BAT will be installed on the root directory, as usual. This also includes the hidden DOS files IBMBIO.COM and IBMDOS.COM.

Then follow the normal OS/2 V1.3 installation procedure.

3.6 Upgrading from OS/2 V1.1

The groups defined in the Start Program will be included in the Desktop Manager. The applications defined in the group will be included in each group as previously defined. The last step of OS/2 installation will use the MAKEINI.EXE file to rebuild the OS2.INI file. The OS2SYS.INI, which contains your hardware definition, is then created.

The printer device drivers already installed will be updated with the new printer device drivers of OS/2 V1.3.

All setups defined in OS/2 V1.1 from the control panel such as color screen, printer definition, and queue definition will remain in OS/2 V1.3.

The STARTUP.CMD file is unchanged and can be used in OS/2 V1.3. With OS/2 Version 1.3, you have new features such as:

- Dual Boot facility
- High Performance File System(HPFS)
- Some OS/2 utilities (for example: Format Diskette, Copy Diskette Compare Diskette, Fixed Disk Utility)
- Print Manager (enhancement)
- File Manager (enhancement)
- XGA device driver (for PS/2 Model 90 and Model 95).

With OS/2 V1.3, you have the opportunity to install Dual Boot facility, which allows you to start your system with OS/2 or DOS; therefore, there are two installation procedures:

3.6.1 Without Dual Boot

This is the same procedure as explained in section 3.5.1, "Without Dual Boot."

3.6.2 With Dual Boot

This is the same procedure as explained in section 3.5.2, "With Dual Boot."

For an example see also Appendix E, "Install Log Files" on page 177. The LOG file reflects exactly which files are affected while upgrading from OS/2 V1.1 to V1.3.

3.7 Upgrading from OS/2 V1.2

Compared to OS/2 V1.2, Version 1.3 provides many new and enhanced features:

- New program properties
- Procedure Language 2/REXX
- Selective install
- Type 1 fonts
- XGA Antialiased fonts (also available for V1.2)
- Improved Print Manager
- Improved spooler
- Enhanced printer device drivers
- etc.

However, the V1.3 installation process takes care of the most important system dependant modules.

- OS2.INI and OS2SYS.INI stay intact
- Extended attributes stay intact
- HPFS partitions stay intact
- STARTUP.CMD stays intact
- OS2INIT.CMD stays intact
- Existing printer drivers will be upgraded.

Warning

OS/2's CONFIG.SYS and AUTOEXEC.BAT will be replaced! All previous definitions will be gone. Therefore, it is recommended that a backup version of these two files be kept and any modifications needed be made to the new ones after the installation is complete.

3.7.1 Without Dual Boot

This is the same procedure as explained in section 3.5.1, "Without Dual Boot" on page 16.

3.7.2 With Dual Boot

If a V1.2 Dual Boot system is found by the V1.3 installation procedure, the new version will be installed with Dual Boot also.

If Dual Boot wasn't installed before, it is necessary to install a bootable DOS version first.

This will be the same procedure as explained in section 3.5.2, "With Dual Boot" on page 16.

Dual Boot can be used after OS/2 is installed.

Warning

Do not select the FORMAT option while installing OS/2 V1.3.

Chapter 4. Performance Considerations

One of the design goals for Version 1.3 of OS/2 was to provide better performance, especially in memory constraint situations. This tuning and redesign took place in certain areas within the operating system and is discussed in the following sections.

4.1 Memory Management and Swap Control

Swapping uses the hard disk as an external storage device in order to provide the requested amount of virtual memory, this has a dramatically negative impact on the overall system performance. Therefore, it seems logical to avoid any swap activity wherever possible. Again, there are several areas where a carefully designed and tuned operating system can avoid or at least minimize any swapping.

4.1.1 "Moving Window" Segment Compaction

During memory segment compaction, the system looks through the memory segments for space not being used by applications, and attempts to move around those parts of the segment not being used into a more compact block of memory, thus freeing a larger contiguous free block of memory for other applications. All of this is done without the applications losing track of their segments, because the system still presents the same addresses in the descriptor table.

Previous versions of OS/2 did perform segment compaction, however not as efficiently as the current design. Version 1.3 improves the process in the following ways:

- Flexible segment allocating algorithm. Compacting is stopped, when a block becomes large enough to satisfy the allocation for which the compaction is occurring.
- When scanning through the segments to do its compaction, OS/2 V1.2 started at the highest memory point (for example, 8 MB) and moved downwards, compacting as it went. Of course, the system does not compact all memory, that would be too time consuming. Instead it gives up after about a megabyte or when a large enough free block is created. In OS/2 V1.3, the current "window" of compaction activity is marked, so that when this process runs again, it can start from where it left off. This means that some segments are not necessarily compacted several times while other segments are left untouched. Overall, this obviously has the effect of freeing more contiguous memory and improving performance. However, the performance increase due to this is not a real big one.

4.1.2 Segment Swap Control

- Least Recently Used (LRU) sweep

  The OS/2 swap algorithm works by looking through the Global Descriptor Table (GDT) and currently active Local Descriptor Table (LDT) to find the least recently used (LRU) segment to swap out to disk. In OS/2 V1.2, this sweep was done at intervals dictated by the timer. Now in OS/2 V1.3, the LDT is swapped during task switch time, reducing the amount of swap activity. Also the fairer aging is an advantage. Aging is the process, when the system has to decide, which segments are LRU candidates. Previously, it was possible (and even likely) that some processes' LDTs could go unswept. OS/2 would havᵉ treated those segments as LRU when they really were recently used. That causes extra swapping.

- LRU Segment Marking - DosUnlockSeg

  The DosUnlockSeg API call allows the programmer to mark a segment as LRU, to allow the swap algorithm to pick it up and swap it if necessary. This effectively declares a preference for certain segments to be swapped before others, to give other (more critical) segments a greater chance of remaining unswapped. Applications can now collaborate with OS/2's segment swap control to help influence the performance of certain critical parts of that application. OS/2 uses this technique itself on returning from the DOS Compatibility Session.

- Save Logical Video Buffer (LVB) on Entry to DOS

  When the DOS Compatibility Session is invoked in Version 1.3, the LVB is saved, so that on return from the DOS compatibility session, this buffer is restored, and only the PM application window that has foreground focus gets a repaint message. In V1.2, all windows got the repaint message, which inevitably caused more swapping in constrained environments and takes much more time.

  The effect of this change is to redraw the screen much quicker on returning from the DOS Compatibility Session. Also, the DOS Compatibility session (if SWAPDOS in CONFIG.SYS is specified) and LVB segments are marked as LRU when exiting back to protected mode, causing less of the system working set to be swapped out.

- Group Swap Out

  Most segments used by applications use only between 64 bytes and 2 KB. Previously, the swapper performed disk I/O for each of these "mini-segments" in turn. In Version 1.3, groups of them are collected into buffers of 14 KB and then swapped out in one I/O operation only. This reduces the number of I/O operations during swapping, which reduces the time taken by the swap operation. In fact, this modification is one of the biggest performance boosters of Version 1.3.

4.1.3 Swap Algorithm

- Remove Discardable Segments from Calculation

  In V1.2, space was allocated in the swap file for discardable segments, making the swap file bigger than was necessary. In Version 1.3, the swap file size is limited to the amount of swappable memory allocated. This helps to keep the size of the swap file smaller, especially on large memory configuration machines.

- Allocate Swap File at IPL Based on Physical Memory

  OS/2 Version 1.2 always starts with the same size swap file. In Version 1.3, the initial swap file size is dependant on the amount of memory installed. At IPL time, OS/2 calculates how much swap space is likely to be needed and creates a swap file of that size. Enough space is allocated to accommodate typical system load on memory constrained machines. Systems with less than 4 MB of memory will start out with larger swap files. The smaller the memory size, the larger the swap file will be. This avoids the overhead of increasing the swap file in several small steps. It also reduces disk fragmentation which improves swapping performance.

Note: This is not such a problem with an HPFS partition, because of its built in optimization algorithm regarding file space allocation on the physical drive. Also, HPFS is more aware of the special needs of the entire swap operation. On the other hand, the difference is not very big and HPFS consumes much more memory, which is usually not available on such machines.

- Re-Using Unused Space in the Swap File

  Unlike the previous versions of OS/2, V1.3 makes much better use of the swap file. It can, to a certain extent, re-use free space in the swap file better than previous versions do. This doesn't mean, that the swap file shrinks down if swap segments are no longer needed. But at least, it doesn't grow as fast as before if there are free segments in the swap file. This does definitely improve the DASD situation on systems which are running for long hours without being restarted, for example LAN servers remote database servers and communication gateway machines.

- Calculate Memory Overcommitment

  The overcommit calculations were changed to take into account the notion that OS/2 always keeps some of the swappable segments in memory. Which ones may vary, but some percentage are always present. Version 1.2 presumed that it was possible to swap everything out and allocated swap space accordingly. Version 1.3 presumes that half of the available memory will contain swappable segments and reduces the swap file requirement by that amount. Should the system reach the point of being unable to swap a segment out, it leaves the more recently used discardable segments present and discards the least recently used swappable segments instead. This change was almost solely responsible for the 40% (approximate) reduction in swap file size. This change essentially superseded the first one (limiting the swap file size) above.

Comparisons between V1.3 and V1.2 show that V1.2's swap file begins to grow much quicker than V1.3's in the same situation, and that several operations can be performed in a constrained environment in V1.3 before swapping needs to take place, long after V1.2 has done several swap operations under the same circumstances.

4.2 Performance

OS/2 V1.3 gives significant improvements in performance, particularly in memory-constrained environments. Depending on the actual situation, it may be 5% to 25% faster than previous versions of OS/2.

- Memory management is improved by reducing the need to swap to disk. The swap file size is 30% to 40% smaller than previous versions in identical situations. Swapping is also done much faster and more efficiently in terms of disk space.
- The minimum system requirements have been reduced from 3 MB to 2 MB for a Standard Edition based system.

Not only have the memory requirements been reduced but also the DASD requirements. The full installation requires 11 MB of disk space, while the selective installation options can reduce this to as little as 8 MB for the Standard Edition. So, up to 5 MB of disk space can be saved over previous versions.

The improved memory management techniques not only save storage space, but lead to increased performance, as more real storage is freed for OS/2 operations.

4.2.1 Presentation Manager

- Code Reorganization

  The code of PM has been reorganized under Version 1.3. The PM DLLs have been "swap tuned" and related functions have been grouped together. This reduces the amount of memory needed to perform a given task.

- Intelligent Outline Fonts

  The new outline font technology on the Adobe Type Manager (ATM) gives better quality outline fonts, which display up to ten times faster than their V1.2 equivalent outline fonts as well as improved legibility at smaller sizes. The ATM gives OS/2 access to the more than 600 type 1 fonts already distributed by Adobe and other vendors.

- Intelligent Initialization Application Processing

  In V1.2, if an application requested a DLL that didn't exist on the system (that is, it hasn't been installed), V1.2 would go through an error handling process for each application that asked for the DLL.

  In V1.3, an indicator is set showing that the DLL doesn't exist. This saves on error handling and therefore on overall speed of operation.

- More than 1200 Windows

  When programming in OS/2 Presentation Manager, every dialog box, radio button, help, etc. is considered a window. In V1.2, some experience showed an effective limit of around 900 windows for PM applications. This limit has been totally removed, allowing as many windows to be created as needed. There is negligible impact for such "heavy" applications. Since there is now a dynamic organization instead of a "hard coded" limit, some very slight performance degradation can be expected if applications require so many PM windows.

4.2.2 Loader

- Packed Executable

  New LINKER options, EXEPACK and PACKDATA, can produce reduced .EXE file sizes. This was done for all the executable files coming with the operating system itself, leading to some significant reduction in overall DASD usage. You can see this when you compare file sizes between V1.3 and V1.2. The loader itself stayes unchanged, it just works as before even with these new packed executables.

- IPL Startup Time Improvement

  Relocation information is now read from the .EXE and .DLL files in larger blocks. This speeds up program loading (and reloading) significantly. Version 1.3 will load 50% faster than an equivalently configured V1.2 system.

- Loading Programs from the LAN

Loading programs over a LAN is significantly faster in OS/2 V1.3. In Version 1.2, remote media was treated the same as removable media. This means that segments in .EXE and .DLL files were preloaded and marked swappable rather than discardable. This takes a long time and can cause the swap file to grow. In Version 1.3, remote media is treated the same as fixed media. Segments are only loaded as they are needed, and can be discarded/reloaded as necessary. Additionally, if a server does disconnect, OS/2 will attempt to reconnect to satisfy a remote segment load. The safety we had in Version 1.2 is retained while load performance roughly equivalent to that of a local fixed disk is achieved.

4.2.3 File Manager

- Startup Performance

  The File Manager now loads more quickly than in V1.2. Part of this is because of the more selective loading of resources.

  Instead of loading everything at once, the File Manager only loads resources as they are needed. For example, it doesn't automatically open the help file, but waits until it is requested by the user.

- Listing Files and Directories

  The enhanced DosFindFirst and DosFindNext API calls are used within the File Manager to find only directory entries first, rather than every single file. If a directory is then selected by the user, the appropriate files can be searched for. This makes scanning directories much quicker than in V1.2.

- COPY, MOVE and DELETE Faster

  The logic of these operations has been moved to a single DLL to remove duplication of code (for example, verifying existence of the file before operating on it). These new functions can be accessed by applications via an API.

4.2.4 Intelligent ABIOS Install

The root directory of V1.2 had a large number of .BIO files, the ABIOS patches to support OS/2, most of which did not apply to the machine being used. Version 1.3 copies only the appropriate .BIO file(s), reducing DASD requirements and improving IPL time (because it does not have to scan a large number of irrelevant .BIO files).

4.2.5 Selective Install/Reinstall

Version 1.3 allows you to select the parts of the system you wish to install or reinstall. This not only saves DASD space, but also on system working set size, and therefore improves performance.

4.2.6 Summary

OS/2 V1.3 runs faster and smoother, especially in memory constraint situations. The system uses much less DASD space and allows for specifying minimum system installations, in order to run on small machines but still supporting all specific application requirements. In general, it can be said that OS/2 V1.3 provides a much better system base than its predecessors. This is especially true for the existing 16-bit system base with the INTEL 80286 processor.

Chapter 5. Procedures Language 2/REXX

REXX is the SAA[17] (Systems Application Architecture) language for writing procedures under OS/2.

REXX was first implemented between 1979 and 1982 by Mike Cowlishaw of IBM. During this time, REXX was widely disseminated within the company. Consequently it was improved by the feedback of hundreds of users world wide. REXX was first made commercially available as the systems-procedure language for VM/CMS in 1983. When SAA was announced in 1987, REXX was included as the standard systems-procedure language.

A REXX version for MVS was introduced in 1988 and for OS/400[18] , the operating system for the AS/400[19] , it was made available in 1990. The OS/2 implementation was first shipped with OS/2 V1.2 Extended Edition in March 1990.

With Version 1.3 REXX also became part of the OS/2's Standard Edition.

High-quality procedure programming can now be achieved by using common English words in a natural syntax flow, that both beginning and experienced programmers can understand.

REXX uses powerful general-purpose programming functions, common arithmetical abilities, as well as OS/2 commands within a simple framework.

Existing batch files can now be modified, given more function, and replaced with very powerful REXX procedures.

REXX can be very efficient and handy when string manipulation is required. On the other hand, users won't be able to do all the mathematics they want to do. In those cases they might want to program under C or FORTRAN. However, REXX always allows you to include and call those external programs in order to perform specific tasks.

Even more powerful is the ability to program those special functions under C, put them in their own Dynamic Link Library (DLL), and make them available for REXX.

The fact that REXX supports many interfaces and functions under OS/2, such as Dialog Manager and Database Manager, makes it a very powerful prototyping tool. A sample of how to program the interface to the Dialog Manager is given in *OS/2 V1.2 Standard Edition Internals and Evaluation* manual, reference number *GG24-3466*. For samples of the Database Manager refer to the appropriate Extended Edition documentation.

REXX also offers powerful tracing and debugging capabilities. Due to that and the fact it is an interpreter, the user gets a very flexible application development utility that has a handy prototyping environment as well as a nice application "front end".

[17] SAA and Systems Application Architecture are trademarks of International Business Machine Corporation.

[18] OS/400 is a trademark of International Business Machines Corporation.

[19] AS/400 is a trademark of International Business Machines Corporation.

5.1 Portability

Since Procedures Language 2/REXX is part of IBM's System Application Architecture, portability of REXX procedures to the other SAA platforms (on MVS/ESA,[20] VM/XA[21] , etc.) is possible as long as the user doesn't use specific features of these environments.

5.2 Batch Files Versus REXX Procedures

5.2.1 Batch Files

A batch file is a common processing file containing commands for the operating system (OS/2 or DOS).

When the user runs a batch file, the OS/2 or DOS Command Processor will process its commands one at a time. So batch files reduce typing time and typing errors when controlling the system.

Batch files for OS/2 must have the ".CMD" filename extension.

Batch files for DOS (and the OS/2 "DOS compatibility session") must have the ".BAT" filename extension.

5.2.2 REXX Procedures

REXX commands are extraordinarily more powerful than ordinary batch file commands because REXX adds many easy-to-handle programming language facilities to batch files/procedures.

This also makes REXX useful for prototyping in various situations.

REXX procedures are handled by a special REXX command processor and interpreter.

5.3 Implementation

The REXX interpreter is now part of OS/2 Standard Edition Version 1.3.

It is automatically installed with the base operating system. Basically, it consists of several DLL modules as well as some header files that allow users to implement REXX functions in their own programs. See also section 5.5.9, "Other Programming Languages" on page 31.

When installed, the "REXX interpreter" is **automatically invoked** by the OS/2 Command Processor **if the first line of the OS/2 Batch File is a REXX comment line:**

```
/* This is a REXX procedure comment line */
```

Note: In the first line the *"/*"* must be in columns 1 and 2. In the second and following lines comments may begin anywhere.

[20] MVS/ESA is a trademark of International Business Machines Corporation.

[21] VM/XA is a trademark of International Business Machines Corporation.

Note: If you do not have REXX installed, but nevertheless start a REXX procedure, the message *"Procedures Language 2/REXX is not installed"* will appear.

5.4 Overview

5.4.1 REXX Features at a Glance

1. Free format
   - Instructions can span multiple lines
   - Multiple instructions per line
   - Instructions can start in any column
   - Mixed-case support
2. Typeless data
   - Everything, including numbers, is a string
3. Parsing capabilities
   - Extensive character string manipulation capabilities
4. String-oriented expression evaluation
   - Numeric precision, as much as you like
   - Automatic concatenation of adjacent terms in expressions
   - Undefined variables treated as literals instead of errors
5. Structured programming constructs
   - Conditionals (if..else)
   - Blocks (do..end)
   - Loops (do..repetitor..end)
   - Functions and procedures
-with arguments and local variables
6. Special instructions for parsing strings into component words or fields
7. Hierarchy for resolving function and procedure names
   - User-defined functions and procedures
   - Built-in functions and procedures
   - External routines
     - Other REXX programs
     - Operating system commands
     - Compiled programs
   - Debugging
     - Trace for run-time animation
8. Operation is transparent to the OS/2 batch file executor.

5.5 Interfaces to Other Products

Procedures Language 2/REXX can extend its service to the user by using other applications and system utilities via the REXX Application Programming Interface (REXX API). Therefore, REXX includes handlers for subcommand, external function, and system exit processing.

5.5.1 Function Calls and Subroutines

REXX programs normally make extensive use of function calls and subroutines. Functions and subroutines are written in the same manner. The difference is that a routine must return a value when called as a function, whereas the return value is optional when called as a subroutine. Up to 20 variable-length parameters may be passed by value to REXX functions and subroutines. Unlike many languages, REXX does not use a formal parameter list on a function or subroutine prototype statement to access the actual parameters. Instead, the parameters are read through the Parse Arg and Arg instructions, as if they were being read from a special device or buffer.

REXX has several types of functions:

- Built-in functions

  REXX provides over 60 built-in functions. The functions cover such areas as:

  - Character-oriented string manipulation
  - Word-oriented string manipulation
  - Numeric operations
  - File input and output
  - OS/2-specific system operations

- Internal functions

  Functions may be included within the file containing the main program. They begin with a label (which gives them their name) and end with a return instruction.

- External functions

  Functions may be written in REXX, stored in individual files, and called by any REXX program without declaring the functions, which are identified by their file names.

Functions may also be written in C, Pascal, or MASM and called from REXX. This requires that the functions be "registered," as discussed below.

5.5.2 Commands

OS/2 commands can be issued by REXX programs. The full text of a command may be hard-coded, or the power of REXX expression and function processing may be used to form the command.

REXX can also be used as a macro processor by any application that "registers" itself with REXX. The REXX procedure replaces a user typing a command at the keyboard. In this case, commands issued by the REXX program are routed to the application for handling rather than to OS/2.

5.5.3 OS/2 Interfaces

REXX can be used with the applications for interfacing with OS/2 through the REXX APIs. Subcommands and functions can be used for adding new features to REXX that enhance the capabilities for interfacing with OS/2. These and other features for interfacing with OS/2 are described below.

5.5.4 Using REXX APIs

REXX allows an application to provide extensions to many parts of REXX, including handlers for subcommands, external functions, and system-exit processing. These handlers are all similar in how they must be coded, compiled, and packaged. All handlers may be packaged as part of an .EXE module or as a Dynamic Link Library (DLL). REXX requires that a handler be written as a large model program. These handlers must be registered with REXX before they can be used. Registration informs REXX of the location of the handler and how to invoke it. For example, the information provided in the registration of a dynamic-link external functions includes the name of the function, the name of a dynamic link library containing the handler, and the name of the procedure within the dynamic link library that implements the function.

There are some differences in usage between handlers packaged as .EXE modules and dynamic link libraries. .EXE handlers are local to the registering process, whereas DLL handlers are global to the OS/2 system. That is, handlers packaged as part of an .EXE module can only be invoked by a REXX procedure running in the same process that registered the handler. Handlers packaged as part of a DLL can be invoked from a REXX procedure running in any process in any session in OS/2.

5.5.5 Extending REXX (Subcommands and Functions)

There are two ways to add new features to REXX: subcommands and functions. They may be used individually or in combination. By writing subcommands or functions, end users and application writers can enhance and change the capabilities of REXX.

Subcommands are defined as single clauses consisting of an expression. The expression is evaluated and the result is passed as a command string to the currently "addressed" environment. Subcommands are used in the context of REXX running as a macro processor under some environment or program.

Functions are direct extensions to the capabilities of REXX. They are used when an application developer wants to code an application in REXX, but needs specialized features that are not provided in the language. Where subcommands generally correspond to the application's normal command set, functions are generally REXX-specific and many only have a meaning from within REXX.

Functions are used to generate a result based on given inputs (a function in the mathematical sense) or to produce certain side effects within the REXX environment. Functions may be as simple as an encapsulation of several base REXX instructions or as complex as a large library of functions external to REXX written in some other programming language.

5.5.5.1 Subcommands

The subcommand interface to REXX is well suited to applications using REXX as a macro processor, especially if the application has a rich set of commands. Since subcommands are intimately associated with a given application, they are written as part of the application. The subcommand code may reside in the module (either EXE or DLL) with the application or in an accessory DLL. Any separate procedure that can be registered with REXX and follows the appropriate conventions can be a handler.

There are five programming interfaces for REXX subcommands. Two of these interfaces, RxSubcomDrop and RxSubcomRegister, are used quite frequently by application programs. RxSubcomRegister registers the subcommand handler with REXX before starting a REXX procedure. RxSubcomDrop is used to drop the handler upon completion. The other three interfaces are needed less frequently by applications. RxSubcomQuery can be called to query if a particular subcommand handler is currently registered with REXX. RxSubcomLoad forces REXX to load a handler (DLL handlers are not normally loaded until they are executed the first time). RxSubcomExecute is used to duplicate the REXX calling interface to execute a subcommand handler.

5.5.5.2 Functions

External functions written in REXX do not need to be registered with REXX; they are located by searching the disk for a file name that matches the function name being called. Functions written in other languages must be registered. Whether they are packaged as part of an EXE module or as a procedure within a dynamic link library, REXX must be told where they are so that the correct invocation mechanism can be used.

These non-REXX external functions are more difficult to write than REXX external functions, but there are circumstances where significant benefits can result from the extra effort. Performance for specific actions is one common reason. Any action that does not need the generality or string-handling capabilities of REXX would be a candidate for implementation in another language.

There are four programming interfaces for non-REXX external functions that are very similar to the APIs provided for subcommands. These APIs include a registration, deregistration, query, and call (execute) interface. REXX also provides some functions that can be used from within REXX that parallel these APIs. A common use for these functions is developing a REXX language profile that loads commonly used external function packages into memory at system startup.

5.5.6 Macrospace

There are times when REXX is the best language to implement an external function, such as string manipulation. REXX provides a method to let applications store and execute high-usage REXX procedures in memory by loading them into a macrospace. There are several APIs for manipulating macrospace. RxMacroChange takes a REXX procedure that is on disk and adds it to the macrospace. RxMacroDrop deletes any individual macro from the macrospace and RxMacroErase erases all functions in the macrospace.

An application can have a library of unique macros, not available as individual REXX procedures on the hard disk. RxMacroSave saves either all, or a specified subset of all, functions in the macrospace to a specified file. RxMacroLoad then loads either all, or a specified subset of all, functions in the macrospace to a specified file. RxMacroLoad then loads with all, or a specified subset, of all functions from a specified file into the macrospace.

5.5.7 System Exits

System exits allow REXX to operate under various host operating systems. When the exits are used, certain system-dependent interpreter activities are executed as calls to routines provided by the caller of the interpreter, instead of using code of the interpreter itself. There are several types of exits:

- For the administration of resources, at the beginninq and end of interpretation
- For assisted linkages to general classes of external facilities, that is, commands and functions
- For special language features (for example, I/O to standard resources)
- For those that poll for external interrupts and conditions.

Like subcommands, system-exit handlers must be registered with REXX before a system exit can be used. The interpreter exits are defined on invocation of the interpreter via the SYSEXIT parameter of the call to the REXX interpreter.

5.5.8 Variable Pool

REXX provides an interface (RxVar) to allow certain applications (such as subcommands and external functions) to read and modify the current set of variables in use by a REXX procedure. Certain private REXX information (such as the version number of the interpreter) can be queried. The applications provide control information instructing REXX what changes to make. REXX then processes the request internally. Applications do not need to know the internals of the REXX variable pool to make direct changes.

5.5.9 Other Programming Languages

REXX applications can "call" programs that are written in other programming languages such as:

- C
- COBOL
- FORTRAN
- Pascal
- Assembler.

These programming languages also can "call" REXX procedure(s).

But there is more than just that. Programmers can build in powerful functions, which are supported by REXX, within their own applications. The way to do this is to "call" those REXX functions within the Dynamic Link Libraries (DLL). These REXX DLLs are provided and installed with the base operating system.

Examples are provided in the *OS/2 Programming Tools and Information V1.3* package.

5.5.10 Using REXX with DLLs (Dynamic Link Library)

As mentioned before, it is possible to add more functions to REXX by writing them in an SAA-supported language and placing them into a DLL. These functions can then be registered in a REXX program and used as any REXX function. Therefore REXX can have the functionality of any other SAA language.

There are only two requirements on the structure of functions in a DLL that are to be used by REXX. The function must be called with five parameters and must return a short integer directly and return information in a calling parameter.

The format for an external function is:

```
SHORT APIENTRY dll_rexx_function (PSZ, SHORT, PRXSTRING, PSZ, PRXSTRING)

SHORT APIENTRY dll_rexx_function (name, argnum, argarr, queue, retstr)
        PSZ         name
        SHORT       argnum
        PRXSTRING   argarr
        PSZ         queue
        PRXSTRING   retstr
```

- Where name is a pointer to a string. This is the name by which the function is invoked.
- Where argnum is the number of arguments passed in the call to the function.
- Where argarr is the arguments passed to the function. It is defined as a pointer to structure that contains a character string and an integer representing the length of the string.
- Where queue is the name of the queue to be used by the function and is a pointer to a string.
- Where retstr is the return information required by the REXX program. It is defined in the same way as the arguments passed to the function.

All these definition types are given in the include file *REXXSAA.H* and this file should be included in your program.

All results should be returned in the final parameter passed to function. This parameter is defined as a structure and includes a character array and integer definition. The character array is used to contain the string result that is returned to the REXX program and the integer is used to contain the length of the string.

All REXX external functions require that information is returned to the REXX program in this way. If no value is returned from the function, all these values can be empty. For these cases, the interpreter will generate error code 44, "Function did not return data".

The function should pass a short integer back to the REXX program when exited to indicate the success or failure of the function. This should be done by the *return* command.

Examples are provided in the *OS/2 Programming Tools and Information V1.3* package.

5.5.11 Dialog Manager

The Dialog Manager, coupled to and co-operating with REXX, can be an attractive prototyping tool in various situations.

It is relatively easy to combine these two products and have automatic Common User Access (CUA) screen input, output, and handling invoked from a powerful and easy-to-use procedures language.

However, the following services are not supported:

VDEFINE
VDELETE
VREPLACE
VRESET.

Since ARRAY variables must be explicitly defined (VDEFINE), which is not possible via REXX, there is a problem passing arrays from REXX to Dialog Manager.

A suggestion could be to create a public DM variable access exit that understands how to get/put variables to subscripted REXX variables. For example:

```
<varlist class=RexxVarAccess>
  <vardcl name=array1 maxdim=100>
  ...
</varlist>
```

The variable access exit would have to be publicly registered, since REXX can not register window procedures (which is how DM implements variable exits).

When using the Dialog Manager directly from REXX it is very advisable to check the Dialog Manager return codes during testing.

The Dialog Manager return codes are returned in the DM communication area.

Example:

```
ADDRESS ISPCIR 'DISPLAY PANEL(PREPORT) DMCOMM(DMC)'
```

DMC is in this case the DM communication area.

Variable	**Contains**
DMC.1	= Return Code (Same as RC)
DMC.2	= Reason Code
DMC.3	= Instance Id
DMC.4	= OS/2 Return code
DMC.5-16	= Error information, if any

When testing the procedure from "Process STATus Report - Application Example 3" after each call to the Dialog Manager, a check is done for the various return codes. The error-handling is done by procedure ERROR_HANDLER:

```
/* Error Handle Procedure */

ERROR_HANDLER: PROCEDURE EXPOSE RC SIGL DMC DMC.2
  IF rc=8 & dmc.2=801700 THEN RETURN /* The user selected exit */
  x=sigl-1
  SAY "-----------------------------------------"
  SAY "Error return from line" x "- Return Code =" rc
  SAY "Dialog Manager Reason code =" DMC.2
  SAY "-----------------------------------------"
  /* Close the Dialog Manager Services */
  ADDRESS ISPCIR 'DMCLOSE DMCOMM(DMC)'
  /* Drop the connection from REXX with the Dialog Manager */
  RXSUBCOM DROP ISPCIR
  EXIT
RETURN
```

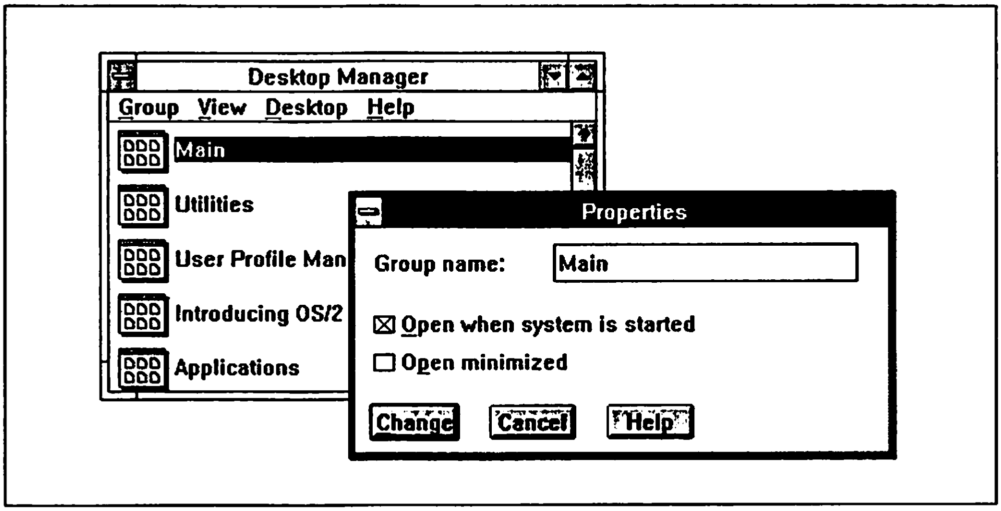
For more information about application development using the Dialog Manager see *OS/2 Dialog Manager Application Development.*

Chapter 6. Presentation Manager Shell

In OS/2 Version 1.3, the Presentation Manager application properties have been improved in the following areas:

- Group Properties
- Program Properties.

6.1 Group Properties



Group name: Main

☒ Open when system is started

□ Open minimized

Figure 4. Desktop Manager Group Properties

When creating or changing a group entry under Desktop Manager, it is possible to specify the properties and tell the system how to handle this group during system start. There are three choices:

- **Don't open**
- **Open when system is started.** The system will open and display the group window.
- **Open minimized.** The system will open and display the group window, but it will be minimized as an icon.

See Figure 4 for an illustration of these new group properties in OS/2 V1.3. Of course, like many other settings under Presentation Manager, all those definitions and settings are stored in the **OS2.INI** file. Replacing this file will result in a totally different system appearance. It is important to understand that this file together with its counterpart, **OS2SYS.INI** is absolutely essential and therefore critical to the overall system appearance and behavior.

Usually this file is locked and the user cannot delete or overwrite it. Access to this file is only allowed through the profile APIs and is thereby sequenced and controlled by the system. However, system failures such as sudden power loss or switching off the system without performing a regular SHUTDOWN or at least the warm boot sequence CTRL-ALT-DEL, can corrupt these two files. The effect may not even be visible and a CHKDSK operation may not report any bad files. The users may only encounter problems when they try to do things like:

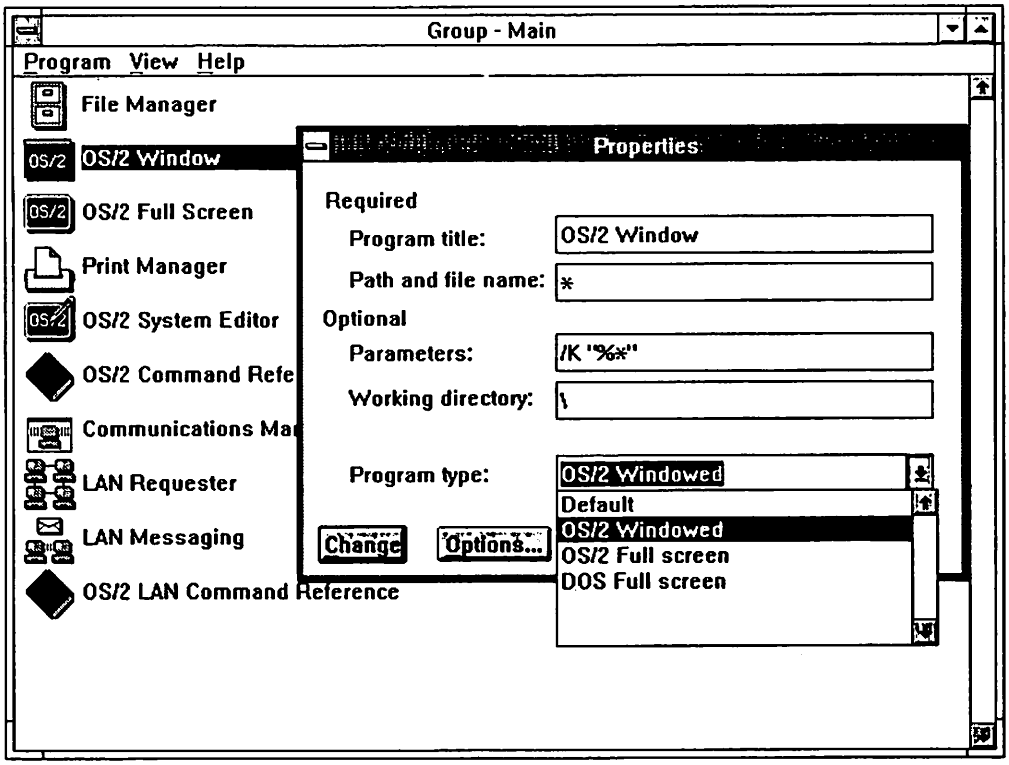
- Modify printer settings
- Create a new program entry
- Change colors
- Just any normal task, which involves access to these two profiles.

The situation then usually gets worse with every try. In really bad circumstances, Communications Manager and other components may fail to start, without any visible reason and no obvious explanation!

Recommendation

> Respect the .INI files and keep a working backup of them at all times. Of course, the system has to be restarted from the OS/2 installation disk in order to gain access to these files.

6.2 Program Properties



Required

Program title: OS/2 Window

Path and file name: *

Optional

Parameters: /K "%*"

Working directory: \

Program type: OS/2 Windowed

Default
OS/2 Windowed
OS/2 Full screen
DOS Full screen

Figure 5. Program Properties: Program Type

When creating or changing a program, the following information can be entered:

- **Program title** (required)
- **Path and file name** (required)
- **Parameters** (optional)

- **Working directory** (optional)
- **Program type** (required).

  This is a new feature of OS/2 V1.3. It specifies how to run a program. All possible program types are listed in the program type drop-down combination box, when a program title is added or changed.

The program can be one several types:

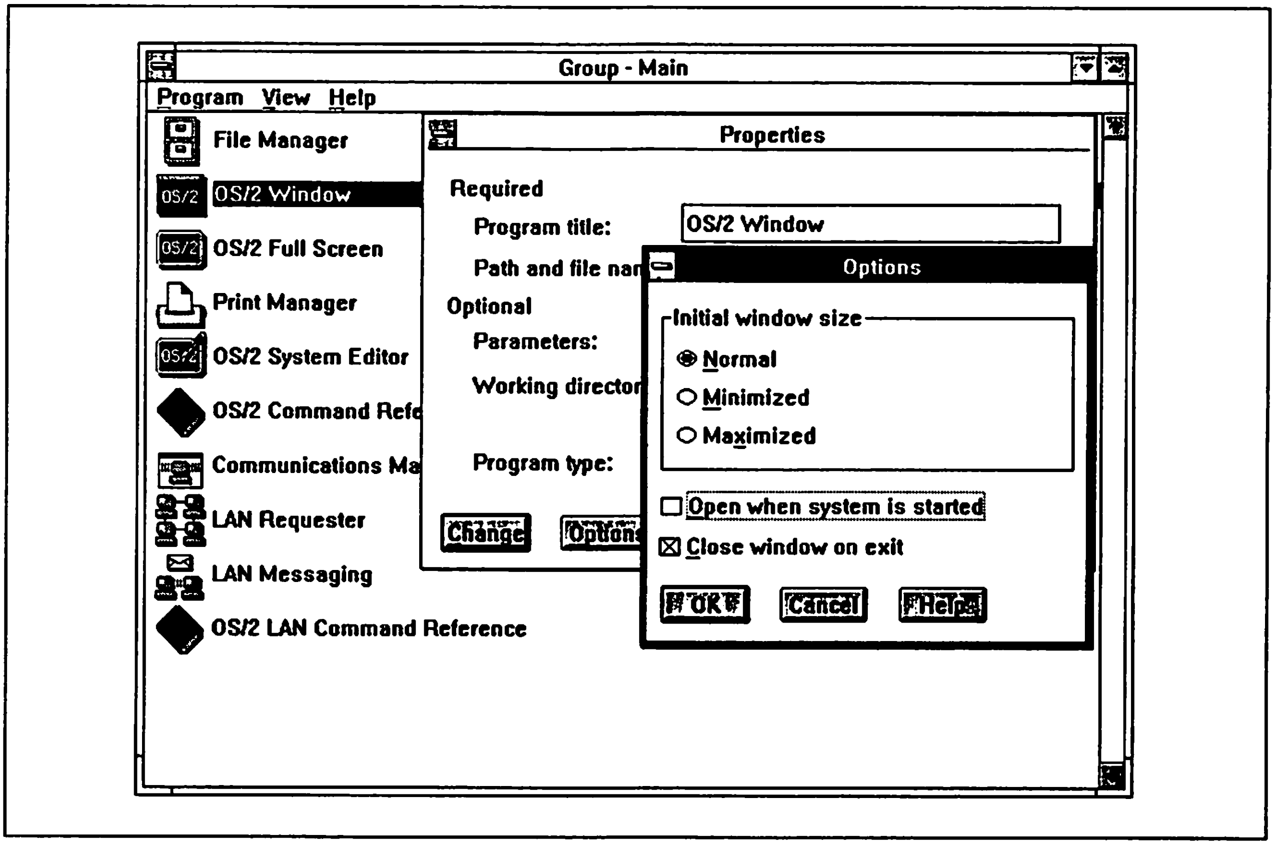
- **Presentation Manager**

  If chosen, Desktop Manager tries to find and use the ICON information which is stored as a PM resource file within the executable file. If not found, as well as in the following cases, the appropriate default ICON is selected if no .ICO file with the same filename can be found.

- **Windowed**
- **Full-screen**
- **DOS** program that runs in the DOS Compatibility Session. Unlike OS/2 programs, only one DOS program can run at a time.

See Figure 5 on page 36 for an illustration of program type.

6.3 Program Properties Options



Program title: OS/2 Window

Initial window size
● Normal
○ Minimized
○ Maximized

□ Open when system is started
☒ Close window on exit

Figure 6. Program Properties: Options

In V1.3, a new program properties option allows you to start the program when the system is started. You can also specify the initial window size and an automatic window close on exit. See Figure 6 for an illustration of Program Properties.

Under program properties, there are two options:

- **Open when system is started.**

  The program will start when the system is started. This has the same effect as the START command in the STARTUP.CMD file.

- **Close window on exit.**

When using the option *Open when system is started*, you have three options:

- **Normal window**
- **Minimized window**
- **Maximized window.**

See Figure 6 on page 37 for an illustration of program properties options.

For the inexperienced user this might be a more convenient way of configuring the system. On the other hand, a system administrator or user help desk person may find it difficult to maintain such a system. Unfortunately, there is no way of listing all the specified startup options for the user's desktop environment. That means one would have to check all entries in the entire desktop in order to understand what the startup configuration is supposed to be. The old way of editing a simple ASCII file like STARTUP.CMD is still much more maintainable. Also, there are almost no corrupted STARTUP.CMD files but many magic instances of OS2.INI files. Also, in a complex Extended Edition environment, there might be some timing dependencies between several program initializing sequences. This can be handled very well in a STARTUP.CMD command file, but not through those Desktop Manager entries.

6.4 Icons

6.4.1 Icon Size Reduction

A design change to the Presentation Manager shell in OS/2 V1.3 reduces the size of the icons.

6.4.2 Visual Enhancement

The 64 X 64 icons in OS/2 Version 1.2 required too much desktop space and allowed only seven icons across the bottom of the screen and only nine icons vertically in a group window.

Aesthetically the icons appeared out of proportion and were not commensurate with the resolution of the device on which they were displayed. It was decided that the icons would be reduced to achieve a cohesive appearance with the rest of the screen elements (max/min buttons, title/scroll bar, etc.).

6.4.3 Design Enhancement

The design and planning reasons for the icon size reduction are many and varied:

- It ensures that new applications are developed with the new icons
- It makes 16 bit applications compatible with future generations of OS/2
- It makes it easier to migrate Windows applications to OS/2
- It ensures consistency across display devices.

Basically, the size of the icon has been reduced from 64 pels to 40 pels square. The result is an icon which, when displayed on an 8514/A, is relatively the same size as a 32 pel icon displayed on a VGA screen. AT 64 pels, icons had twice the number of pels as the 32 pel icons used on the VGA screen. The 8514/A however, does not have twice the resolution of the VGA, so fewer icons could be displayed on the 8514/A across the bottom as iconized applications.

The 40 pel square size for the new icons is the result of usability tests and studies, which concluded that they project an improved visual appearance for OS/2 on 8514/A and the new XGA display adapters.

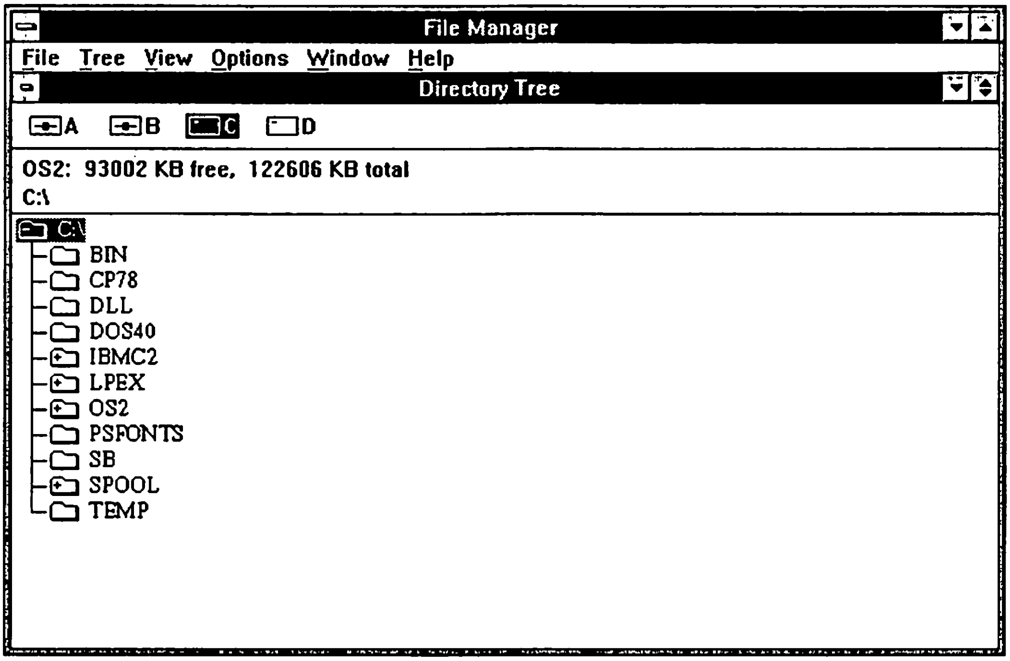
Chapter 7. File Manager

In OS/2 Version 1.3, the File Manager has been improved in the following areas:

- Displaying the directory tree
- Copying and moving files
- Creating subdirectories.

7.1 Displaying Directory Tree

When File Manager is started, it comes up with its directory tree window. This window displays the root directory and one level of directories underneath it on the current disk drive. See Figure 7. When File Manager was started in V1.2, this window displayed all subdirectories instead of only one level. Of course, this can have a positive impact on the start up performance of File Manager. Also, usually not everybody is interested in a fully exploded directory tree in the first place. Now with V1.3 the user can select the directory of interest or request a fully expanded tree via the pull-down menu.



OS2: 93002 KB free, 122606 KB total
C:\

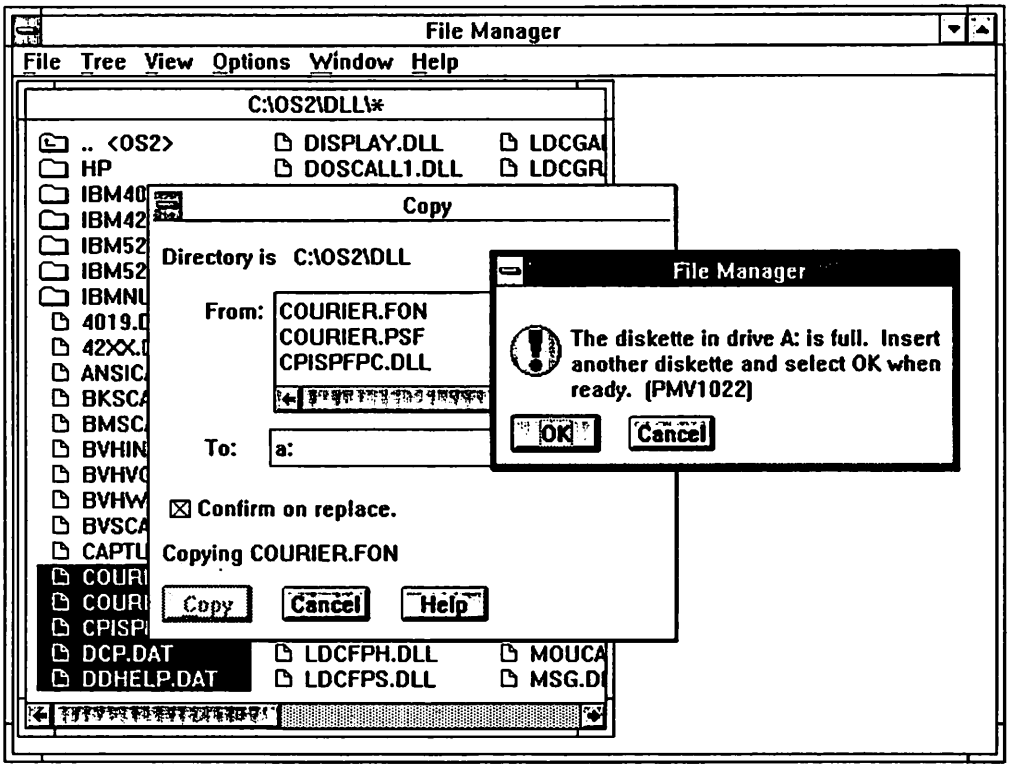
Figure 7. File Manager: Directory Tree

To display an outline of a directory on the current disk drive, click on the plus symbol (+) next to that directory name. The plus symbol (+) always appears next to names of directories that contain one or more subdirectories. See Figure 7.

Also File Manager now only runs through the absolutely necessary initialization steps instead of loading the entire application. Functions like COPY, MOVE and others, have been moved into separate DLLs and are only loaded on demand.

As a sideffect, some of these functions are now available and published as regular APIs.

7.2 Copy and Move Files



Directory is C:\OS2\DLL

From: COURIER.FON
COURIER.PSF
CPISPFPC.DLL

To: a:

☒ Confirm on replace.

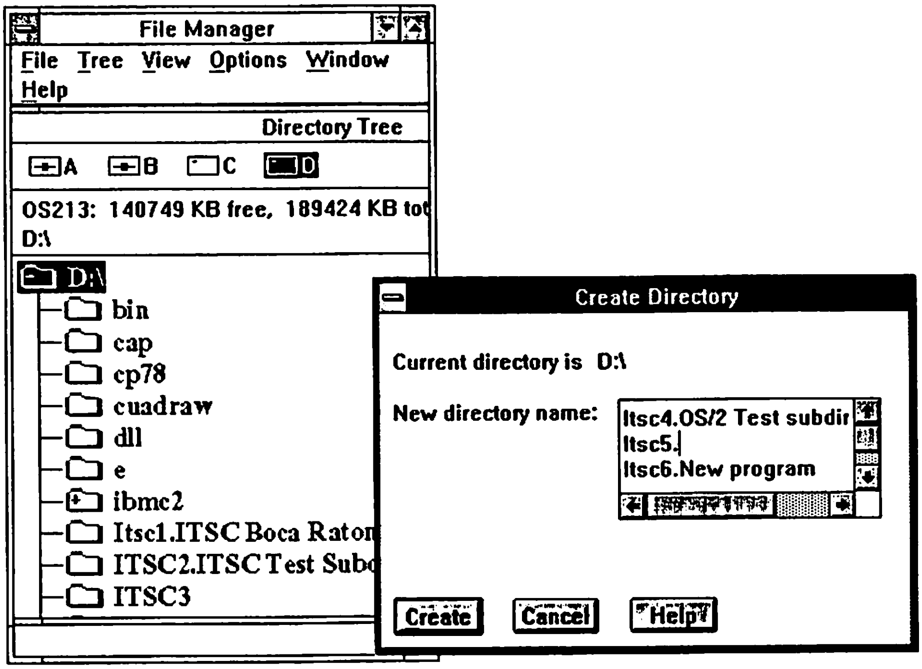
Copying COURIER.FON

The diskette in drive A: is full. Insert another diskette and select OK when ready. [PMV1022]

Figure 8. File Manager: Copy File

With OS/2 V1.3, Copying and moving files is also faster. If the users try to copy or move some files onto another drive and that drive is full, they will receive a warning message which will allow for another diskette on the target drive so that the operation can be completed. See Figure 8.

7.3 Create Directory



OS213: 140749 KB free, 189424 KB tot
D:\

D:\
bin
cap
cp78
cuadraw
dll
e
ibmc2
Itsc1.ITSC Boca Raton
ITSC2.ITSC Test Subd
ITSC3

Current directory is D:\

New directory name:
Itsc4.OS/2 Test subdir
Itsc5.
Itsc6.New program

Figure 9. File Manager: Create Directory

It is now also possible to create more than one directory within one step. See Figure 9.

7.4 Fonts

Of course, File Manager does support the new Type 1 fonts as well as any other installed PM fonts. This allows users to tailor this application to their personal preferences.

The same is true for the system editor, which also supports any available PM font.

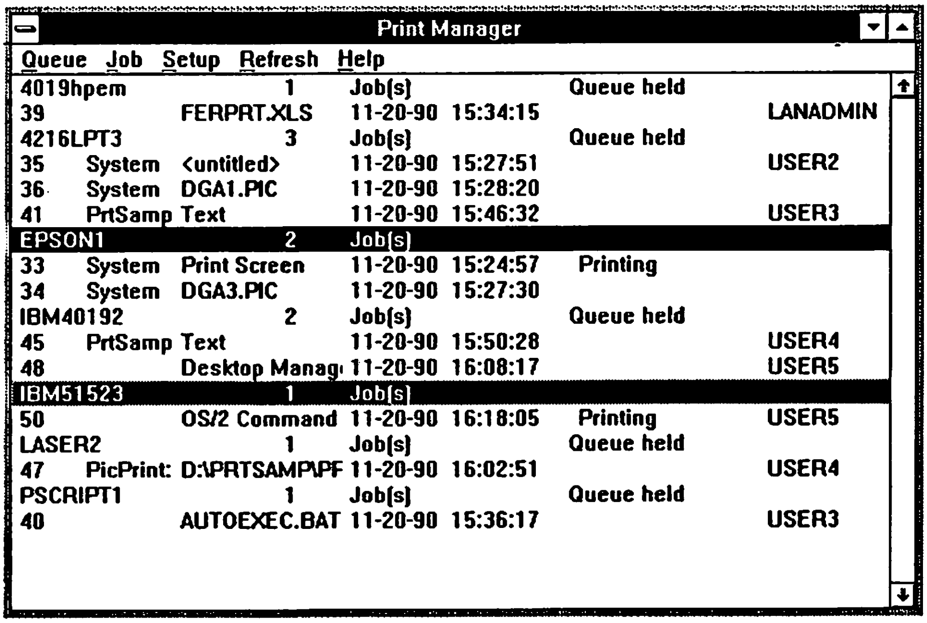
Chapter 8. Print Manager

The Print Manager is an interface for managing printers and printer queues. The Print Manager can:

- Install printers.
- Install new printer device drivers.
- Install fonts.
- Configure printers, plotters, and queues.
- Control the whole print process.

In general, the Print Manager is the user's interface to the entire print subsystem.

The following figure is an example of what a busy Print Manager dialog box might look like on a LAN that does a lot of printing. Each of the queues that are listed will show how many jobs they have on them, the status of the queue, the status of the job if it is not simply waiting to print, the name of the user that sent the job (if on the LAN) and, if the application has been coded correctly, there should also be a job number and name.



```
4019hpem                  1    Job(s)              Queue held
39             FERPRT.XLS      11-20-90  15:34:15               LANADMIN
4216LPT3                  3    Job(s)              Queue held
35   System    <untitled>      11-20-90  15:27:51               USER2
36   System    DGA1.PIC        11-20-90  15:28:20
41   PrtSamp   Text            11-20-90  15:46:32               USER3
EPSON1                    2    Job(s)
33   System    Print Screen    11-20-90  15:24:57   Printing
34   System    DGA3.PIC        11-20-90  15:27:30
IBM40192                  2    Job(s)              Queue held
45   PrtSamp   Text            11-20-90  15:50:28               USER4
48             Desktop Manag   11-20-90  16:08:17               USER5
IBM51523                  1    Job(s)
50             OS/2 Command    11-20-90  16:18:05   Printing    USER5
LASER2                    1    Job(s)              Queue held
47   PicPrint: D:\PRTSAMP\PF   11-20-90  16:02:51               USER4
PSCRIPT1                  1    Job(s)              Queue held
40             AUTOEXEC.BAT    11-20-90  15:36:17               USER3
```

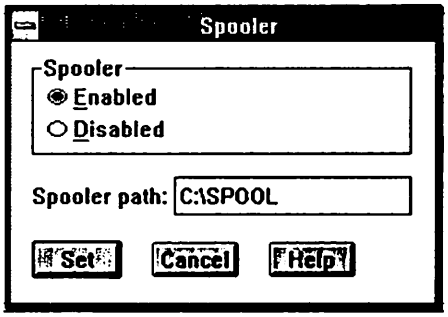
Figure 10. Print Manager, Main Window

Notice that you will also see the file names of the jobs in the queues.

In the following sections we will discuss, in more detail, some of the new features related to OS/2 V1.3 and the Print Manager.

8.1 Spooler

When you select the spooler option on any non-LAN system you will only be allowed to *Enable or Disable* the spooler or change the drive and path used by the spooler.



Spooler

● Enabled

○ Disabled

Spooler path: C:\SPOOL

Figure 11. Standard spooler Dialog Box

If you want to disable the spooler:

- You must select the *Disabled* button then select Set.
- The system MUST then be restarted before the spooler will be disabled.

If the spooler has been disabled and is now required, you have to select the *Enabled* button then select Set and the spooler will start.

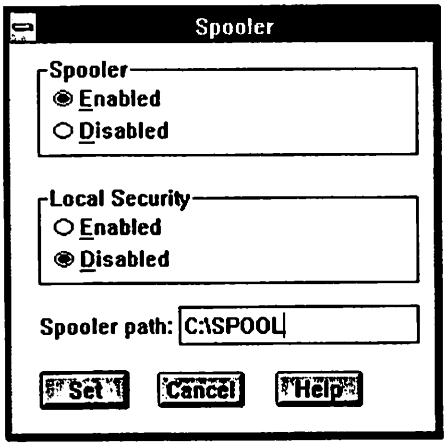
- It is not necessary to restart the system to enable the spooler.

If you wish to change the drive and or path to the spooler, all that is required is that the new drive and or path be put in the *Spooler path* field of the SPOOLER dialog box and then select Set. This is the actual place on the hard disks where all the spool files will be stored while being spooled and held in the queue!

NOTE

You will NOT be allowed to change the spooler path *if there are one or more jobs in any of the queues.*

Now, if the machine you are working on is a **LAN server** and **the server is started**, there will be an additional option in the spooler dialog box, which will allow you to *Enable or Disable* **Local Security.**



Spooler
● Enabled
○ Disabled

Local Security
○ Enabled
● Disabled

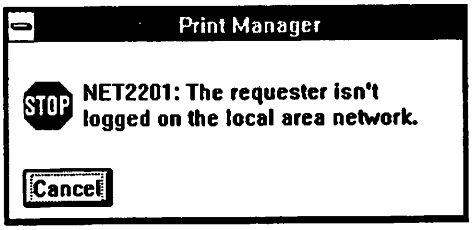
Spooler path: C:\SPOOL

Figure 12. LAN Server spooler Dialog Box.

Anyone can *enable* Local Security by going into the Print Manager and selecting *Setup - spooler* and then choosing the *Enabled* radio button in the *Local Security* window and then selecting *Set*. You do not have to be signed on to the LAN in order to enable the Local Security, but you **MUST be signed on as an administrator** to be allowed to disable it.

Once Local Security is enabled, if you try to cancel or manipulate any jobs or queues without signing onto the LAN you will find the following:

- The Printer Install works and will allow you to install additional printers and drivers.
- If you select spooler, printers, or queues from the Setup menu or try to manipulate any jobs or queues in the Print Manager, you will receive the message that is displayed in the following figure.



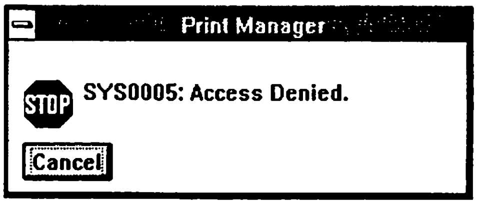
NET2201: The requester isn't logged on the local area network.

Figure 13. Message Box NET2201

- Application Defaults does bring up its dialog box and you can select a different printer to be the application default printer and then select SET. However, you will find that when you go back into the Application Defaults, your selection was actually ignored and the default printer will still be what it was before you tried to change it.
- Dos Timeout will allow you to change the value of the Dos Timeout and also turn it on or off for the ports on your machine.

If you logon to the LAN as a USER and then try to use the SETUP options in the Print Manager you will find that you are basically in a **READ ONLY** mode and you will get the following results:

- Printer Install will work normally.
- SPOOLER will bring up the SPOOLER dialog box and you can select different radio buttons in the windows, but when you try to save your choices by selecting SET you will get the SYS0005 error as displayed in the following Figure.



SYS0005: Access Denied.

Figure 14. Message Box SYS0005

- PRINTERS will bring up the Printers dialog box, but the ADD and DELETE buttons will be unavailable. You can select CHANGE and you will get the Change Printer dialog box but the PRINTER PROPERTIES and the CHANGE buttons are unavailable. So you are basically in a **READ ONLY** mode.
- QUEUES will react the same as the Printers option and work basically in a READ ONLY mode.
- APPLICATION DEFAULTS is also in a READ ONLY mode as any choices you make will be ignored.
- DOS TIMEOUT works as it normally does.

If you logon to the LAN as an **ADMINISTRATOR**, all functions will be returned to normal operations. You will also be allowed to disable the Local Security.

NOTE

The Local Security selection in the SPOOLER option of the Print Manager will **only appear on a LAN SERVER machine and ONLY if the SERVER is running.**

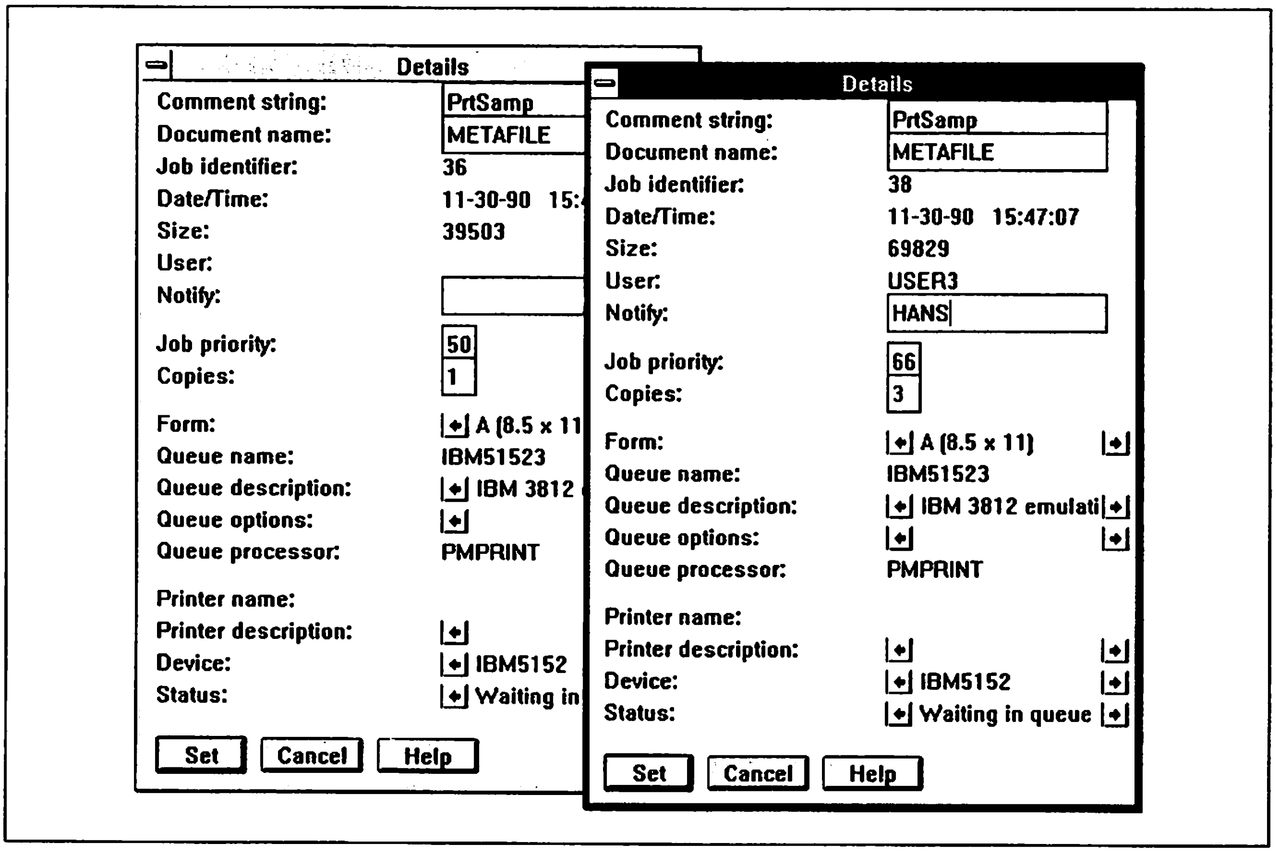
8.2 Jobs

One or more jobs in the spooler can be manipulated at the same time and selected to be:

- Canceled
  - This will remove the selected job from the queue.
- Hold
  - This will hold the selected job but allow the rest of the jobs on the queue to print normally.
- Released
  - The selected job with the status of *Job Held* will be released.
  - Will not release jobs that are held due to a *Queue Hold*.
- Started Again
  - If a Job has been paused by the system you can select it and then use this option to restart it.
- Printed Next
  - If there are multiple jobs on a queue, the system starts on the top one on the list and processes them in order.
  - If there is a job low on the queue, which needs to be printed out quickly, you can select the job and then use this option to move the job to the top of the queue where it will be the next job printed.
- Job Details

  This will give the details of the selected job, such as title, when it was created, who created it, and the status of the job.

The following figure shows the details of two print jobs waiting in one of the queues:



Comment string: PrtSamp
Document name: METAFILE
Job identifier: 36
Date/Time: 11-30-90 15:
Size: 39503
User:
Notify:
Job priority: 50
Copies: 1
Form: ↓ A (8.5 x 11
Queue name: IBM51523
Queue description: ↓ IBM 3812
Queue options: ↓
Queue processor: PMPRINT
Printer name:
Printer description: ↓
Device: ↓ IBM5152
Status: ↓ Waiting in

Comment string: PrtSamp
Document name: METAFILE
Job identifier: 38
Date/Time: 11-30-90 15:47:07
Size: 69829
User: USER3
Notify: HANS
Job priority: 66
Copies: 3
Form: ↓ A (8.5 x 11) ↓
Queue name: IBM51523
Queue description: ↓ IBM 3812 emulati ↓
Queue options: ↓ ↓
Queue processor: PMPRINT
Printer name:
Printer description: ↓ ↓
Device: ↓ IBM5152 ↓
Status: ↓ Waiting in queue ↓

Figure 15. Print Manager, Job Details

Besides the fact that the Job Details have been slightly reordered under V1.3, one may notice some new and helpful fields in there:

Notify Only active on a LAN though, it tells the LAN server who to inform once the job is printed, or gets stuck because of any problems. This field can also be modified by the user on the server.

Copies Tells Print Manager how many copies of this output the user wishes to get. This field can also be modified by the user on the server.

Status This will actually reflect the current status of this print job in the queue.

The two jobs in this sample have actually been printed by the same application but show two minor differences:

Size The second job is almost twice as big as the first one. The difference is that it was created in RAW format rather than STD format as the first one. As you can see, this can have some impact on performance, when it comes to heavy graphics printing.

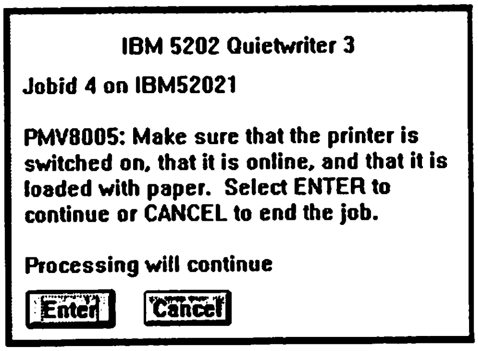
User Since the second job was printed while being logged on to the LAN, you can see the user's name.

Job Priority The user changed the second job's priority to a higher number. That means, this job will be printed before the other one in this queue.

Copies The user requested three copies of this second print job.

NOTE

If a job gets sent to a queue, which has no printer attached to it, or the printer is off line for what ever reason, a PMV8005 message will be generated and it will list the printer name and job number and offer the choice of selecting *enter* to continue (if the printer is now on line) or else select *cancel* and the job will be deleted from the queue.



IBM 5202 Quietwriter 3

Jobid 4 on IBM52021

PMV8005: Make sure that the printer is switched on, that it is online, and that it is loaded with paper. Select ENTER to continue or CANCEL to end the job.

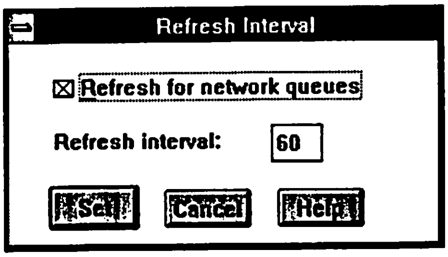
Processing will continue

8.3 Refresh

Refresh is what the system does to the Print Manager window to show the latest changes. This includes jobs that have been added to, or which have finished processing and have disappeared from the queues. Basically, it is the system updating the list of jobs and queues.

If you are using a Standard Edition system, the only option on the Refresh menu is *Refresh Now*, which will refresh your Print Manager window immediately and not wait for the next automatic update from the system. However, on a Standard Edition system, as well as, on an Extended Edition system without an active LAN connection, refresh is performed almost in real time.

If you are using an Extended Edition system on a LAN, the Refresh menu will have an additional option on it called *Refresh Interval*. Selecting the Refresh Interval option will bring up the *Refresh Interval* dialog box.



Refresh Interval

☒ Refresh for network queues

Refresh interval: 60

Figure 16. Refresh Interval on the LAN

A network queue is a printer queue, which is available to the network.

There are two things that can be done in this dialog box.

1. Set the *Refresh for network queues* on or off by selecting the check box, which will either put an "X" in, or take one out.

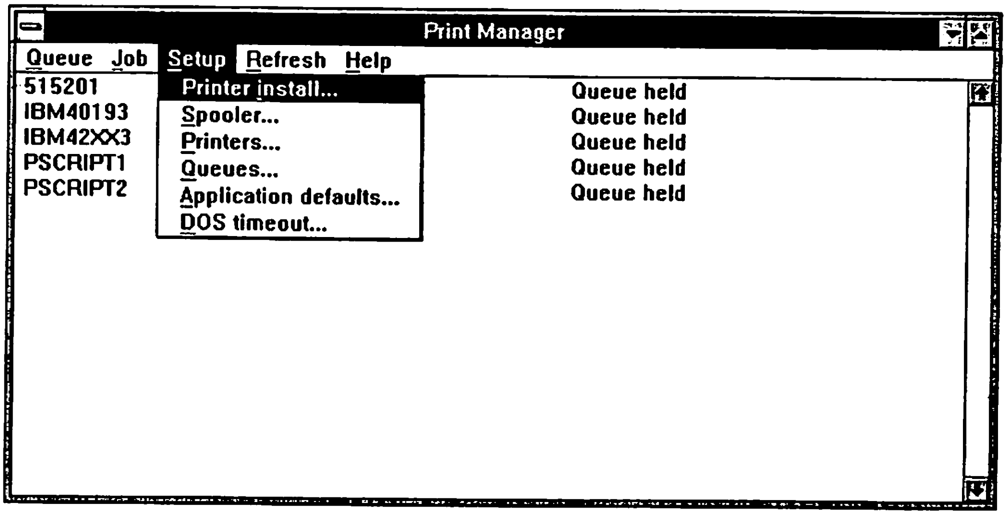
- "X" in the check box means the network refresh interval is on.
- A blank check box means the network refresh interval is off.

2. Enter the interval time (in seconds) you want between the Print Manager updates of the network queues. This interval only applies to the network queues and does not affect the local queues. Local queues are still updated in real time. If you are on a server and have its local queue shared, it is considered to be a network queue.
   - You can set the interval to be any number of seconds between 1 and 999.
   - The lower this number, the more frequently the servers will have to update status of the network queues.

Caution

By setting this option to a low number you will increase the number of times the updates have to be sent across the network, which will increase the workload on the servers as well as the requester that has the low setting on its refresh interval. If this is set to a very low interval, the server and the requester will be so busy sending and receiving updates to the Print Manager that they will not be able to do much else.

8.4 Printer Install



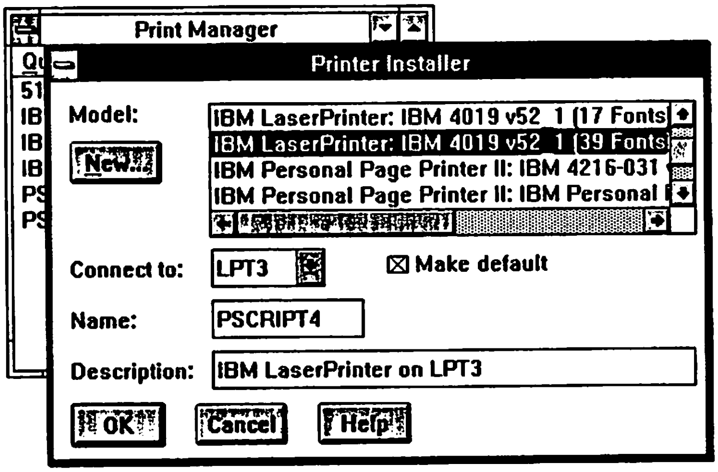
515201 Queue held
IBM40193 Queue held
IBM42XX3 Queue held
PSCRIPT1 Queue held
PSCRIPT2 Queue held

Figure 17. Print Manager: Printer Install

In OS/2 V1.2, the only way to install a printer device driver was to use the Control Panel. But, as the latest printer device drivers became more complex and had to be installed with additional DLL, help, font and other files, this method was no longer sufficient. On top of that it was just confusing.

In OS/2 V1.3, you now have a new function under Print Manager called *Printer install....* See Figure 17.

This function allows you to install a printer device driver without using the Control Panel anymore. If the printer device drivers are already copied on the hard disk, Print Manager displays a list of those device drivers available.



Model:
IBM LaserPrinter: IBM 4019 v52_1 (17 Fonts
IBM LaserPrinter: IBM 4019 v52_1 (39 Fonts
IBM Personal Page Printer II: IBM 4216-031
IBM Personal Page Printer II: IBM Personal

Connect to: LPT3 ☒ Make default

Name: PSCRIPT4

Description: IBM LaserPrinter on LPT3

Figure 18. Print Manager: Printer Installer

In Figure 18, Print Manager displays all printer device drivers available. This information can be obtained from a diskette or from a collection of printer device drivers in a subdirectory of its own.

If you want to install a new printer device driver, select the option **New...** in the Printer Installer dialog box. Print Manager will then ask for the drive letter and subdirectory, where to find any printer device drivers. The printer can be connected to *LPT1, LPT2, LPT3* and *COM1, COM2, COM3* or *NONE*. The selection for the serial ports will only be displayed if they are configured through their device driver (COM0x.SYS) in the CONFIG.SYS file. So, the user has to make sure that the port is available before installing the printer device driver.

The name is going to be the "QUEUE NAME" and some default is provided. However it can be changed to anything else. The same is true for the description field.

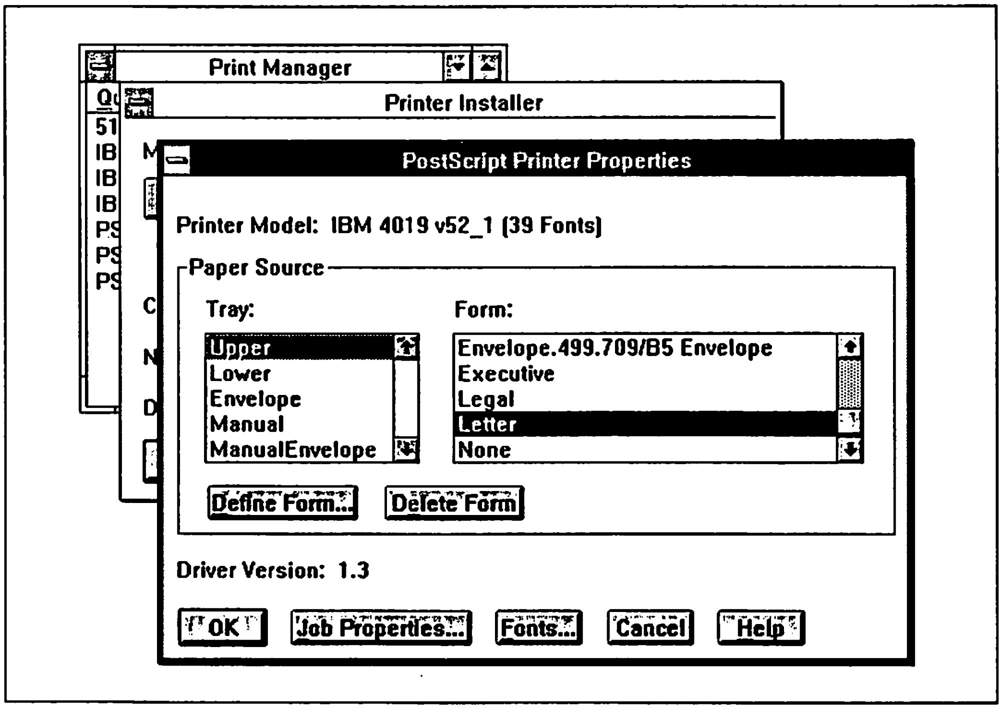
When OK is selected, Print Manager will define the printer and the queue. It will also copy all the necessary files for this particular printer device driver onto the system into their appropriate subdirectories. It will then install these files; that means it will perform all the necessary definitions and updates in the two profiles OS2.INI and OS2SYS.INI.

Note

The only way to uninstall any printer device driver is still through the *delete printer driver* option in the **Control Panel**. This will ensure that the proper updates to the profile files are performed!

The next step is to define the printer properties.

8.5 Printer Properties



Printer Model: IBM 4019 v52_1 (39 Fonts)

Paper Source

Tray:

- Upper
- Lower
- Envelope
- Manual
- ManualEnvelope

Form:

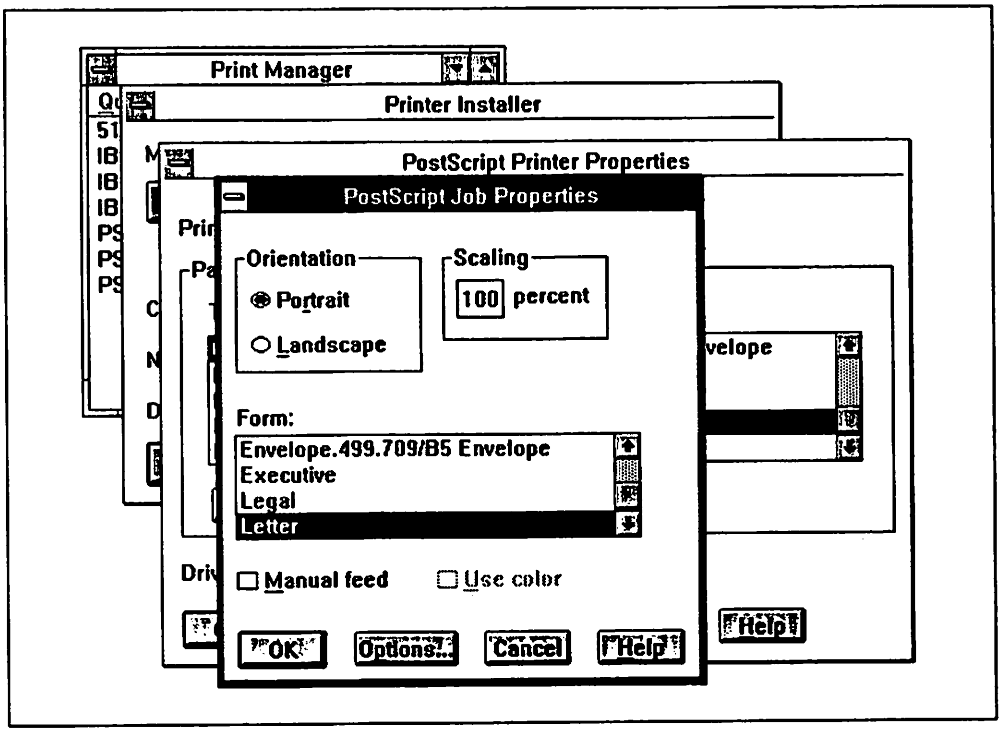
- Envelope.499.709/B5 Envelope
- Executive
- Legal
- Letter
- None

Driver Version: 1.3

Figure 19. Print Manager: PostScript Printer Properties

In Figure 19, Print Manager asks the user to define the printer properties. The Printer Properties dialog box will allow you to determine how the printer will operate. Each printer driver has its own set of printer properties. Depending on the printer or plotter that you are using, the Printer Properties describe optional parameters such as paper size, orientation, font cartridge, paper drawer to use, amount of memory installed in the printer or plotter, etc.

8.6 Job Properties



PostScript Job Properties

Orientation

● Portrait

○ Landscape

Scaling

100 percent

Form:

Envelope.499.709/B5 Envelope
Executive
Legal
Letter

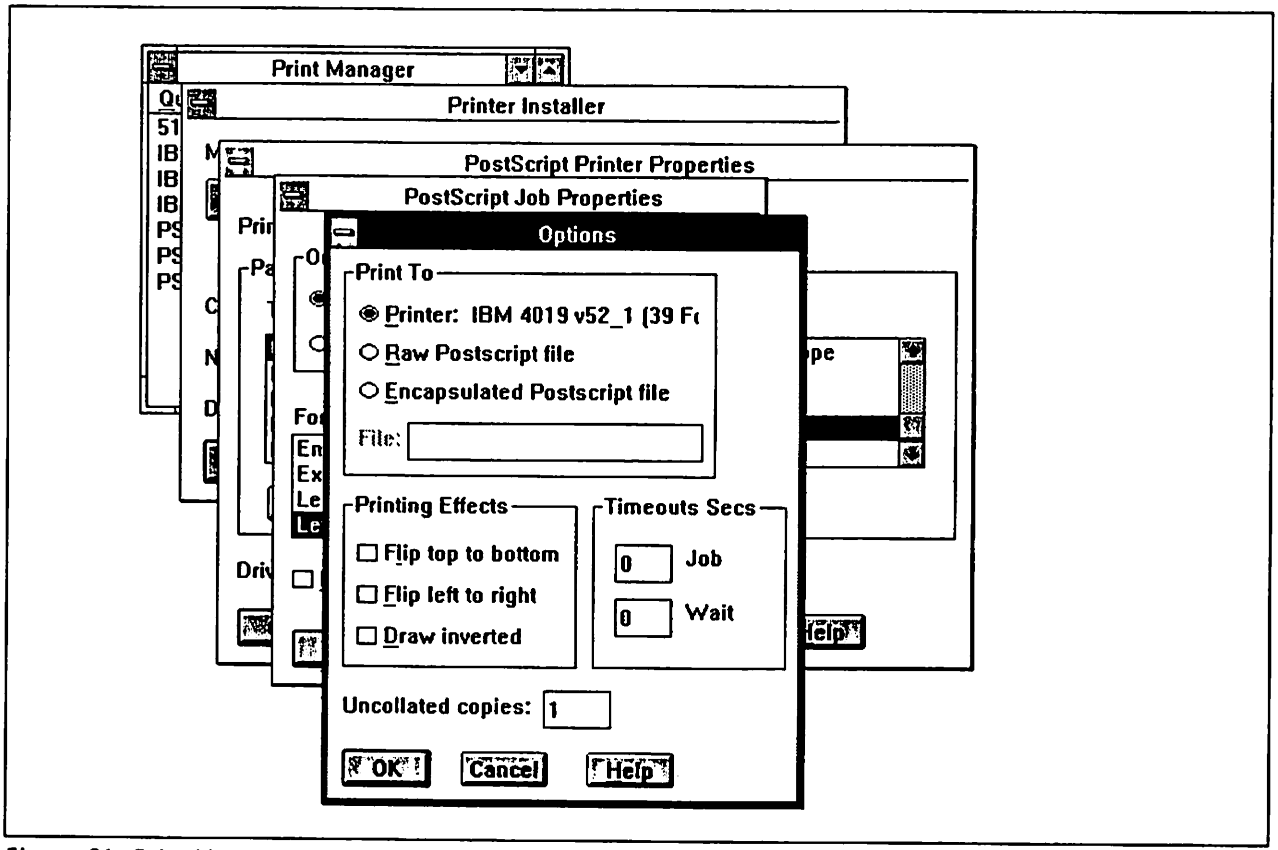
□ Manual feed

□ Use color

Figure 20. Print Manager: PostScript Job Properties

The Job Properties can either complement or override the Printer Properties. The Job Properties describe the same items as the Printer Properties. Figure 20 shows a sample of the Job Properties for a PostScript printer.

8.7 Job Properties Options



Print To

● Printer: IBM 4019 v52_1 (39 F(

○ Raw Postscript file

○ Encapsulated Postscript file

File:

Printing Effects

□ Flip top to bottom

□ Flip left to right

□ Draw inverted

Timeouts Secs

0 Job

0 Wait

Uncollated copies: 1

Figure 21. Print Manager: PostScript Job Properties Options

Some of the printer drivers may also provide more than one dialog under the printer and job properties. Figure 21 shows a sample for the PostScript printer options.

Printer Destination

- Select printer to print directly to this PostScript printer.
- Select "Raw PostScript file", if you would like to create a PostScript file, which could be sent to a service bureau for high quality typesetting, for instance.
- Select "Encapsulated PostScript file", if you would like to create an Encapsulated PostScript file, which could be imported into an application that accepts EPS format file, for instance.

Note

For "Raw PostScript file" and "Encapsulated PostScript file" you must enter a file name.

Note the differences between Raw and Encapsulated. Encapsulated means that a *"%%BoundingBox"* statement is included in the output file, which permits scaling. For example, you can upload a graphic as EPS to a host, there use the

".po" (SCRIPT) command to embed and scale the graphic any number of times in the same SCRIPT/GML document, unlike a PSEG file which is a fixed size.

For a detailed discussion about the PostScript language see also the ITSC bulletin *Printing PostScript Language.*

Printing Effects

The printing effects allows you to alter the appearance of the printed image.

- *Flip top to bottom* and the drawing is printed upside down.
- *Flip left to right* and the drawing is printed as a mirror image.
- *Draw inverted*, the drawing is printed as if it were a photographic negative.

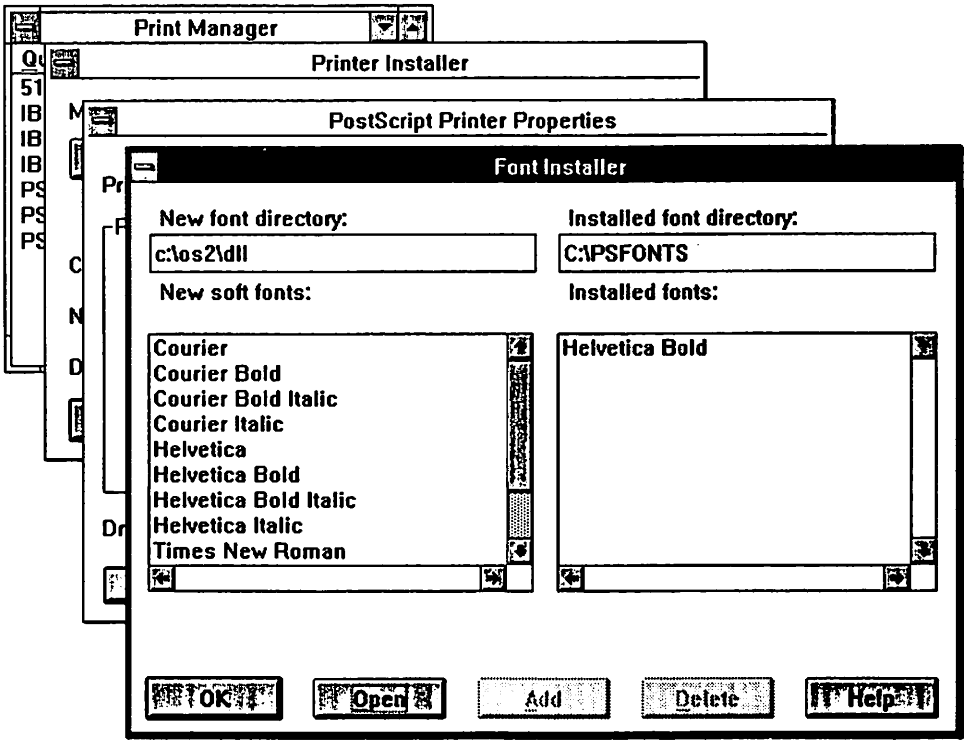
Timeout Secs

- *Job Timeout.* You should use this to set a maximum limit on how long Print Manager should wait if communication with the PostScript printer is interrupted or lost, before the print job will be aborted. A value of zero means that there is no limit and you may wait forever.
- *Wait Timeout* sets the limit on how long the printer will wait to receive the next character before aborting a print job. It may be useful to increase the wait-timeout value if your system is very busy and is slow in sending information to the printer. Zero means there is no limit and the PostScript printer may wait forever.

Uncollated Copies

You may specify how many copies of each page get printed. See Figure 21 on page 56.

8.8 Font Install



New font directory: c:\os2\dll

Installed font directory: C:\PSFONTS

New soft fonts:

Courier
Courier Bold
Courier Bold Italic
Courier Italic
Helvetica
Helvetica Bold
Helvetica Bold Italic
Helvetica Italic
Times New Roman

Installed fonts:

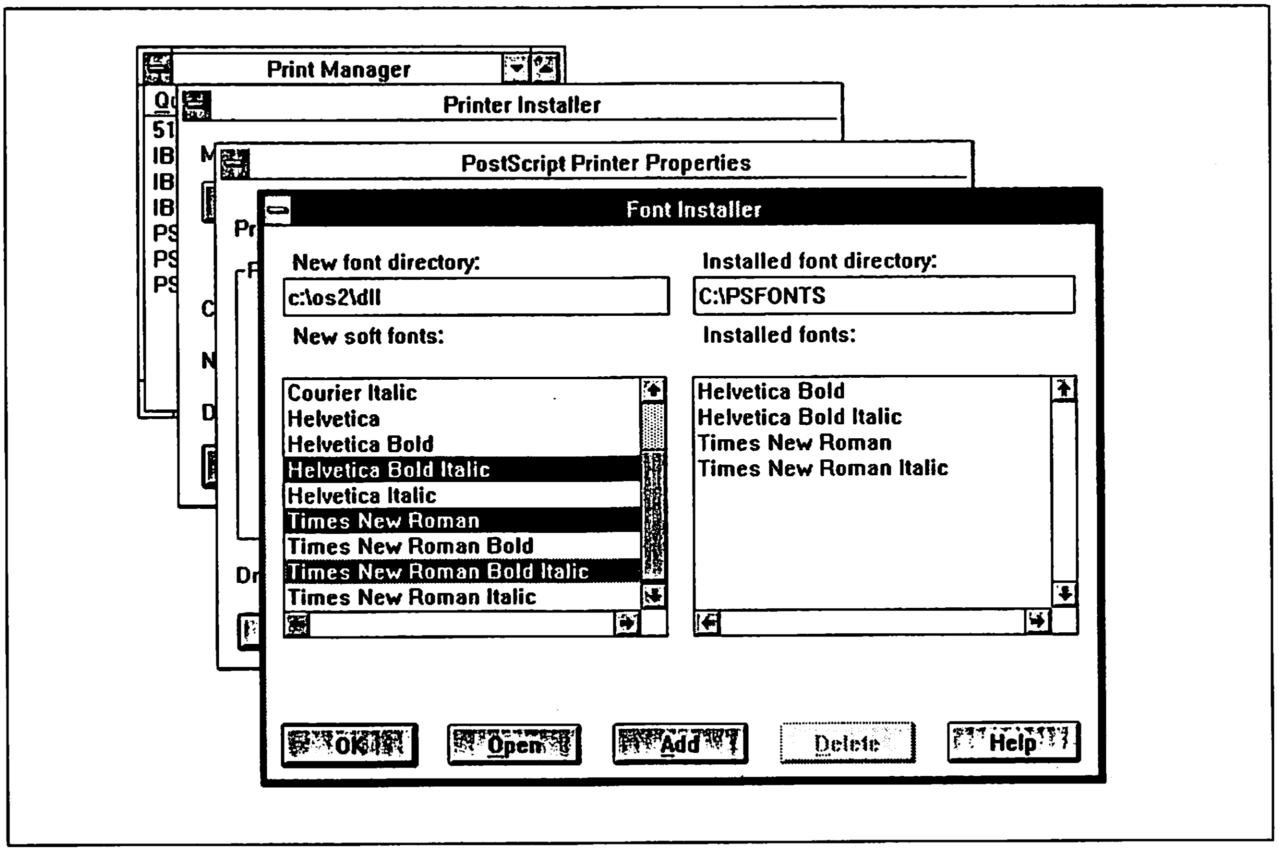
Helvetica Bold

Figure 22. Print Manager: Font Install

Depending on the printer driver, you can install fonts from the Print Manager. For example, the Type 1 fonts, which can also be used by a PostScript printer may be located in the:

```
C:\PSFONTS
```

subdirectory. If you want to install new fonts, which are not in the PSFONTS subdirectory, you can select another subdirectory where these fonts may reside.



Print Manager

Printer Installer

PostScript Printer Properties

Font Installer

New font directory: c:\os2\dll

Installed font directory: C:\PSFONTS

New soft fonts:

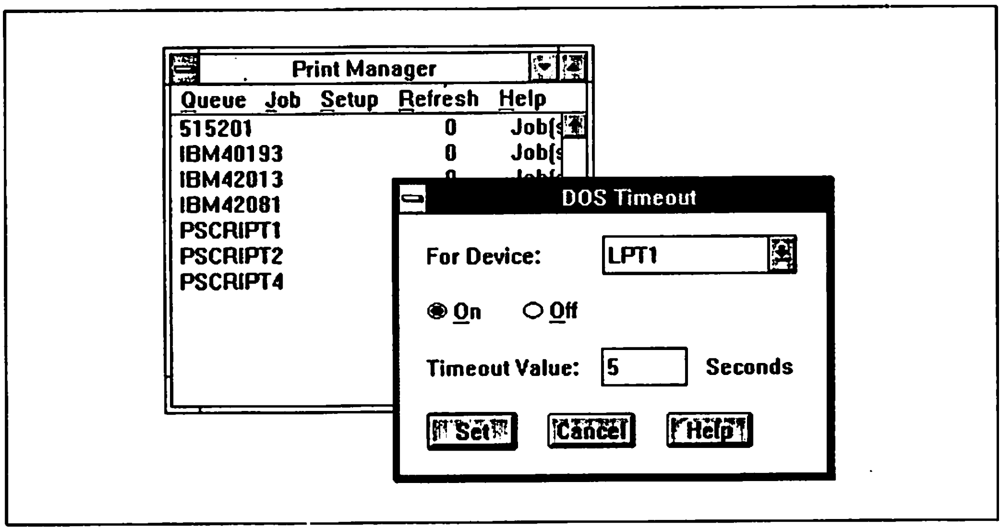
- Courier Italic
- Helvetica
- Helvetica Bold
- Helvetica Bold Italic
- Helvetica Italic
- Times New Roman
- Times New Roman Bold
- Times New Roman Bold Italic
- Times New Roman Italic

Installed fonts:

- Helvetica Bold
- Helvetica Bold Italic
- Times New Roman
- Times New Roman Italic

Figure 23. Print Manager: Font Installer

8.9 DOS Timeout



Print Manager

```
515201       0   Job(
IBM40193     0   Job(
IBM42013
IBM42081
PSCRIPT1
PSCRIPT2
PSCRIPT4
```

DOS Timeout

For Device: LPT1

● On ○ Off

Timeout Value: 5 Seconds

Figure 24. Print Manager: DOS Timeout.

OS/2 has its own print spooler to control and manage information going to the printer from different sources. This allows OS/2 programs to send output to a printer while your DOS program is printing. Under certain circumstances, sending print jobs from a DOS application to the spooler may not come to an end and the spool job never gets closed. To force the close of a spooled (DOS) printer job it may be necessary to press the Ctrl, Alt and PrtSc keys together. Another way to end a DOS program print job is to set the **DOS Timeout** parameter in Print Manager. This will close the spool job after a specified amount of inactivity on this port and Print Manager will start to print the job. You can specify a time-out value of 1 to 65,535 seconds. A value of 0 means that there is no timeout at all. This parameter can be specified differently for each available port.

8.10 Use of Extended Attributes

Since Version 1.2 of OS/2, a file system can save additional information about files and directories. This information is called extended attributes. Through extended attributes, a program can attach information to a file system object (files or directories) describing the object to another program, to the operating system, or to the file system driver (FSD) managing that object.

With Version 1.3, some of the more complex printer device drivers are associated with some extended attributes. The printer installation procedures may heavily rely on this information.

Note

Don't copy these printer drivers under DOS, for example. You would loose those extended attributes and therefore some module in the print subsystem may not function properly!

The ITSC bulletin *OS/2 V1.3 Volume 2: Print Subsystem* has a much more detailed discussion about the whole print subsystem of OS/2. This also includes some sample code, which shows how to program print functions under OS/2.

Chapter 9. Fonts Usage in OS/2 V1.3

Nowadays, all microcomputers use graphics screens and printers. Font support in their operating systems is one of the major parts. In OS/2 V1.3, fonts are in different places and in different forms. This section will try to give an overview of what fonts are used in OS/2 V1.3 and where to find them. Some of them are display-only fonts, others are printer-only fonts and some can be used for both.

Fonts are classified in three major categories depending on their usage:

1. VIO full-screen fonts: these are the basic fonts used by the video-controller.
2. PM fonts: here we will find all the fonts we can see under the graphical user interface of OS/2, the Presentation Manager.
3. Downloadable printer fonts: today's printers get more and more "intelligent" and are able to support downloadable fonts, and a variety of font cartridges. Afterwards, we will explain the way OS/2 V1.3 supports these fonts.

Table 13 on page 176 summarizes the font usage in OS/2. The following paragraphs provide more details.

9.1 VGA Hardware Video Display Fonts

Usage: At power-on in DOS or OS/2 VIO full-screen sessions.,
Font Type: Bitmap.
Char. Size: 12 lines of 40 characters to 60 lines of 80 characters in VGA.
Font Size: 4 to 18 KB per font.
Source: Video controller.
Installed by: Hardware.
Deleted by: N/A.
Location: Video controller.

9.2 Presentation Manager Fonts

9.2.1 VGA VIO Window Fonts

Usage: OS/2 VIO sessions in a window.
Font Type: Bitmap.
Char. Size: 8x8,10x8 ,12x8 ,14x8 ,16x8 ,18x8 pels.
Font Size: The VGA DISPLAY.DLL is 135 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation.
Deleted by: N/A.
Location: C:\OS2\DLL\DISPLAY.DLL.
See Chapter 10, "More Fonts for OS/2 V1.3 VIO Windows" on page 67.

9.2.2 8514 VIO Window Fonts

Usage: OS/2 VIO sessions in a window.
Font Type: Bitmap.
Char. Size: 8x8, 12x7, 12x8, 14x6, 14x8, 15x7, 16x12, 17x8, 20x12, 22x12, 25x7, 30x12 pels.
Font Size: The 8514 DISPLAY.DLL is 284 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation.
Deleted by: N/A.
Location: C:\OS2\DLL\DISPLAY.DLL.

9.2.3 XGA VIO Window Fonts

Usage: OS/2 VIO sessions in a window.
Font Type: Bitmap.
Char. Size: 8x8, 10x6, 10x8, 12x8, 14x6, 14x8, 15x7, 16x8, 16x12, 18x8, 18x20, 20x12, 22x12, 25x7, 30x12 pels.
Font Size: The XGA DISPLAY.DLL is 468 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation.
Deleted by: N/A.
Location: C:\OS2\DLL\DISPLAY.DLL.
See section 11.5.5.7, "Type 1 Fonts Versus XGA Antialiased Fonts" on page 108 and Chapter 12, "Antialiased Fonts" on page 113.

9.2.4 System Font

Usage: Presentation Manager title bars, pull-downs, menus, dialog boxes, etc.
Font Type: Bitmap.
Char. Size: 10 points.
Font Size: In DISPLAY.DLL.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation.
Deleted by: N/A.
Location: C:\OS2\DLL\DISPLAY.DLL.
Remarks: With the XGA device driver, the system fonts can use the antialiased fonts Roman, Swiss or Courier. The program SYSFONT.EXE which allows to do that, can be found on the XGA device support diskette for OS/2.
See Chapter 12, "Antialiased Fonts" on page 113.

9.2.5 OS/2 Bitmap Fonts

Usage: Presentation Manager applications.
Font Type: Bitmap.
Char. Size: 8, 10, 12, 14, 18, 24 points.
Face Names:
System Monospaced, Courier, Helv, Tmn Rmn.
Font Size: System Monospaced (10,12 pt) 20 KB
Courier (8,10,12,14): 120 KB
Helvetica: 360 KB
Times Roman: 330 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation program.

Deleted by: Control Panel.
Location: C:\OS2\DLL\COURIER.FON
C:\OS2\DLL\HELV.FON
C:\OS2\DLL\TIMES.FON
C:\OS2\DLL\SYSMONO.FON.
See section 11.5.4, "Outline Versus Bitmap" on page 98.

9.2.6 OS/2 Outline Fonts

Usage: Presentation Manager applications.
Font Type: Outline.
Char. Size: Any.
Face Names:
Courier, Helv, Tmn Rmn.
Font Size: Courier 119 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation program.
Deleted by: Control Panel.
Location: Built into the PM graphics engine (PMGRE.DLL).
Remarks: In OS/2 V1.3, there is only one OS/2 outline font available. If the Type 1 fonts are installed, the Type 1 Courier font replaces this standard OS/2 outline font.
See section 11.5.5.6, "Type 1 Fonts Versus OS/2 V1.2 Outline Fonts" on page 108.

9.2.7 IBM Core Fonts (Type 1 Fonts)

Usage: Presentation Manager applications.
Font Type: Outline.
Char. Size: Any.
Face Names:
Symbol, Courier, Helvetica, Times New Roman.
Font Size: Symbol+Courier 160 KB, Helvetica 120 KB, Times New Roman 130 KB.
Source: OS/2 installation diskettes.
Installed by: OS/2 installation program.
Deleted by: Control Panel.
Location: C:\OS2\DLL\COURIER.PSF
C:\OS2\DLL\HELVETIC.PSF
C:\OS2\DLL\TIMESNRM.PSF.
Remarks: See section 11.4.1, "ATM Standard IBM Core Fonts" on page 87.

9.2.8 Other Type 1 Fonts

Usage: Presentation Manager applications.
Font Type: Outline.
Char. Size: Any.
Face Names:
About 1000 (end of 1990).
Font Size: About 50 KB per font.
Source: Adobe and other suppliers.
Installed by: Control Panel.
Deleted by: Control Panel.
Location: C:\OS2\DLL\*.PFB.
C:\OS2\DLL\*.AFM.
(The location can actually be selected by the users).

Remarks: The PFB file is the font description and the .AFM file the font metrics.
See section 11.2.1, "Type 1 Fonts" on page 76.

9.2.9 XGA Antialiased Fonts

Usage: Presentation Manager applications and system fonts.
Font Type: Bitmap.
Char. Size: 8, 10, 12, 14, 18, 24 point.
Face Names:
Swiss AA (8,10,12,14,18,24 point)
Swiss AA Italic (8,10,12,14,24 point)
Swiss AA Bold (8,10,12,14,24 point)
Roman AA (8,10,12,14,18,24 point)
Roman AA Italic (8,10,12,14,24 point)
Roman AA Bold (8,10,12,14,24 point)
Courier AA (8,10,12,14 point).
Font Size: 8 point 38 KB
10 point 48 KB
12 point 61 KB
14 point 76 KB
18 point 118 KB
24 point 190 KB.
Source: XGA device support diskette.
Installed by: XGA device support diskette (AAINSTAL procedure).
Deleted by: Control Panel.
Location: C:\OS2\DDFONTS\*.FNT.
Remarks: See Chapter 12, "Antialiased Fonts" on page 113.

9.3 Downloadable Printer Fonts

9.3.1 IBM 4019 Printer Fonts

Font Type: Bitmap.
Char. Size: 6, 8, 10, 12, 14, 18, 24, 30 point.
Font Size: 100 KB per font.
Source: IBM and other suppliers.
Installed by: Print Manager + IBM4019 driver.
Deleted by: Print Manager + IBM4019 driver.
Location: C:\4019FONT\*.DLF
C:\4019FONT\*.FMF.
Remarks: These software fonts, consist of two files. The .DLF file is the font description which is actually downloaded to the printer. The .FMF file contains the font metrics, which describes the font to Presentation Manager.

The IBM 4019 can also use 41 font cards, two at one time. During the installation, of those font cartriges, the printer device drivers will read the .PMF file and unpack its content. It will create two small files per font and store them in a separate directory, such as C:\4019FONT.
The same mechanismn exists for the IBM 5202 and IBM 5204 printers and their font cartridges, using the new IBM52XX printer device driver.

See also *OS/2 V1.3 Volume 2: Print Subsystem* for a much more detailed discussion of printer fonts and the IBM 4019 Laser Printer.

9.3.2 PostScript Printers Fonts

Font Type: Outline.

Char. Size: Any.

Face Names:

About 1000 (end of 1990).

Font Size: 100 KB per font.

Source: Adobe and other suppliers.

Installed by: Print Manager + PostScript driver.

Deleted by: Print Manager + PostScript driver.

Location: C:\PSFONTS\*.PFA
C:\PSFONTS\*.PFM.

Remarks: The .PFA file is the font description and the .PFM file the font metrics.
See section 11.4.2, "PostScript Downloadable Fonts" on page 88.

9.3.3 HP LaserJet Soft Fonts

Char. Size: 4 to 200 point.

Font Type: Bitmap.

Font Size: 20 to 300 KB per Font.

Source: Hewlett Packard and others.

Installed by: Print Manager + HP Laser driver.

Deleted by: Print Manager + HP Laser driver.

Location: C:\OS2\DLL\HP\PCL\*.SFP
C:\OS2\DLL\HP\PCL\*.FNT

Remarks: The .SFP file is the font description and the .FNT the font metric. In HP LaserJet emulation, the IBM 4216 and the IBM 4019 are able to print these HP LaserJet soft fonts while running in HP LaserJet emulation.

See also Table 13 on page 176 for a comparison of these fonts.

Chapter 10. More Fonts for OS/2 V1.3 VIO Windows

OS/2 V1.3 provides a choice of six fonts for OS/2 applications in a VIO window with the VGA display device driver. The 8514 driver gives you a choice of twelve fonts and the XGA driver fourteen. With V1.2, the choice was only between a small font and a large font. The new fonts start from an 8 x 8 font cell, which lets us see 57 lines of text, to an 18 X 8 font cell giving 25 lines in a VIO window.

The font selection is chosen using the "Font Size..." option from the system menu of an OS/2 VIO window (Figure 25).

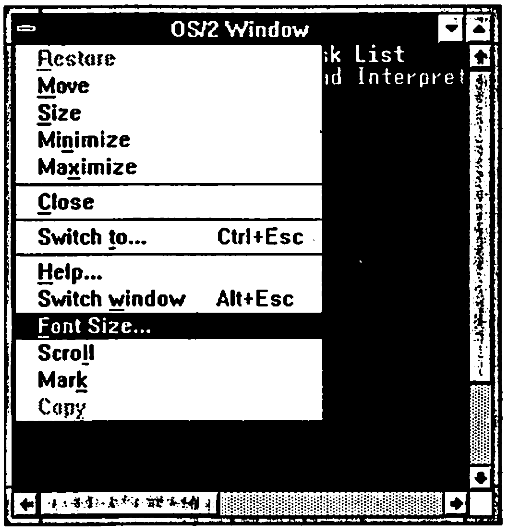
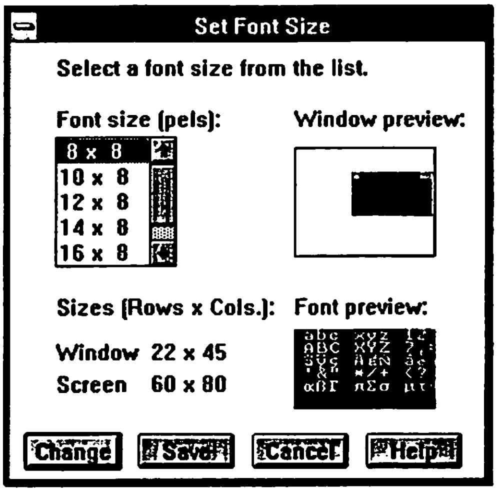


Figure 25. Font Size Selection

The font size option gives access to the "Set Font Size" dialog box (Figure 26 and Figure 27 on page 68).



Set Font Size

Select a font size from the list.

Font size (pels):

8 x 8
10 x 8
12 x 8
14 x 8
16 x 8

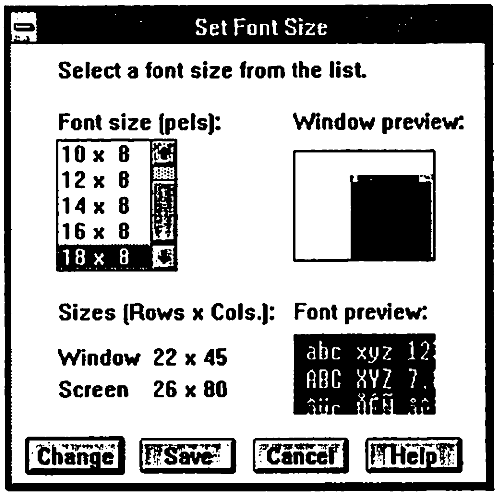
Window preview:

Sizes (Rows x Cols.):

Window 22 x 45
Screen 60 x 80

Font preview:

Figure 26. Font and Window Preview for the 8 X 8 Font



Set Font Size

Select a font size from the list.

Font size (pels):

10 x 8
12 x 8
14 x 8
16 x 8
18 x 8

Window preview:

Sizes (Rows x Cols.):

Window 22 x 45
Screen 26 x 80

Font preview:

Figure 27. Font and Window Preview for the 18 X 8 Font

This dialog box allows you to change the font size for text in OS/2 VIO windows. The font sizes are shown in the format: height x width in pixels, in ascending order of height.

The **Font preview** window displays some sample text in the font size selected.

The **Window preview** shows the relative area of the screen that the current OS/2 VIO window would occupy to display the same amount of text in the selected font size.

The selected font size is used to show:

- The number of rows and columns in the current VIO window
- The maximum number of rows and columns that will fit the full screen.

The OS/2 command MODE can be used to specify the number of rows you would like to see in an OS/2 window. For example:

```
MODE CO80,row
```

The maximum valid specification for row is 102 for a windowed OS/2 session. Consult the IPF help function (F1) any time for more information.

To change the font size for the current OS/2 VIO window, click on (or select) the font size. Then click on Change (or select it and press Enter).

To change the default font size, click on (or select) the font size. Then click on Save (or select it and press Enter).

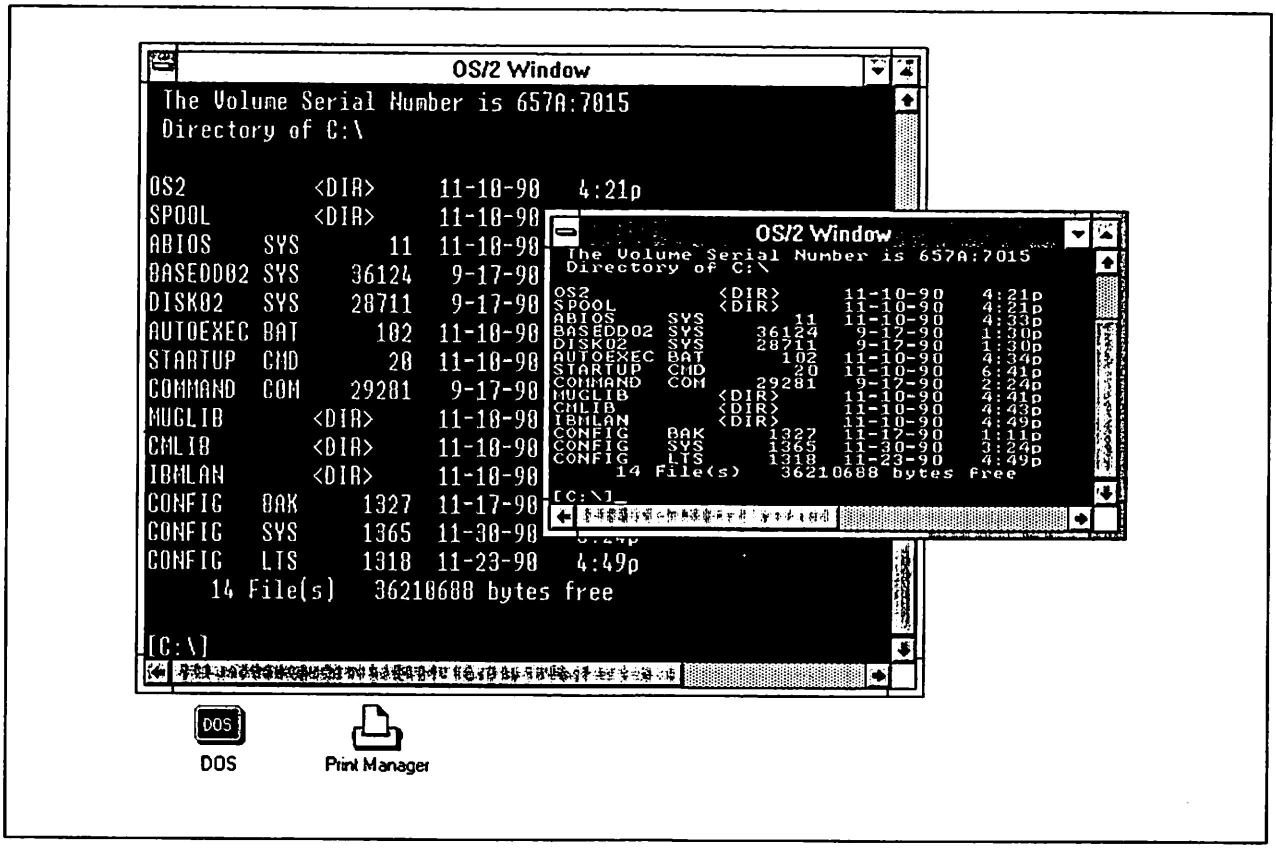
Pushbutton actions:

Change Changes the font size for this window only. The default font size is not changed.

Save Changes the default font size. The font size is changed in this window and in all new OS/2 VIO windows. The font size in existing windows is not changed.

This will save the new default size in the OS2.INI file.

Cancel Removes the pop-up window. No information is saved.



```
The Volume Serial Number is 657A:7015
Directory of C:\

OS2          <DIR>      11-10-90   4:21p
SPOOL        <DIR>      11-10-90
ABIOS    SYS         11 11-10-90
BASEDD02 SYS      36124  9-17-90
DISK02   SYS      28711  9-17-90
AUTOEXEC BAT        102 11-10-90
STARTUP  CMD         20 11-10-90
COMMAND  COM      29281  9-17-90
MUGLIB       <DIR>      11-10-90
CMLIB        <DIR>      11-10-90
IBMLAN       <DIR>      11-10-90
CONFIG   BAK       1327 11-17-90
CONFIG   SYS       1365 11-30-90
CONFIG   LTS       1318 11-23-90   4:49p
     14 File(s)   36210688 bytes free

[C:\]
```

```
 The Volume Serial Number is 657A:7015
 Directory of C:\

OS2           <DIR>     11-10-90   4:21p
SPOOL         <DIR>     11-10-90   4:21p
ABIOS    SYS         11 11-10-90   4:33p
BASEDD02 SYS      36124  9-17-90   1:30p
DISK02   SYS      28711  9-17-90   1:30p
AUTOEXEC BAT        102 11-10-90   4:34p
STARTUP  CMD         20 11-10-90   6:41p
COMMAND  COM      29281  9-17-90   2:24p
MUGLIB        <DIR>     11-10-90   4:41p
CMLIB         <DIR>     11-10-90   4:43p
IBMLAN        <DIR>     11-10-90   4:49p
CONFIG   BAK       1327 11-17-90   1:11p
CONFIG   SYS       1365 11-30-90   3:24p
CONFIG   LTS       1318 11-23-90   4:49p
     14 File(s)   36210688 bytes free

[C:\]_
```

Figure 28. OS/2 VIO Windows with the 18 X 8 and the 8 X 8 Font

Chapter 11. Adobe Type Manager (ATM)

11.1 IBM's Direction for Fonts

On March 5th, 1990, at the Seybold Desktop Publishing Conference, IBM announced that it would implement Adobe[22] Systems font-rendering technology and Type 1 font format across its SAA systems. This represents part of a unified font strategy across all SAA platforms.

The first stage of the implementation of this strategy has now been reached with the licensing of Adobe Type Manager (ATM[23]), and the inclusion of ATM and 3 new Type 1 fonts in OS/2 V1.3, plus the ability to support the thousands of Type 1 fonts already available.

IBM's font strategy is derived from the need of many customers to be able to print a document on a different system than the one used to create it.

This strategy can be summarized as:

1. Print anywhere via SAA core font standardization.
2. Produce the capability to give the users a real WYSIWYG (What You See Is What You Get).
3. Support today's best font technology.
4. Be able to support any new fonts technology.

IBM Core Font Standardization

Type 1 fonts technology will be implemented across SAA compliant systems. The use of Type 1 fonts in OS/2 V1.3 is consistent with IBM's SAA direction for fonts, helping achieve greater consistency in the use of fonts across SAA platforms.

IBM Core Fonts defined for SAA are those that are implemented in OS/2 V1.3.

A Real WYSIWYG (What You See Is What You Get)

As the same algorithm and font definition is used to rasterize text for displays or printers, we can expect no difference between what is on the screen and what is printed.

Today's Best Font Technology

Type 1 fonts are widely used in all IBM computing environments. IBM support for Type 1 fonts is not a technology-driven decision but was made in response to overwhelming requests from our customers.

[22] Adobe is a trademark of Adobe Systems, Inc.

[23] ATM is a registered trademark of Adobe Systems, Inc.

Support any New Fonts Technology

OS/2 V1.3 includes an Intelligent Font Interface(IFI), which allows support of additional font rendering technologies as they emerge.

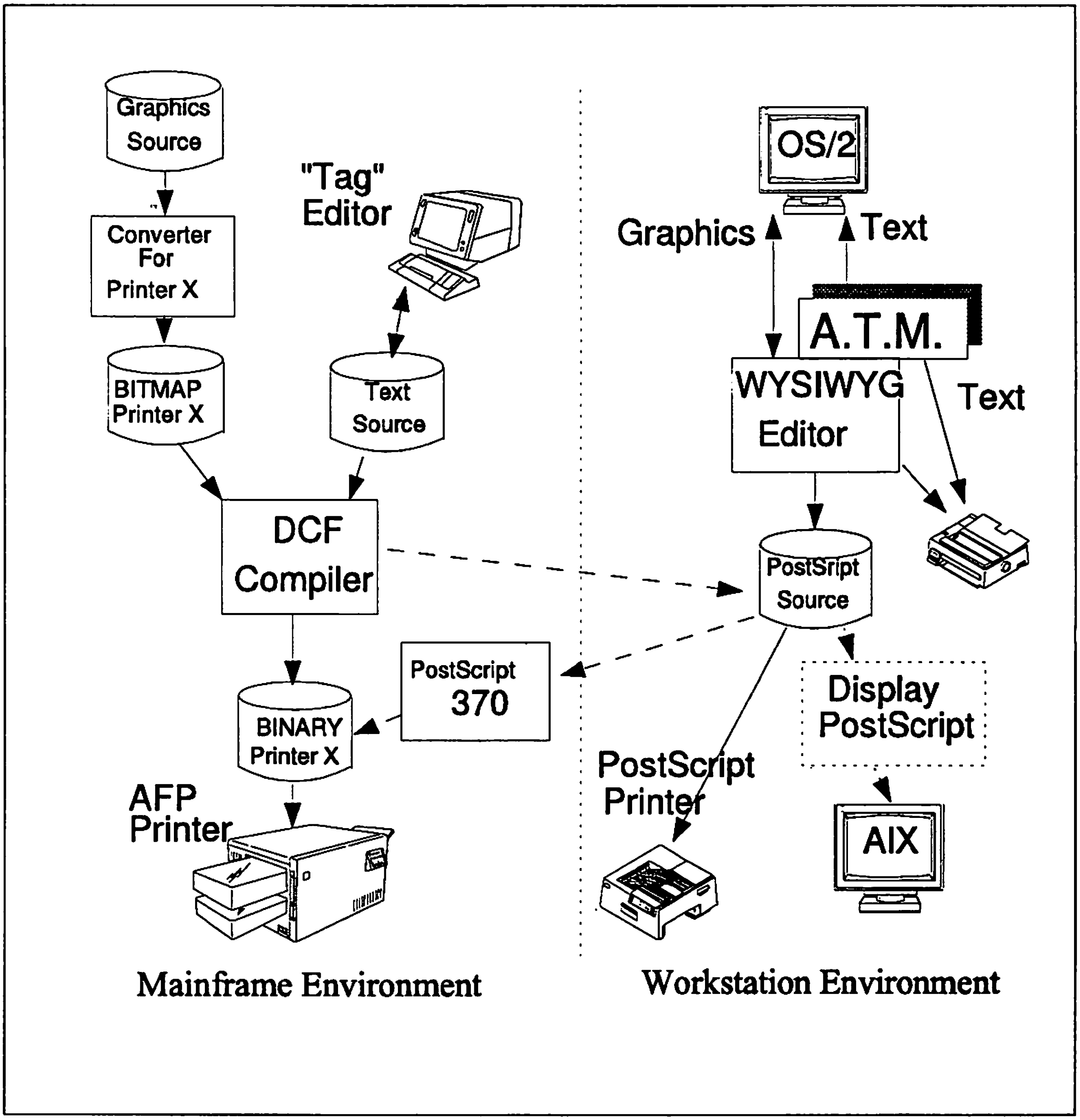


Figure 29. ATM, PostScript and DCF. Adobe Type Managers Place in the IBM Composition Facilities

11.1.1 Why Adobe?

It is interesting to examine the reasons why IBM chose Adobe's technology instead of others that were available. Here are some of the reasons why:

Customer Demand

In supporting Type 1, IBM is meeting the needs of its customers, many of whom already have a large investment in Adobe technology: over $4 billion, according to Lee Reiswig, IBM's VP of Software Strategy. OS/2's support of Type 1 will protect and add value to this investment.

Mature Technology

Adobe has been producing digital fonts since 1984; not only licensing fonts from font foundries like Linotype and ITC, but also designing their own (the "Adobe Originals") including well-known typefaces like Adobe Garamond.[24] Adobe has an excellent record of continuing to improve their technology. For example, in June 1990, Adobe announced PostScript Level 2 - a major revision of the PostScript language and Adobe's PostScript interpreter; it incorporates composite fonts, features of Display PostScript, and device independent color, as well as performance improvements, better memory management and various functional extensions for improved halftoning, forms and pattern caching. Adobe has stated that PostScript Level 2 products will be compatible with the installed base of existing PostScript products.

Thousands of Type 1 Fonts Already Available

And these fonts come from a wide variety of vendors, not just Adobe (see section 11.2.3, "Other Suppliers of Type 1 Fonts" on page 78). In fact, the Adobe Type Library is claimed to be the largest collection of typefaces in the same format ever assembled in the history of the printing and publishing industry (expected to be 1000+ by the end of 1990). OS/2's support of this format opens up this vast library to the user of Presentation Manager applications, giving the user more choices than ever before.

An Emerging ISO Standard

Type 1 is already a de facto standard in the font arena, but is also beginning to be recognized as an ISO standard. The ISO font standard committee recently voted for Type 1 as the model for representing glyphs, or font shapes, in ISO 9541.

Transportable

Type 1 fonts are already widely used across a variety of environments, from PC to mainframe, all linked by their common use of Type 1 fonts.

[24] Adobe Garamond is a trademark of Adobe Systems Inc.

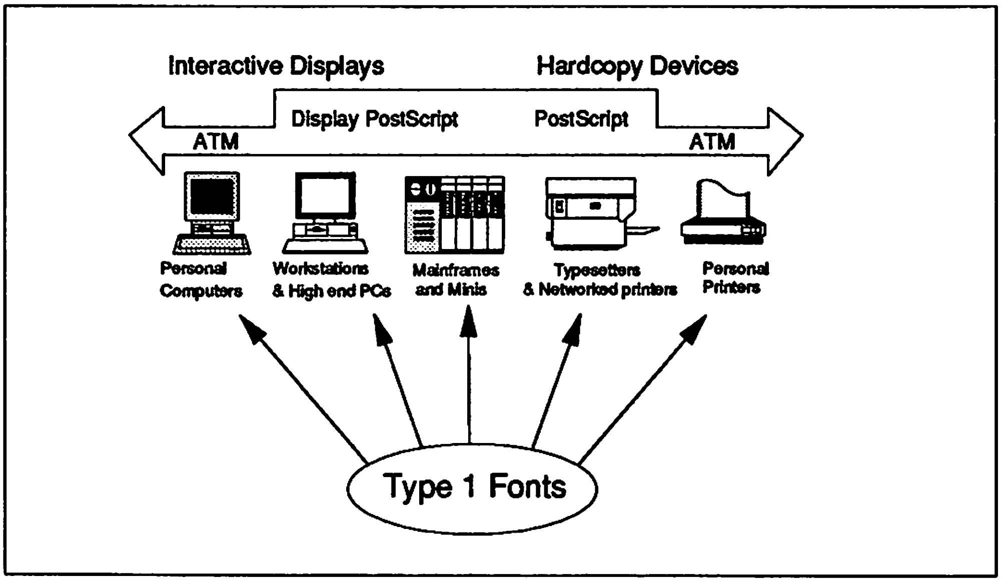


Figure 30. Adobe PostScript/Type 1 Usage Across Computing Environments

This gives users the potential to share documents containing Type 1 fonts across systems, and use different devices for document creation, drafting and printing.

11.1.2 The Alternatives

Other vendors besides Adobe have announced outline font technologies. The principle behind most of them is similar: they all describe fonts in terms of mathematical outlines, and provide hinting mechanisms to improve quality at low resolution. There is no proven inherent performance advantage in any one technology.

Perhaps the biggest differences are:

1. **Availability** - there are hundreds of Type 1 fonts already available. This is partly owing to Adobe's success in providing tools for typeface vendors to convert their fonts to Type 1 format, which greatly speeds up the process of adding hints.
2. **Device Independence/Transportability** - Type 1 fonts are available across a number of platforms and devices.

Other vendors of intelligent fonts include Apple[25] and Microsoft's[26] TrueType (previously known as "Royal"), Agfa-Compugraphic's Intellifont (used in Hewlett-Packard PCL-5 printers), and Bitstream's Fontware and Speedo.

Although Apple's TrueType is well known (at least in the computer industry) among the alternative font rendering technologies, to date it does not exist in any available commercial product, and there are no TrueType fonts yet available from Apple or from any other font foundries. True Type fonts are expected to ship in the initial form in Apple's System 7.0.

[25] Apple is a registered trademark of Apple Computer, Inc.

[26] Microsoft is a trademark of Microsoft Corporation.

The principal technical difference between TrueType and Type 1 is that the Apple approach places more emphasis on the hinting code within the font and therefore less intelligence in the rasterizer. For more information, read the August 1990 issue of *Byte* page 289, which compares the two technologies in more depth. See also section 11.6.2, "Intelligent Font Interface and Microsoft's TrueType" on page 111.

11.2 What Is the Adobe Type Manager for OS/2 V1.3?

OS/2 V1.3 now includes the Adobe Type Manager (ATM), providing high-quality typographic text to all OS/2 V1.3 PM-supported printers and displays, in a variety of styles and sizes. OS/2 V1.3 also provides support for the Type 1 fonts format, providing WYSIWYG for PM applications, and the ability to install and use any of the hundreds of Type 1 fonts, compatible with the PostScript Page Description Language, which is widely used at many customer sites.

Thirteen Type 1 fonts are included in OS/2 V1.3 in four font groups (Times New, Helvetica,[27] Courier and Symbol), replacing the outline versions of these fonts from previous releases, and adding one new font family (symbol). This provides a satisfactory basic working set, to which extra fonts can be added with ease.

Users now have a wider choice of fonts, and can display and print them even on lower-cost devices. For example, PostScript-quality fonts can now be printed on non-PostScript devices such as the IBM 4019, without the need to add the additional PostScript interpreter option. Even an IBM 4201 or an IBM 5152 printer can print Type 1 fonts, but of course, not with the same quality as laser printers. OS/2 V1.3 includes the Adobe Type Manager as a standard feature. This gives OS/2 V1.3 the ability to generate high-quality screen and printer fonts. The fonts that can be used by the ATM are the same fonts that PostScript printers recognize. Fonts currently available that are intended for downloading to PostScript printers can be installed in OS/2 V1.3 using the Control Panel and can be used on the screen. These same fonts can be "installed" by the PostScript device driver for downloading to PostScript printers. There are over 1000 different typeface styles available today in the right format: The "Type 1 font" format.

The ATM fonts don't need a PostScript printer to print. With an HP LaserJet or a Proprinter we still get excellent results, though slower performance, as the fonts will be rasterized in the PS/2 rather than the printer.

Tests with Lotus 1-2-3/G[28] , Microsoft Excel, Aldus PageMaker, and Describe indicated that all of these applications were able to display and print the new fonts, and include them in their font choice list, as soon as additional fonts were installed from the Control Panel. The same applies to two OS/2 PM applications included with OS/2, the File Manager and System Editor. However, the latter do not provide any special functions to also print the selected Type 1 fonts. They will only print their data, using the system font.

[27] Helvetica is a registered trademark of Linotype Company.

[28] Lotus is a trademark of the Lotus Development Corporation.

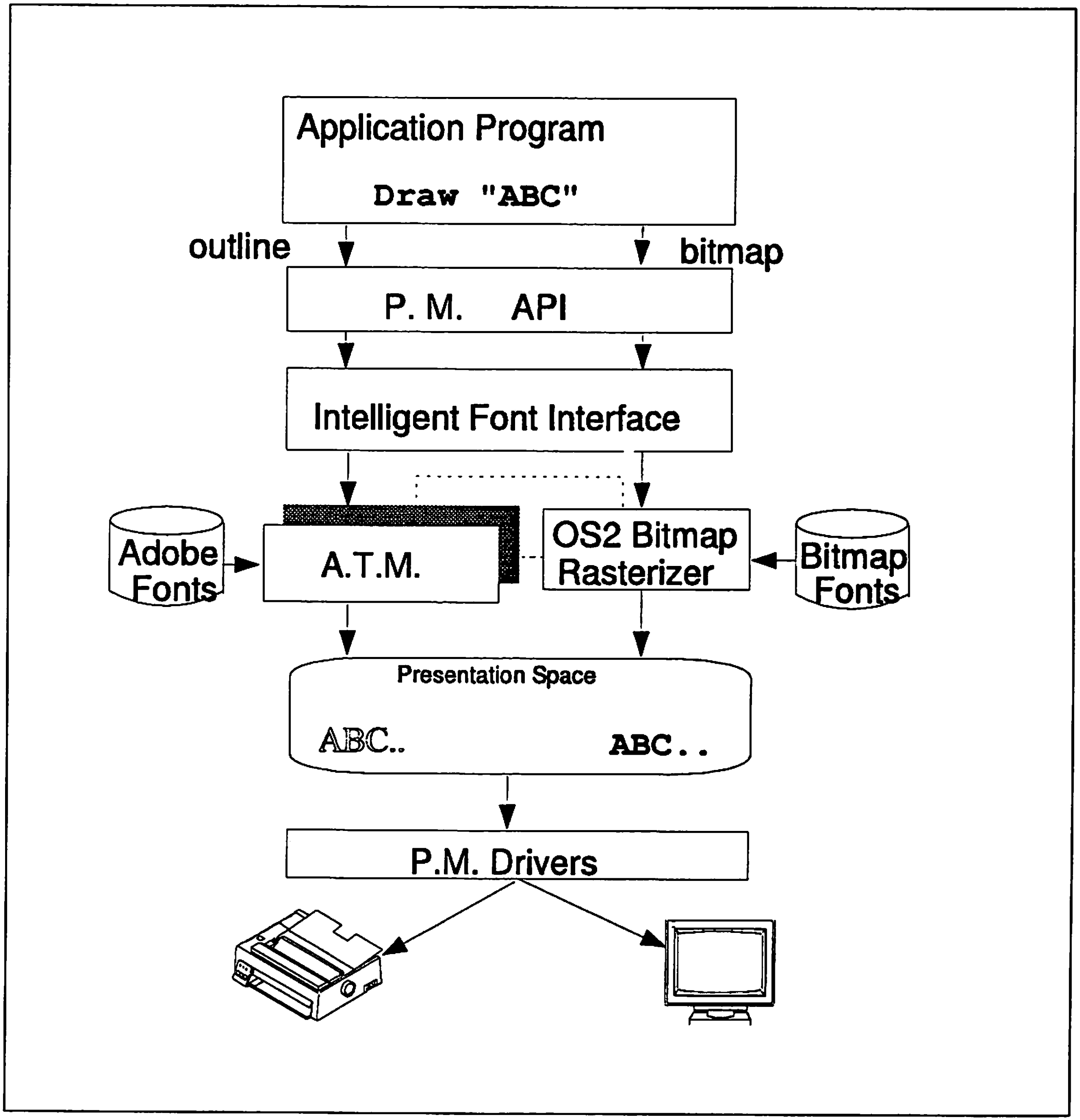


Figure 31. ATM in the Presentation Manager Architecture. Adobe Type Manager is a new font rasterizer for PM. The Intelligent Font Interface has been added to support any new font technology.

11.2.1 Type 1 Fonts

There are two main types of fonts: bitmap and outline. Bitmap fonts were used first because they are easy to define, give an excellent quality, and don't need a lot of processing time to be displayed. In the first generation of microcomputers, fonts were hard-wired in the video controller chips and didn't need any CPU processing.

As the graphics screens and printers became cheaper, bitmap fonts became more difficult to handle, because they require a complete set of redefinitions for each size, attribute (bold, italic, etc.) and device resolution. We needed a new font technology: outline fonts appeared.

In outline fonts, each letter shape, called a glyph, is defined as a series of graphics orders for a particular typeface. Type 1 fonts are defined on a 1000 x 1000 matrix by a series of straight lines and Bezier curves in PostScript language. It became easier then to create new letter shapes from the base shapes for a particular resolution, attribute and size.

What Makes Type 1 Fonts Different?

It is less difficult to re-size and transform an outline font than a bitmap font. The last process of sizing a font is to create the corresponding bitmap for each character (screens and printers are always bitmap devices). When you create a small font for a low resolution device you finally have to choose to put one or two pixels, or to put or not to put a pixel somewhere. This makes the Type 1 fonts different. Adobe introduced (and protected) in their fonts, some "hints" that inform the rasterizer where to put these pixels. Therefore, at small sizes, Type 1 fonts look better than other outline fonts, which is one of the reasons Type 1 fonts became a de-facto standard.

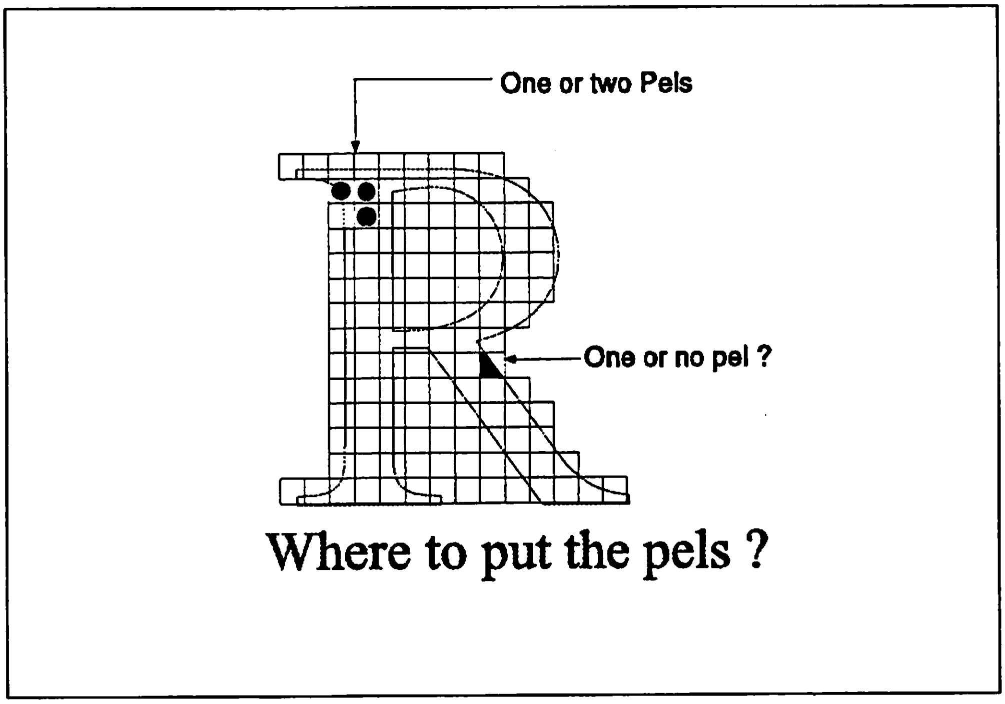


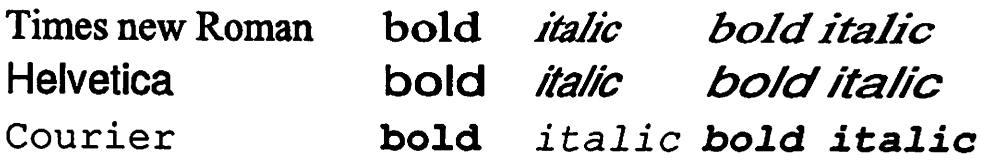
Figure 32. Outline to Bitmap Transformation

11.2.2 Fonts Shipped with OS/2 V1.3

As standard with OS/2 V1.3, thirteen SAA Core fonts will be provided. These come in the four families:

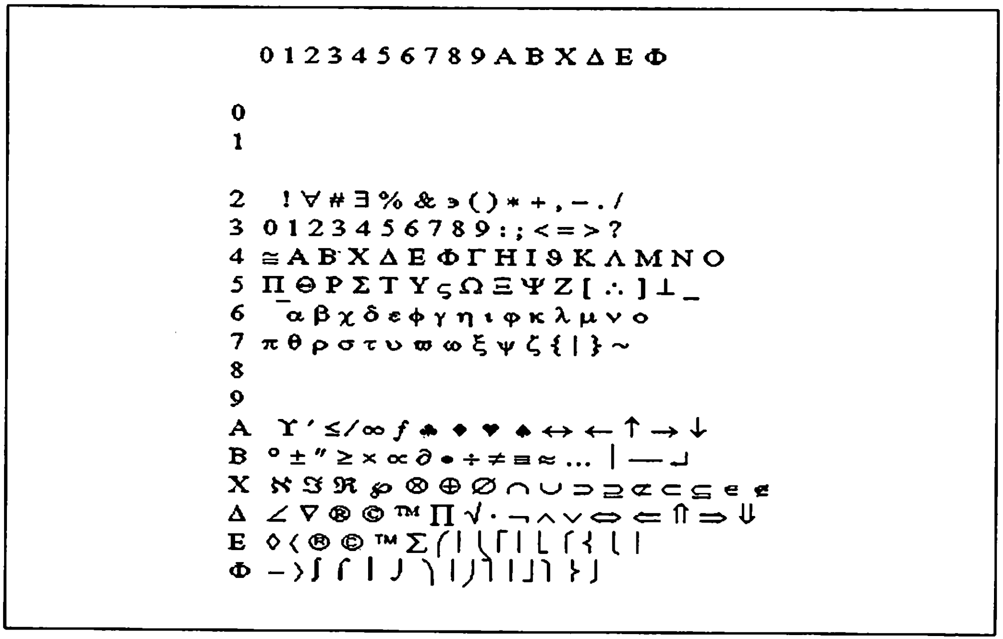
```
- Times New Roman : Normal, bold, italic, bold italic (previously TmsRmn)
- Helvetica       : Normal, bold, italic, bold italic (previously Helv)
- Courier         : Normal, bold, italic, bold italic (previously Courier)
- Symbol          : (new)
```

The bitmap fonts are still available under their old names: Courier, Helv and TmsRmn.



Times new Roman	**bold**	*italic*	***bold italic***
Helvetica	**bold**	*italic*	***bold italic***
Courier	**bold**	*italic*	***bold italic***

Figure 33. OS/2 V1.3 Type 1 Fonts Shipped with OS/2 V1.3



	0	1	2	3	4	5	6	7	8	9	A	B	X	Δ	E	Φ
0																
1																
2		!	∀	#	∃	%	&	∋	(	)	∗	+	,	−	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	≅	Α	Β	Χ	Δ	Ε	Φ	Γ	Η	Ι	ϑ	Κ	Λ	Μ	Ν	Ο
5	Π	Θ	Ρ	Σ	Τ	Υ	ς	Ω	Ξ	Ψ	Ζ	[	∴	]	⊥	_
6	‾	α	β	χ	δ	ε	φ	γ	η	ι	ϕ	κ	λ	μ	ν	ο
7	π	θ	ρ	σ	τ	υ	ϖ	ω	ξ	ψ	ζ	{	|	}	∼	
8																
9																
A		ϒ	′	≤	⁄	∞	ƒ	♣	♦	♥	♠	↔	←	↑	→	↓
B	°	±	″	≥	×	∝	∂	•	÷	≠	≡	≈	…	|	—	┘
X	ℵ	ℑ	ℜ	℘	⊗	⊕	∅	∩	∪	⊃	⊇	⊄	⊂	⊆	∈	∉
Δ	∠	∇	®	©	™	∏	√	⋅	¬	∧	∨	⇔	⇐	⇑	⇒	⇓
E	◊	〈	®	©	™	∑	⎛	⎜	⎝	⎡	⎢	⎣	⎧	⎨	⎩	⎪
Φ	–	〉	∫	⌠	⎮	⌡	⎞	⎟	⎠	⎤	⎥	⎦	⎫	⎬	⎭	

Figure 34. OS/2 V1.3 Type 1 Font Glyphs from the Symbol Font

11.2.3 Other Suppliers of Type 1 Fonts

Adobe supplies tools (BuildFont) to allow existing font foundries to convert their existing outline font data to the Type 1 format, allowing other vendors to distribute their own fonts in Type 1 format, and adding to the large number of fonts already available from Adobe themselves. Some of the companies who are converting their libraries into Type 1 format are:

Linotype
Autologic
Agfa-Compugraphic
Monotype
Morisawa
Bitstream
Varityper.

11.2.4 International Character Support

Most of Adobe's Type 1 fonts include 228 glyphs, which cover most of the standard Latin character set, and all of the code page 850 set except one glyph (the Greek mu character - µ - 230). However, some PM code pages will include characters that are not available in Adobe's standard set of glyphs; additional glyphs may be obtained from a standard set of glyphs built into PM, but they will not necessarily be stylistically related to the other characters in the typeface (you can see this if you print out the code page 850 character set in an additional Type 1 font like ITC Zapf Chancery - the mu character will look different from the other characters).

The Type 1 fonts included in OS/2 (IBM Core Fonts: Times New, Helvetica and Courier) have a wider range of glyphs, so there will not be quite as many discrepancies. Greek characters from the code page 437 set will be the most apparent omissions from the standard set of 228, supplied in the additional Type 1 fonts that users may install. In this respect, support of existing Type 1 fonts represents a compromise - the characters will be available, but not necessarily matching the typeface and attributes (bold, italic etc.) required. But, all of the accented alphabetic characters occurring in code page 850 are supported.

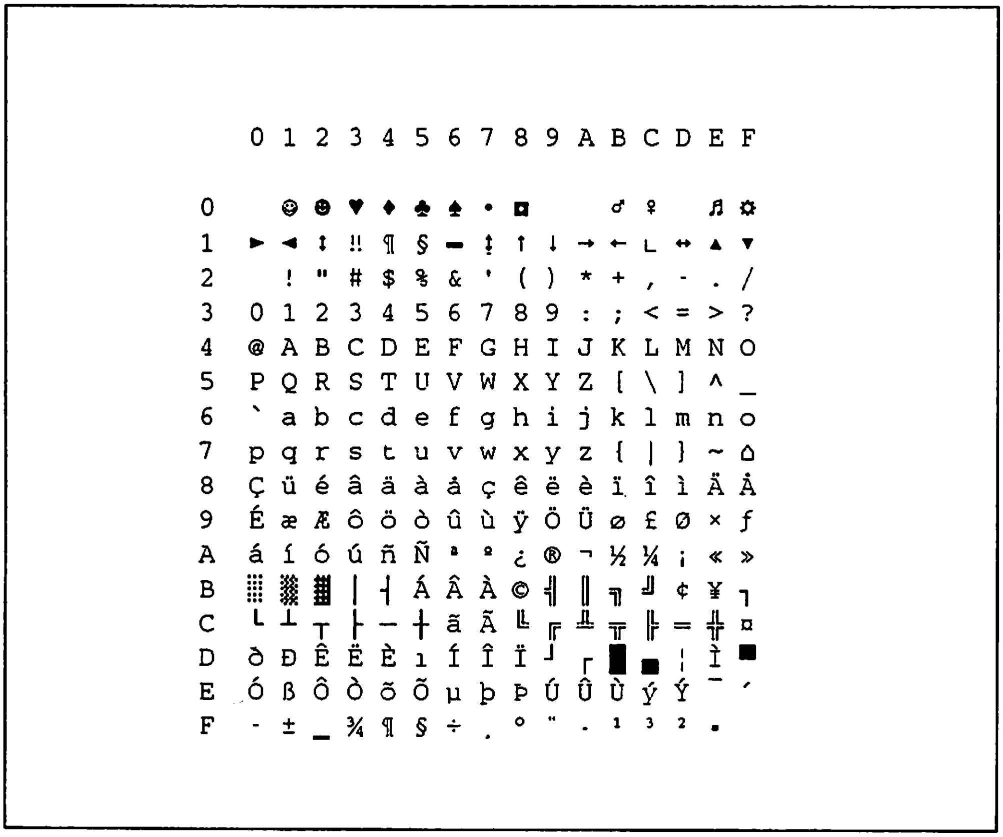
Furthermore, there are few Type 1 fonts currently available in non-Roman faces (except for a few Script Arabic and Far East character sets). However, changes would be needed anyway to the current version of OS/2 to recognize the additional glyphs for non-Roman faces.

11.2.4.1 Code Pages and PostScript Printers

When printing on a PostScript printer, with fonts that are resident in the printer and the user asks to print in a code page that does not exist in the printer's device font set, missing glyphs will be replaced by a space. If the font is one of the IBM Core Fonts, which can be downloaded to the printer, then all of the code pages supported by Presentation Manager can be printed because all the glyphs exist.

11.2.4.2 Code Pages and Non-PostScript Printers

If printing on a non-PostScript printer and the font is an off-the-shelf Type 1 Font, then all of the code pages supported by PM can be printed. However, there may be glyphs missing from the font definition. In this case, ATM substitutes a glyph from somewhere so that you get something, although eventually in the wrong typeface.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	•	◘			♂	♀		♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	⌂
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
C	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
D	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
E	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F	-	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	

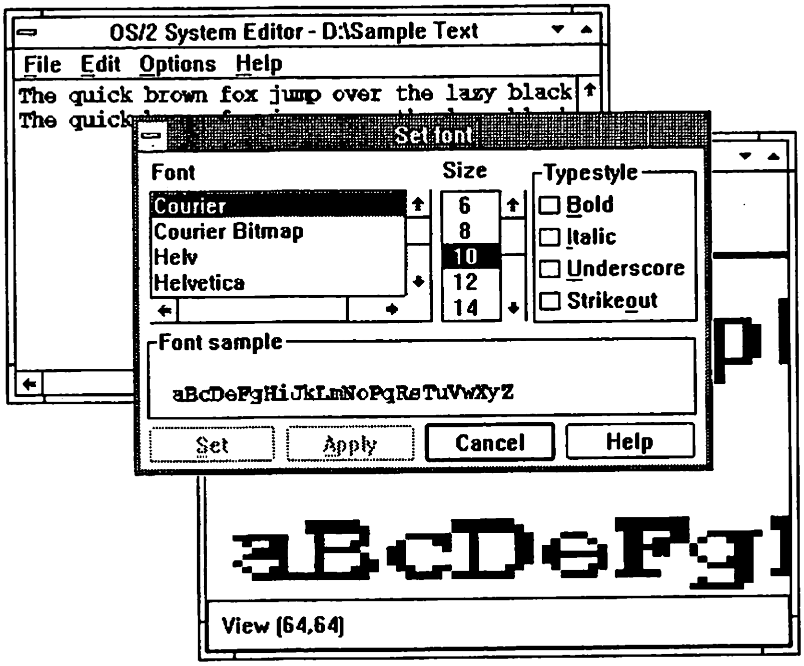
Figure 35. OS/2 V1.3 Type 1 Font Support for Code Page 850 (Courier)

11.2.5 Viewing of the Type 1 Fonts With the OS/2 System Editor

Figure 36 on page 81 and Figure 37 on page 81 are examples showing the difference between the OS/2 outlines fonts and the new Type 1 fonts.

Both examples have been taken from the OS/2 System Editor's "Set Font" dialog box and show the Courier 10 font, using the old OS/2 PM outline font and the new Type 1 font, which comes standard with OS/2 V1.3.

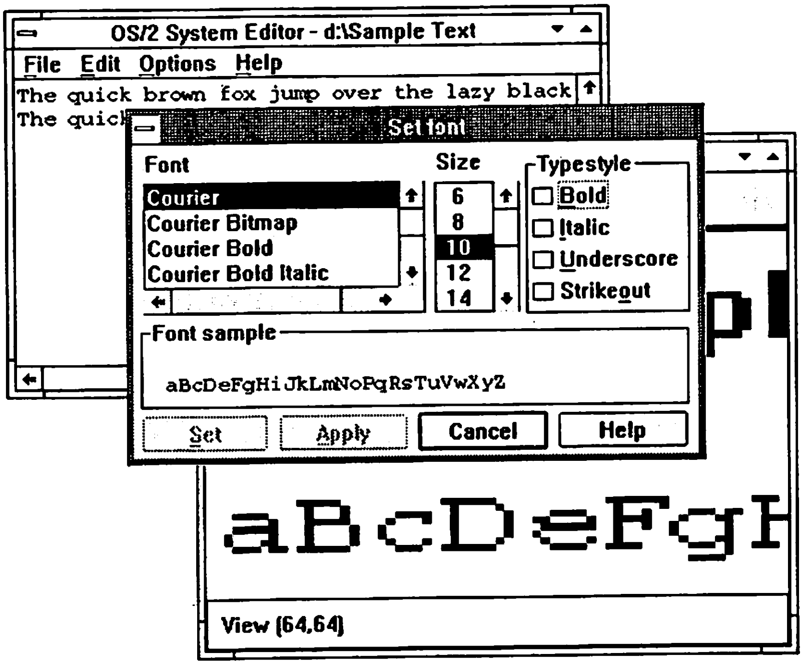
Figure 36 on page 81 shows the Courier 10 font on an OS/2 V1.2 system and Figure 37 on page 81 shows the same typeface on an OS/2 V1.3 system with the Adobe Type Manager installed.



The quick brown fox jump over the lazy black
The quick

aBcDeFgHiJkLmNoPqRsTuVwXyZ

Figure 36. OS/2 PM Outline Font. Set Font Dialog Box and Magnify[29]



The quick brown fox jump over the lazy black
The quic

aBcDeFgHiJkLmNoPqRsTuVwXyZ

Figure 37. OS/2 V1.3 Type 1 Fonts. Set Font Dialog Box and Magnify

[29] Magnify is an OS/2 utility for IBM internal use only.

11.2.6 Development Impact

No new programming interfaces!

The best feature of ATM is that there are absolutely no new programming interfaces for applications to gain access to the Type 1 fonts. All existing applications which work with OS/2 V1.3 PM outline fonts will work with these new ones. Therefore, the OS/2 V1.3 File Manager and System Editor will work with the ATM fonts immediately. OS/2 is now able to support a lot of new fonts so applications should be prepared to cope with a large number of fonts. As the new and the old font names are different, applications that use the old outline fonts and have their names "hard coded" may need to be modified and recompiled.

Table 5. Old and New Font Names under OS/2's Presentation Manager

OS/2 1.2 Outline Fonts	Type 1 fonts
Tms Rmn	Times New Roman
Tms Rmn Bold	Times New Roman Bold
Tms Rmn Italic	Times New Roman Italic
Tms Rmn Bold Italic	Times New Roman Bold Italic
Helv	Helvetica
Helv Bold	Helvetica Bold
Helv Italic	Helvetica Italic
Helv Bold Italic	Helvetica Bold Italic
Courier	Courier
Courier Bold	Courier Bold
Courier Italic	Courier Italic
Courier Bold Italic	Courier Bold Italic

If PM applications are able to query PM for all fonts available, there should be no need for changing the code. A good sample about font handling and printing in general can be found in the ITSC bulletin *OS/2 V1.3 Volume 2: Print Subsystem.*

11.3 Benefits

So how does the user benefit from the introduction of this new font technology with OS/2 Version 1.3?

11.3.1 Better Looking Screens and Printouts

This is the most apparent benefit that applies to all OS/2 users, even if they do not use desktop publishing or install extra fonts. Spreadsheets and word processing applications will display faster, scroll quicker, and produce better looking printouts when using outline fonts.

Note

If the applications use only bitmap fonts for screen display, there will be no difference in speed. Indeed, bitmap fonts may display slightly faster than their outline counterparts.

The text will be sharper, smoother and correspond more closely to printed output.

The improvement in quality will be especially obvious at smaller point sizes, but the faster display will be noticed whatever typeface is used and at whatever point size, leading to a faster "feel" for PM applications overall.

11.3.2 WYSIWYG

This takes WYSIWYG beyond our previous expectations: where some years ago it was considered enough to show bold or italic on a screen, or offer a screen preview function. We now have true WYSIWYG, where the screen and printer are actually using the same fonts, at the same sizes.

OS/2's WYSIWYG is much more than the approximate process of displaying raster fonts at specific (not necessarily the same) point sizes as the printer's outline fonts (as can be done today in environments like Microsoft Windows using bitmap files supplied by Adobe when the ATM extension is not installed).

This helps to bring users nearer the goal of not just being able to print anywhere, but to view-and-print anywhere on a variety of devices of differing resolution. It also reduces the number of iterative draft print outs that are needed en-route to the desired final result.

Note

Since November 1990, the Adobe Type Manager is also available for Microsoft Windows V3.0. It is supposed to deliver the same functions as OS/2's PM. ATM however, does not come standard with Windows V3.0, rather it is a separate add-on product.

11.3.3 Improved Output Without Investing in New Hardware

ATM and OS/2 now print high quality text on a large range of printers, even lower-cost devices like non-PostScript laser printers, or dot-matrix printers. This allows users to benefit from the new fonts and get a wider choice of typefaces without having to invest in a PostScript cartridge, for example. So now, quality text is available to every OS/2 user.

Users who already have PostScript printers also benefit, by the fact that they can now see on their screen what they will print. They can also add extra fonts without the need to master a cumbersome and time-consuming font download process or printer modification.

The result is that all users can be assured of identical fonts and spacing across different output devices. For example, one could design a document under OS/2 PM on a VGA screen (72 dpi), proof it by printing on an IBM 4019 Laser Printer (300 dpi) and send the text (by modem) to a professional typesetter for high quality print out (1200 dpi) or even film separation, suitable for reproduction (2500 dpi or higher) - using the same font through the whole process. This shows how output can be more consistent across devices, and therefore less device-dependent.

11.3.4 Huge Selection of Add-On Fonts - An Industry Standard

Type 1 fonts come in hundreds of styles and sizes, to suit any application. This gives the user a wider choice of fonts, and the opportunity to choose the one that is most appropriate to the context.

11.3.5 Investment Protection

Type 1 fonts are already widely used among IBM's customers. OS/2's support for this industry standard makes use of existing installed Type 1 fonts at thousands of customer sites, and enables safe continued investment in that technology, backed by both Adobe and IBM.

11.3.6 Performance

Display and printing of outline fonts is much faster than in previous releases, saving the user time, increasing overall productivity, and giving a faster feel to OS/2 PM applications.

11.3.7 Available to PM Applications WITHOUT ANY CHANGES NEEDED

For developers, this means they do not have to produce a new version simply to take advantage of the new functions. It also means new applications can be written to take advantage of the new fonts without a fundamental change in programming style.

For users, this means they can take advantage of this new quality immediately. They get a wider choice of fonts, of higher quality and superior performance, from the applications used today.

11.3.8 Intelligent Font Interface - No "Lock In" to One Technology

The Intelligent Font Interface allows other font technologies to be supported as they emerge, so that users are not restricted solely to the Adobe technology, but can take advantage of others if they provide additional benefits. In this way, IBM is providing "open" font support as well as supporting existing standards.

11.3.9 Ready for the Future

The Type 1 outline technology is well suited to take advantage of the inevitable improvements in screen and printer resolution as they emerge. This is because the outline font can always deliver the sharpest type the device is capable of creating.

Currently, VGA has about 72dpi (dots per inch), 8514/A and XGA 92dpi, an average PostScript laser printer 300dpi, but a typesetter already has at least 1200dpi of resolution.

WYSIWYG will therefore become even better as resolution nears the "ideal" resolution in which the font was created.

Type 1 fonts also preserve your investment in existing hardware - now you don't need to throw out old fonts when you change to a new device; and you don't need to throw out your current printer just to get a wider choice of fonts.

11.3.10 Cost Saving

Since ATM is included in OS/2 version 1.3, you get all these benefits without an extra charge. ATM is available in other environments, but usually has to be obtained as a separate cost item. OS/2 also includes four font families to get you started which will serve for most basic needs anyway.

11.3.11 Ease of Use

Type 1 fonts are easy to install and use. Their use is transparent to most users, who will simply appreciate the higher quality and improved speed. With regards to installation, there is no longer any need to use separate font download routines for printers and another procedure for screen font installs; one install does both - a unified technology for screens and printers. This means that when adding new fonts, you don't have to worry about downloading fonts to the printer as you might in DOS; now "one font fits all".

Note

So far, the OS/2 PM printer device drivers still have their own "downloadable fonts" dialog boxes, depending on the target printer. In the case of non-PostScript printers this might still be required due to any special user requirements. In the case of real PostScript printers, this is still necessary if the user installs more fonts for the screen than the printer already has built-in. This will enable the driver to download these additional Type 1 fonts if required by an application.

See also *OS/2 V1.3 Volume 2: Print Subsystem* for more details.

11.3.12 Portability

Adobe's font-rendering technology is a consistent standard, already available across a wide variety of different devices and platforms, including IBM's VM, MVS and AIX,[30] and other platforms such as the Apple Macintosh[31] and DEC VAX/VMS[32]

This gives the potential for greater device independence and portability (create the document in one place, print in another) and for sharing documents among departments and even organizations, retaining the full formatting and typeface features.

This is part of the potential - for consistency across platforms - that made IBM choose Adobe's technology as a part of its SAA plans. The aim can be summed up as "view and print anywhere".

All in all, the new Type 1 fonts give the OS/2 user high quality fonts, a wide choice of typefaces, provide the basis for true WYSIWYG, device-independence, and portability across platforms.

[30] AIX is a registered trademark of International Business Machine Corporation.

[31] Macintosh is a registered trademark of Apple Computer, Inc.

[32] DEC and VAX/VMS are registered trademark of Digital Equipment Corporation.

11.4 Adobe Type Manager: Installation

In the following section we will discuss how and where to install the Type 1 fonts.

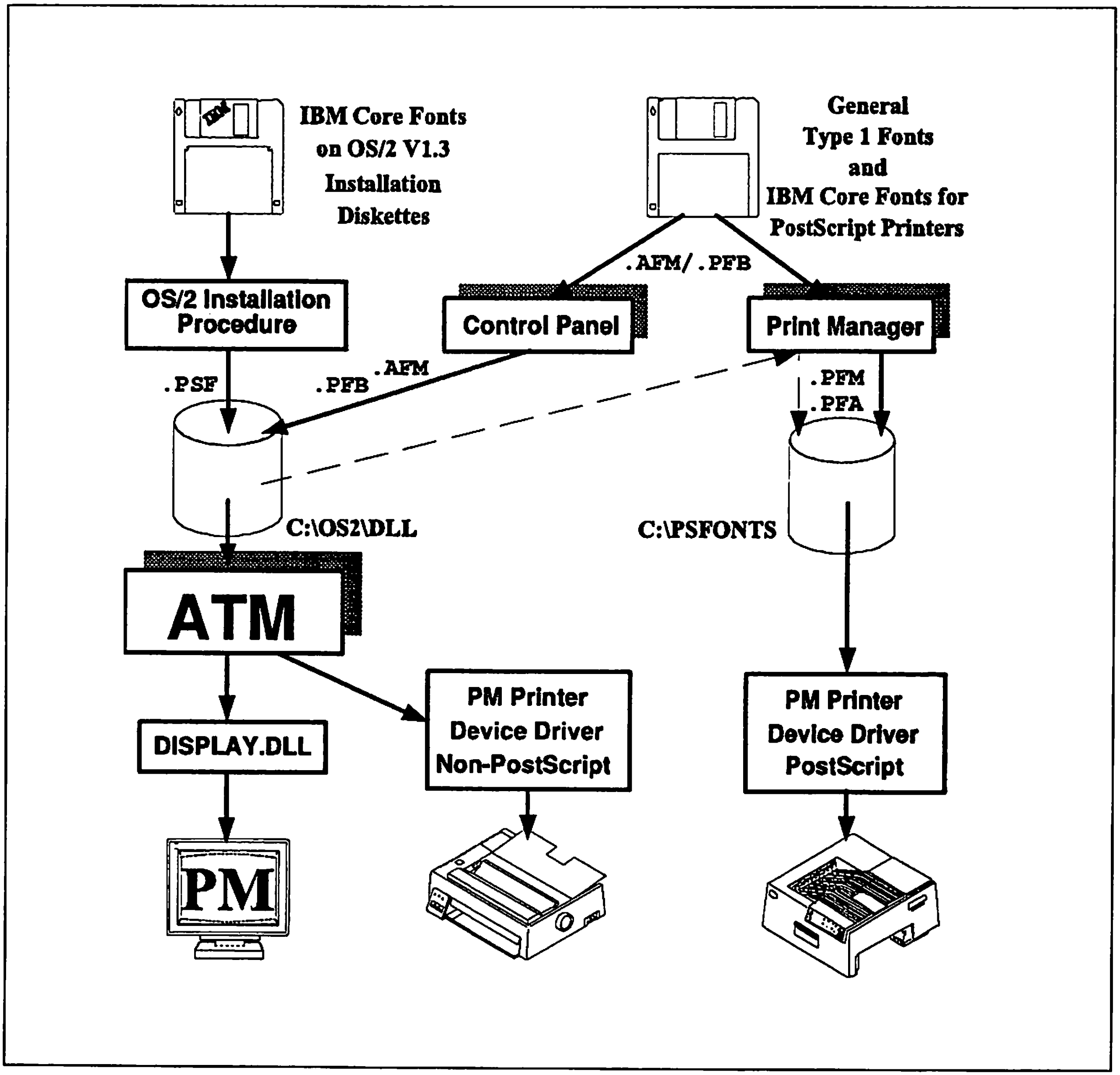

IBM Core Fonts on OS/2 V1.3 Installation Diskettes
General Type 1 Fonts and IBM Core Fonts for PostScript Printers
.AFM/.PFB
OS/2 Installation Procedure
Control Panel
Print Manager
.PSF
.PFB
.AFM
.PFM
.PFA
C:\OS2\DLL
C:\PSFONTS
ATM
DISPLAY.DLL
PM Printer Device Driver Non-PostScript
PM Printer Device Driver PostScript
PM

Figure 38. Adobe Type Manager: Font Installation

Most of Adobe's Type 1 fonts include 228 glyphs, which cover most of the standard Latin character set, and all of the code page 850 set except one glyph (the Greek mu character - μ - 230). However, some PM code pages will include characters that are not available in Adobe's standard set of glyphs; additional glyphs may be obtained from a standard set of glyphs built into PM, but they will not necessarily be stylistically related to the other characters in the typeface (you can see this if you print out the code page 850 character set in an additional Type 1 font like ITC Zapf Chancery - the mu character will look different from the other characters).

The Type 1 fonts included in OS/2 (IBM Core Fonts: Times New, Helvetica and Courier) have a wider range of glyphs, and so there will not be quite so many discrepancies. Greek characters from the code page 437 set will be the most apparent omissions from the standard set of 228, supplied in the additional Type 1 fonts that users may install. In this respect, support of existing Type 1 fonts represents a compromise - the characters will be available, but not necessarily matching the typeface and attributes (bold, italic etc.) required. But, all of the accented alphabetic characters occurring in code page 850 are supported.

Furthermore, there are few Type 1 fonts currently available in non-Roman faces (except for a few Script Arabic and Far East character sets). However, changes would be needed anyway to the current version of OS/2 to recognize the additional glyphs for non-Roman faces.

Code Pages and PostScript Printers
When printing on a PostScript printer, with fonts that are resident in the printer and the user asks to print in a code page that does not exist in the printer's device font set, missing glyphs will be replaced by a space. If the font is one of the IBM Core Fonts, which can be downloaded to the printer, then all of the code pages supported by Presentation Manager can be printed because all the glyphs exist.

Code Pages and Non-PostScript Printers
If printing on a non-PostScript printer and the font is an off-the-shelf Type 1 Font, then all of the code pages supported by PM can be printed. However, there may be glyphs missing from the font definition. In this case ATM substitutes a glyph from somewhere so that you get something, although eventually in the wrong typeface.

11.4.1 ATM Standard IBM Core Fonts

Fonts in OS/2 V1.3 are, as in the previous release, installed in the C:\OS2\DLL directory. Type 1 fonts are optional. If you want to install them on your system you have to execute the following steps:

1. Boot the OS/2 V1.3 installation diskette.
2. Insert diskettes as required until you get the "Refresh Installation Configuration" menu.
3. Select the "Add selectable operating-system options" item.
4. Select the "Font..." item.
5. Press *Enter* on the "Select Fonts" menu.
6. Insert diskettes as required.
7. Re-start the system.

The 13 "IBM Core Fonts" come on a separate diskette with OS/2 V1.3, *IBM Core Fonts for PostScript Printers*. There are two files for each font. One file has the extension .AFM the other has .PFB. The AFM/PFB format files are the format that Adobe (and other typeface foundries) deliver their "Type 1" format fonts in. The first are the "Font Metrics" in an ASCII file which describes the font. The second one is the "PostScript Font Binary" which is the font itself. The OS/2 Adobe Type Manager requires both. After installing OS/2 V1.3 and the Type 1 fonts, the directory C:\OS2\DLL will contain these font related files.

C:\OS2\DLL\

```
COURIER  FON  119776  11-02-90  12:00p
HELV     FON  355440  11-02-90  12:00p
TIMES    FON  329840  11-02-90  12:00p
SYSMONO  FON   39424  11-02-90  12:00p
COURIER  PSF  158208  11-02-90  12:00p
HELVETIC PSF  122368  11-02-90  12:00p
TIMESNRM PSF  134656  11-02-90  12:00p
```

Files: .FON

These files hold the standard OS/2 V1.3 bitmap fonts and are copied there by the OS/2 V1.3 install procedure or the "Control Panel" program ("Installation/Add Font..." menu).

Files: .PSF

PostScript font files are the new Type 1 Font files in internal-to-PM format which is designed (for efficiency) for the core fonts only (Times New Roman[33] , Courier and Helvetica).

That's all you need to know to use Type 1 fonts under PM, or on any non-PostScript printer.

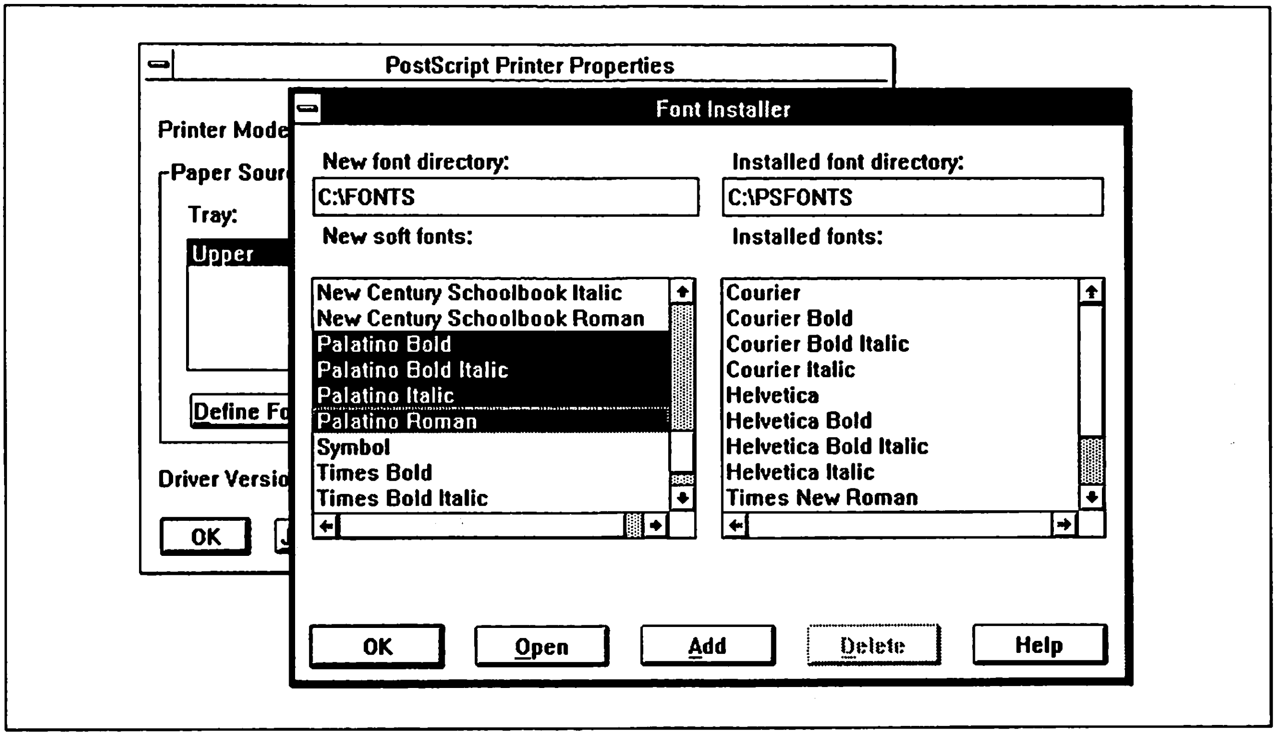
If you want to use PostScript printers, you have to install a PostScript printer device driver. You can install printer drivers with the "Control Panel" but it is simpler, and advisable, to do it with the new "Printer Installer" option of the Print Manager.

The Adobe Type Manager itself is contained in a dynamic link library C:\OS2\DLL\PMATM.DLL, which is automatically installed on the system. It delivers its function through a hook into the operating system and can also be called by PM for its special services.

11.4.2 PostScript Downloadable Fonts

The PostScript printer device driver now also supports downloadable fonts. You can install downloadable fonts from the "Font Installer" dialog box either at installation time, if you are using the "Printer Install" option of the Print Manager Setup menu, or at a later time using the "Printer Properties" dialog. The maximum number of downloadable fonts that a given printer can use on a document is limited by the amount of memory installed in that printer.

[33] Times New Roman is a registered trademark of Monotype Corporation Limited.



PostScript Printer Properties

Printer Mode

Paper Sour

Tray:

Upper

Define Fo

Driver Versio

Font Installer

New font directory:

C:\FONTS

New soft fonts:

New Century Schoolbook Italic
New Century Schoolbook Roman
Palatino Bold
Palatino Bold Italic
Palatino Italic
Palatino Roman
Symbol
Times Bold
Times Bold Italic

Installed font directory:

C:\PSFONTS

Installed fonts:

Courier
Courier Bold
Courier Bold Italic
Courier Italic
Helvetica
Helvetica Bold
Helvetica Bold Italic
Helvetica Italic
Times New Roman

Figure 39. Print Manager: Font Installer Dialog Box

To make downloadable fonts available to your PostScript printers:

1. Select the Setup menu in Print Manager, and select the Printers command.
2. Select a printer associated with the PostScript device(s), and select Change.
3. Click on the Printer Properties button.
4. Click on the Fonts button. In the Font Installer dialog box, add the downloadable fonts to be used by your printer.

Note

If you have multiple PostScript printers, the installation of downloadable fonts need only be done once. Once installed, downloadable fonts are available to any PostScript printer connected to your system.

Once you have installed the downloadable PostScript fonts, a new directory C:\PSFONTS is created and may contain some of these files:

```
C:\PSFONTS\

COU     PFA    92725  12-09-90  11:06a
COU     PFM     1380  12-09-90  11:08a
COUB    PFA    94126  12-09-90  11:06a
COUB    PFM     1384  12-09-90  11:08a
COUBI   PFA    95987  12-09-90  11:07a
COUBI   PFM     1390  12-09-90  11:08a
COUI    PFA    93492  12-09-90  11:07a
COUI    PFM     1388  12-09-90  11:08a
HEL     PFA    82846  12-09-90  11:07a
HEL     PFM     2012  12-09-90  11:09a
HELB    PFA    85896  12-09-90  11:07a
HELB    PFM     1992  12-09-90  11:09a
HELBI   PFA    90374  12-09-90  11:07a
HELBI   PFM     1986  12-09-90  11:09a
HELI    PFA    89579  12-09-90  11:07a
HELI    PFM     1978  12-09-90  11:09a
TNR     PFA    94442  12-09-90  11:07a
TNR     PFM     2072  12-09-90  11:09a
TNRB    PFA    97797  12-09-90  11:07a
TNRB    PFM     2076  12-09-90  11:10a
TNRBI   PFA   105223  12-09-90  11:07a
TNRBI   PFM     2082  12-09-90  11:10a
TNRI    PFA   103386  12-09-90  11:08a
TNRI    PFM     2080  12-09-90  11:10a
```

The PFA file holds the same data as the PFB, but is translated from binary to ASCII. It contains the ASCII PostScript description of fonts. The PFM file contains the same data as the AFM file, but is translated from ASCII to binary. It contains the font metric of the fonts.

The PostScript printer device driver expects that all downloadable fonts are in the C:\PSFONTS directory. As both the Adobe Type Manager and the PostScript device driver accept the Type 1 fonts file format, it is recommended that any fonts installed later with the Control Panel also be copied to this directory. Those fonts can then be installed for use by the PostScript printer device driver. See the PSCRIPT.DAT file on the PM printer devices driver diskette, for more information. Be aware though, that after a complete installation these files have different internal formats.

11.4.3 Installing Additional Type 1 Fonts

Adobe and other vendors supply hundreds of Type 1 fonts. Adobe distributes them in their Adobe Type Library (usually a pack of one or two font families with normal, bold, italic and bold italic variants). These contain fonts which are mainly used for downloading to PostScript printers. The theory of OS/2 V1.3's support of the ATM is that these fonts should be installable via the Control Panel in the usual way.

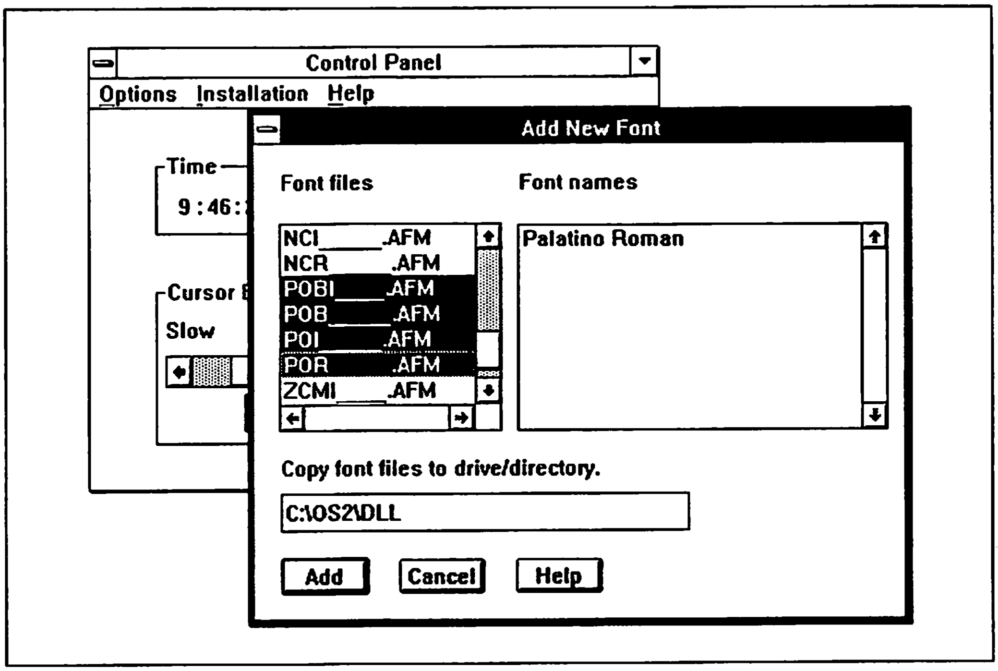
They are - except for a small problem. OS/2 V1.3's Control Panel looks for two files on the Type 1 font diskettes, one with an "AFM" extension, and another with a "PFB" extension. Adobe currently ships these files on two separate diskettes in the Type Library package. You have to put the right disk in first (the one with the .AFM file(s) - usually labelled the "Screen Font And Metrics Diskette"), otherwise the Control Panel thinks there are no fonts there. If you put the right disk in, you get a list of .AFM files (usually at least three or four per diskette) with the family font name alongside each, and you can then select one or more to install. It copies the .AFM file okay, but displays an error message saying it can't find the corresponding .PFB file. The "advice" which accompanies the message suggests that you are out of space on the drive, or that the drive is not ready. Neither is true - you just haven't gotten the disk with the .PFB file in the drive (this disk is usually labelled "Printer Font Diskette"), but it doesn't give you a chance to put that disk in and retry. The way around this problem is to copy the relevant .AFM and .PFB files to a separate directory (for example C:\FONTS) and install using C:\FONTS as the source directory. This works fine and allows you to install several fonts in one step. It is also much faster than installing from the diskette. The problem is that the users have to work this out for themselves.

See also *Using Type 1 Fonts.*

Information

Adobe is going to change the packaging of their Adobe Type 1 Library to include the required AFM and PFB files on the same disk.

However, once installed, the additional Type 1 fonts are easily picked up by all the applications added to the menus, and then treated in the same way as Times New or Helvetica, available in the same variety of point sizes. This includes the File Manager and System Editor.



Font files:

NCI_____.AFM
NCR .AFM
POBI____.AFM
POB_____.AFM
POI .AFM
POR .AFM
ZCMI____.AFM

Font names: Palatino Roman

Copy font files to drive/directory.

C:\OS2\DLL

Figure 40. Control Panel: Add New Font Dialog Box

For example, these files could have been added to the C:\OS2\DLL\ after installing the Palatino Roman fonts.

C:\OS2\DLL\

```
POBI____ AFM    16252  5-01-90   8:00a
POBI____ PFB    44696  5-01-90   8:00a
POB_____ AFM    16097  5-01-90   8:00a
POB_____ PFB    43224  5-01-90   8:00a
POI_____ AFM    16173  5-01-90   8:00a
POI_____ PFB    44586  5-01-90   8:00a
POR_____ AFM    16198  5-01-90   8:00a
POR_____ PFB    43693  5-01-90   8:00a
```

Note: The internal file format is different than the standard core fonts delivered with OS/2 V1.3. Those have been modified for performance reasons, where these fonts are installed as standard Type 1 fonts.

Important

As information is added to the OS2.INI file, don't try to install or remove fonts just by copying or erasing files.

Adding or deleting fonts must be done with the Control Panel and the Print Manager. Don't try to install fonts just by copying their files.

The core fonts delivered with OS/2 V1.3 are loaded in a different format. **Don't** install them with the Control Panel; you must use the OS/2 V1.3 installation diskettes.

See also *Using Type 1 Fonts*.

Don't forget to install these fonts as downloadable fonts if you plan to use them on a PostScript printer, which does not have them already built-in. The process for installing them is the same as the one for installing the core fonts supplied with OS/2 V1.3. Afterwards, the directory C:\PSFONTS\ may contain the following files:

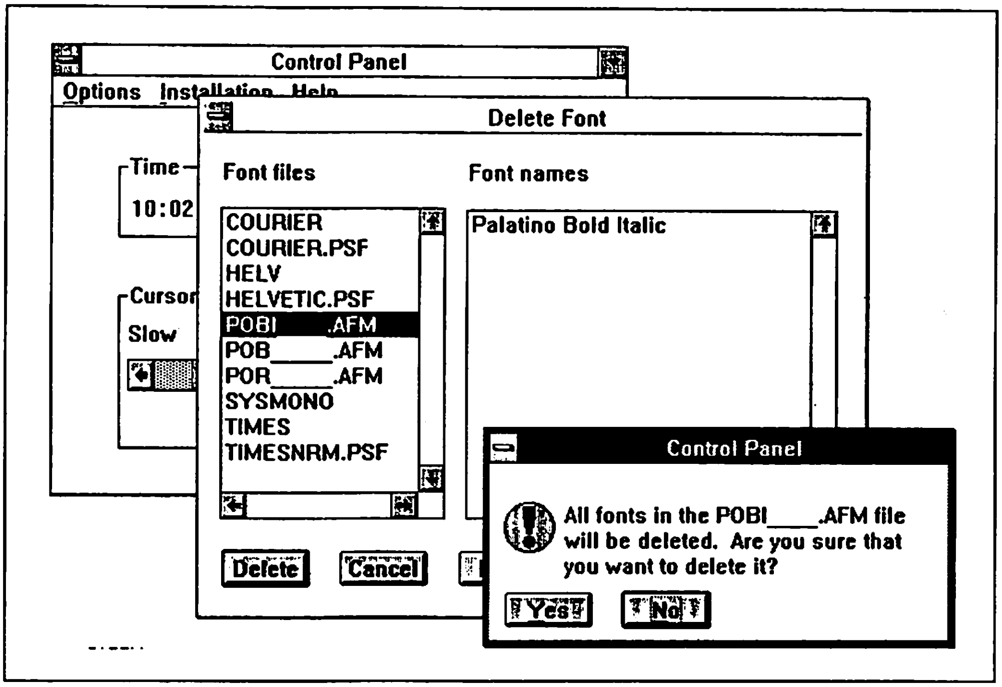
C:\PSFONTS\

```
POB_____ PFA    86780  11-01-90  11:05a
POB_____ PFM     2020  11-01-90  11:06a
POBI____ PFA    89806  11-01-90  11:05a
POBI____ PFM     2068  11-01-90  11:06a
POI_____ PFA    89582  11-01-90  11:05a
POI_____ PFM     2054  11-01-90  11:06a
POR_____ PFA    87744  11-01-90  11:06a
POR_____ PFM     2082  11-01-90  11:06a
```

11.4.4 Deleting Fonts

The standard core fonts or the additional Type 1 fonts have to be deleted with the Control Panel. To delete them, select "Installation/Delete font..." from the Control Panel. In the Delete Font dialog box, select the font you want to delete and press the Delete pushbutton. Confirm the delete on the message box and confirm too that you want to remove the file from the C:\OS2\DLL\.

Note: To reinstall fonts: the OS/2 V1.3 core fonts have to be reinstalled with the installation diskette and all other fonts with the Control Panel.



COURIER
COURIER.PSF
HELV
HELVETIC.PSF
POBI .AFM
POB_____.AFM
POR_____.AFM
SYSMONO
TIMES
TIMESNRM.PSF

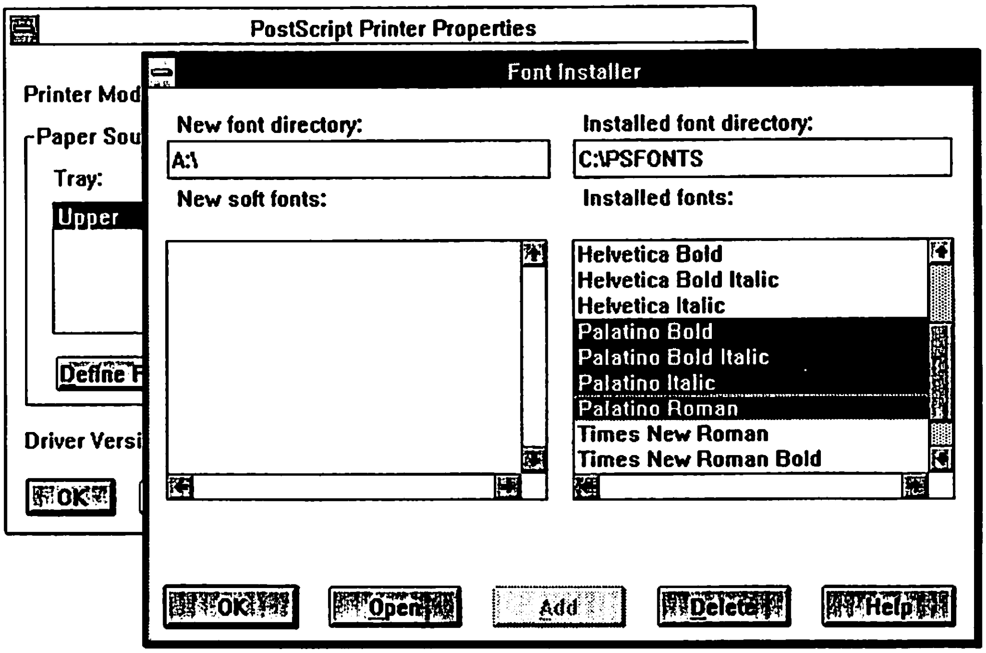
Palatino Bold Italic

All fonts in the POBI____.AFM file will be deleted. Are you sure that you want to delete it?

Figure 41. Control Panel: Delete Font Dialog Box

To delete a downloadable (printer) font:

1. Select the Setup menu in Print Manager, and select the Printers command.
2. Select the printer associated with a PostScript device(s), and select Change.
3. Click on the Printer Properties button.
4. Click on the Fonts button. In the Font Installer dialog box, select the font you want to delete and press the Delete pushbutton.



PostScript Printer Properties

Printer Mod

Paper Sou

Tray:

Upper

Driver Versi

Font Installer

New font directory:

A:\

New soft fonts:

Installed font directory:

C:\PSFONTS

Installed fonts:

Helvetica Bold
Helvetica Bold Italic
Helvetica Italic
Palatino Bold
Palatino Bold Italic
Palatino Italic
Palatino Roman
Times New Roman
Times New Roman Bold

Figure 42. Print Manager: Delete Font Dialog Box

If you have multiple PostScript printers, the deletion of downloadable fonts need only be done once.

11.5 About Fonts in General

First of all, it is worth examining why fonts and the use of different typefaces have become important, and what different font types are used.

11.5.1 Why Fonts?

The rapid growth in all forms of electronic publishing, and particularly desktop publishing on personal computers, has created an increasing demand for, and an ever more sophisticated use of, a variety of high quality typefaces. Even the "print-for-profit" publishing industry (including a growing proportion of the press) often uses Adobe's PostScript as their communications format.

Letterforms can be one of the most important means of helping to convey your message effectively in print. That is because the fonts or typefaces you use can say almost as much as the words themselves.

11.5.1.1 The Importance of Good Font Design

A lot of expertise goes into designing typefaces. Type design itself goes back hundreds of years, to the days of hand-carving and setting up printing by hand. Centuries before computers were ever used in printing or publishing, people were creating typefaces with various proportions of straight and round strokes, of varying thickness, to create a pleasing effect on the eye of the reader.

Today, digital font vendors like Adobe, license most of their designs from type suppliers ("foundries") like Linotype, Agfa-Compugraphic and International Typeface Corporation. Interestingly enough, Linotype was the first licensee of Adobe's PostScript font creation technology, showing the process is two-way; other foundries will convert their libraries into Type 1 format as well. Now the creators of type designs and those who convert those designs into digital format work very closely together, as the typeface industry relies ever more on computer technology. In fact, Adobe designed many new typefaces of their own, which they distribute in Type 1 format (the "Adobe originals").

11.5.1.2 Expressive of Mood/Tone

Typefaces can be used in various ways to invoke a certain tone or mood in the reader, whether authoritative, or modern-looking, or informal or even humorous. Some typefaces have names that associate them with certain professions (for example New Century Schoolbook for education, or Chancery for legal) and which also matches their look and feel. Many thousands of different typefaces exist, many with only very subtle differences from each other.

Some font vendors (for example Bitstream with their Fontware) produce a list of fonts giving suggestions of which fonts to use for different types of applications: obviously, certain fonts are suitable for office correspondence and financial reports while others are more suited to invitations, labels and newsletters.

Customers are using types with more sophistication and therefore demanding more from the font technology they use.

Just think of the difference that Graphic User Interfaces have brought to people's expectations: the gulf between the fixed pitch, fixed space font in a character-based interface, compared to the Graphic User Interface, which shows text more like it is meant to be seen - proportionally spaced (like newspapers and books and everything else we read).

11.5.2 Font Terminology

The terms used in describing fonts can sometimes be confusing. In this chapter, we have attempted to use the following terms fairly consistently, to distinguish between them:

11.5.2.1 Glyphs

Although not a common term, this is a very useful one. A glyph is the shape or form of an individual character which is unique to that character: for example, an "upper case A", a "lower case e acute". The word is intended to refer to the basic shape of the character (such as a capital H which consists of two perpendicular bars joined by another at right angles to them, joining them at or near the middle of each vertical bar.)

11.5.2.2 Typeface

A typeface is a collection of individual glyphs, with certain variations in shape and proportion defined by the designer of the typeface, and additions of items like serifs (the "edging" applied to characters like "R", below:)

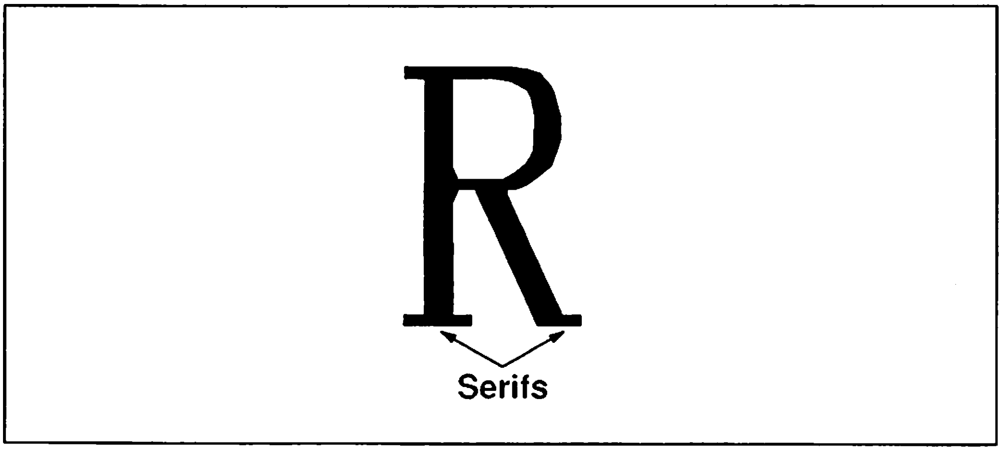


Figure 43. Serifs on Capital R

The term covers the full collection of glyphs defined, rather than any individual glyph.

11.5.2.3 Font

A font is the embodiment of a specific typeface in a physical or machine-readable form. Therefore, it refers not only to computer (digital) form, but can also be used to refer to stencils and even letters etched in brass, for example.

In the rest of this chapter, we will be concerned solely with digital font technology.

11.5.3 Font Types

There are two principal types of digital fonts: bitmap and outline. The difference between them is fundamental to the way fonts are used in Graphic User Interface environments like OS/2's Presentation Manager.

11.5.3.1 Bitmap Fonts

The first digital fonts were produced by scanning font masters at high resolution to create fonts at 1000 dots per inch (dpi). Although these fonts worked well in the high-resolution devices (typesetters) for which they were designed, they could not be effectively scaled for use on lower resolution devices such as display screens.

These fonts were stored in bitmap format: a specific pattern of dots and spaces in a matrix of a given size.

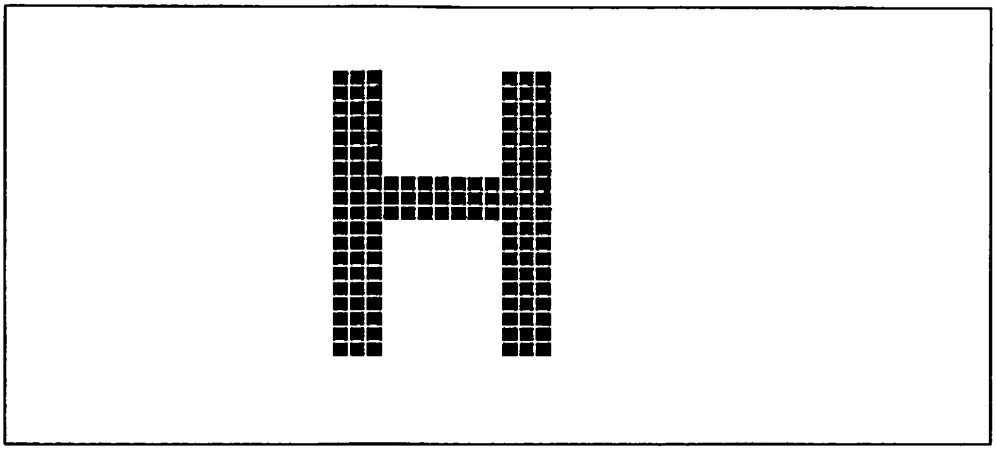


Figure 44. Bitmap Capital H

11.5.3.2 Outline Fonts

On the other hand, outline fonts are what their name suggests: a series of co-ordinates defining an outline, often on a 1000x1000 matrix (as is the case of Adobe's Type 1 format), joined with a series of straight lines and Bezier curves.

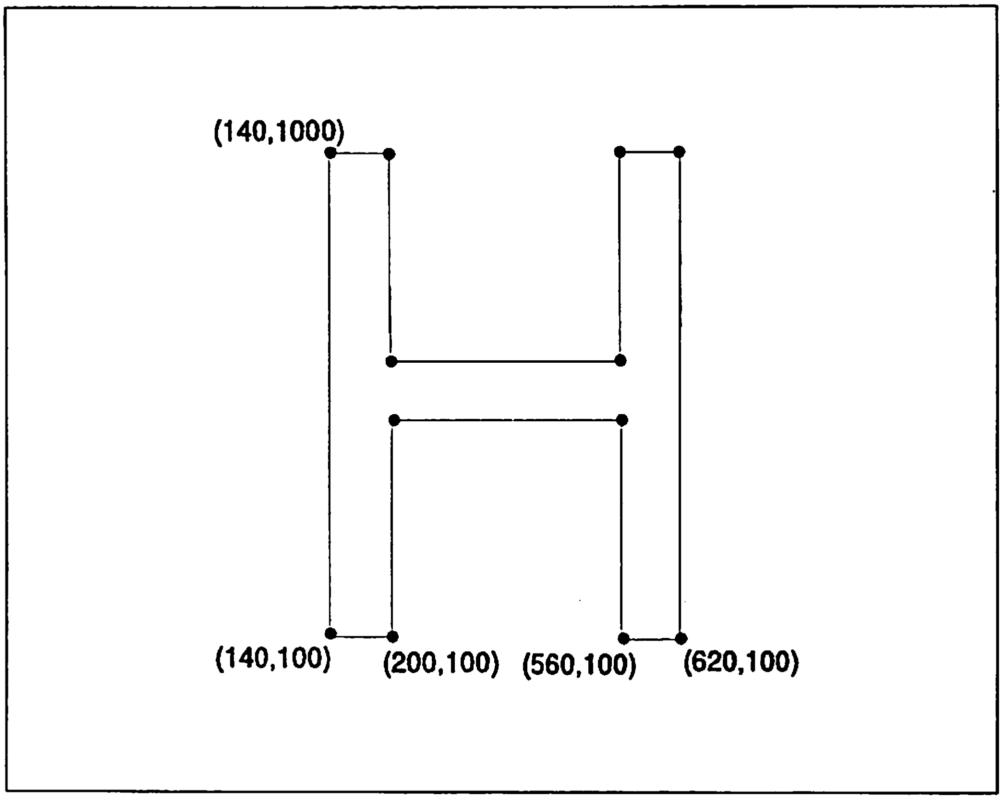


Figure 45. Outline Capital H

To create characters at a particular resolution on a particular device, the basic outline is scaled and transformed to the hardware device co-ordinates. Then the outline is filled with dots at the device resolution. This is basically how PM displays outline fonts.

OS/2 Version 1.3 includes four families of outline fonts: Courier, Helvetica, Times New Roman, and Symbol. The first three families are available in four faces: normal, bold, italic, and bold italic.

11.5.4 Outline Versus Bitmap

Both types of fonts are used widely in various types of computers, and PM uses both. So what are the relative advantages and disadvantages of each format?

First, let's look at how outlines are preferable to bitmaps.

11.5.4.1 Outlines Are Device and Resolution Independent

This means that only one representation of the font is needed whatever the device and resolution, because the basic outline can be scaled to the new resolution as required. So, even new devices can be handled easily by outline fonts, providing the basic hardware device driver is present. On the other hand, bitmaps have to be created for a specific combination of device and resolution. This means that bitmaps have very restricted portability - a separate bitmap is needed for each size and resolution required, which takes up more disk space.

This also means that you are likely to have a wider choice of fonts and point sizes among outline fonts than you are for bitmaps (as all the effort for bitmaps goes into creating variants of the same font for different resolutions), and also that it is generally easier to provide immediate outline font support for a new device than bitmaps.

11.5.4.2 Outlines Are More Flexible than Bitmaps

Outlines can be scaled, rotated, filled, and made oblique quite easily: it is just a fairly straightforward case of transforming the co-ordinates. This is not the case for bitmaps, where you effectively start from scratch with each new size, or orientation (this also makes it more difficult to retain consistency for a typeface through a range of point sizes). This makes it more difficult to provide bitmap fonts in a wide range of point sizes, whereas there is no difficulty for outlines. That's why outline fonts are essential for applications like desktop publishing, which require greater flexibility in point sizes. You will probably notice in OS/2 that bitmap versions of the fonts (such as Helv or TmsRmn) are only available in a limited range of point sizes, whereas applications can use outlines in other point sizes.

11.5.4.3 Outlines Take Less Memory and Disk Space

This is because they are stored as a formula, rather than as the entire pattern. Note however, that because of this, bitmaps will often be quicker, unless the font caching (see section 11.5.5.2, "Font Caching" on page 104) is very efficient. Bitmap file sizes also tend to be much larger and therefore take up more disk space.

See also Appendix D, "Fonts Under OS/2 V1.3" on page 175 for a comparison of font file sizes.

If the only fonts used were bitmaps, a large amount of disk storage would be needed. Indeed, the larger the point size, the larger the bitmap font files need to be; so for large point sizes, such as 72 point and above, outlines are usually the more practical option.

11.5.4.4 Jagged Edges at Larger Sizes

If a bitmap is not available in a given point size, and there are no outline fonts available for the typeface required, it may be necessary to scale up the bitmap. As well as not being such a straightforward exercise as scaling an outline, the result may be less attractive, giving jagged edges, like the "R" below:

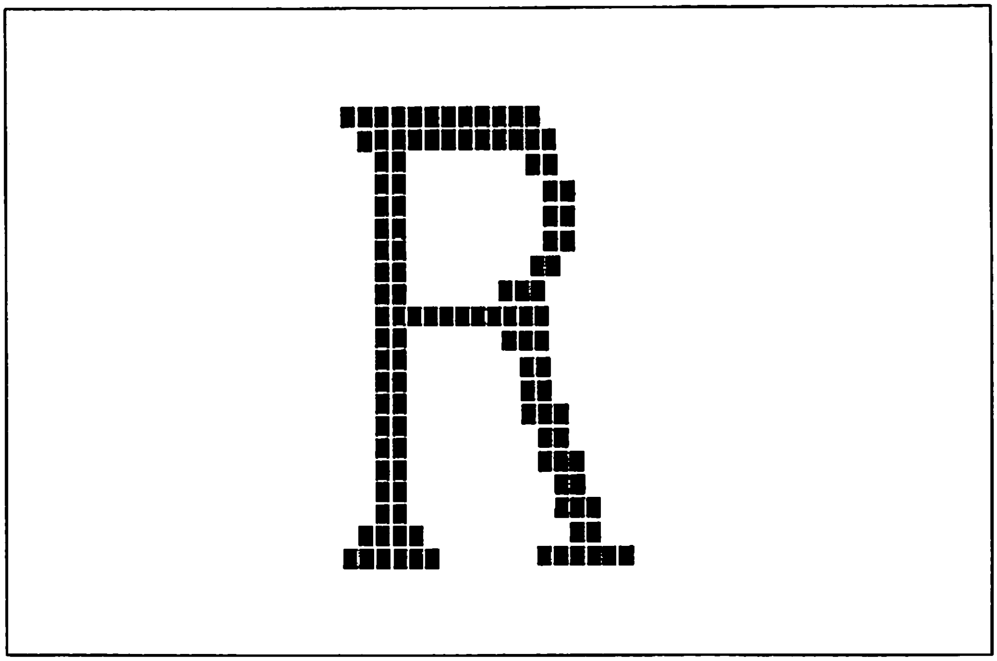


Figure 46. Jagged Edges When Scaling Up a Capital R

This effect is sometimes seen where large point sizes are used within applications in graphical environments that do not provide outline fonts as a default, such as Microsoft Windows.

11.5.4.5 Hand-Edited Bitmaps

However, bitmaps do have some advantages. Despite the quality problems of scaling bitmap fonts, and their relative inflexibility, they can be "hand-edited" and customized for the best possible quality for that particular device and resolution (as they have been in Presentation Manager). This will obviously take more work than scaling an outline, but the results will sometimes be better, and therefore a more practical choice, at very low resolutions and very small point sizes, for example 6 point text on a VGA screen. Furthermore, bitmaps that do not have to be scaled can be displayed very quickly, as it is only a matter of copying an array of bytes to the screen.

11.5.4.6 Outline Font Limitations

One of the other problems with outline fonts is that the scaling process can cause problems of both quality and speed:

Quality: The quality problems occur when scaling an outline definition at lower resolutions. Simply dividing by a given factor (for example 100) and rounding to the nearest point (i.e. pixel), rounding errors will occur which lead to incorrect shapes being formed, such as for "H", where you can have unequal widths of the vertical stems. The diagrams below show the problem, where the "ideal" shape of the "H" is misformed by rounding down to a lower resolution (in this case dividing the co-ordinates by 100):

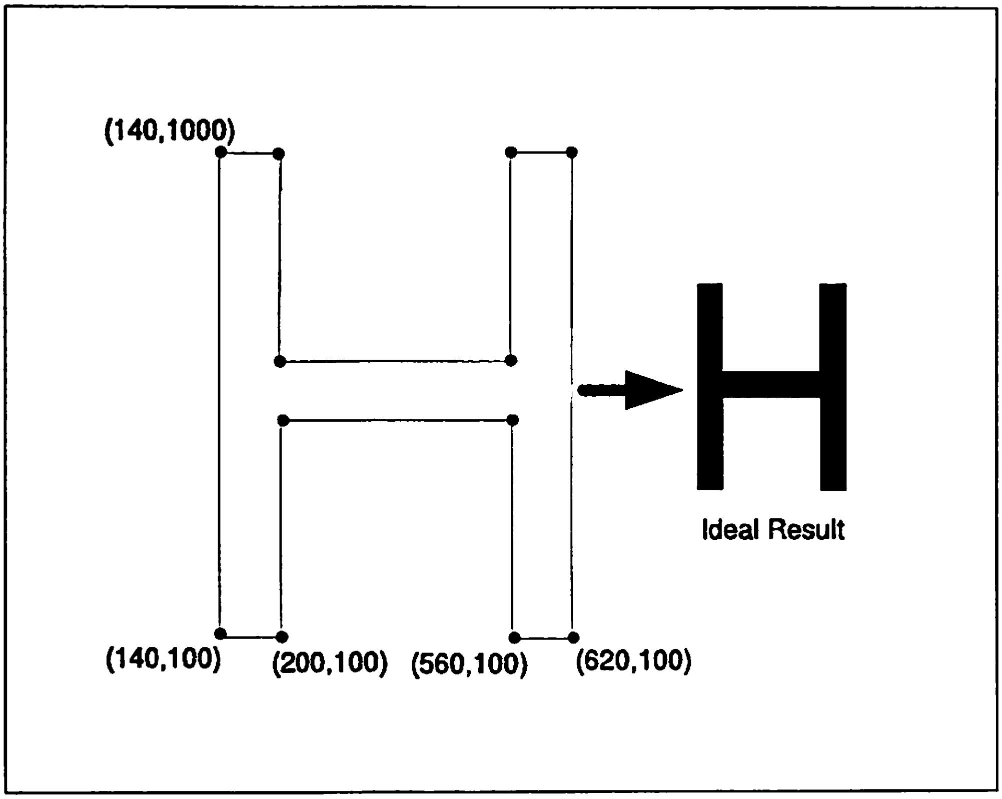


Figure 47. Outline Capital H Before Scaling

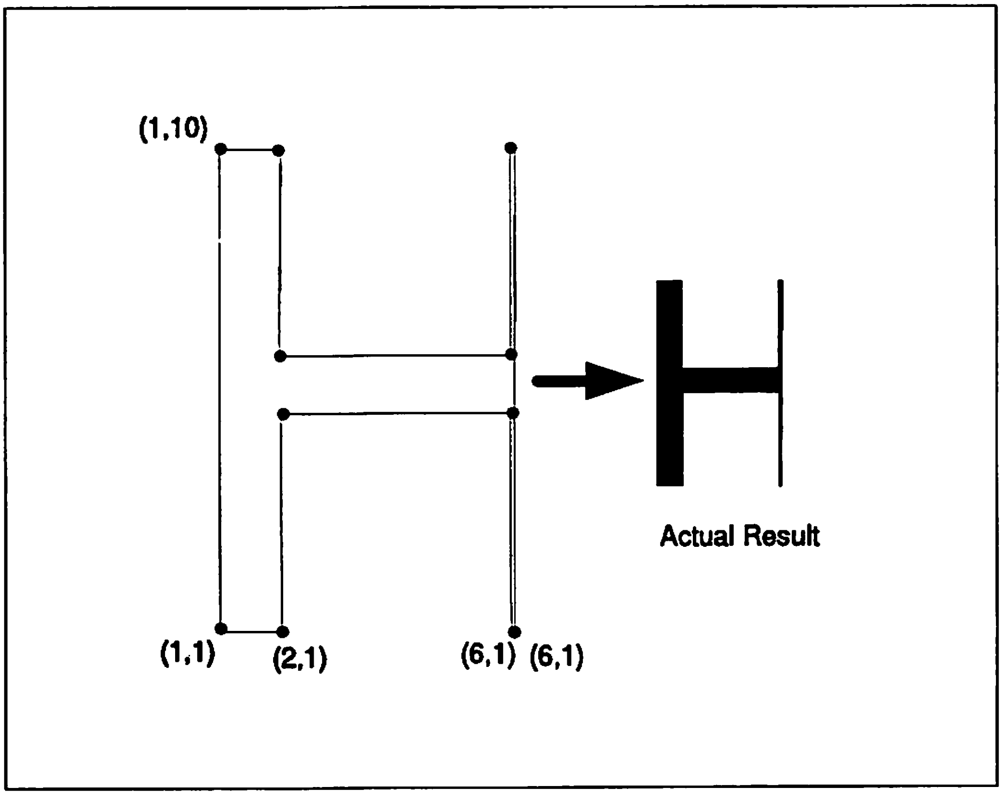


Figure 48. Outline Capital H After Scaling

The problem gets even worse with curves. Usually the thickness of a curve should be wider than the straight strokes (For example on a "D"), and also the bottom of an "O" usually will go lower than the bottom of an "L". These are effects designed into the typeface for pleasing proportion and readability. But rounding errors can grotesquely exaggerate these effects, causing individual characters to appear misformed and of variable size (See Figure 36 on page 81)

Other problems include:

- Pixel fill-in (sometimes seen on characters like "e")

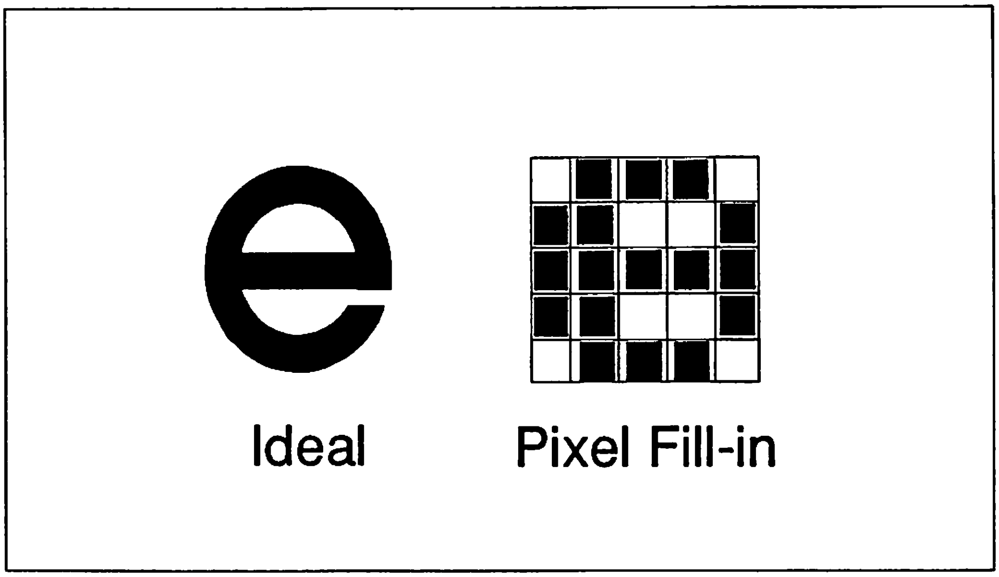


Figure 49. Pixel Fill-In on "e"

- Pimples and dimples on curves ("o", "p", "D", "C")

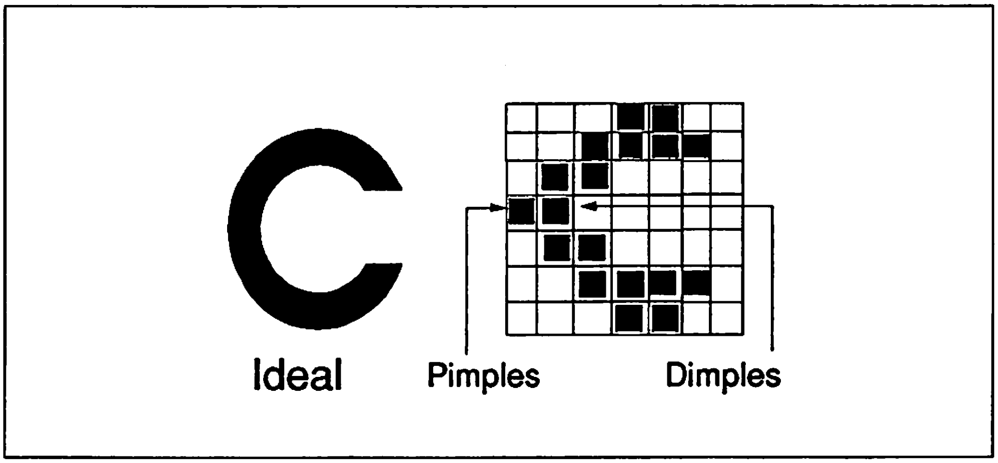


Figure 50. Dimples on "c"

- Serif collision (for example, on "n")

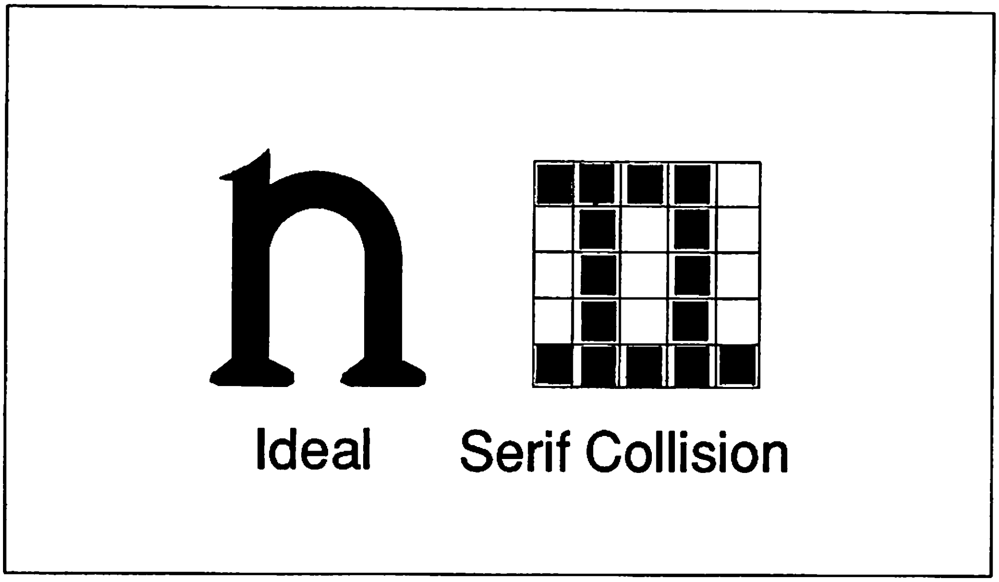


Figure 51. Serif Collision on "n"

- Dropout (for example, on "R")

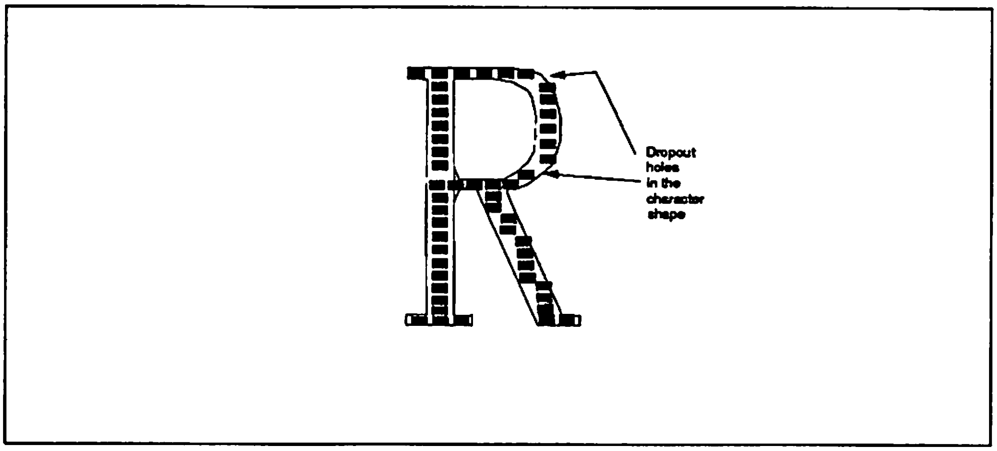


Figure 52. Dropout on Capital R

It is also important for similar characters (such as "m" and "n") in a font to be rendered in a similar way. For example, stems of letters should be same thickness. This does not always occur when rounding causes inconsistencies.

Performance: As well as the quality problems that can occur, outlines can sometimes be slower to display on screen than bitmaps. As stated earlier, displaying a bitmap font simply means copying the appropriate bytes to the screen.

Outline fonts, on the other hand, require a conversion process into the appropriate bitmap before displaying, involving the following steps:

1. Scale co-ordinates.
2. Calculate lines and curves to connect co-ordinates.
3. Draw outlines using lines and curves and fill with dots at appropriate resolution.

Therefore, bitmap fonts should, in theory, be faster than an outline at a given point size and resolution (providing the bitmap is available in that point size). This is usually the case, but font caching can make the difference much smaller (see section 11.5.5.2, "Font Caching" on page 104).

11.5.5 Intelligent Fonts - How They Work

This is why vendors like Adobe, who produce outline fonts, have developed a more "intelligent" scaling process to reduce the imperfections caused by rounding errors. Central to this process is the use of hints.

11.5.5.1 Hints

Hints tell the scaling routines what features in a typeface are most important to preserve, so that it looks good at all resolutions, and give especially improved results at low resolution such as on displays and lower resolution (dot-matrix) printers.

Hints include global information that would apply across all characters in the font; for example, the desired heights of uppercase and lowercase, the distance rounded edges are allowed to overshoot past flat edges ("O" versus "L") and the widths of stems and serifs on letters like "n" and "m". There would also be hints for particular characters, to allow the rasterizer to apply global hints to those characters, and thus avoid the problem with the "H" described earlier, by ensuring both vertical stems of the "H" were of equal width (even if this means the character has a "thinner" look).

Hints, therefore, make the characters in a font of higher quality and more consistent with each other, when scaled to lower resolutions.

11.5.5.2 Font Caching

Font caching addresses the speed issue for outline fonts. Once bitmaps have been generated from the outline ("rasterized"), they can be stored in memory (a "cache") for future use, therefore making subsequent uses of that font much faster, as the rasterization process is not repeated.

This technique is used in printers using Adobe's PostScript interpreter. Have you ever noticed that when printing a document using two or three fonts on a PostScript printer, the first page takes a little time to print, - while the rasterizer generates and saves the bitmaps from the outline definitions - but subsequent pages are printed much quicker?

Efficient font caching can make outlines display and print as fast as bitmaps.

11.5.5.3 Adobe's Intelligent Font Technology

Many of the problems described above with outline fonts occur in earlier releases of OS/2 Presentation Manager, because the outline fonts used in previous releases were not hinted. The designers of OS/2 Version 1.3 wanted to include a more intelligent scaling process incorporating hints, but first they looked at what technology was already available: the result was that IBM chose Adobe's.

The Adobe technology used in OS/2 Version 1.3 is Adobe Type Manager (ATM) and the Adobe Type 1 format.

ATM is a font rasterizer (a rasterizer converts outlines to bitmaps) that has added intelligence. It builds type "on the fly", that is on request, from Type 1 outline fonts. This means that applications are not limited just to the pre-defined bitmap fonts that are available. Therefore, ATM basically performs the same font-rendering functions as the PostScript interpreter inside a printer, but of course not located in the printer, but in the computer memory (see the diagram below):

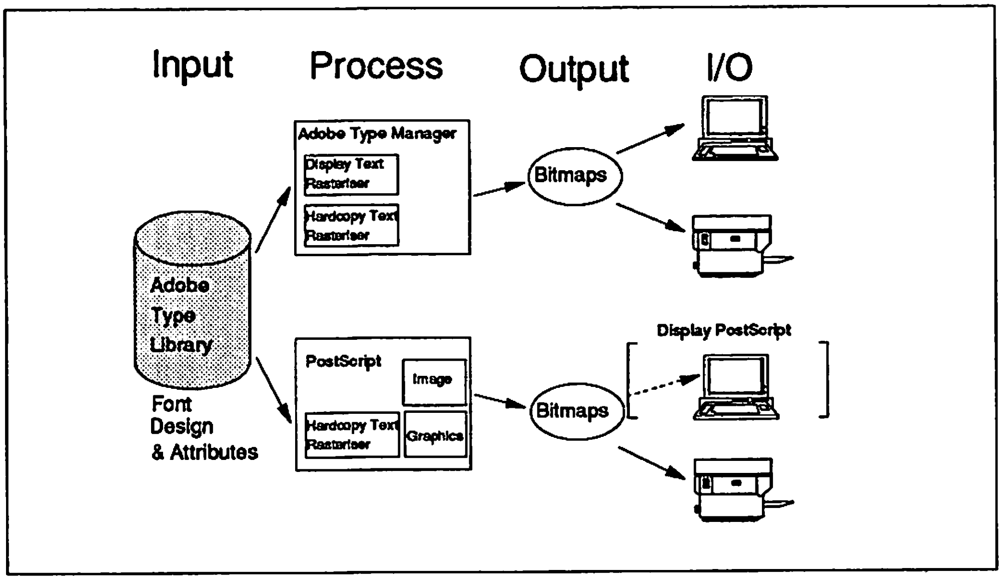


Figure 53. ATM and PostScript Rasterising Technologies

Type 1 is simply a particular format defined by Adobe, who now license tools to other type foundries for them to create their own fonts in the same format (see section 11.2.3, "Other Suppliers of Type 1 Fonts" on page 78).

No confusion must be made; PostScript or Display PostScript is **not** part of OS/2 V1.3. Adobe Type Manager is the sub-part of PostScript that transforms a request to print or display a text in a certain size and typeface in a bitmap adapted to the screen or printer resolution (often called a rasterizer).

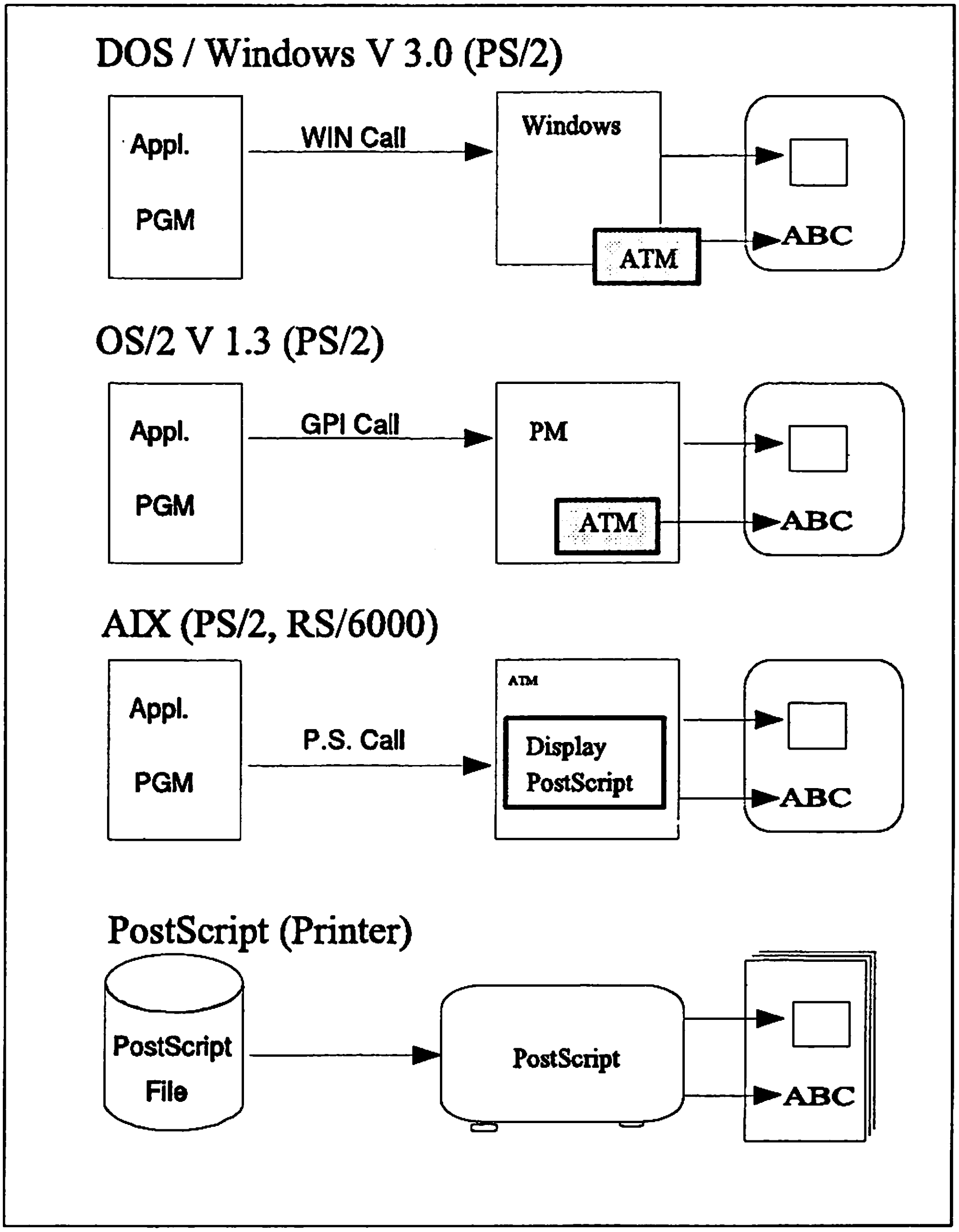


Figure 54. ATM and Display PostScript. ATM's Place in the Different Systems

The PostScript language was designed by Adobe as a high-level programming language to describe, in a device independent standard, a formatted page of data that can be output to an all-points-adressable device, like laser printers.

PostScript is an interpreted language, like Basic or APL, so an interpreter must exist in the system that controls the output device (laser printer, screen, or other devices). OS/2 V1.3 doesn't provide any PostScript interpreter. ATM can only interpret Type 1 fonts. As a matter of fact, they are defined with a subset of the PostScript language.

11.5.5.4 What Is Display PostScript?

As under OS/2's Presentation Manager, the Display PostScript system provides a device-independent imaging model for displaying information on a screen. In Display PostScript, this imaging model is fully compatible with the imaging model used on PostScript printers. It frees the programmer from display-specific details such as screen resolution and the number of available colors. Display PostScript is part of a window system (like Xwindow under AIX). The application uses the window system's features for window arrangement and Display PostScript to take care of imaging tasks inside the window.

Display PostScript has no new language. The heart of Display PostScript is a PostScript interpreter with some extensions. These extensions fall into a few major categories:

1. Memory management has been added to allow more than one task to call Display PostScript concurrently.
2. The language has been extended to optimize executions and to serve the special needs of the window system.
3. An interface library, called Client Library, as been added to transform the application call to a PostScript language input stream.
4. A translator, called pswrap, is provided to create C callable routines from PostScript ASCII source code.
5. To optimize memory and CPU usage, the interpreter accepts a new language binary encoding that can be mixed with the normal ASCII PostScript source code.

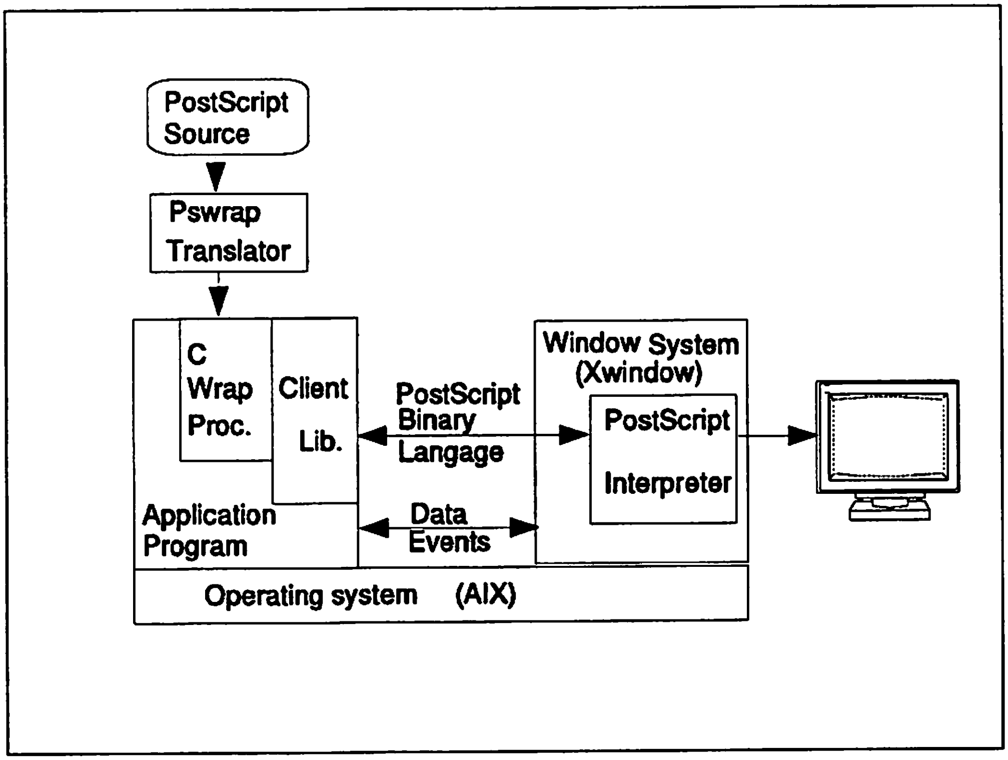


Figure 55. The Display PostScript System

11.5.5.5 IBM's Use of PostScript

IBM has been working with Adobe since 1987, when the first PostScript software licence agreement was signed. Currently Type 1 fonts are supported by:

4216, 4019, 4019E	PostScript printers
Document Composition Facility (DCF 3.2)	Document composition and formatting software
370 AFP PostScript (VM, MVS)	PostScript interpreters
Display PostScript for AIX/X-Windows (RS/6000) AIX/NeXtStep (PS/2, RS/6000)	Display PostScript

11.5.5.6 Type 1 Fonts Versus OS/2 V1.2 Outline Fonts

Speed:

Tests show that the new Type 1 outline fonts under OS/2 V1.3 display half as fast as V1.2 bitmapped fonts, but about ten times as fast as the V1.2 outline fonts.

The addition of improved outline font technology will require some additional memory, but this will be offset by ATM's caching ability and the general benefits of the font technology (more choices, better quality, higher speed).

Type 1 fonts are compressed, so less memory is used and the "hints" are included with the font file. This means less disk space is used with the new Type 1 fonts than the previous outline fonts, and substantially less than the bitmap fonts used in Version 1.3 (see section 11.5.4.3, "Outlines Take Less Memory and Disk Space" on page 99).

However, bear in mind that no matter how good the caching mechanism, use of a large number of different fonts on the same page (considered poor typographic practice anyway) may have a negative effect for overall display and printing performance.

Quality:

The quality difference between OS/2 V1.3 and V1.2 is obvious. Especially with characters like "H", "O", and "e". The 1.3 Versions seem to be lighter, but of more uniform quality. The V1.2 fonts look squashed and have irregular stem widths (for example "H") and show evidence of pixel fill-in (see the "e" in the Figure 36 on page 81).

11.5.5.7 Type 1 Fonts Versus XGA Antialiased Fonts

Antialiased fonts are a set of bitmap fonts that are shipped on the diskettes that accompany the XGA hardware and will work with OS/2 V1.3 on XGA adapters. These fonts can only be used on systems that have an XGA adapter with 512 KB of video memory configured as 640 X 480 with 256 colors or with the 1 MB video memory XGA adapter. These fonts have been designed to offer improved appearance and readability of text displayed on CRT display screens. Character pixels are not displayed in two colors as the OS/2 bitmap or Type 1 fonts are, but with a range of eight color shades per pixel. For more information see Chapter 12, "Antialiased Fonts" on page 113.

Speed:

The antialiased fonts are bitmap fonts. Outline fonts can hardly be as fast as bitmap fonts because they request more CPU time to be converted to the screen resolution.

As they use three bits per pixel, they should be three times slower than the one bit per pixel bitmap font. However, they are optimized for their usage on XGA adapters only and therefore take advantage of some hardware features on this adapter.

Quality:

Studies have shown that, although human reading speed from CRT displays is slower than from printed output, anti-aliased typefaces can allow much of the lost screen-reading speed to be regained. Results will vary from person to person; therefore, the choice of the most appropriate font depends on the individual, the application and the display.

The antialiased fonts have been specially designed for black-on-white text display screens. They are approaching the printed font quality on XGA, but inherit all the problems of the bitmap fonts, like needing to be redefined for each output device resolution. As they use a set of color shades to be defined, they can hardly be transposed to black and white printers which have to use another kind of font.

Applications that request a lot of screen reading from the user must consider these fonts in their solutions (see section 12.3, "Guidelines on Using Antialiased Fonts" on page 116).

Note

These fonts are not recognized by the ATM and, therefore, cannot be printed as they appear on the screen.

11.5.5.8 Adobe Font Systems in Other Environments

ATM for the Apple Macintosh was the first implementation of ATM by Adobe, and has been available since October 1989 as a separate product.

ATM for Microsoft Windows V3.0 has been available since November 1990, and will be supplied by Adobe. Unlike OS/2, ATM for Windows has not been incorporated into the core graphics API, and will be supplied as a separate product.

OS/2 and Windows users will be able to supplement the basic group of fonts (SAA Core Fonts) with seven other font families, by purchasing the Adobe Plus Pack. This supplies an additional 22 fonts in the following seven families:

- ITC Avant Garde Gothic[34]

[34] ITC Avant Garde Gothic is a registered trademark of International Typeface Corporation.

- ITC Bookman[35]
- Helvetica Narrow
- New Century Schoolbook
- Palatino[36]
- ITC Zapf Chancery[37] Medium Italic
- ITC Zapf Dingbats[38]

These 22 fonts, together with those supplied with ATM, or with OS/2 version 1.3, supply the full set of fonts found in most PostScript printers from Adobe. It is already available for the Apple Macintosh and under Windows V3.0, also in IBM PC format.

Adobe fonts are also packaged in various other forms, from the Adobe Type Library, which usually includes one or two fonts in up to four faces each, to collections of three to six font families in packages like Adobe Type Set[39] or Adobe Publishing Packs.

The Adobe Type Library is available in Macintosh, DEC and IBM 370 formats as well as IBM PC format.

11.6 The Intelligent Font Interface (IFI)

The Intelligent Font Interface allows other font technologies to be supported as they emerge, so that users are not restricted solely to the Adobe technology, but can take advantage of others if they provide additional benefits. In this way, IBM is providing "open" font support as well as supporting existing standards.

Although IBM has implemented Type 1 fonts rendering technology in OS/2 V1.3, it has not precluded alternative font support. IBM's use of Adobe technology represents its support for a standard font format that already exists. But IBM is also committed to supporting other font technologies, and will do so in OS/2 via the Intelligent Font Interface that is part of OS/2 V1.3. As Type 1 will give user's standard font support, the Intelligent Font Interface will supply open font support.

The Intelligent Font Interface is the mechanism for installing font rasterizers other than the ATM in OS/2 V1.3. The Intelligent Font Interface is not formally documented and won't be formally announced, but is available to vendors who n׀ay want to produce font rasterizers.

No Change to PM Application Programs

OS/2 applications will not need to be aware of the technology difference between the fonts when multiple technologies are used. And, more importantly, nor will the users.

[35] ITC Bookman is a registered trademark of International Typeface Corporation.

[36] Palatino is a registered trademark of Linotype Company.

[37] ITC Zapf Chancery is a registered trademark of International Typeface Corporation.

[38] ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

[39] Adobe Type Set is a trademark of Adobe Systems Inc.

11.6.1 How It Works

The Intelligent Font Interface allows other font vendors to implement "drivers", which link through the interface to the Presentation Manager API, allowing its fonts to be declared to PM and, therefore, available to applications that ask for fonts through normal PM calls, such as GpiQueryfonts. This means that Type 1 fonts and fonts from other technologies can be used in the same way.

During IPL, OS/2 attempts to load all the fonts listed in OS2.INI. As well as the fonts themselves, there is also a list of "font engines" in OS2.INI. Currently, there is only one, PMATM.DLL. During the loading process for a font, the Intelligent Font Interface asks each font engine whether the font that is being loaded is in its format or not. If so, then it gets loaded, if not, then the IFI asks the next font engine, and so on.

11.6.2 Intelligent Font Interface and Microsoft's TrueType

TrueType (sometimes called Royal font technology), developed by Apple and Microsoft, is a different way to describe fonts. Instead of creating another high level language, like PostScript, it provides a low-level language, a sort of assembly language specialized to describe fonts.

TrueType and Type 1 fonts are outline fonts. TrueType fonts are not defined with a series of drawing instructions as Type 1 fonts are, but as a collection of control points. These control points are used by the TrueType statements to draw curves. These curves, when connected, form the character outline.

The hinting code in the TrueType fonts influence the final result by moving the control points.

TrueType is really two languages: a basic stack-oriented langage to define and manipulate the points and a set of high level routines to manipulate outlines. This allows TrueType to be more flexible. TrueType is a lower level language than PostScript. A PostScript interpreter could be written in TrueType like a BASIC compiler can be written in C.

We can say that Adobe put a lot of intelligence in their rasterizer and TrueType put it in the font description.

OS/2 V1.3 does not implement TrueType (for the simple reason that there is no version of TrueType available yet, and certainly no quantity of typeface styles) but the Intelligent Font Interface is ready to accept this new font technology as soon as it emerges.

Chapter 12. Antialiased Fonts

This chapter covers a new capability that the higher resolution of the new video subsystem (XGA adapter) will exploit. This is the ability to use antialiased fonts for system and application displays.

12.1 OS/2 Software Considerations

One of the considerations in choosing between the 512 KB and the 1 MB video memory option for the XGA adapter lies in whether you elect to install and use antialiased fonts.

Note: In some countries the XGA adapter may only be available with the complete 1 MB video memory already installed.

The major benefit of an antialiased font is an improved reading speed capability that approximates that of printed material. As more and more product documentation, online help and manuals are delivered on magnetic media, it is increasingly important that we can access and read such information as quickly as, or even more quickly, than we can in a printed manual. The following two figures help to show the differences between existing fonts and antialiased fonts.

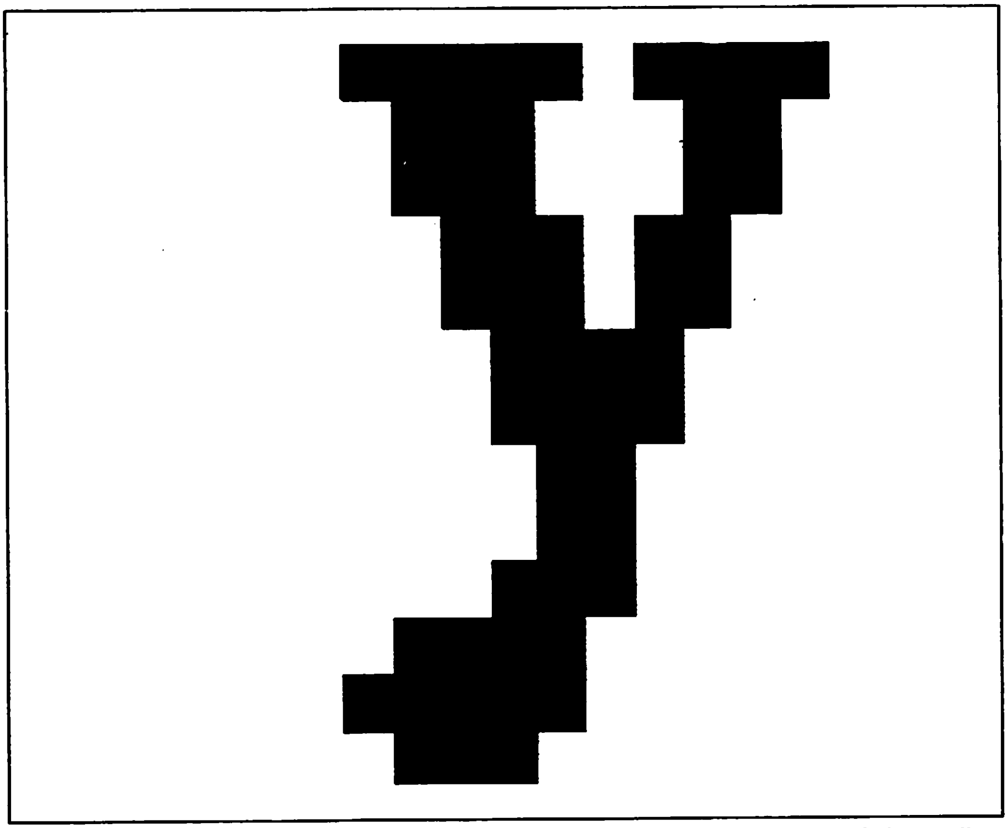


Figure 56. Times Roman lower case "y". Notice the jagged appearance of the outline. This has been the character set used by text applications to date.

This first example is taken from the actual Presentation Manager Times Roman font. Each black square is an enlargement of a pixel. The second example is taken from the antialiased Times Roman font. It is the same letter, at the same point size. Each square is an enlargement of a pixel. Some squares have been modified manually to approximate gray scales for the printing process.

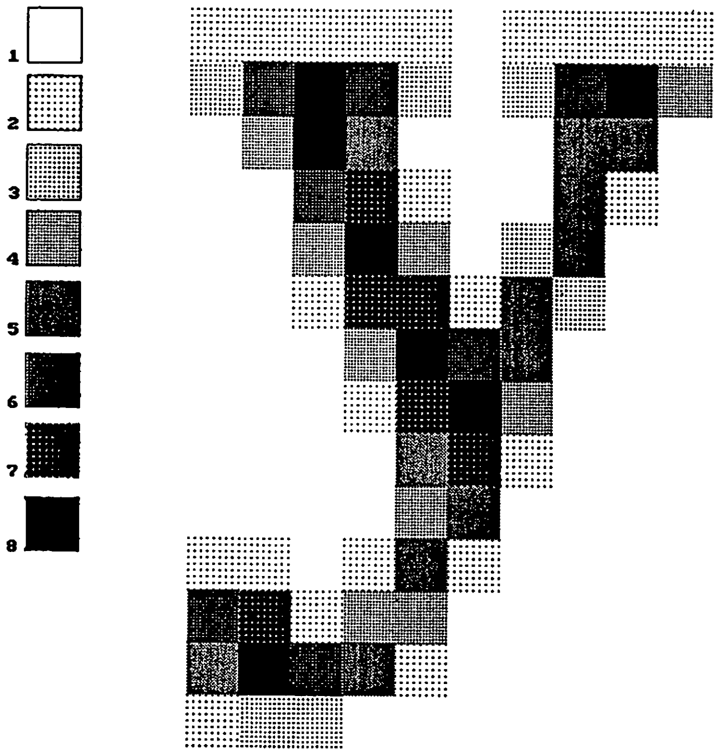


Figure 57. Antialiased Times Roman lower-case "y". This is an approximation of the eight levels of intensity used in the antialiased font.

12.2 Guidance for Users of Antialiased Text

12.2.1 Introduction

This font software supports antialiased text, which offers improved appearance and readability of text displayed on CRT display screens.

Conventional text on display screens is displayed as an array of dots or picture elements (pixels or pels). Each pixel is displayed in one of two colors: a foreground color - the color of the text characters - or a background color. With antialiased text, pixels can be displayed using a range of color shades (up to eight in this implementation) that can have values between the foreground and background color. Antialiased fonts are designed to use this capability to improve the appearance and readability of text.

Antialiased fonts are available on systems that support 256 colors as follows:

Video Memory	Configured As	Antialiased Fonts
1 MB	1024 x 768 with 256 colors (default)	Available
512 KB	640 x 480 with 256 colors	Available
512 KB	1024 x 768 with 16 colors	Not Available

Table 6. Availability of Antialiased Fonts by Configuration

You can select an antialiased system font after you have installed the device driver and fonts as follows: In the OS/2 Desktop Manager, select the *screen setup* option. Then, in the screen setup group, run *select system font*. This program lets you view antialiased system fonts and compare them with the normal system font before deciding which to use. You can rerun the program at any time if you want to select a different font.

Note: Antialiased fonts use an optimized set of colors that are loaded by the device driver. If you use a program that changes the default set of colors loaded by the device driver (by using a Realize Color Table call), the color shades selected for antialiased fonts will be incorrect. You should, therefore, not use antialiased fonts with programs that change the default colors. For example, you should observe the effect of using the OS/2 PM Control Panel to change the window background to a dark shade.

The remainder of this section gives background information about the display of text, describes some of the advantages of using antialiased fonts, and explains how to install and to use them. It is optional reading for end users with an interest in antialiased fonts. It will be of particular interest to application programmers who plan to use antialiased fonts in their applications.

12.2.2 Background

Typeface design has been studied and taught for over 500 years. Over the years, printed typefaces have evolved into two distinct classes, display faces and text faces designed for general reading. The text faces were refined over the years, mainly to optimize their appearance and readability when printed on paper. Reading text from a video display screen is a recent mode of human communication. Cathode ray tube (CRT) displays with their typefaces have been used to produce text for about three decades. This new technology has advantages and limitations never considered for ink on paper typeface design. We know that human reading speed from CRT displays is slower than from printed output. Despite many improvements to display technology, the difficulty of reading from screens, until recently, remained an unsolved problem. However, reports now show that some typefaces that have had the antialiasing technique applied to their design allow much of the lost screen-reading speed to be regained. Antialiasing technology enables precise control of line width, line placement, and "color" (also known as line density). This control permits diagonal lines to be shown as visually straight lines without a jagged (or stair-step) appearance. It also allows the typographic designer to smoothly control line width. With antialiasing techniques, a typographer can design CRT typefaces that are close visual matches to some of the well-known classical printed designs. Photometric measurements have quantified this control of placement width and shape. Antialiased typefaces restore some of the high spatial frequencies that are lost on CRT displays.

Many factors should be considered when selecting a typeface for use with a software application. Among these factors are typestyle (serif or sans serif), size (10 point or 12 point), and color or polarity (black characters on a white background or white characters on a black background). The ability to select different styles, sizes, and colors is built into this software to permit the users the flexibility of choosing those characteristics they need. The typeface designs included with the antialiasing fonts were created for the purpose of optimizing the appearance and text readability characteristics. Tests have shown readability and preference is most improved with the antialiased Roman face using black characters on a white background.

12.3 Guidelines on Using Antialiased Fonts

A number of complex factors are involved in the choice of a font for your display, quite apart from the availability of fonts. The best choice depends on you, your applications, and your display:

- Each individual has different oreferences because of previous experience and eyesight.
- The display you use, particularly its sharpness, affects the appearance of the fonts.
- The application (that is, how your software displays the font and what you use it for) can also have a marked effect.

If you are an end user, select the system font by viewing the examples in the Select System Font program. Choose the one that looks best to you, on your display, in the colors that you and your applications use most often. If you are using an application that gives you a choice of fonts, you should adopt the same approach. Also, application developers should design applications that offer a choice of fonts (including antialiased) to end users. When making the choice of font, remember that:

- The antialiased fonts are designed to be close in appearance to printed fonts that you may use often.
  - Consider a match to your favorite printer font, for example.
- Text applications can be expected to benefit most from the typographical qualities of antialiased fonts.
  - Non-text applications, for example, programming with many codes that need recognition of individual characters, may be better with normal non-serif fonts.
  - Consider whether the text looks smooth or jagged, bold or thin, sharp or fuzzy. These factors depend on you and the display, so different people make different choices.
- The antialiased fonts are optimized for black-on-white display.
  - If you use much white-on-black text, look at white-on-black examples carefully before deciding.
  - Other color combinations will also work, but check them out.
- Make the choice on the display that you will actually use.
- If you are setting up for someone else, get them to make the choice of font.

- Remember that you can run the Select System Font program at any time, so that you can review your choice of system font if your application changes or you think another font might be better.

12.3.1 Available Antialiased Fonts

Antialiased fonts are provided for use as OS/2 Presentation Manager *system* fonts and OS/2 Presentation Manager *application* fonts.

- The *system* font is used by Presentation Manager in window components such as title bars and menus, in help panels, and as a default font for applications that do not explicitly select a font. Four alternative antialiased system fonts are supplied on the device driver diskette:

  10 point Swiss

  12 point Swiss

  10 point Roman

  12 point Roman.

  Swiss fonts are sans serif, Roman fonts have serifs.

- The *application* fonts are available for selection by application programs. Antialiased application fonts provided are similar to the normal image fonts provided by the OS/2 Presentation Manager. They are in the same font families, styles and sizes as normal Presentation Manager image fonts.

32 antialiased application fonts are provided on the device driver diskette in the following sizes:

Roman normal:	8, 10, 12, 14, 18, 24 point
Roman italic:	8, 10, 12, 14 point
Roman bold:	8, 10, 12, 14 point
Swiss normal:	8, 10, 12, 14, 18, 24 point
Swiss italic:	8, 10, 12, 14 point
Swiss bold:	8, 10, 12, 14 point
Courier normal:	8, 10, 12, 14 point.

Chapter 13. New and Revised APIs

This chapter deals with the new and revised APIs that OS/2 Standard Edition Version 1.3 provides.

These APIs are available and documented with the *OS/2 Programming Tools and Information V1.3* package.

13.1 Overview

The following is a description of the functions in the OS/2 Control Program Application Programming Interface (API) which are added or enhanced by OS/2 Standard Edition Version 1.3.

Detailed API function call information may be obtained from the ***Control Program Programming Reference***.

13.2 Direct Manipulation

The SAA and CUA *Advanced Interface Design Guide* has established a definition of what is known as the workplace model. A key feature of the workplace user interface is the ability to drag objects across the screen and drop them on other objects in order to execute some function. This ability is known as **direct manipulation**.

Previously, direct manipulation was limited to a single application or to members of a group which were programmed to be aware of each other's presence. This approach forced applications to define their own protocols. The presence of a diverse group of interfaces thereby inhibited integration on a system-wide scale.

In OS/2 V1.3, a protocol that is defined by the system and a set of APIs enable applications, that know nothing of each other, with the ability to utilize direct manipulation. Direct manipulation is now a feature of the system rather than of the application.

In addition, a complete PM sample application is now provided with the *OS/2 Programming Tools and Information V1.3*.

13.2.1 Direct Manipulation APIs Added, Revised and Deleted

Note: Though these APIs are new to the OS/2 programmer, parts of them have been available under OS/2 V1.2 but have never been published.

DrgDrag() Two reserved fields were added to allow for future changes to the user interface.

DrgQueryStrName() Replaces DrgQueryStrHandle().

DrgQueryStrNameLen() Replaces DrgQueryStrHandleLen().

DrgAllocDragTransfer() Memory allocation API for the DRAGTRANSFER structure.

DrgFreeDragTransfer()	Memory deallocation API. Complements DrgAllocDragTransfer().

The following APIs were no longer considered needed due to the dropping of hProtocol and were removed:

- DrgVerifyProtocol
- DrgVerifyProtocolSet.

13.2.1.1 Other Revised APIs

DrgSetDragImage()

Formerly called DrgSetDragPointer().

Changes to DrgDrag() and DrgSetDragImage():

- Bitmaps are now dragged rather than mouse pointers so that the (cx, cy) offset of the mouse pointer hotspot relative to the origin of the bitmap is now free to be passed.
- Another flag (DRG_IMAGEBITMAP) was added to indicate if the caller wishes to make the icon being dragged have a transparent center from which to view the target onto which it is being dropped.
- The user interface to both APIs has been changed to allow more than one icon or bitmap to be dragged.
- Polygons can now be dragged as well as bitmaps and icons.
- Icons and bitmaps can be stretched to any size while dragging.

"RendMechFmt"

All references to "RendMechFmt" are now changed to "RMF". The following functions are affected:

- DrgQueryNativeRendFmt()
- DrgQueryNativeRendMechFmtLen()
- DrgVerifyNativeRendMechFmt()
- DrgVerifyRendMechFmt().

13.2.2 Direct Manipulation Messages Added and Revised

WM_DDE_REQUEST	This message is used to establish the conversation (not WM_DDE_ADVISE).
DM_PRINTSELF	This message will be sent when an item is dropped onto a printer.
DM_RENDERPREPARE	This message is used to alert the source (object being dragged) of an upcoming data transfer.
DM_ENDCONVERSATION	Replaces DM_RENDERCOMPLETED to avoid being confused with DM_RENDERCOMPLETE.
DM_PRINT	Formerly DM_PRINTSELF.

13.2.2.1 Direct Manipulation Sending and Posting of Messages

DRAG_LEAVE

DRAG_LEAVE will always be sent to allow the use of target emphasis in order to provide the user with a visual cue that shows where it is permissible to do a drop. The target can be emphasized even when the drop is not allowed.

The following messages will be **sent**:

- DM_RENDERPREPARE
- DM_RENDER
- DM_ENDCONVERSATION
- DM_PRINTSELF.

The following message will be **posted**:

- DM_RENDERCOMPLETE.

13.2.3 Direct Manipulation Fields Added and Revised

hAppName

Removed from the DRAGITEM structure as it is no longer needed to establish a conversation for a drag operation.

pSharedSeg

Added this field to the DRAGTRANSFER structure to pass the address of a shared memory segment when the selected rendering mechanism is DRM_SHAREDMEMORY.

pDragInfo

This parameter was added to allow including the string handles as pointers in the future but will not be used in this release of drag. The following functions were affected:

- DrgAddStrHandle
- DrgDeleteStrHandle
- DrgQueryStrName
- DrgQueryStrNameLen.

hProtocol

This field was removed from the DRAGINFO structure in favor of ordered pairs, i.e. <rendering mechanism, format>.

13.2.4 Drag Drop from the Users View

The most important part of OS/2, the user interface, is the Presentation Manager (PM). Under Version 1.2 and Version 1.3, PM has become more object-oriented than in the previous version. With V1.2, the "pick, drag, and drop" technique was incorporated.

Notes:

1. The old "point and click" technique is still used to select an object and/or start a program.
2. Under V1.1, only mouse button 1 was used for any type of interaction. Under V1.2 and V1.3, button 1 is used for selecting an object (clicking on it), and button 2 has to be used for manipulating an object (dragging it).

As shown in Figure 58, objects on the Presentation Manager desktop can now be moved (dragged) around the screen. The action that will be performed on those objects is dependent on where they are released (dropped). That means, when the user selects an object, PM doesn't know at that moment what the user wants to do with it. That decision is made when the user releases that object. This kind of application design is also called "event-driven" programming.

Some samples are given in the following picture and will be discussed below:

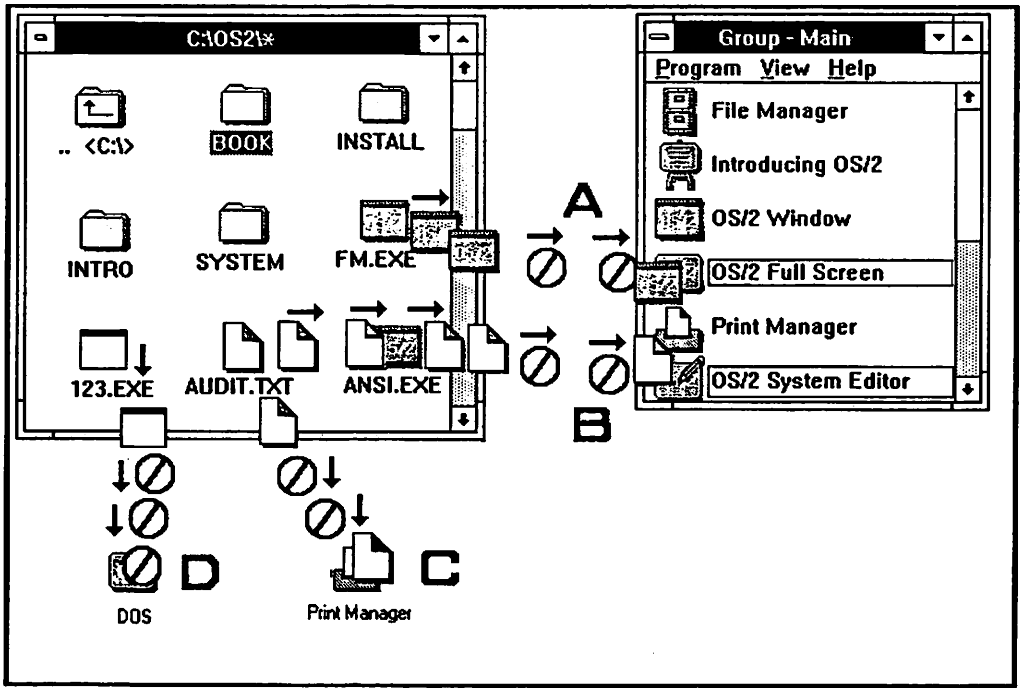


Figure 58. Object-Oriented Interaction with the OS/2 Presentation Manager

There are basically three different ways of performing any kind of action on an object:

1. Within an application
2. With the Desktop Manager
3. Through the PM message interface.

Method one:

In this case, it is the application's own responsibility to handle that interface function. The programmer decides which kind of operation can be performed. As a sample, take a look at the File Manager.

Method two:

As mentioned before, this method has now been implemented in the Desktop Manager. The implementation at this time is relatively simple. The user can drop an object over any of the entries in either the Desktop Manager or one of the groups. PM simply tries to start that program, which was found (installed) at that place and passes the object as an argument. Currently that name would be a file or program name. Also, the "provider" of such a filename to pick up can only be the File Manager. In the future, other applications may be enhanced to support this interface.

Note: If the user picks up anything within the Desktop Manager itself or one of its groups, PM thinks that the user wants to rearrange the entries. Following are some sample scenarios:

- Dropping a program file (executable) over the "full screen" icon (Example A, Figure 58 on page 122).

  PM starts another screen group and loads the OS/2 command interpreter "CMD.EXE". If the argument is an executable program, it just gets executed by the command interpreter. The same thing would happen, if an OS/2 batch file or REXX procedure is passed.

- Dropping a non-program file over the "full screen" icon.

  PM starts another screen group and loads the OS/2 command interpreter "CMD.EXE". Then the OS/2 command interpreter tries to execute the file which was passed as an argument. This attempt will not be successful. The user receives an error message and will be returned to the OS/2 command prompt.

- Suppose the same non-program file is dropped over the System Editor (Example B). That will make sense.

  PM loads and runs the System Editor. Then the filename is passed as an argument for the System Editor. The System Editor accepts any filename, passed as a command argument, and tries to load and edit this file.

  If the file was a plain ASCII file, like a C-source code file, the System Editor will be successful. It pops up as a new window under PM and displays that file.

  If the file was a binary file, like an executable program, the System Editor stops loading that file and displays a warning message, because it doesn't make sense to edit such a file.

As an example, the user can write a simple batch file that contains only one line:

```
ERASE %1
```

Then that batch file is installed under the Desktop Manager.

When a file object is dropped over the icon of that batch procedure, it gets immediately deleted. In addition, if the user wants to tell the Desktop Manager where to insert the name of the file represented by the icon that is being dropped on a Desktop Manager entry he can use the "%*" parameters in the Parameters field. If an entry is selected without an icon drop, then the "%*" is ignored by the Desktop Manager when it starts the application.

Here is an example of using the "%*" parameter as an entry in the Desktop Manager:

```
Program Title: My Favorite Editor

Path and file name: C:\EDIT\EDITOR.EXE

Parameters: %* /NOTABS
```

If the user drops the icon for the file ABC.TXT over this entry, the Desktop Manager starts the program as follows:

```
C:\EDIT\EDITOR.EXE ABC.TXT /NOTABS
```

In the above example, if the user simply double-clicks on this entry, the Desktop Manager starts the program as follows:

```
C:\EDIT\EDITOR.EXE /NOTABS
```

Method three (Example C):

This method of accepting objects is performed via the Presentation Manager message interface.

When an object is dropped over a running application, this application receives a message from PM. It doesn't matter if this application window is minimized (iconized) or maximized at that moment. It is then the responsibility of that application to interpret that message and react to it in the appropriate way (method).

A first example of that kind of implementation is the new Print Manager. As soon as the user drops a file object over it, the Print Manager accepts that filename as an input and prompts the user for the print queue to select.

The current implementation isn't perfect at the moment, because every filename will be accepted. The result can be worse than in the previous sample with the System Editor.

- If a plain ASCII file is passed, it will be printed without any problems.
- If a binary file is passed, it could cause a lot of trouble on the printer.

A better implementation of the object-oriented approach for instance, would have been to analyze a file ;irst. Based on the results, another program could be called to assist in printing such a file. For example, the Print Manager could look into the extended attributes of such a file. It could load the program that created that file and send a message to that program in order to generate the appropriate printer data stream. Such a behavior assures that all files are printed, based on the appropriate procedures. The user doesn't have to worry about all the different file formats and procedures.

As mentioned before, the System Editor does some file analyzing. Unfortunately it doesn't react like the Print Manager. If the user drops a file object over an open System Editor window, or over its icon on the desktop (not the Desktop Manager), it doesn't react to that kind of interaction. Obviously, it doesn't intercept that "new" PM message.

The only applications that know how to handle this "new" PM message are the Print Manager, the File Manager, and the Desktop Manager.

While dragging an object over an unacceptable area (application), the mouse pointer changes to the "invalid option" icon (example D, Figure 58 on page 122).

While dragging over an application which accepts that kind of input, the mouse pointer changes to another more meaningful icon. The appearance of that icon can be different, depending on the object being dragged and the application, which reacts to that event.

In the past, this "new" message interface was not documented. It is now available to programmers as the new Drag-Drop API. Now a programmer can develop applications which will conform to that new standard of user interaction.

For a more general discussion about that subject, please see *Understanding Object Oriented Techniques.*

13.3 Spin Button Control

The spin button model displays lists or data that are viewed through what is seemingly a slot in a window. Real-life examples are old style gas pumps which rotate digits, digital clocks, and thermostat temperature selectors.

Spin buttons are used to select from a list by scrolling through the consecutively ordered choices.

One example is to set the date as shown in the following figure:

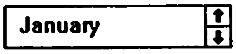


Figure 59. Spin Button

As with the value set, this control must be programmed by the application developer using static and scroll bar windows. The component is created using the WinCreateWindow() function with the following parameters:

Class	WC_SPINBUTTON
Window Style	WS_VISIBLE

Other presentation parameters include:

- One of the following:
  - SPBS_MASTER
  - SPBS_SLAVE
- One of the following:
  - SPBS_ALLCHARACTERS
  - SPBS_NUMERICONLY

- SPBS_READONLY

- One of the following:
  - SPBS_JUSTLEFT
  - SPBS_JUSTRIGHT
  - SPBS_JUSTCENTER
- Any of the following:
  - SPBS_NOBORDER
  - SPBS_FASTSPIN
  - SPBS_PADWITHZEROS.

The component is destroyed by a call to WinDestroyWindow() with the component handle returned at creation time as the input parameter.

13.4 Increased Windows Limit

In previous versions of OS/2, a PM application was limited to a maximum of 1200 windows. Keep in mind that one window is actually composed of many smaller windows. Although the theoretical limit was a number greater than 1200 windows, in reality, the system would run out of other resources before getting that far. This limitation was a restriction for some applications.

With a change in the process for dynamically linking to PMWIN.DLL, OS/2 V1.3 is able to break the 1200 window barrier. Obviously, an application must dynamically link to PMWIN.DLL to call its Application Program Interfaces (APIs).

The limit therefore disappeared totally and rather depends now on the availability of other system resources, such as memory.

There might also be a design implication for existing PM applications. The information in the top word of the window handle is now required! Under OS/2 V1.2, that information was rarely used. In such a case, the application would cause a Trap-D. However, this is simply one of many ways in which an application can crash itself. As the rest of the system keeps running, it is less serious than many other ways of causing a crash.

13.5 Print Spooling

The spooler for OS/2 Version 1.3 is now the so called LanMan spooler as introduced by Microsoft for their Lan Manager product. In migration from the OS/2 V1.2 spooler to the LanMan spooler used in OS/2 V1.3, some of the functions of the OS/2 V1.2 spooler were lost. Particularly notable was the loss of the ability for the applications to provide their own **SPL1A** and **SPL2A** dynamic link libraries which were used to hook certain functions of the spooler.

In OS/2 Version 1.3, as a result of a design change to the print subsystem, applications can still hook the lost spooler functions. The design extends the **DosPrint()** API to control the Presentation Manager print spooling system as well as the print spooling system on LAN servers which it had been exclusively devoted to previously.

Most applications will continue to work without modification but they will need to be linked with **OS2.LIB**. The APIs have been moved to **PMSPL.DLL** from

NETSPOOL.DLL. Header files are supplied for other applications that need to be recompiled or linked. No changes need to be made to DosPrint() unless additional function is desired.

13.5.1 Down Level Support

Previous information levels for all API functions continue to be supported. Additionally, new information levels will work when directed to downlevel Microsoft Lan Manager versions and IBM Lan Server versions. The new fields in the APIs will either be filled with a NULL value or will be given a value which will be interpreted as not available.

DosPrintJobEnum

See DosPrintJobGetInfo

DosPrintQGetInfo

Level 3

pszPrProc If no special print processor is specified, LM10 is returned as the print processor.

pszPrinters Returns a list of ports separated by a comma and assumes that the printers have the names of the ports.

pszDriverName NULL

pDriverData NULL

DosPrintQSetInfo

The parameter number constant (PARMNUM), is the last parameter in this API. If either of the following:

- PRO_DRIVERNAME_PARMNUM
- PRO_DRIVERDATA_PARMNUM

are used, ERROR_NOT_SUPPORTED will be returned.

DosPrintQAdd

See DosPrintQGetInfo and DosPrintQSetInfo

13.5.1.1 Fields that Are Unavailable

DosPrintJobGetInfo

Level 2:

uPriority Always PRJ_NO_PRIORITY

pszDocument NULL

Level 3:

pszQProcName NULL

pszQProcName NULL

pszQProcParms	NULL
pszPrinterName	NULL
pszDriverName	NULL
pDriverData	NULL

13.5.1.2 APIs Requiring No Special Down-Level Support

- DosPrintJobPause
- DosPrintJobContinue
- DosPrintJobDel
- DosPrintJobGetId
- DosPrintQContinue
- DosPrintQDel
- DosPrintQPause
- DosPrintQPurge

13.5.1.3 APIs Having No Emulation

- DosPrintDestEnum
- DosPrintDestControl
- DosPrintDestGetInfo
- DosPrintDestAdd
- DosPrintDestSetInfo
- DosPrintDestDel

13.5.2 The DosPrint() API Function

The DosPrint() API provides the following classes of actions:

- Add
- Del
- Enum
- GetInfo
- SetInfo.

13.5.2.1 Add

An **Add** function adds a resource to a particular set of items.

13.5.2.2 Del

A **Del** function removes a resource from a particular set of items.

13.5.2.3 Enum

The **Enum** functions list information about system resources. The parameters include:

- pbBuf
- cbBuf
- pcReturned
- pcTotal.

If **pbBuf cannot** store all returning data, the Enum function returns the error code NERR_BufTooSmall. All data in **pbBuf** is rendered invalid and **pcTotal** will contain the number of resources available.

If **pbBuf can** store all returning data, but not all available variable-length data, Enum returns ERROR_MORE_DATA. The data in pbBuf is valid and pointers to any incomplete variable-length data are set to NULL.

If the value of **cbBuf** is zero, Enum returns a valid **pcTotal** parameter.

13.5.2.4 GetInfo

The **GetInfo** function retrieves information about a resource. The parameters include:

- pbBuf
- cbBuf
- pcbNeeded.

If **pbBuf** cannot store all returning data, GetInfo returns the error code NERR_BufTooSmall. All data in **pbBuf** is rendered invalid but pcbNeeded contains the number of bytes needed to store the information.

If **pbBuf can** store all returning data, but not all available variable-length data, GetInfo returns ERROR_MORE_DATA. The data in pbBuf is valid and pointers to any incomplete variable-length data are set to NULL.

If the value of **cbBuf** is zero, Enum returns a valid **pcbNeeded** parameter.

13.5.2.5 SetInfo

The **SetInfo** function sets the parameters of a resource. The parameter is:

- uParmNum.

uParmNum specifies whether one specific component of the data structure is going to change, in which case the information would be passed in **pbBuf**, or the whole data structure is passed, in which case **pbBuf** is equal to zero.

13.5.3 Print Queue Functions

The following functions are supported for print queues in the PROINFO data structure:

- DosPrintQAdd
- DosPrintQContinue
- DosPrintQDel
- DosPrintQEnum
- DosPrintQGetInfo
- DosPrintQSetInfo
- DosPrintQPause
- DosPrintQPurge.

A much more detailed discussion about OS/2's print subsystem and some sample code can be found in the ITSC bulletin *OS/2 V1.3 Volume 2: Print Subsystem*.

Chapter 14. New Features in Extended Edition

Several major enhancements have been provided with OS/2 Extended Edition Version 1.3.

The main Version 1.3 enhancements are as follows:

- Increased performance in memory-constrained environments. The minimum memory requirement has been reduced from 3.5MB to 3.0MB.

  For memory and fixed disk requirements worksheets, refer to Appendix F, "OS/2 V1.3 Extended Edition Memory and Fixed Disk Requirements" on page 183.

- Improved program load facility
- Communications Manager enhancements for:
  - 3270 emulation font selection
  - 3270 printer emulation text mode support
  - Batch Configuration utility
  - Asynchronous communication support (ACDI redirection)
  - ETHERAND* LAN support for LUA applications
- Database DOS Requester support for ETHERAND LAN.

This section will provide information about some of these enhancements.

14.1 Program Load Facility

The program load facility of OS/2 Extended Edition Version 1.3 has been improved.

For many applications, it provides a performance enhancement of two to three times faster than OS/2 EE Version 1.2 when loading programs from LAN Servers to LAN Requesters.

The performance enhancement for any given application may vary outside this range, as it is influenced by the amount of code the application preloads at start-up time.

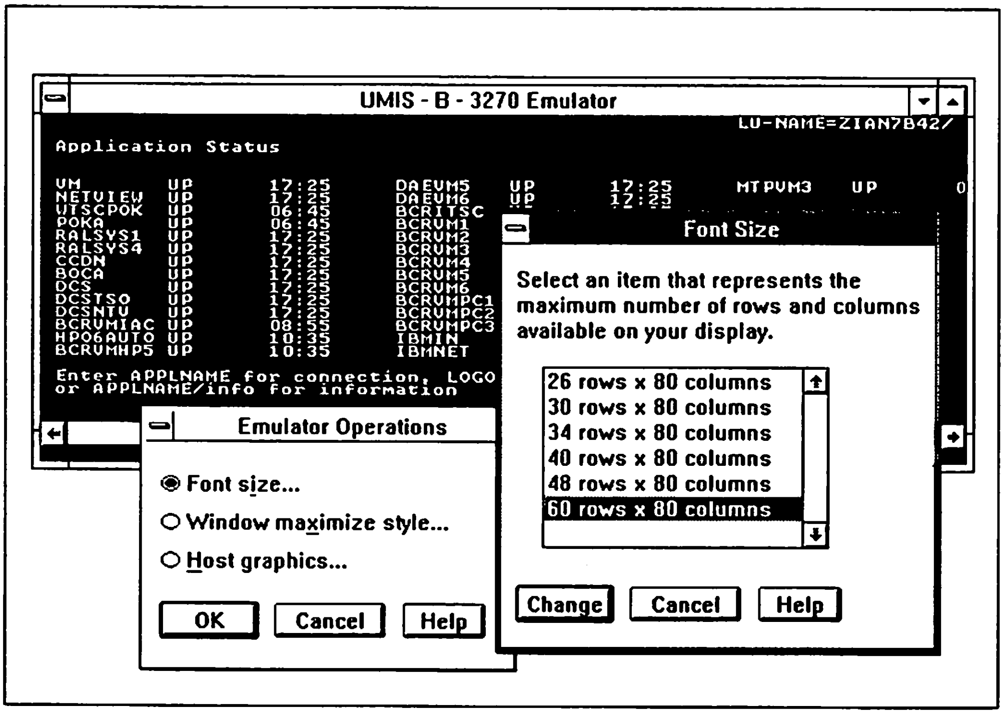
For the purpose of the OS/2 LAN Server performance test cases, we made tests with a common market spreadsheet. The amount of code loaded through the LAN was approximately 800KB for an OS/2 Extended Version 1.2 LAN Requester and 400KB for an OS/2 Extended Version 1.3 LAN Requester. If the LAN Requester workstation uses OS/2 Extended Edition V1.3, the amount of application code loaded from an OS/2 LAN Server (Version 1.2 or Version 1.3) is the same as if the application was installed locally on the workstation.

* ETHERAND is a trademark of the International Business Machines Corporation.

14.2 3270 Emulation Font Selection

OS/2 Extended Edition V1.3 gives the user more choices for the font in their 3270 emulation sessions. The number of font size options for a regular VGA display increased from two with Version 1.2 to six in Version 1.3. This enhancement gives the user the choice of fonts ranging from very compact (Figure 60) to very large (Figure 61 on page 133), allowing full utilization of the entire screen.

This can be done by selecting **Emulator Operations...** from the system icon on the emulator session.



UMIS - B - 3270 Emulator

```
                                                    LU-NAME=ZIAN7B42/
Application Status

VM         UP       17:25     DAEVM5     UP      17:25     MTPVM3    UP      0
NETVIEW    UP       17:25     DAEVM6     UP      17:25
VTSCPOK    UP       06:45     BCRITSC
POKA       UP       06:45     BCRVM1
RALSYS1    UP       17:25     BCRVM2
RALSYS4    UP       17:25     BCRVM3
CCDN       UP       17:25     BCRVM4
BOCA       UP       17:25     BCRVM5
DCS        UP       17:25     BCRVM6
DCSTSO     UP       17:25     BCRVMPC1
DCSNTU     UP       17:25     BCRVMPC2
BCRVMIAC   UP       08:55     BCRVMPC3
HPO6AUTO   UP       10:35     IBMIN
BCRVMHP5   UP       10:35     IBMNET

Enter APPLNAME for connection, LOGO
or APPLNAME/info for information
```

Emulator Operations

● Font size...

○ Window maximize style...

○ Host graphics...

OK | Cancel | Help

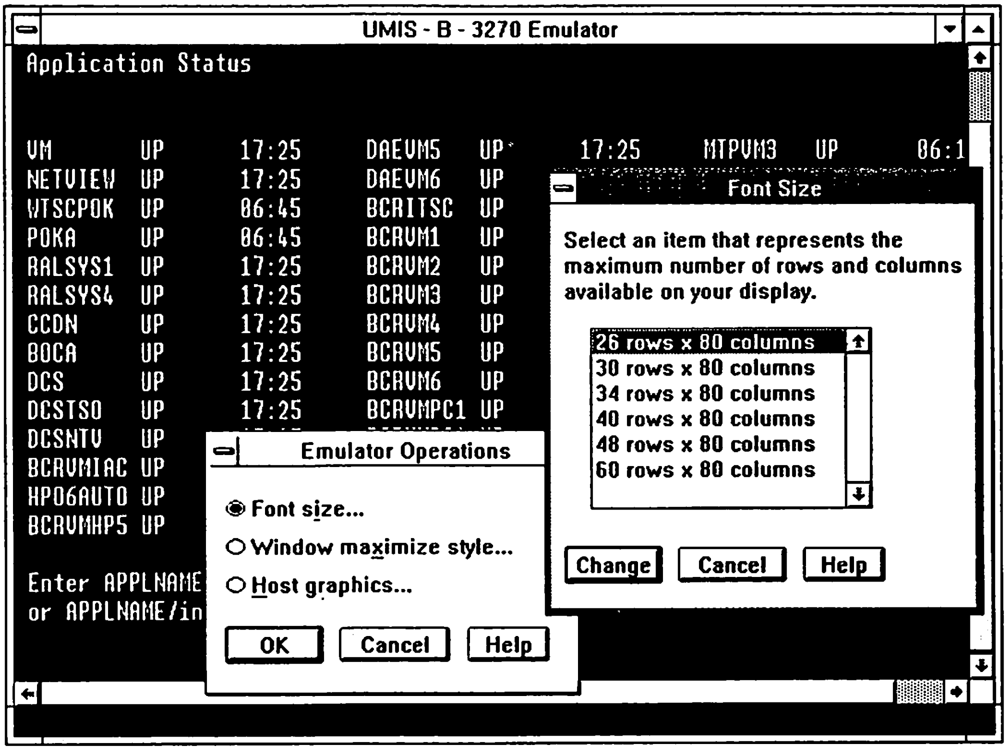
Font Size

Select an item that represents the maximum number of rows and columns available on your display.

26 rows x 80 columns
30 rows x 80 columns
34 rows x 80 columns
40 rows x 80 columns
48 rows x 80 columns
60 rows x 80 columns

Change | Cancel | Help

Figure 60. Example of Most Compact 3270 Emulator Font



```
Application Status

VM        UP   17:25   DAEVM5     UP   17:25   MTPVM3   UP   06:1
NETVIEW   UP   17:25   DAEVM6     UP
WTSCPOK   UP   06:45   BCRITSC    UP
POKA      UP   06:45   BCRVM1     UP
RALSYS1   UP   17:25   BCRVM2     UP
RALSYS4   UP   17:25   BCRVM3     UP
CCDN      UP   17:25   BCRVM4     UP
BOCA      UP   17:25   BCRVM5     UP
DCS       UP   17:25   BCRVM6     UP
DCSTSO    UP   17:25   BCRVMPC1   UP
DCSNTV    UP
BCRVMIAC  UP
HP06AUTO  UP
BCRVMHP5  UP

Enter APPLNAME
or APPLNAME/in
```

Select an item that represents the maximum number of rows and columns available on your display.

26 rows x 80 columns
30 rows x 80 columns
34 rows x 80 columns
40 rows x 80 columns
48 rows x 80 columns
60 rows x 80 columns

Figure 61. Example of Largest 3270 Emulator Font

14.3 3270 Printer Emulation Text Mode Support

OS/2 Extended Edition Version 1.3 has been provided with a new default print mode used by the 3270 printer emulation.

With OS/2 Extended Edition Version 1.2 the default print mode was graphic.

3270 Printer Device Drivers now support, by default, text mode for:

- Host directed print
- Screen print.

14.4 Batch Configuration Utility (BCU)

The Batch Configuration Utility is a productivity aid to assist the administrator in managing a large number of users and configurations.

The Administrators have to create an ASCII input file in which they specify the name of each new Communications Manager configuration file they want to create and the parameter values that are specific to each Communications Manager configuration fi.'e.

The Batch Configuration Utility uses the information in the ASCII input file to create the new configuration files.

OS/2 Extended Edition Version 1.3 brings some enhancements to the Batch Configuration Utility:

- It is now possible to change a configuration file. In order to use the Batch Configuration Utility to change an existing Communications Manager configuration file, users must code both fields of the file name record, as well as the profile records they want to change, in the ASCII input file.
- A significant extension of the Communications Manager profiles the user can create or change has been provided, as described in the following tables.
- It is also now possible, with OS/2 Extended Edition Version 1.3, to delete profiles.

14.4.1 Profiles that Can Be Created

Here is the list of the Communications Manager profiles than can be created using the Batch Configuration Utility :

Table 7 (Page 1 of 2). BCU Create. Communications Manager Profiles that Can Be Created

Profile	Parameter	New in 1.3
APPC logical unit (LU) profile	• Model profile name • Profile name (LU alias) • Comment	 x
APPC partner LU profile	• Model profile name • Profile name • Comment	 x
APPC transmission service mode profile	• Model profile name • Profile name (mode name) • Comment	x x x
APPC initial sessions limits profile	• Model profile name • Profile name • Comment	x x x
APPC remotely attachable transaction program (TP) profile	• Model profile name • Profile name • Comment	x x x
APPC conversation security profile	• User ID • Password • Comment	x x x
SDLC connection DLC adapter 1 profile	• Uses DLC adapter 0 profile as a model to create DLC adapter 1 profile	x
LAN DLC adapter 1 profile	• Uses DLC adapter 0 profile as a model to create DLC adapter 1 profile	x
IEEE 802.2 (LAN) adapter 1 profile	• Uses adapter 0 profile as a model to create adapter 1 profile	x
3270 logical terminal session profile	• Model session number • Session number • Session ID/LU name • Short session ID • Comment • LU local address (non-DFT)	x x x x x x

Table 7 (Page 2 of 2). BCU Create. Communications Manager Profiles that Can Be Created

Profile	Parameter	New in 1.3
3270 logical printer session profile	• Model session number • Session number • Session ID/LU name • Short session ID • Comment • LU local address (non-DFT)	x x x x x x
SNA gateway workstation LU profile	• Model profile name • Profile name • Pool class • LU local address at host • PU name at workstation • LU local address at workstation • Comment	x x x x x x x

14.4.2 Profiles that Can Be Changed

Here is the list of the Communications Manager profiles that can be changed using the Batch Configuration Utility :

Table 8 (Page 1 of 5). BCU Change. Communications Manager Profiles that Can Be Changed

Profile	Parameter	New in 1.3
Workstation profile	• Comment • Machine type / Model number • IBM plant of manufacture • Machine sequence number • Enable auto-start options • Emulators to auto-start • Display this screen first	 x x x
SNA base profile	• PU name • Network name • Node ID • Auto-activate APPC attach manager	 x
APPC local LU profile	• Profile name (LU alias) • LU name • LU local address (NAU address)	x

Table 8 (Page 2 of 5). BCU Change. Communications Manager Profiles that Can Be Changed

Profile	Parameter	New in 1.3
APPC partner LU profile	• Profile name (partner LU alias)	
	• Fully qualified partner LU name	
	• Partner LU uninterpreted name	
	• LU-LU session security password	
	• DLC type	
	• Adapter number	
	• LAN destination address	
	• X.25 directory entry name	
	• Profile name (local LU alias)	
	• Partner LU session limit	x
	• LU-LU session security	x
	• Conversation security	x
	• Conversation security already verified	x
	• Permanent connection	x
	• Solicit SSCP session	x
	• Add or Remove	x
	– Transmission service mode profile names	x
	– Initial session limit profile names	x
APPC transmission service mode profile	• Profile name (mode name)	x
	• Minimum RU size	x
	• Maximum RU size	x
	• Receive pacing limit	x
	• Session limit	x
APPC initial session limits profile	• Profile name	x
	• Min. number of contention winners	x
	source	x
	• Min. number of contention winners target	x
	• Number of automatically activated sessions	
APPC remotely attachable TP profile	• TP profile name	
	• Sync level	x
	• Conversation type	x
	• Conversation security	x
	• Service TP	x
	• First character of service TP	x
	• TP name	x
	• TP filespec	
	• TP operation	x
	• Queued allocates timeout	x
	• TP receive timeout	x
	• Maximum queue depth	x
	• Program type	x
	• TP start-up parameters	x
APPC conversation security profile	• User ID	x
	• Password	x

Table 8 (Page 3 of 5). BCU Change. Communications Manager Profiles that Can Be Changed

Profile	Parameter	New in 1.3
SDLC connection DLC profile	• Adapter number	
	• Local station address	
	• Free unused link	
	• Line mode	
	• NRZI	
	• Maximum RU size	x
	• Send window count	x
	• Receive window count	x
	• Line type	x
	• Link station role	x
	• Modem rate	x
	• Data set ready timeout	x
	• XID repoll count	x
	• Non-XID repoll count	x
Twinaxial connection DLC profile	• Controller address	
LAN DLC Profile	• Adapter number	x
	• DLC type	x
	• C&SM LAN ID	x
	• Maximum RU size	x
	• Send window count	x
	• Receive window count	x
	• Maximum number of link stations	x
	• Percent of incoming calls	x
	• Free unused link	x
	• Congestion tolerance	x
IEEE 802.2 (LAN) adapter profile	• Adapter number	
	• Adapter type	
	• Shared RAM location	x
	• ETHERAND network protocol	x
	• Adapter address	
	• Maximum number of link stations	x
	• Transmit buffer size	x
	• Receive Buffer size	x
	• Minimum réceive buffers	x
	• Maximum number of SAPs	x
	• Maximum number of users	x
	• Number of queue elements	x
	• Number of global descriptor table selectors	x x
	• Number of transmit buffers	x
	• Group 1 response timer (T1)	x
	• Group 1 acknowledgment timer (T2)	x
	• Group 1 inactivity timer (T1)	x
	• Group 2 response timer (T1)	x
	• Group 2 acknowledgment timer (T2)	x
	• Group 2 inactivity timer (T1)	
NETBIOS profile	• Adapter number	x
	• Maximum number of link stations	x
	• Universally administrated address reversed	x x
	• Maximum sessions	x
	• Maximum commands	x
	• Maximum names	x
	• Query timeout value	

Table 8 (Page 4 of 5). BCU Change. Communications Manager Profiles that Can Be Changed

Profile	Parameter	New in 1.3
3270 DFT connection profile	• Adapter type	x
	• Adapter address	x
3270 DFT terminal session profile	• Session number	x
	• Session ID	x
	• Short session ID	x
	• Data transfer buffer size override	x
3270 DFT logical printer session profile	• Session number	x
	• Session ID	x
	• Short session ID	x
3270 non-DFT connection profile	• Adapter type	
	• Adapter address	x
	• Destination address	
	• X.25 directory entry name	
3270 non-DFT terminal session profile	• Session number	x
	• Session ID	
	• Short session ID	x
	• Data transfer buffer size override	x
	• Lu local address	
3270 non-DFT logical printer session profile	• Session number	x
	• Session ID	
	• Short session ID	x
	• Lu local address	
SNA gateway host connection profile	• Permanent connection	x
	• Auto-logoff timeout	x
	• DLC type	x
	• Adapter number	x
	• LAN destination address	x
	• X.25 directory entry name	x
	• Pools (add or remove)	x
SNA gateway workstation LU profile	• LU name at workstation	x
	• PU name at workstation	x
	• LU local address at workstation	x
	• DLC type	x
	• Adapter number	x
	• LAN destination address	x
	• X.25 directory entry name	x
	• LU type (dedicated or pooled)	x
	• LU pool class	x
	• LU local address at host	x
	• Auto-logoff	x
SNA LUA profile	• Profile name	
	• Local (NAU) address	
	• DLC type	
	• Adapter number	
	• LAN destination address	
	• X.25 directory entry name	
X.25 link profile	• Link profile name	
	• Local DTE address	
X.25 routing table entry	• Routing table entry name	
	• Called address	
	• Calling address	

Table 8 (Page 5 of 5). BCU Change. Communications Manager Profiles that Can Be Changed

Profile	Parameter	New in 1.3
X.25 directory entry	• Directory entry name • Link profile name • Remote DTE address	

14.4.3 Profiles that Can Be Deleted

Here is the list of the Communications Manager profiles that can be deleted using the Batch Configuration Utility :

Table 9. BCU Delete. Communications Manager Profiles that Can Be Deleted

Profile	Parameter	New in 1.3
APPC local LU profile	• Profile name (LU alias)	x
APPC partner LU profile	• Profile name (partner LU alias)	x
APPC transmission service mode profile	• Profile name (mode alias)	x
APPC initial session limits profile	• Profile name	x
APPC remotely attachable TP profile	• Profile name	x
APPC conversation security profile	• User ID	x
SDLC connection DLC profile	• Adapter number	x
LAN DLC adapter profile	• Adapter number	x
IEEE 802.2 (LAN) adapter profile	• Adapter number	x
3270 DFT connection profile and associated 3270 DFT terminal and printer sessions profiles		x
3270 non-DFT connection profile and associated 3270 non-DFT terminal and printer sessions profiles		x
3270 logical terminal session profile	• Session number	x
3270 logical printer session profile	• Session number	x
SNA gateway workstation LU profile	• LU name at workstation	x
SNA gateway host profile and all workstation LU profiles for the SNA gateway connection		x

14.4.4 BCU Example

The Batch Configuration Utility executable file CMCONFIG, shipped with IBM Operating System/2 Extended Edition V1.3, can be used only with Version 1.3 or Version 1.2 configuration files.

The purpose of our test was to create a complete OS/2 Gateway configuration for 112 3270 terminal sessions on 28 workstations (four sessions per workstation). All the sessions are pooled in the same pool class. Gateway connection to the host is via Token Ring. The gateway also has two 3270 terminal sessions.

Here is an overview of the procedure followed for this example. More detailed step-by-step information is provided in Sections 14.4.4.1, "Creation of the PROTO.CFG File" on page 141 and Appendix G, "GATEWAY.DAT File" on page 187.

1. Create a prototype configuration file, using the Basic Configuration Services (BCS). The prototype file need only contain default data to have two 3270 sessions on the workstation using a LAN connection via Token Ring. This configuration file created by BCS is stored in the OS2\INSTALL directory.
2. Copy the .CFG file created by BCS from the OS2\INSTALL directory to the CMLIB directory, so it can be used by Communications Manager.
3. Because BCS is unable to create the SNA gateway host connection, the newly created configuration file must be modified using the **Advanced Configuration** service of Communications Manager. The default values should be used whenever possible.
4. Prepare an ASCII file to update the prototype configuration file. Refer to the *IBM OS/2 Extended Edition Version 1.3 System Administrator's Guide for Communications (SO1F-0302)* for detailed information on each statement.
5. The Batch Configuration Utility was run using the following steps:
   a. Start Communications Manager with any valid configuration file. It is not mandatory to use the file just created.
   b. From the Group-Main window, select **OS/2 Full Screen**
   c. At the OS/2 prompt, enter the following command :

      ```
      CMCONFIG configfile inputfile
      ```

      where :

      configfile is the name of the prototype file that was created. It must be a Version 1.2 or Version 1.3 configuration file. The .CFG file name extension is taken by default. The file may have been found inconsistent by the verification procedure of OS/2 Communications Manager. This is not a problem, as the errors will be corrected by using the Batch Configuration Utility.

      inputfile is the name of the ASCII input file created at step 4.

      The CMCONFIG program assumes CMLIB as the current subdirectory, so the full file specification must be provided for any file not in the CMLIB subdirectory.

      For example, to use the file PROTO.CFG in the CMLIB subdirectory and the GATEWAY.DAT ASCII file in the root directory, the following must be entered:

      ```
      CMCONFIG PROTO C:\GATEWAY.DAT
      ```

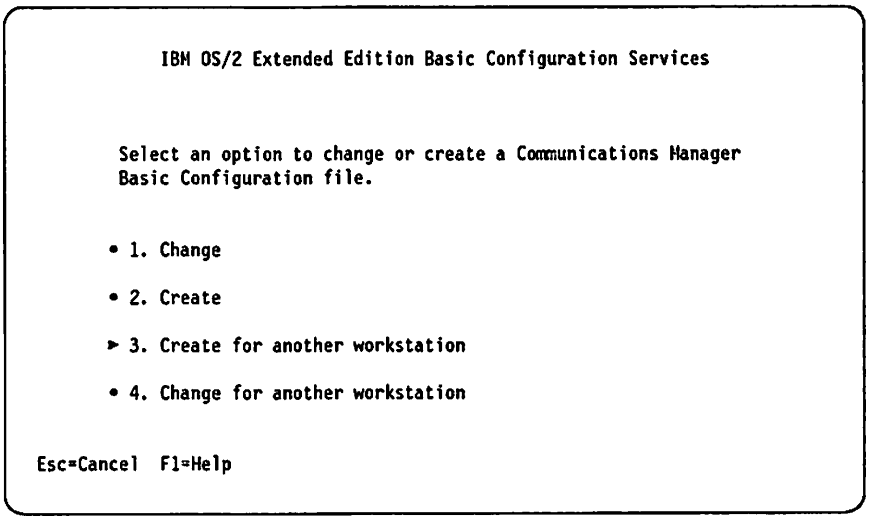
The utility program creates and verifies the new (or updated) configuration file and puts it in the CMLIB directory.

The utility creates a LOG file of the operations (CMCONFIG.LOG) in the current directory.

14.4.4.1 Creation of the PROTO.CFG File

In order to create an initial configuration file, the Basic Configuration Services (BCS) program (EECFG) must be used.

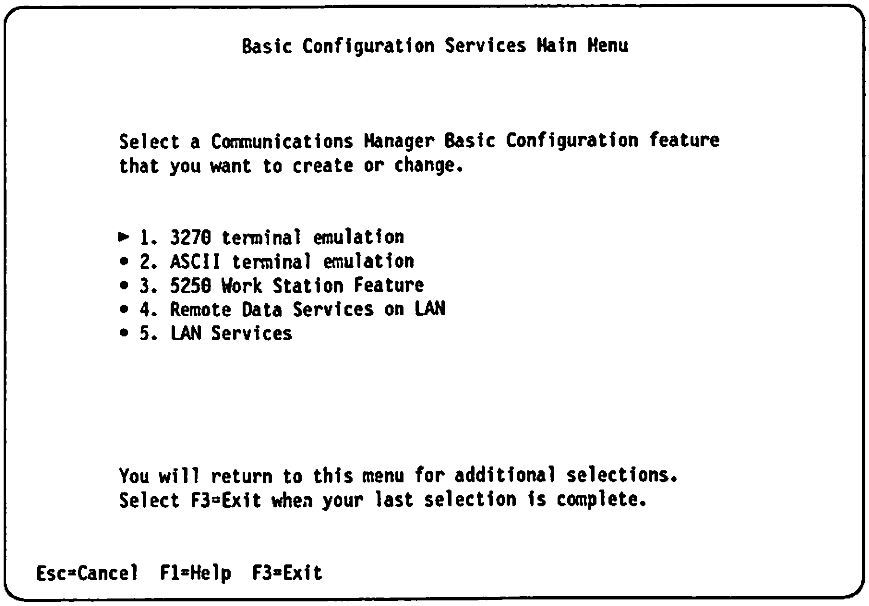
The configuration file created is for another workstation, as indicated in Figure 62 below.



```
            IBM OS/2 Extended Edition Basic Configuration Services


       Select an option to change or create a Communications Manager
       Basic Configuration file.

      • 1. Change

      • 2. Create

      ► 3. Create for another workstation

      • 4. Change for another workstation

 Esc=Cancel  F1=Help
```

Figure 62. Use of BCS to Create a Configuration File

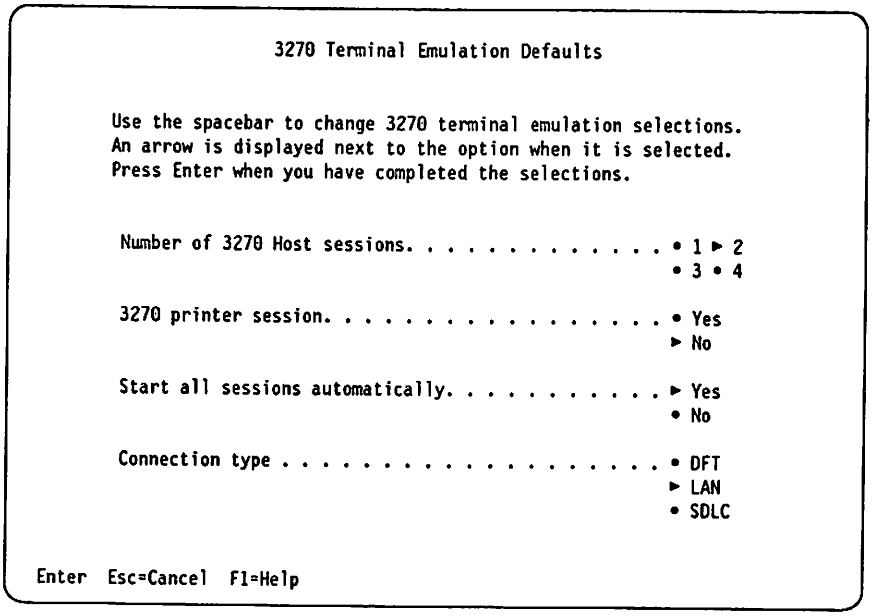
The 3270 terminal emulation feature is needed:



```
                 Basic Configuration Services Main Menu


       Select a Communications Manager Basic Configuration feature
       that you want to create or change.


       ► 1. 3270 terminal emulation
       • 2. ASCII terminal emulation
       • 3. 5250 Work Station Feature
       • 4. Remote Data Services on LAN
       • 5. LAN Services



       You will return to this menu for additional selections.
       Select F3=Exit when your last selection is complete.

 Esc=Cancel  F1=Help  F3=Exit
```

Figure 63. 3270 Terminal Emulation Selection

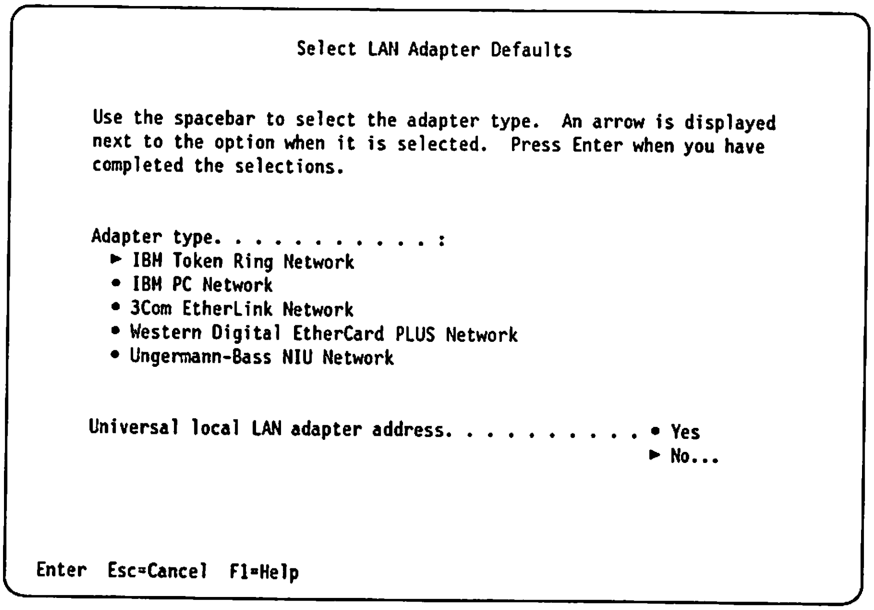
Two LAN based 3270 terminal sessions are configured with no printer session:



```
                       3270 Terminal Emulation Defaults


        Use the spacebar to change 3270 terminal emulation selections.
        An arrow is displayed next to the option when it is selected.
        Press Enter when you have completed the selections.


         Number of 3270 Host sessions. . . . . . . . . . . . . • 1 ▶ 2
                                                               • 3 • 4

         3270 printer session. . . . . . . . . . . . . . . . . • Yes
                                                               ▶ No

         Start all sessions automatically. . . . . . . . . . . ▶ Yes
                                                               • No

         Connection type . . . . . . . . . . . . . . . . . . . • DFT
                                                               ▶ LAN
                                                               • SDLC


   Enter  Esc=Cancel  F1=Help
```

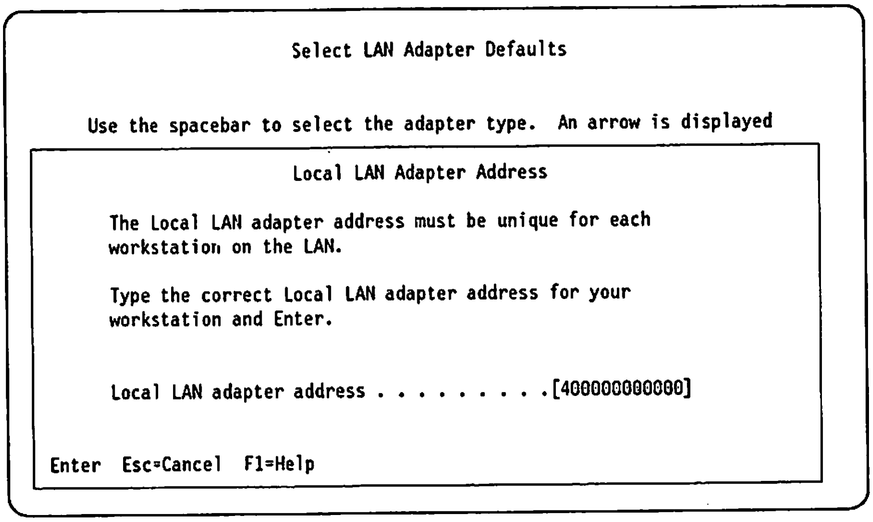
Figure 64. Terminal Emulation Defaults Selection

The workstation will use a Token Ring adapter (address = 400000000000), as indicated in the next two figures.



```
                         Select LAN Adapter Defaults


      Use the spacebar to select the adapter type.  An arrow is displayed
      next to the option when it is selected.  Press Enter when you have
      completed the selections.


      Adapter type. . . . . . . . . . . :
        ▶ IBM Token Ring Network
        • IBM PC Network
        • 3Com EtherLink Network
        • Western Digital EtherCard PLUS Network
        • Ungermann-Bass NIU Network


      Universal local LAN adapter address. . . . . . . . . . • Yes
                                                             ▶ No...



   Enter  Esc=Cancel  F1=Help
```

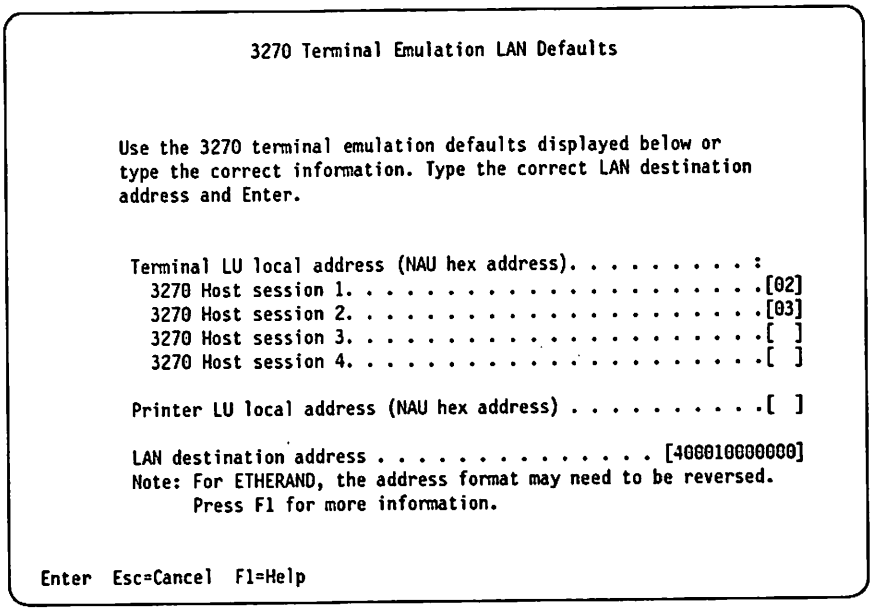
Figure 65. LAN Adapter Defaults Selection



```
                     Select LAN Adapter Defaults

    Use the spacebar to select the adapter type.  An arrow is displayed
                       Local LAN Adapter Address

       The Local LAN adapter address must be unique for each
       workstation on the LAN.

       Type the correct Local LAN adapter address for your
       workstation and Enter.


       Local LAN adapter address . . . . . . . . .[400000000000]


  Enter  Esc=Cancel  F1=Help
```

Figure 66. LAN Adapter Address Information

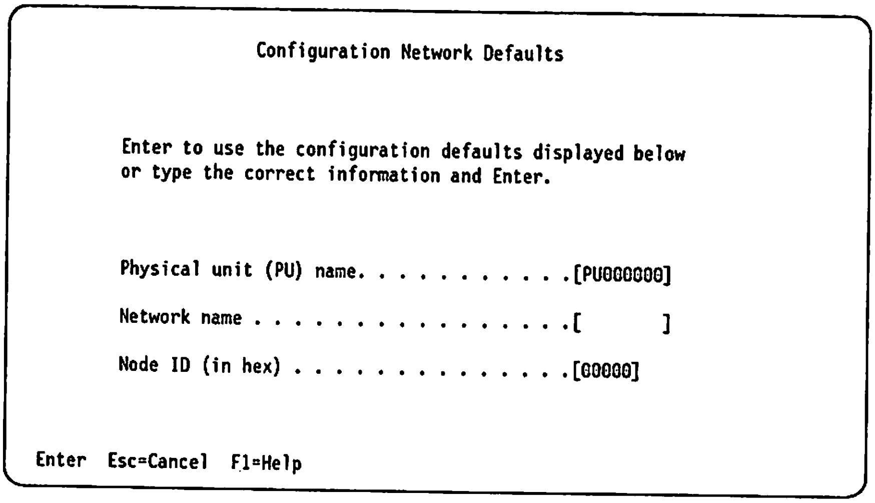
The default LU local addresses are kept, and the LAN address of the host is specified:



```
                   3270 Terminal Emulation LAN Defaults


        Use the 3270 terminal emulation defaults displayed below or
        type the correct information. Type the correct LAN destination
        address and Enter.


         Terminal LU local address (NAU hex address). . . . . . . . . :
           3270 Host session 1. . . . . . . . . . . . . . . . . . . . .[02]
           3270 Host session 2. . . . . . . . . . . . . . . . . . . . .[03]
           3270 Host session 3. . . . . . . . . . . . . . . . . . . . .[  ]
           3270 Host session 4. . . . . . . . . . . . . . . . . . . . .[  ]

         Printer LU local address (NAU hex address) . . . . . . . . . .[  ]

         LAN destination address . . . . . . . . . . . . . . [400010000000]
         Note: For ETHERAND, the address format may need to be reversed.
               Press F1 for more information.


  Enter  Esc=Cancel  F1=Help
```

Figure 67. Terminal Emulation Defaults Selection

The default values for the PU name, Network name and Node ID are kept:



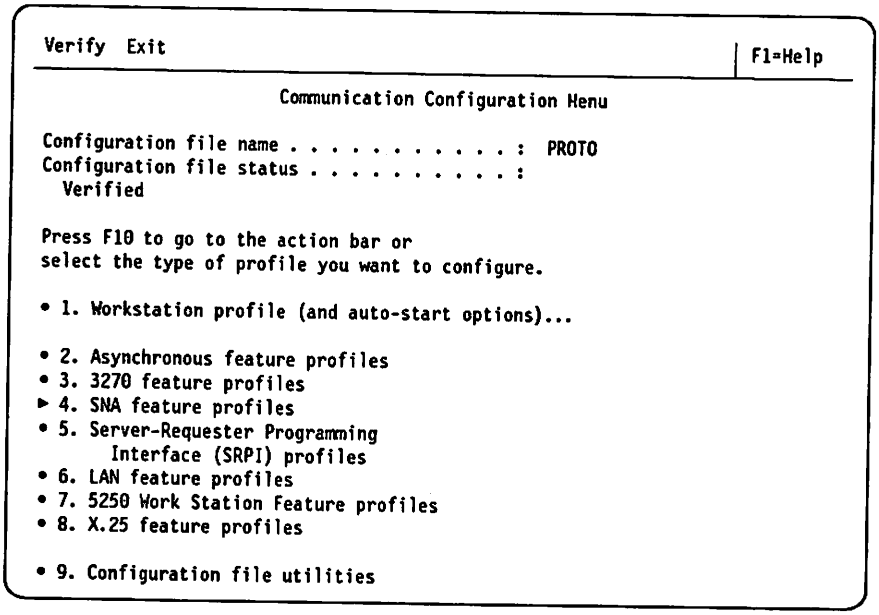
```
                    Configuration Network Defaults


        Enter to use the configuration defaults displayed below
        or type the correct information and Enter.


        Physical unit (PU) name. . . . . . . . . . .[PU000000]

        Network name . . . . . . . . . . . . . . . .[        ]

        Node ID (in hex) . . . . . . . . . . . . . .[00000]


 Enter  Esc=Cancel  F1=Help
```

Figure 68. Configuration Network Defaults Selection

After completing these steps, the new configuration file must be verified. If no errors occur, BCS stores the file created in the OS2\INSTALL directory. However, to be started, a configuration file must be located in the CMLIB directory. Therefore, the PROTO.CFG file must be copied from the OS2\INSTALL directory to the CMLIB directory.

Communications Manager is then started and the **Advanced configuration** function is used to modify the PROTO.CFG created so that it contains a SNA gateway host connection profile.

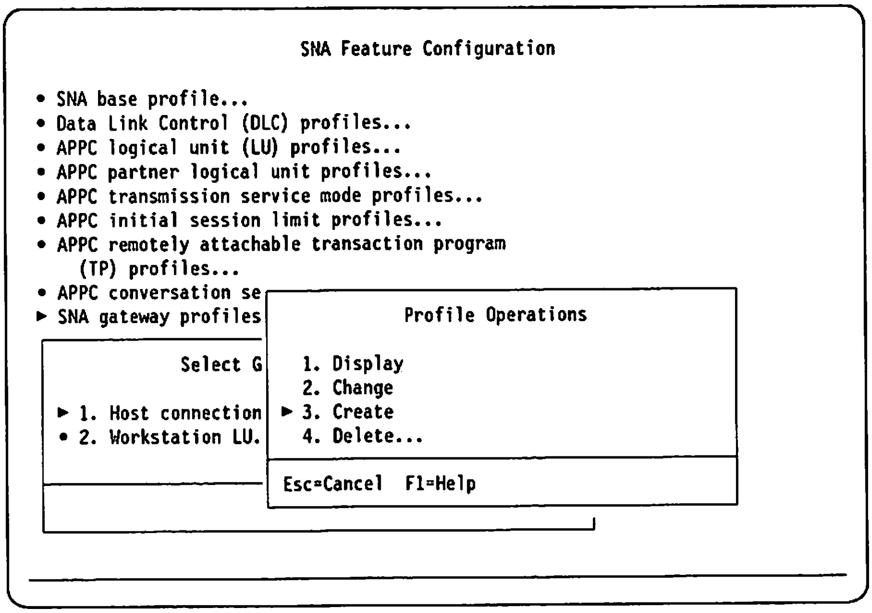
SNA feature profiles is first selected from the **Communication Configuration Menu**:



```
 Verify  Exit                                             | F1=Help
 ------------------------------------------------------------------
                   Communication Configuration Menu

 Configuration file name . . . . . . . . . . . :  PROTO
 Configuration file status . . . . . . . . . . :
   Verified

 Press F10 to go to the action bar or
 select the type of profile you want to configure.

 • 1. Workstation profile (and auto-start options)...

 • 2. Asynchronous feature profiles
 • 3. 3270 feature profiles
 ► 4. SNA feature profiles
 • 5. Server-Requester Programming
        Interface (SRPI) profiles
 • 6. LAN feature profiles
 • 7. 5250 Work Station Feature profiles
 • 8. X.25 feature profiles

 • 9. Configuration file utilities
```

Figure 69. SNA Feature Profiles Selection

SNA gateway profile is then selected to create the **Host connection.**

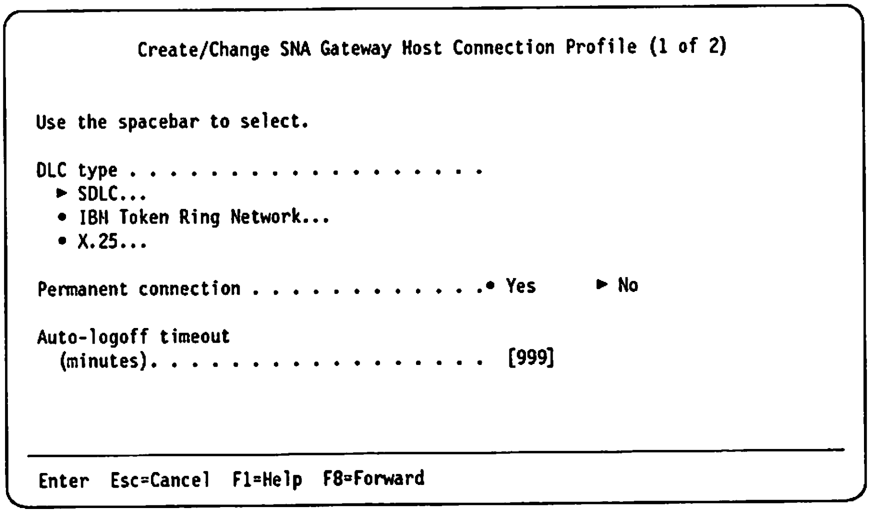


```
                     SNA Feature Configuration

• SNA base profile...
• Data Link Control (DLC) profiles...
• APPC logical unit (LU) profiles...
• APPC partner logical unit profiles...
• APPC transmission service mode profiles...
• APPC initial session limit profiles...
• APPC remotely attachable transaction program
    (TP) profiles...
• APPC conversation se
► SNA gateway profiles      Profile Operations

        Select G     1. Display
                     2. Change
  ► 1. Host connection ► 3. Create
  • 2. Workstation LU.   4. Delete...

                     Esc=Cancel  F1=Help
```

Figure 70. SNA Gateway Host Connection Profile Selection

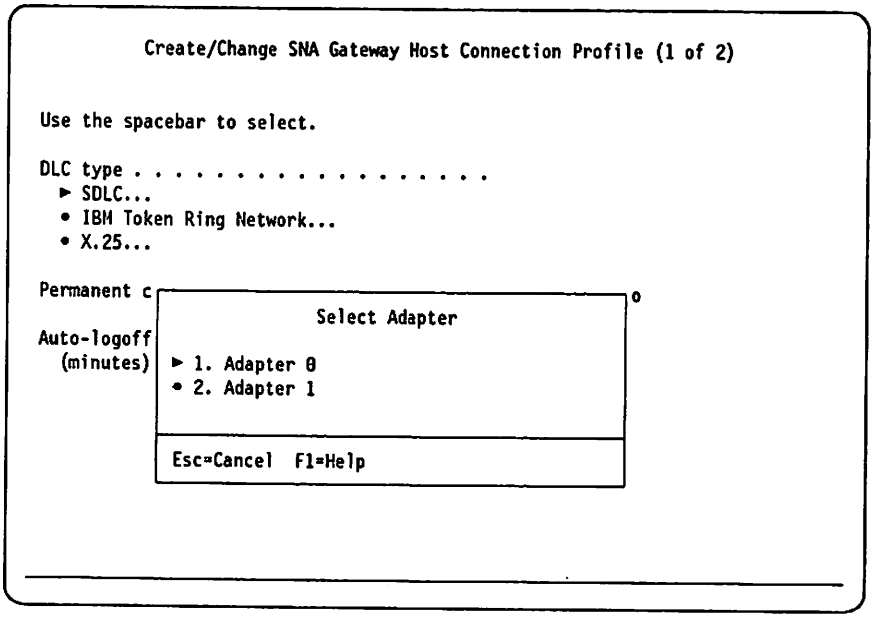
Because this configuration is going to be modified using the Batch Configuration Utility, the selections made are not important. For example, in this example an **SDLC type** adapter is chosen even though there isn't one in the machine. The important thing is that the profile is created.

Therefore, it is sufficient to select the default values given, as illustrated in the following two figures:



```
       Create/Change SNA Gateway Host Connection Profile (1 of 2)


Use the spacebar to select.

DLC type . . . . . . . . . . . . . . . . . . .
  ► SDLC...
  • IBM Token Ring Network...
  • X.25...

Permanent connection . . . . . . . . . . . .• Yes     ► No

Auto-logoff timeout
  (minutes). . . . . . . . . . . . . . . . .  [999]


Enter  Esc=Cancel  F1=Help  F8=Forward
```

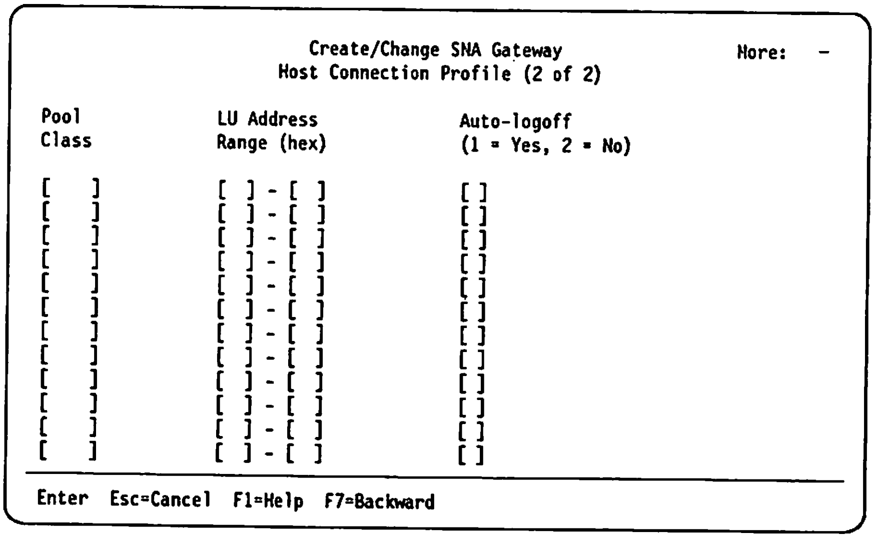
Figure 71. SNA Gateway Host Connection Profile Creation



```
Create/Change SNA Gateway Host Connection Profile (1 of 2)

Use the spacebar to select.

DLC type . . . . . . . . . . . . . . . . . .
  ► SDLC...
  • IBM Token Ring Network...
  • X.25...

Permanent c                                             o
                         Select Adapter
Auto-logoff
  (minutes)  ► 1. Adapter 0
             • 2. Adapter 1

             Esc=Cancel  F1=Help
```

Figure 72. SNA Gateway Host Connection Profile Creation (Part 2)

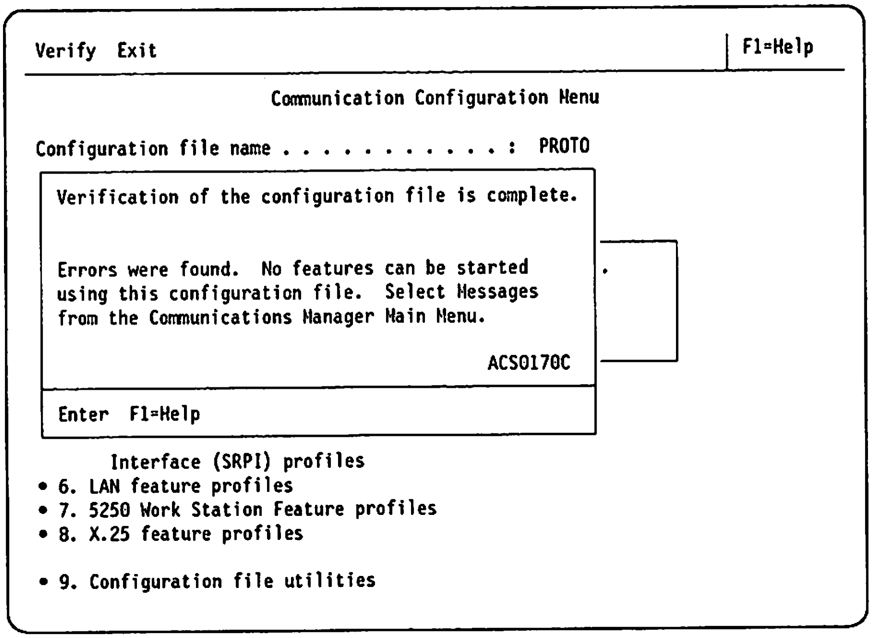
As shown in the following figure, no information is entered regarding the pool classes. This information will be added by the BCU.



```
                      Create/Change SNA Gateway              More:  -
                    Host Connection Profile (2 of 2)

Pool             LU Address          Auto-logoff
Class            Range (hex)         (1 = Yes, 2 = No)

[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]
[    ]           [ ] - [ ]           [ ]

Enter  Esc=Cancel  F1=Help  F7=Backward
```

Figure 73. Pool Class Creation

The verification of the file (below) shows that errors are found and that the configuration file cannot be started. This is not a problem, as the errors will be corrected by using the Batch Configuration Utility.



```
Verify  Exit                                                 | F1=Help

                      Communication Configuration Menu

Configuration file name . . . . . . . . . . . :  PROTO

 Verification of the configuration file is complete.

 Errors were found.  No features can be started          .
 using this configuration file.  Select Messages
 from the Communications Manager Main Menu.

                                            ACS0170C

 Enter  F1=Help

        Interface (SRPI) profiles
• 6. LAN feature profiles
• 7. 5250 Work Station Feature profiles
• 8. X.25 feature profiles

• 9. Configuration file utilities
```

Figure 74. Configuration File Verification

14.4.4.2 GATEWAY.DAT File

For a listing of the ASCII file used by the Batch Configuration Utility to create the GATEWAY.CFG file, please refer to Appendix G, "GATEWAY.DAT File" on page 187.

14.5 ACDI Redirection Facility

The Asynchronous Communications Device Interface (ACDI) Redirection Facility is a new function provided by the Communications Manager in OS/2 Extended Edition Version 1.3.

14.5.1 Overview

The redirection facility works in conjunction with both ACDI and the Local Area Network Asynchronous Connection Server (LANACS).

The ACDI interface is an OS/2 Communications Manager service which provides asynchronous communication capability. A LAN workstation on which the ACDI service has been installed can be configured so that ACDI ports are redirected to a LAN Asynchronous Connection Server.

The Local Area Network Asynchronous Connection Server (LANACS) is a PC DOS product that provides asynchronous communication server capability to a Local Area Network.

Serial ports, asynchronous communications equipment and phone lines located at an asynchronous connection server can be shared by workstations on the LAN.

ACDI Redirection allows the redirection of serial ports COM1, COM2 and COM3 on a LAN OS/2 workstation to the asynchronous resources provided at the server. The server and the OS/2 workstation use the IBM NETBIOS protocol to exchange information.

14.5.2 Installing and Configuring ACDI Redirection

This section describes the activities for planning, installing and configuring the ACDI Redirection facility on an OS/2 workstation.

14.5.2.1 Hardware and Software Requirements

- Software
  - Communications Manager component of OS/2 Extended Edition Version 1.3.
  - The IBM Local Area Network Asynchronous Connection Server (LANACS) program must be installed on a DOS workstation on the LAN.

    PC LAN Program or DOS LAN Requester are not necessary.
- Hardware
  - The OS/2 workstation does not require an asynchronous adapter.
  - The OS/2 workstation must have a LAN adapter.
  - ACDI Redirection requires 36KB of fixed-disk space and 20KB of RAM.

14.5.2.2 Process for Planning, Installing, and Configuring

Step 1. Server installation and configuration.

You must first ensure that the IBM Local Area Network Asynchronous Connection server program has been installed and configured on the asynchronous server machine.

Step 2. Planning your NETBIOS names.

The administrator who configures the asynchronous server assigns a server NETBIOS name to each asynchronous resource that the server provides. If resources are pooled, then the server NETBIOS name is assigned to the pool.

Plan a local NETBIOS name for each OS/2 workstation that will use ACDI Redirection. Each local NETBIOS name must be unique within the Local Area Network.

If an OS/2 workstation has both the LAN requester and ACDI Redirection installed, then the local NETBIOS name, used by ACDI Redirection, must be different from the "workstation" NETBIOS name used by the LAN requester.

Step 3. Planning the Communications Manager configuration file.

Here are the requirements for ACDI Redirection configuration :

a. The asynchronous feature must be configured to match the asynchronous hardware that is attached to the serial port on the asynchronous server machine.

   For each remote serial port (or pool), a separate asynchronous port profile must be created in the OS/2 Communications Manager configuration file.

b. When calculating resources used by all the NETBIOS applications on the workstation (in the Communications Manager NETBIOS profile), consider the following requirements for ACDI Redirection :

```
ACDI Redirection uses : 3 NETBIOS sessions
                        9 NETBIOS commands
                        1 NETBIOS name
```

Step 4. Install Communications Manager on the OS/2 workstation.

The files required to run ACDI Redirection are installed together with ACDI.

There are several ways to install the workstation :

- If you have a configuration file that fulfills the requirements in step 3, then you can install using this configuration file.
- If you have not created a configuration file, you may want to use the Basic Configuration Services. You have to configure for both the LAN (with NETBIOS), and the asynchronous feature.

  You may need to modify this configuration file for the NETBIOS parameters discussed in step 3, and for any asynchronous feature you need to modify to match the asynchronous server.
- If you have not created a configuration file and you do not want to use the Basic Configuration Services, then you must :

  a. Install with one of the default configuration files (ACSCFG.CFG or ACSCFGUS.CFG) and select ACDI from the additional features panel.
  b. Create a configuration file that fulfills the requirements in step 3.
  c. Reinstall Communications Manager with the new configuration file, using the REINST command.

Step 5. Edit CONFIG.SYS.

Use any ASCII editor to add the following statement at the end of the CONFIG.SYS file :

```
DEVICE=d:\CMLIB\ASYNCDDE.SYS local_NETBIOS_name
```

where

- **d** is the target drive where Communications Manager is installed on this workstation.

- **local_NETBIOS_name** is the unique NETBIOS name, planned in step 2, that you assign to this workstation.

14.5.3 Using ACDI Redirection

Before an ACDI application is started, the REDIRECT command has to be used to redirect the local serial port that the application expects to use.

14.5.3.1 The Two REDIRECT Commands

There are two different REDIRECT commands:

- The ACDI Redirection REDIRECT command, provided with Communications Manager, for use on OS/2 workstations.
- The REDIRECT command provided with the LAN Asynchronous Connection Server program for use on DOS workstations.

You can configure redirection to be temporary or permanent.

ACDI is initialized to the permanent redirection configuration when Communications Manager is started. While ACDI is running, the redirection for a port can be changed temporarily. However, a serial port cannot be redirected while it is in use.

14.5.3.2 Virtual Serial Ports and Physical Serial Ports

An OS/2 workstation need not have a physical serial port (an asynchronous adapter) in order to use ACDI redirection. ACDI creates a virtual serial port.

For example, if a user redirects COM1 to the NETBIOS name *REMOTECOM*, ACDI creates a virtual serial port named COM1 and associates this serial port with the server NETBIOS name *REMOTECOM.*

An application written to ACDI can use the virtual serial port COM1 at the same time that another application (or device, such a mouse) is using the physical serial port COM1.

ACDI can handle :

- Two virtual serial ports (COM1 and COM2) on an IBM Personal Computer AT workstation
- Three virtual serial ports (COM1, COM2 and COM3) on an IBM Personal System/2 workstation.

14.5.3.3 Using the REDIRECT Command

The *REDIRECT* command is issued at the OS/2 command prompt.

Syntax for the *REDIRECT* statement is as follows:

```
>> REDIRECT─┬──────────────────────────────────────────────┬─┤
            │                                              │
            │          ┌ COMy ─────────────────────────┐   │
            ├ COMx TO──┤                               ├───┤
            │          └ server_name ──┬───────────────┤   │
            │                          │               │   │
            │                          └ ON LANAn ─────┘   │
            │                                              │
            ├ RESTORE ─────────────────────────────────────┤
            │                                              │
            └ SAVE ────────────────────────────────────────┘
```

The following statements should be used:

- To redirect a local serial port to another serial port on the same workstation:

  ```
  REDIRECT COMx TO COMy
  ```

- To redirect a local serial port to the server:

  ```
  REDIRECT COMx TO server_name
  ```

  where

  server_name is the server NETBIOS name assigned to the asynchronous resource at the asynchronous server machine.

- To redirect a serial port to an asynchronous resource on a specific LAN, for an OS/2 workstation which has two LAN adapters:

```
REDIRECT COMx TO server_name ON LANAn
```

If no LAN adapter is specified, then LAN adapter 0 is assumed.

- To cancel redirection by redirecting the serial port to itself:

  ```
  REDIRECT COMx TO COMx
  ```

- To get the redirection permanent, as it is lost when ACDI stops:

  ```
  REDIRECT SAVE
  ```

- To reinitialize ACDI to the permanent redirection configuration:

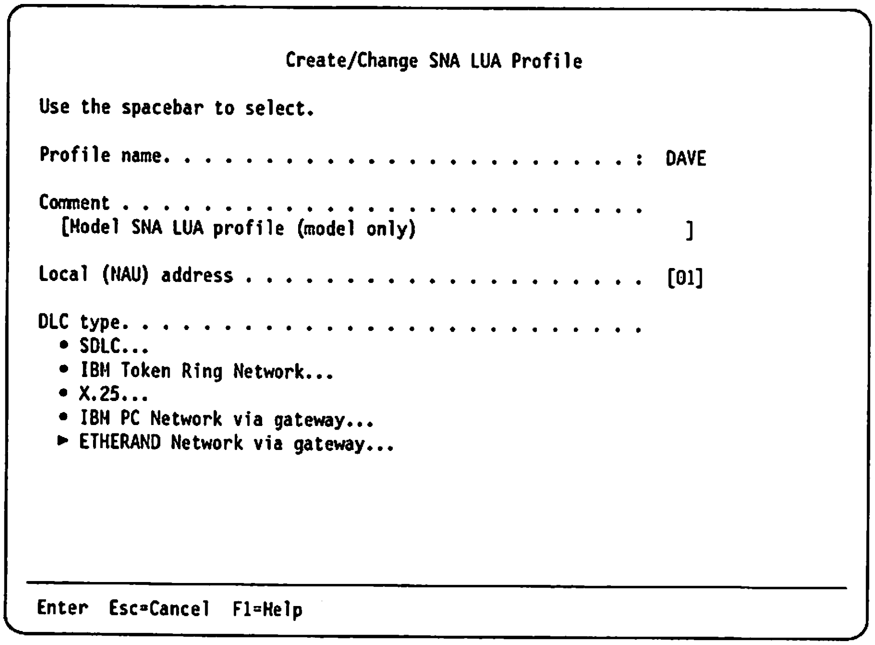
  ```
  REDIRECT RESTORE
  ```

- To display the current configuration:

  ```
  REDIRECT
  ```

14.6 ETHERAND LAN Support for LU Application (LUA) Protocols

OS/2 Extended Edition Version 1.3 adds the support for LUA (LU 0, 1, 2, and 3) protocols over an ETHERAND LAN. This support is demonstrated by the inclusion of the **ETHERAND Network via gateway...** selection on the **Create/Change SNA LUA Profile** screen in Communications Manager, as shown in Figure 75.



```
                    Create/Change SNA LUA Profile

Use the spacebar to select.

Profile name. . . . . . . . . . . . . . . . . . . . . . . : DAVE

Comment . . . . . . . . . . . . . . . . . . . . . . . . . .
  [Model SNA LUA profile (model only)                        ]

Local (NAU) address . . . . . . . . . . . . . . . . . . . . [01]

DLC type. . . . . . . . . . . . . . . . . . . . . . . . . .
  • SDLC...
  • IBM Token Ring Network...
  • X.25...
  • IBM PC Network via gateway...
  ► ETHERAND Network via gateway...




Enter  Esc=Cancel  F1=Help
```

Figure 75. Create/Change SNA LUA Profile Screen in Communications Manager

14.7 ETHERAND LAN Support for Remote Data Services (RDS) DOS Requesters

New to OS/2 Extended Edition with Version 1.3 is support for RDS DOS requesters on an ETHERAND (IEEE 802.3 or DIX 2.0 Ethernet** based) LAN.

The DOS Database Requester Workstation requires the installation of the LAN Support Program Version 1.2 and DOS 3.3, or later, to support the NETBIOS session used to communicate with the OS/2 Extended Edition Database Manager server. The LAN Support Program Version 1.2 code can be purchased separately or comes with the IBM OS/2 LAN Server Version 1.3 code.

Instructions for the implementation of this configuration can be found in Section 14.7.1.1, "LAN Support Program Configuration for an ETHERAND* DOS Requester" on page 153. In most respects, this information also applies to the use of LAN Support Program for DOS LAN Requester.

14.7.1.1 LAN Support Program Configuration for an ETHERAND* DOS Requester

As mentioned in Chapter 1, IBM has added the support for RDS DOS Requester over an ETHERAND network in OS/2 Extended Edition Version 1.3. Together with the previous section, this section is to aid in the configuration of a DOS requester/OS/2 server ETHERAND* RDS network.

The DOS Database Requester Workstation requires the installation of the LAN Support Program Version 1.2 and DOS 3.3, or later, to support the NETBIOS session used to communicate with the RDS server. The LAN Support Program can be purchased separately or comes with OS/2 LAN Server Version 1.3 code.

The different device drivers for the DOS requester diskette are as follows:

DXMA0MOD.SYS	For IEEE 802.2 support
DXME0MOD.SYS	For ETHERAND support
DXMT0MOD.SYS	For NETBIOS support
MACWD.DOS	For a Western Digital EtherCard PLUS** Network Adapter
ELNKII.DOS	For the 3COM EtherLink II** Network Adapter
ELNKMC.DOS	For the 3COM EtherLink/MC** Network Adapter
MACETH.DOS	For the IBM PS/2 Adapter/A for Ethernet Networks
UBNEI.DOS	For the Ungermann-Bass NIUpc or NIUps Network Adapters.

An example CONFIG.SYS file for the DOS Requester machine is as follows:

```
BREAK=ON
BUFFERS=20
FILES=100
LASTDRIVE=Z
SHELL=A:\COMMAND.COM /E:2000 /P
DEVICE=A:\ANSI.SYS
INSTALL=A:\FASTOPEN.EXE C:=(50,25)
DEVICE=\LANMAN\PROTMAN.EXE
DEVICE=\MACWD.DOS
DEVICE=\DXMA0MOD.SYS
DEVICE=\DXME0MOD.SYS ,,1
DEVICE=\DXMT0MOD.SYS
FCBS=16,8
```

Notes:

1. The ,,1 parameter for the DXME0MOD.SYS device driver indicates that the DIX 2.0 Ethernet protocol will be used. If this parameter was removed the default IEEE 802.3 protocol would be used. The protocol used must be consistent with that specified on the RDS Server. **This is very important, as the defaults in the LAN Support Program 1.2 and OS/2 Version 1.3 Extended Edition Communications Manager are DIFFERENT.**
2. Detailed information concerning the DXM drivers can be found in the DXMINFO.001 (or .DOC) file on the LAN Support Program diskette. A printed copy of this file would be useful during the installation process.
3. The DEVICE=\MACWD.DOS statement specifies the device driver for the LAN adapter card, which in this case is a Western Digital card. This driver must correspond with the installed card.

4. In the tested configuration, the default parameter values for the NETBIOS device driver, DXMT0MOD.SYS, were acceptable. If any parameters are specified after this driver, the O=N parameter must also be specified. This parameter means ***Open=No***, or do not open the network adapter at IPL time. The default value for this parameter is ***Open=Yes***. However, the O=Y parameter does not work if any other parameters are specified.

   For example, if it was required to increase the number of sessions from the default value of 16 to 22, and increase the number of link stations from its default value of 16 to 20, the device driver statement should look like this:

   ```
   DEVICE=\DXMT0MOD.SYS S=22 ST=20 O=N
   ```

The PROTOCOL.INI file is a text file which provides parameters for the PROTMAN.EXE device driver. This file must specify the proper parameters and bindings. The following is a portion of an example PROTOCOL.INI file used on the DOS Requester machine.

```
; --------------------- Protocol Manager Definition ----------------------

[PROTOCOL_MANAGER]
   DriverName = PROTMAN$

; ------------------ IBM Ethernet Protocol Definition -------------------

[LSP12]
   DriverName = DXME0$

; -------------------------- Bindings Statement --------------------------
   Bindings = WDMAC

; The modules listed below are valid choices for the "Bindings =" field.
;
;      (1)  TCMAC2  - 3Com Micro Channel Adapters
;      (2)  STCMAC2 - 3Com Micro Channel Adapters (2nd 3COM adapter in one
;                     machine)
;      (3)  TCMAC   - 3Com PC I/O Bus Adapters
;      (4)  STCMAC  - 3Com PC I/O Bus Adapters (2nd 3COM in one machine)
;      (5)  WDMAC   - Western Digital Micro Channel and PC I/O Bus Adapters
;      (6)  IBMAC   - IBM PS/2 Adapter/A for Ethernet Networks
;      (7)  IBMAC2  - IBM PS/2 Adapter/A for Ethernet Networks (2nd IBM
;                     adapter in one machine)

; ------------- Western Digital Network Adapter Definition ---------------

[WDMAC]
   DriverName = MACWD$
;
; On Micro Channel adapters, the following 3 parameters are read from the
; POS registers on the adapter.  Therefore, they are not needed and will
; be ignored.
   IRQ = 3
   RamAddress = 0xD000
   IOBase = 0x280     ; This value must match adapter jumpers
   ReceiveBuffers = 16
   ReceiveChains = 16
   MaxRequests  = 10
   MaxTransmits = 10
   ReceiveBufSize = 256
```

Notes:

1. The BINDINGS= field must correspond to the adapter used. A list of the choices is given in the PROTOCOL.INI file, as indicated.

2. The Bindings = WDMAC indicates that this adapter is ***adapter 0***. If the address for this adapter is ***adapter 1***, this module should be specified as Bindings = ,WDMAC. If there are two adapters in the same machine, they can be specified as, for example:

   ```
   Bindings = TCMAC2,STCMAC2
   ```

3. Two adapters from the same manufacturer may, in some cases, not be used in the same machine. This is currently the case for adapters manufactured by both the Western Digital Corporation and Ungermann-Bass, Inc. Refer to the documentation provided with the adapter to be used for further information.

4. For more information on the PROTOCOL.INI file, refer to the ***OS/2 Extended Edition Version 1.3 System Administrator's Guide for Communications (S01F-0302).***

The relevant portions of the CONFIG.SYS file for the OS/2 RDS Server are listed below:

```
DEVICE=C:\CMLIB\LANDD.SYS
DEVICE=C:\CMLIB\PROTMAN.OS2 /I:C:\CMLIB
DEVICE=C:\CMLIB\MACWD.OS2
DEVICE=C:\CMLIB\ETHERDD.SYS CFG=C:\CMLIB\SRVETH.CFG
RUN=C:\CMLIB\ACSEPSYS.EXE
RUN=C:\CMLIB\NETBIND.EXE
DEVICE=C:\CMLIB\NETBDD.SYS  CFG=C:\CMLIB\SRVETH.CFG
```

Note: The Communications Manager configuration file (SRVETH.CFG) was configured using Basic Configuration Services. The name of this file is, of course, up to the user.

The OS/2 PROTOCOL.INI file must also contain the appropriate statements. Following are a few lines from this file.

```
; --------------------- Protocol Manager Definition ----------------------

[PROTOCOL_MANAGER]
   DriverName = PROTMAN$

; ------------------- IBM ETHERAND Protocol Definition --------------------

[ETHERAND]
   DriverName = OS2EE$
   Bindings = WDMAC

; The modules listed below are valid choices for the Bindings field.
;
;        (1)  TCMAC2 - 3Com Micro Channel Adapters
;        (2)  TCMAC - 3Com IBM PC-AT Adapters
;        (3)  WDMAC - Western Digital Micro Channel and IBM PC-AT Adapters
;        (4)  UBMAC - Ungermann-Bass Micro Channel and IBM PC-AT Adapters
;
;             NOTE: If you choose UBMAC, please check the AdapterType
;                   definition to insure that it is correct for your
;                   adapter.
; -------------- Western Digital Network Adapter Definition ---------------

[WDMAC]
   DriverName = MACWD$
   IRQ = 3
   RamAddress = 0xC400
   IOBase = 0x280
   ReceiveBuffers = 16
   ReceiveChains = 16
   MaxRequests  = 10
   MaxTransmits = 10
   ReceiveBufSize = 256
```

Note: Please refer to the notes for the DOS PROTOCOL.INI file for more information.

Chapter 15. Support for New Hardware

The IBM PS/2 Model 90 XP 486 and Model 95 XP 486 provide new functions and features to provide maximum performance. They are advanced computing systems, ideal for use in environments that require high performance and expandability.

The IBM PS/2 Model P75 486 is a high-performance workstation with built-in display. It can be used as a quick-to-install backup unit as well as a professional workstation. Due to its portability, it will find many applications.

15.1 The 80486 Machines, PS/2 Model 90 and 95

The Model 90 and 95 utilize several technical improvements to provide you with higher resolution, faster throughput, larger capacity, and greater reliability.

When OS/2 V1.3 is installed with the XGA driver, the system will use the VGA mode or the XGA mode depending on the screen connected to the system without changing the CONFIG.SYS. VGA mode will be used when the low resolution display is connected, such as the IBM 8512 and IBM 8513. XGA mode will be used for the high resolution display, such as IBM 8514 and IBM 8515. These new models have:

- The new XGA Video Subsystem, which provides higher resolution and better graphics performance.
- The PS/2 external SCSI storage, which allows you to store up to 2.2GB of data on their hard disks.

OS/2 Version 1.3 now also supports up to 24 SCSI devices.

15.2 The Portable, PS/2 Model P75

The Model P75 486 provides full XGA graphics and can have external SCSI devices attached.

The PDP (Plasma Display Panel) display adapter card supports the following three configurations:

1. Using the PDP only - all VGA modes are supported
2. Using an external CRT only - all VGA and XGA modes are supported
3. Using the PDP and a CRT at the same time - all VGA modes are supported.

You can change the display configuration by using the reference diskette. When OS/2 V1.3 is installed with the XGA device driver, the system will use the VGA mode or the XGA mode depending on the screen connected to the system without changing the CONFIG.SYS. VGA mode will be used for low resolution if a display such as the IBM 8512 or IBM 8513 is connected. XGA mode will be used for a high resolution display such as the IBM 8514 or IBM 8515.

15.3 The Extended Graphics Adapter/A (XGA)

15.3.1 OS/2 Considerations

The PS/2 Model 90, Model 95 and Model P75 do not support VGA directly, therefore after restoring OS/2 V1.3, from a backup diskette, the CONFIG.SYS file must be changed. This modification must be done before starting your system.

On the IBM PS/2 Model 50, 55, 60, 65, 70, P70 and 80 with VGA adapter (no 8514/A adapter installed), OS/2 V1.3 creates the following statements:

```
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
SET VIDEO_DEVICES=VIO_IBMVGA
SET VIO_IBMVGA=DEVICE(BVHVGA)
```

On the IBM PS/2 Model 90 and 95 with XGA adapter, OS/2 V1.3 creates the following statements:

```
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
SET VIDEO_DEVICES=VIO_IBMXWY
SET VIO_IBMXWY=DEVICE(BVHVGA,BVHXGA)
DEVICE=C:\OS2\XGARING0.SYS
```

The XGA palette is not the same as the 8514/A palette. It is a 6x6x6 (linear) palette plus 40 greys, and is optimized for antialiased text and dithering. The XGA PM display device driver supports palette realisation.

Note

The IBM 8514/A adapter is not supported on the IBM PS/2 Model 90, 95 and P75.

Refer to Appendix C, "XGA Installation under OS/2 V1.2" on page 171 for more details on XGA installation.

15.3.1.1 Limitation

Windows V3.0 is not compatible with the XGA DOS adapter interface. If you wish to run Windows V3.0, you must not install the XGA DOS adapter interface. If you have already done so, remove the following line from your CONFIG.SYS file:

```
DEVICE=C:\XGAPCDOS\XGAAIDOS.SYS
```

and reboot your system.

Note!

On the XGA driver support diskette, you will find a special DOS device driver **XGAAIDOS.SYS**. This device driver is intended to be used under DOS only. Its purpose is to provide full XGA support to the Windows V3.0 environment. This device driver will not function if Windows is run in the DOS compatibility box of OS/2.

15.4 How to Install and Use Antialiased Fonts

15.4.1 System Fonts

The four antialiased system fonts provided are copied to your hard disk when you install the device driver. They are stored in the \OS2\DDFONTS directory on your C drive. You can select an antialiased system font after you have installed the device driver, as follows:

1. In the OS/2 Desktop Manager, select Screen Setup.
2. In the Screen Setup group, run Select System Font. This program lets you view antialiased fonts and compare them with the regular system font before deciding which to use. You can rerun the program at any time if you want to select a different font.

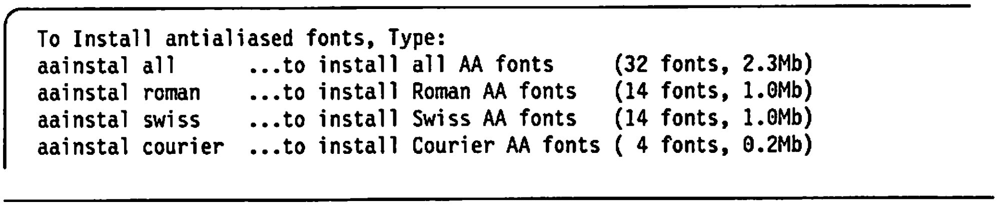
Antialiased fonts are available on systems that support 256 colors, as described in Table 6 on page 115.

15.4.2 Application Fonts

Antialiased fonts supplied on the device driver diskette are in compressed form. To unpack them and copy them to your hard disk do the following:

1. Insert the *XGA Device Driver Diskette 2* in drive A.
2. Select "OS/2 Full Screen" from OS/2 Group-Main.
3. Type **a:aainstal** and press Enter.
4. Follow the on-screen instructions to install some or all of the antialiased fonts.

You can expect to see a screen that looks like this:



```
To Install antialiased fonts, Type:
aainstal all      ...to install all AA fonts     (32 fonts, 2.3Mb)
aainstal roman    ...to install Roman AA fonts   (14 fonts, 1.0Mb)
aainstal swiss    ...to install Swiss AA fonts   (14 fonts, 1.0Mb)
aainstal courier  ...to install Courier AA fonts ( 4 fonts, 0.2Mb)
```

Figure 76. Prompt for Antialiased Font Installation, Showing Disk Requirements

15.4.2.1 Selection of Antialiased Application Fonts

Antialiased application fonts are selected by application programs. Although the fonts are device fonts and are in a different format from Presentation Manager fonts, you use them in the same way as you use Presentation Manager image fonts. The GPI calls "GPIQueryFonts" and "GPICreateLogicalFont" work with antialiased fonts in the same way as with normal Presentation Manager fonts. The OS/2 File Manager and the OS/2 System Editor both have options to select fonts. Any antialiased fonts that have been installed can be selected by the end user by using the font options provided by the OS/2 File Manager and the OS/2 System Editor.

15.5 Programming Considerations

- Antialiased font face names are different from normal font face names. Antialiased face names are:

  Swiss AA

  Swiss AA Italic

  Swiss AA Bold

  Roman AA

  Roman AA Italic

  Roman AA Bold

  Cour AA.

- The OS/2 Presentation Manager Font Editor cannot be used with antialiased fonts because the font editor assumes 1 bit per pel whereas antialiased fonts use 3 bits per pel.
- Antialiased font characters cannot be used to define an area fill pattern.
- Drawing text using an antialiased font into a (1,1) format bitmap is not supported.
- Memory and disk requirements: Antialiased fonts use more memory and disk space than normal fonts. Figure 76 on page 159 shows disk space requirements. Each antialiased character definition uses 3 bits to define a character pixel as compared with 1 bit for normal text. Therefore, antialiased fonts use up to three times more space than normal fonts in memory and on your hard disk. The size of antialiased fonts is shown in Table 10. The size depends only on the point size of the font and not on the font family or font style.

Table 10. Memory and Disk Space Requirements by Font Size

Point Size	Memory Required
8 point	38334 bytes
10 point	47874 bytes
12 point	61230 bytes
14 point	75540 bytes
18 point	118470 bytes
24 point	190020 bytes

You may, of course, delete any of the antialiased fonts from your hard disk if they are not used.

15.6 SCSI Support

OS/2 Version 1.2 already supported seven SCSI hard disks but Version 1.3 now supports up to 24 SCSI storage devices.

This is only a "logical" limitation, because we only have 26 drive letters (A-Z). Two of them are always used for disk drive A and B. Even if they are not installed, the letters A and B are reserved for them. This leaves us with the remaining 24 drive letters (C-Z). Be aware that any additional physical device such as CD-ROMs, optical disks and any logical devices such as virtual disks and LAN redirected drives will again leave you with fewer than 24 drive letters. Also, any physical hard disk can have several partitions on it, each one of those taking even more drive letters again.

However, DOS 3.3 is limited to two SCSI hard files and DOS 4.0 is limited to seven SCSI hard files, if the user has installed at least the CSD UR31300 for DOS 4.0 (US version).

Appendix A. National Language Support

OS/2 V1.3 has National Language Support for thirteen languages:

- Canadian French
- Danish
- Dutch
- Finnish
- French
- German
- Italian
- Norwegian
- Portuguese
- Spanish
- Swedish
- United Kingdom English
- United States/Universal English.

The following are the NLS files used by the Information Presentation Facility (IPF), in order to support these languages:

Table 11. IPF NLS Language Files

ID	Language	NLS File
DAN	Danish	IPFDAN.NLS
DEU	German	IPFDEU.NLS
ENG	English UK	IPFENG.NLS
ENU	English US	IPFENU.NLS
ESP	Spanish	IPFESP.NLS
FIN	Finnish	IPFFIN.NLS
FRA	French	IPFFRA.NLS
FRC	Canadian French	IPFFRC.NLS
ITA	Italian	IPFITA.NLS
NLD	Dutch	IPFNLD.NLS
NOR	Norwegian	IPFNOR.NLS
PTG	Portuguese	IPFPTG.NLS
SVE	Swedish	IPFSVE.NLS
UND	User Defined	IPFUND.NLS

These files are installed with the Toolkit and are only used during the IPF compiler runs.

The following is a list of the supported code pages:

Table 12. NLS Country Codes and Code Pages

Country	Country Code	Code Pages Supported (primary/secondary)
Arabic	785	864 , 437 , 850
Asian English	099	850 , 437
Australia	061	437 , 850
Belgium	032	437 , 850
Canadian English	001	437 , 850
Canadian French	002	863 , 850
Czechoslovakia	243	852 , 850
Czech (Slovak)	42	852 , 850
Denmark	045	865 , 850
Finland	358	437 , 850
France	033	437 , 850
Germany	049	437 , 850
Hebrew	972	862 , 437 , 850
Hungary	36	852 , 850
Iceland	354	850 , 861
Italy	039	437 , 850
Japan	081	932 , 437 , 850 , 942
Korea	082	934 , 437 , 850 , 944
Latin America	003	437 , 850 , 852
Netherlands	031	437 , 850
Norway	047	865 , 850
Poland	48	852 , 850
Portugal	351	860 , 850
People's Republic of China	086	936 , 437 , 850 , 946
Spain	034	437 , 850
Sweden	046	437 , 850
Switzerland	041	437 , 850
Taiwan	088	938 , 437 , 850 , 942
Turkey	90	857 , 850
United Kingdom	044	437 , 850
United States	001	437 , 850
Yugoslavia	38	852 , 850

Appendix B. CONFIG.SYS Configuration File

B.1 OS/2 Standard Edition V1.3 CONFIG.SYS File for a VGA System

The following is a sample of the CONFIG.SYS file from OS/2 Standard Edition V1.3 with a VGA adapter installed in the system.

There are no new statements in V1.3.

```
PROTSHELL=C:\OS2\PMSHELL.EXE;C:\OS2\OS2.INI;C:\OS2\OS2SYS.INI;C:\OS2\CMD.EXE;
IFS=C:\OS2\HPFS.IFS -C:64 /AUTOCHECK:D                  1
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=C:\DLL;C:\OS2\DLL;C:\
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET PROMPT=$i[$p]
SET HELP=C:\OS2\HELP                                     2
BUFFERS=30
DISKCACHE=64
MAXWAIT=3
MEMMAN=SWAP,MOVE,SWAPDOS                                 3
PROTECTONLY=NO
SWAPPATH=C:\OS2\SYSTEM 512
THREADS=128
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP                        4
SET VIDEO_DEVICES=VIO_IBMVGA                             4
SET VIO_IBMVGA=DEVICE(BVHVGA)                            4
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\IBMMOU02.SYS                               5
DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$                     5
DEVICE=C:\OS2\PMDD.SYS
SET KEYS=ON                                              6
SET BOOKSHELF=C:\OS2\BOOK;                               7
SHELL=C:\OS2\COMMAND.COM /P
BREAK=OFF
FCBS=16,8
RMSIZE=640
DEVICE=C:\OS2\EGA.SYS
DEVICE=C:\OS2\DOS.SYS                                    8
AUTOFAIL=NO                                              9
IOPL=YES
FILES=255
RUN=C:\OS2\CACHE.EXE /LAZY:ON                           10
```

Figure 77. CONFIG.SYS File Sample for a OS/2 V1.3 SE System with VGA

1 Statement for High Performance File System (HPFS).

2 Statement for help information (IPF).

3 Additional parameter for swapping the DOS compatibility box.

4 Statement for video device drivers (VGA).

5 Mouse device driver and configuration.

6 Statement for retrieve function.

7 Statement for online reference information (IPF)

8 Statement for the DOS Box device driver.

9 Statement for error handling.

10 Statement for HPFS cache.

B.2 OS/2 Standard Edition V1.3 CONFIG.SYS File for a XGA System

The following is a sample of the CONFIG.SYS file for a OS/2 Standard Edition V1.3 system with a XGA adapter installed.

```
PROTSHELL=C:\OS2\PMSHELL.EXE C:\OS2\OS2.INI C:\OS2\OS2SYS.INI C:\OS2\CMD.EXE
IFS=C:\OS2\HPFS.IFS -C:64 /AUTOCHECK:D
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=C:\OS2\DLL;C:\;
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET PROMPT=$i[$p]
SET HELP=C:\OS2\HELP
BUFFERS=30
DISKCACHE=64
MAXWAIT=3
MEMMAN=SWAP,MOVE,SWAPDOS
PROTECTONLY=NO
SWAPPATH=C:\OS2\SYSTEM 512
THREADS=128
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP               ] Statements for XGA
SET VIDEO_DEVICES=VIO_IBMXWY                    ] adapter on the
SET VIO_IBMXWY=DEVICE(BVHVGA,BVHXGA)            ] Model 90 and 95.
DEVICE=C:\OS2\XGARING0.SYS                      ] XGA device driver!
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\IBMMOU02.SYS
DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$
DEVICE=C:\OS2\PMDD.SYS
SET KEYS=ON
SET BOOKSHELF=C:\OS2\BOOK
SHELL=C:\OS2\COMMAND.COM /P
BREAK=OFF
FCBS=16,8
RMSIZE=640
DEVICE=C:\OS2\EGA.SYS
DEVICE=C:\OS2\DOS.SYS
DEVICE=C:\OS2\COM02.SYS
IOPL=YES
RUN=C:\OS2\CACHE.EXE /LAZY:ON
```

Figure 78. CONFIG.SYS Sample File for a OS/2 V1.3 SE SYSTEM with an XGA Adapter

Note: The new XGA display adapter is a busmaster adapter. Therefore a special device driver is required in order to allow OS/2 and its applications to communicate with this "intelligent" device on the privilege level 0.

B.3 OS/2 Standard Edition V1.2 Sample CONFIG.SYS File

The following is a sample CONFIG.SYS file for a OS/2 Standard Edition V1.2 system with a VGA adapter installed.

```
PROTSHELL=C:\OS2\PMSHELL.EXE;C:\OS2\OS2.INI;C:\OS2\OS2SYS.INI;C:\OS2\CMD.EXE;
IFS=C:\OS2\HPFS.IFS -C:64 /AUTOCHECK:D
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=C:\DLL;C:\OS2\DLL;C:\
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET PROMPT=$i[$p]
SET HELP=C:\OS2\HELP
BUFFERS=30
DISKCACHE=64
MAXWAIT=3
MEMMAN=SWAP,MOVE,SWAPDOS
PROTECTONLY=NO
SWAPPATH=C:\OS2\SYSTEM 512
THREADS=128
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
SET VIDEO_DEVICES=VIO_IBMVGA
SET VIO_IBMVGA=DEVICE(BVHVGA)
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\IBMMOU02.SYS
DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$
DEVICE=C:\OS2\PMDD.SYS
SET KEYS=ON
SET BOOKSHELF=C:\OS2\BOOK;D:\TOOLKT12\PROGREF;
SHELL=C:\OS2\COMMAND.COM /P
BREAK=OFF
FCBS=16,8
RMSIZE=640
DEVICE=C:\OS2\EGA.SYS
DEVICE=C:\OS2\DOS.SYS
RUN=C:\OS2\CACHE.EXE /LAZY:ON
SET PROGREF=PRINTRO.INF+PRCP.INF+PRGPI.INF+PRWIN.INF+PRDATA.INF
IOPL=YES
FILES=255
```

Figure 79. CONFIG.SYS Sample File for a OS/2 V1.2 SE System.

B.4 OS/2 Standard Edition V1.1 CONFIG.SYS File

The following is a sample of a CONFIG.SYS file for a OS/2 Standard Edition V1.1 system with a VGA adapter installed.

```
PROTSHELL=C:\OS2\PMSHELL.EXE;C:\OS2\OS2.INI;C:\OS2\CMD.EXE;
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=C:\DLL;C:\OS2\DLL;C:\
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET PROMPT=$i [$p]
SET HELP=C:\OS2\HELP
BUFFERS=60
DISKCACHE=64
MAXWAIT=3
MEMMAN=SWAP,MOVE
PROTECTONLY=NO
SWAPPATH=C:\OS2\SYSTEM 512
THREADS=255
TRACE=OFF
SHELL=C:\OS2\COMMAND.COM /P
BREAK=OFF
FCBS=16,8
RMSIZE=640
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
CODEPAGE=437,850
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MOUSEB05.SYS
DEVICE=C:\OS2\PMDD.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
DEVICE=C:\OS2\EGA.SYS
DEVICE=C:\OS2\COM02.SYS
```

Figure 80. CONFIG.SYS Sample File for a OS/2 V1.1 SE System

Appendix C. XGA Installation under OS/2 V1.2

With OS/2 V1.3, the XGA installation is done automatically. If users want to install the XGA drivers and fonts on a V1.2 system they have to do the installation on their own.

The following is an excerpt from the original XGA device support diskette number 2:

(C) Copyright IBM Corporation 1990

```
*****************************************************************************
* XGA OS/2 V1.2 Device Driver.                                              *
*****************************************************************************
```

IBM, Personal System/2 and PS/2 are trademarks of IBM Corporation

The OS/2 XGA display device driver on this diskette is designed to work with OS/2 Standard Edition Version 1.2 and OS/2 Extended Edition Version 1.2. This READ.ME file contains instructions on:

o How to load the device driver, and

o How to select antialiased fonts.

Installing OS/2 1.2 Device Support

If OS/2 is installed with a VGA driver, start at Step 5.

If OS/2 is installed with any other driver, or you are not sure which driver is installed, or OS/2 is not installed, start at step 1.

1. If you have an 8514/A adapter installed in your system unit, ensure that no display is connected to it.

2. Follow the installation instructions supplied with OS/2.

3. When you are asked "Do you have a Device Support diskette to install?" reply "No" and press the Enter key.

4. Follow the instructions displayed to complete the installation of OS/2 with the VGA driver.

5. When the "Group-Main" panel is displayed, select "OS/2 Full Screen" from it.

6. Insert the XGA Device Drivers Diskette #2 in drive A.

7. If you have only one display attached to your system unit, type:

    A:INSTALL and press the Enter key.

If you have more than one display attached to your system unit, type:

A:INSTALL 2 and press the Enter key.

8. When you are told to "Insert a Device Support diskette in drive A:", press the Enter key.

9. The "Select a Device Driver" panel appears. Select "OS/2 1.2 Device Driver for IBM XGA".

10. Follow the instructions displayed to complete the installation of the driver.

11. When you are told to insert the OS/2 Installation Diskette in drive A, do so, and DO NOT remove it until you are asked to.

12. If you have an 8514/A adapter installed in your system unit, and you want to connect a display to it, do so now.

When you have completed the installation of the device driver, there may be configuration options for you to select, depending on the amount of video memory your system has, as follows:

o If your system has 1 MB of video memory (eight Video Ram modules installed), no options are needed. Your system will run at high resolution (displaying 1024x768 pixels - or dots- on the screen) and can also display up to 256 different colors.

o If your system has 0.5 MB of video memory (only four Video Ram modules installed), you can choose to operate at either:

- high resolution (1024x768 pixels) with the ability to display up to 16 colors simultaneously, but without the ability to use antialiased text,

OR

- medium resolution (640x480 pixels) with the ability to display up to 256 colors simultaneously, and the ability to use antialiased text.

If your system has 0.5 MB of video memory, the first time you start OS/2, high resolution (1024x768 pixels) with 16 colors is selected. You can change to medium resolution (640x480 pixels) with 256 colors by running the XGASETUP program, as follows:

In the OS/2 Desktop Manager, select Screen Setup. In the Screen Setup group, run Screen Configuration, which lets you change screen resolution options.

If you do not know how much video memory you have, you can run the XGASETUP program as described above to see if there are any options available to you.

Option choice depends on the type of application programs you use. High resolution lets you get more information on the screen, but colors are limited to 16. Choose this configuration if your applications do not use more than 16 colors. If more than 16 displayed colors are important, for example, to better match color images, or you want to use antialiased text, select medium resolution with 256 colors. Note that the XGASETUP program can be run at any time, so you can change the mode setting as you prefer.

Selection of Antialiased Text

This device driver supports antialiased text, which offers improved appearance and readability of text displayed on CRT display screens.

Antialiased text is available on systems that support 256 colors as follows:

o If your system has 0.5 MB of video memory, and you configure it as high resolution (1024x768 pixels) with 16 colors, antialiased text IS NOT available.

o If your system has 0.5 MB of video memory, and you configure it as medium resolution (640x480 pixels) with 256 colors, antialiased text IS available.

o If your system has 1 MB of video memory, antialiased text IS available.

You can select an antialiased system font after you have installed the device driver as follows:

1) From the OS/2 Desktop Manager panel, select Screen Setup.

2) From the Screen Setup group panel, run Select System Font.

This program lets you view antialiased fonts and compare them with the regular system font before deciding which to use. You can rerun the program at any time if you want to select a different font.

Options to select fonts, including antialiased fonts, are also available with the OS/2 File Manager and the OS/2 System Editor.

For more detailed information on antialiased fonts, with information of particular interest to application developers, please look at the file AAFONTS.TXT on this diskette.

Appendix D. Fonts Under OS/2 V1.3

The table on the next page is a summary of the font related discussion in Chapter 9, “Fonts Usage in OS/2 V1.3” on page 61.

Table 13. Font Usage in OS/2 V1.3

Font Name	Type	Spacing	Chars Size	Font Size	Source	Install.	Remove	Location	File Type
Screen(Hardware)	Bitmap	Fixed	8x8 to 40x16	4 to 20 KB	n/a	n/a	n/a	Video Ctrl	
VIO VGA	Bitmap	Fixed	8x8 to 18x8	135 KB (DLL)	OS/2 Dskt	OS/2 Inst	n/a	\OS2\DLL	DISPLAY.DLL
VIO 8514	Bitmap	Fixed	8x8 to 30x12	284 KB (DLL)	OS/2 Dskt	OS/2 Inst	n/a	\OS2\DLL	DISPLAY.DLL
VIO XGA	Bitmap	Fixed	8x8 to 30x12	468 KB (DLL)	XGA Dskt	XGA Inst	n/a	\OS2\DLL	DISPLAY.DLL
OS/2 System	Bitmap	Prop.	10 pt	DISPLAY.DLL	OS/2 Dskt	OS/2 Inst	n/a	\OS2\DLL	DISPLAY.DLL
OS/2 Bitmap	Bitmap	Prop.	8 pt to 24 pt	20 to 330 KB	OS/2 Dskt	OS/2 Inst	Ctl Pnl	\OS2\DLL	FON
OS/2 Outline	Outline	Prop.	Any	119 KB	OS/2 Dskt	OS/2 Inst	Ctl Pnl	\OS2\DLL	FON
SAA Core	Outline	Prop.	Any	120 to 160KB	OS/2 Dskt	OS/2 Inst	Ctl Pnl	\OS2\DLL	PSF
Adobe Type 1	Outline	Prop.	Any		OS/2 Dskt	Ctl Pnl	Ctl Pnl	\OS2\DLL	AFM/PFB
XGA AntiAliased	Bitmap	Prop.	8 to 24 pt	30 to 190 KB	XGA Dskt	AAINST	Ctl Pnl	\OS2\DDFONTS	FNT
TrueType	Outline	Prop.	Any	N/A	Not Yet Available				
4019 Download	Bitmap	Prop.	6 to 30 pt	100 KB	IBM +	Dev Drv	Dev Drv	\4019FONT	DLF / FMF
Type 1 Download	Outline	Prop.	Any	100 KB	Adobe +	Dev Drv	Dev Drv	\PSFONTS	PFA / PFM
LaserJet	Bitmap	Prop.	4 to 200 pt	20 to 300 KB	HP +	Dev Drv	Dev Drv	\OS2\DLL\HP\PCL	SFP / FNT

Appendix E. Install Log Files

The following two sample listings can be found on any regular OS/2 system in the file C:\OS2\INSTALL\INSTALL.LOG. This should give you a better idea what happens during an OS/2 system installation or update.

E.1 INSTALL.LOG File for OS/2 V1.3

```
                  INSTALL.LOG

Installing Operating System/2.
Model = F8
Submodel = 00
Greater than 12MB partition exists.
Copying files E:\INSTALL.LOG -> C:\
Copying files
 C:\OS2\INSTALL\INSTALL.LOG -> C:\OS2\INSTALL\INSTALL.BAK
Making directory C:\OS2
Making directory C:\OS2\DLL
Making directory C:\OS2\HELP
Making directory C:\OS2\INSTALL
Making directory C:\OS2\SYSTEM
Making directory C:\OS2\BOOK
Making directory C:\OS2\INTRO
Making directory C:\SPOOL
Deleting file c:\clock02.sys
Deleting file c:\disk02.sys
Deleting file c:\kbd02.sys
Deleting file c:\os2\backup.com
Deleting file c:\os2\com01.sys
Deleting file c:\os2\com02.sys
Deleting file c:\os2\fdisk.com
Deleting file c:\os2\install\sysinst.lib
Deleting file c:\os2\mouseb00.sys
Deleting file c:\os2\mouseb01.sys
Deleting file c:\os2\mouseb02.sys
Deleting file c:\os2\mouseb05.sys
Deleting file c:\os2\restore.com
Deleting file c:\os2\system\unp.msg
Deleting file c:\os2\system\unph.msg
Deleting file c:\print02.sys
Deleting file c:\screen02.sys
Deleting file c:\os2\system\swapper.dat
Deleting file c:\os2\viohelp.exe
Deleting file c:\os2\splprmap.pdf
Deleting file c:\os2\dll\splprmap.dll
Deleting file c:\os2\dll\pmc90000.tff
Deleting file c:\os2\system\ispm.msg
Deleting file c:\os2\system\ispd.msg
Deleting file c:\os2\system\dmpc.exe
Deleting file c:\os2\install\sysbas.exe
Deleting file c:\os2\spl2b.pdf
Deleting file c:\os2\dll\spl2b.dll
Deleting file c:\os2\dll\ibmega.dll
Deleting file c:\os2\dll\ibmvga.dll
Deleting file c:\os2\dll\ibmbga.dll
Deleting file C:\OS2\4201.DCP
Deleting file C:\OS2\5202.DCP
Deleting file C:\OS2\KEYB.COM
Deleting file C:\OS2\KEYBOARD.DCP
Dual Boot installed.
OS2 DISK 1 is being copied to your fixed disk
Copying files A:\UNPACK.EXE -> C:\OS2\UNPACK.EXE
C:\OS2\UNPACK.EXE A:\CONVERT.EX@ E:\
A:\CONVERT.EX@
        0 file(s) copied.
        1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\MOVESPL.EX@ E:\
A:\MOVESPL.EX@
        0 file(s) copied.
        1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\DOSRFICO.DL@ E:\
A:\DOSRFICO.DL@
        0 file(s) copied.
        1 file(s) unpacked.
C:\OS2\ATTRIB.EXE
C:\OS2\CHKDSK.COM
C:\OS2\FORMAT.COM
C:\OS2\MAKEINI.EXE
C:\OS2\DLL\OS2SMDUM.DLL
C:\OS2\DLL\PMWINDUM.DLL
        2 file(s) copied.
        4 file(s) unpacked.
C:\OS2\CACHE.EXE
C:\OS2\DLL\STARTLW.DLL
C:\OS2\DLL\UHPFS.DLL
        2 file(s) copied.
        1 file(s) unpacked.
C:\OS2\KEYB.COM
C:\OS2\KEYBOARD.DCP
        2 file(s) copied.
        0 file(s) unpacked.
C:\OS2\BACKUP.EXE
C:\OS2\FDISKPM.EXE
C:\OS2\LABEL.COM
C:\OS2\RECOVER.COM
C:\OS2\RESTORE.EXE
C:\OS2\SORT.EXE
C:\OS2\TREE.COM
C:\OS2\DLL\FDISKPM.DLL
C:\OS2\HELP\FDISKPMH.HLP
        3 file(s) copied.
        6 file(s) unpacked.
OS2 DISK 1 copy is complete
OS2 DISK 2 is being copied to your fixed disk
C:\OS2\UNPACK.EXE A:\IBMMOU02.SY@ C:\OS2
A:\IBMMOU02.SY@
        0 file(s) copied.
        1 file(s) unpacked.
C:\OS2\COM02.SYS
        0 file(s) copied.
        1 file(s) unpacked.
C:\OS2\ANSI.EXE
C:\OS2\COMP.COM
C:\OS2\DISKCOMP.COM
C:\OS2\DISKCOPY.COM
C:\OS2\E.EXE
C:\OS2\EAUTIL.EXE
C:\OS2\FIND.EXE
C:\OS2\MODE.COM
C:\OS2\MORE.COM
C:\OS2\PRINT.COM
C:\OS2\REPLACE.EXE
C:\OS2\SYSLEVEL.EXE
C:\OS2\XCOPY.EXE
C:\OS2\DLL\BMSCALLS.DLL
C:\OS2\DLL\EHXDLMRI.DLL
C:\OS2\DLL\HELPMGR.DLL
C:\OS2\DLL\HPMGRMRI.DLL
C:\OS2\DLL\MONCALLS.DLL
```

```
C:\OS2\DLL\NAMPIPES.DLL
C:\OS2\DLL\PMMLE.DLL
C:\OS2\HELP\EHXHP.HLP
C:\OS2\HELP\HMHELP.HLP
C:\OS2\INSTALL\DDINSTAL.EXE
C:\OS2\SYSTEM\OSO001H.MSG
C:\OS2\SYSTEM\SWAPPER.EXE
       0 file(s) copied.
      25 file(s) unpacked.
C:\OS2\BASIC.COM
C:\OS2\BASICA.COM
C:\OS2\MORTGAGE.BAS
C:\OS2\SETCOM40.EXE
C:\OS2\SUBST.EXE
       0 file(s) copied.
       5 file(s) unpacked.
C:\OS2\BOOT.COM
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\MOUSE.SYS
C:\OS2\POINTDD.SYS
       0 file(s) copied.
       2 file(s) unpacked.
C:\OS2\LINK.EXE
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\DLL\STXTDMPC.DLL
C:\OS2\INSTALL\DMPC.EXE
C:\OS2\INSTALL\INSTAID.CMD
C:\OS2\INSTALL\INSTAID.CNF
C:\OS2\INSTALL\INSTAID.LIB
C:\OS2\INSTALL\INSTAID.PRO
C:\OS2\INSTALL\INSTAIDE.EXE
C:\OS2\INSTALL\ISPD.MSG
C:\OS2\INSTALL\ISPM.MSG
       0 file(s) copied.
       9 file(s) unpacked.
OS2 DISK 2 copy is complete
OS2 DISK 3 is being copied to your fixed disk
C:\OS2\DOSCALLS.LIB
C:\OS2\EXTDSKDD.SYS
C:\OS2\HELP.BAT
C:\OS2\HELP.CMD
C:\OS2\HELPMSG.EXE
C:\OS2\PMCPL.EXE
C:\OS2\PMEXEC.EXE
C:\OS2\PMFILE.EXE
C:\OS2\PMSHELL.EXE
C:\OS2\SYSLOG.EXE
C:\OS2\VIOTBL.DCP
C:\OS2\DLL\PMCPL.DLL
C:\OS2\DLL\PMEXEC.DLL
C:\OS2\DLL\PMFILE.DLL
C:\OS2\DLL\PMSHELL.DLL
C:\OS2\DLL\PMSHLTKT.DLL
C:\OS2\HELP\PMCPLH.HLP
C:\OS2\HELP\PMEXECH.HLP
C:\OS2\HELP\PMFILEH.HLP
C:\OS2\HELP\PMSHELLH.HLP
       0 file(s) copied.
      20 file(s) unpacked.
C:\OS2\VIEW.EXE
C:\OS2\VIEWDOC.EXE
C:\OS2\HELP\VIEWH.HLP
C:\OS2\BOOK\CMDREF.INF
       0 file(s) copied.
       4 file(s) unpacked.
C:\OS2\ANSI.SYS
C:\OS2\APPEND.EXE
C:\OS2\ASSIGN.COM
C:\OS2\COMMAND.COM
C:\OS2\DOS.SYS
C:\OS2\EDLIN.COM
C:\OS2\EGA.SYS
```

```
C:\OS2\GRAFTABL.COM
C:\OS2\JOIN.EXE
       0 file(s) copied.
       9 file(s) unpacked.
C:\OS2\DOS.SYS
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\INTRO\INTROE.EXE
C:\OS2\INTRO\INTROI.TUT
C:\OS2\INTRO\INTROS.TUT
       0 file(s) copied.
       3 file(s) unpacked.
C:\OS2\CREATEDD.EXE
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\DLL\CPISPFPC.DLL
C:\OS2\DLL\DTM.DLL
C:\OS2\DLL\INACALL.DLL
       0 file(s) copied.
       3 file(s) unpacked.
OS2 DISK 3 copy is complete
OS2 DISK 4 is being copied to your fixed disk
C:\OS2\UNPACK.EXE A:\COURIER.PS@ C:\OS2\DLL
A:\COURIER.PS@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\HELVETIC.PS@ C:\OS2\DLL
A:\HELVETIC.PS@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\TIMESNRM.PS@ C:\OS2\DLL
A:\TIMESNRM.PS@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\COURIER.FO@ C:\OS2\DLL
A:\COURIER.FO@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\HELV.FO@ C:\OS2\DLL
A:\HELV.FO@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\TIMES.FO@ C:\OS2\DLL
A:\TIMES.FO@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\OS2SM.PDF
C:\OS2\PMGPI.PDF
C:\OS2\PMPIC.PDF
C:\OS2\PMSHAPI.PDF
C:\OS2\PMSPL.PDF
C:\OS2\SPL1B.PDF
C:\OS2\DLL\OS2SM.DLL
C:\OS2\DLL\PMATM.DLL
C:\OS2\DLL\PMC00000.TFF
C:\OS2\DLL\PMC10000.TFF
C:\OS2\DLL\PMC20000.TFF
C:\OS2\DLL\PMC30000.TFF
C:\OS2\DLL\PMC40000.TFF
C:\OS2\DLL\PMC50000.TFF
C:\OS2\DLL\PMC60000.TFF
C:\OS2\DLL\PMC70000.TFF
C:\OS2\DLL\PMC80000.TFF
C:\OS2\DLL\PMPIC.DLL
C:\OS2\DLL\PMVIOP.DLL
C:\OS2\DLL\SYSMONO.FON
C:\OS2\HELP\PMVIOP.HLP
C:\OS2\HELP\PSMH.HLP
       0 file(s) copied.
      22 file(s) unpacked.
C:\OS2\PICICHG.EXE
C:\OS2\PICPRINT.EXE
C:\OS2\PICSHOW.EXE
C:\OS2\HELP\PICH.HLP
C:\OS2\HELP\PICPH.HLP
```

```
       0 file(s) copied.
       5 file(s) unpacked.
OS2 DISK 4 copy is complete
OS2 DISK 5 is being copied to your fixed disk
C:\OS2\UNPACK.EXE A:\BVHVGA.DL@ C:\OS2\DLL
A:\BVHVGA.DL@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\BVH8514A.DL@ C:\OS2\DLL
A:\BVH8514A.DL@
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\UNPACK.EXE A:\IBMBGA.DL@ C:\OS2\DLL
A:\IBMBGA.DL@
       0 file(s) copied.
       1 file(s) unpacked.
Renaming files C:\OS2\DLL\IBMBGA.DLL -> C:\OS2\DLL\DISPLAY.DLL
C:\OS2\PMDD.SYS
C:\OS2\PMGRE.PDF
C:\OS2\PMWIN.PDF
C:\OS2\DLL\BVHWNDW.DLL
C:\OS2\DLL\FKA.DLL
C:\OS2\DLL\PMAVIO.DLL
C:\OS2\DLL\PMBIND.DLL
C:\OS2\DLL\PMGPI.DLL
C:\OS2\DLL\PMSHAPI.DLL
C:\OS2\DLL\PMTKT.DLL
C:\OS2\DLL\PMWIN.DLL
       0 file(s) copied.
      11 file(s) unpacked.
C:\OS2\DLL\PMTRACE.DLL
       0 file(s) copied.
       1 file(s) unpacked.
OS2 DISK 5 copy is complete
OS2 DISK 6 is being copied to your fixed disk
C:\OS2\PMREXX.EXE
C:\OS2\REXXTRY.CMD
C:\OS2\RXQUEUE.EXE
C:\OS2\RXSUBCOM.EXE
C:\OS2\DLL\PMREXX.DLL
C:\OS2\DLL\REXX.DLL
C:\OS2\DLL\REXXAPI.DLL
C:\OS2\DLL\REXXINIT.DLL
C:\OS2\HELP\PMREXX.HLP
C:\OS2\SYSTEM\REX.MSG
C:\OS2\SYSTEM\REXH.MSG
       0 file(s) copied.
      11 file(s) unpacked.
C:\OS2\DLL\PMDRAG.DLL
C:\OS2\DLL\PMGRE.DLL
C:\OS2\DLL\SPINBTN.DLL
C:\OS2\SYSTEM\LOGDAEM.EXE
       0 file(s) copied.
       4 file(s) unpacked.
C:\OS2\DLL\TUT.DLL
       0 file(s) copied.
       1 file(s) unpacked.
C:\OS2\4201.DCP
C:\OS2\5202.DCP
       0 file(s) copied.
       2 file(s) unpacked.
C:\OS2\PATCH.EXE
C:\OS2\PSTAT.EXE
C:\OS2\TRACE.EXE
C:\OS2\TRACEFMT.EXE
C:\OS2\TRSEL.EXE
       0 file(s) copied.
       5 file(s) unpacked.
C:\OS2\PMPRINST.EXE
C:\OS2\PMSPOOL.EXE
C:\OS2\SPOOL.EXE
C:\OS2\DLL\PMPRINT.QPR
C:\OS2\DLL\PMSPL.DLL
C:\OS2\DLL\PMSPOOL.DLL
C:\OS2\DLL\SPL1B.DLL
C:\OS2\DLL\SPOOLCP.DLL
C:\OS2\HELP\PMPRINST.HLP
C:\OS2\HELP\PMSPOOLH.HLP
C:\OS2\SYSTEM\SPL.MSG
C:\OS2\SYSTEM\SPLH.MSG
       0 file(s) copied.
      12 file(s) unpacked.
OS2 DISK 6 copy is complete
OS2 INSTALL is being copied to your fixed disk
Copying files A:\8514.RC@ -> C:\OS2\INI.RC
C:\OS2\UNPACK.EXE A:\8514.RC@ C:\OS2
A:\8514.RC@
       0 file(s) copied.
       1 file(s) unpacked.
Renaming files C:\OS2\8514.RC -> C:\OS2\INI.RC
C:\OS2\UNPACK.EXE A:\UPINI.RC@ E:\
A:\UPINI.RC@
       0 file(s) copied.
       1 file(s) unpacked.
C:\BASEDD02.SYS
C:\DISK02.SYS
       2 file(s) copied.
       0 file(s) unpacked.
C:\OS2\CMD.EXE
C:\OS2\INISYS.RC
C:\OS2\VDISK.SYS
C:\OS2\DLL\ANSICALL.DLL
C:\OS2\DLL\BKSCALLS.DLL
C:\OS2\DLL\BVHINIT.DLL
C:\OS2\DLL\BVSCALLS.DLL
C:\OS2\DLL\DOSCALL1.DLL
C:\OS2\DLL\KBDCALLS.DLL
C:\OS2\DLL\MOUCALLS.DLL
C:\OS2\DLL\MSG.DLL
C:\OS2\DLL\NLS.DLL
C:\OS2\DLL\QUECALLS.DLL
C:\OS2\DLL\SESMGR.DLL
C:\OS2\DLL\VIOCALLS.DLL
C:\OS2\INSTALL\SYSLEVEL.OS2
C:\OS2\SYSTEM\COUNTRY.SYS
C:\OS2\SYSTEM\HARDERR.EXE
C:\OS2\SYSTEM\OSO001.MSG
      18 file(s) copied.
       1 file(s) unpacked.
C:\OS2\HPFS.IFS
       1 file(s) copied.
       0 file(s) unpacked.
OS2 INSTALL copy is complete
Copying files C:\INSTALL.LOG -> E:\
System files are being copied to your fixed disk
The type of file system for the disk is FAT.
The system files have been transferred.
System file transfer is complete
Copying files E:\INSTALL.LOG -> C:\OS2\INSTALL\INSTALL.LOG
Renaming files C:\OS2\DLL\PMWINDUM.DLL -> C:\OS2\INSTALL\DDINST1.DLL
Renaming files C:\OS2\DLL\OS2SMDUM.DLL -> C:\OS2\INSTALL\DDINST2.DLL
C:\OS2\MAKEINI.EXE C:\OS2\OS2.INI C:\OS2\INI.RC E:\UPINI.RC
C:\OS2\MAKEINI.EXE C:\OS2\OS2SYS.INI C:\OS2\INISYS.RC
Copying files A:\*.DAT -> C:\OS2\DLL
Copying files A:\IBM42XX.DRV -> C:\OS2\DLL\IBM42XX\IBM42XX.DRV
Copying files A:\IBM5202.DRV -> C:\OS2\DLL\IBM5202\IBM5202.DRV
Installation is complete
```

E.2 INSTALL.LOG File for OS/2 V1.1

```
MDL=F8   SUBMDL=00   ADT=7
>9MB PARTITION EXISTS
DO NOT FORMAT PARTITION
FORMATTING
TRANSFERRING THE SYSTEM
The system files have been transferred.
COPYING IBM OS/2 INSTALLATION DISKETTE
CREATE DIRECTORIES
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
C:\*.DLL
SYS0002: The system cannot find the file specified.
        0 file(s) copied.
A:\ANSICALL.DLL
A:\BKSCALLS.DLL
A:\BMSCALLS.DLL
A:\BVSCALLS.DLL
A:\CPISPFPC.DLL
A:\DOSCALL1.DLL
A:\DTM.DLL
A:\KBDCALLS.DLL
A:\MONCALLS.DLL
A:\MOUCALLS.DLL
A:\MSG.DLL
A:\NLS.DLL
A:\QUECALLS.DLL
A:\SESMGR.DLL
A:\VIOCALLS.DLL
       15 file(s) copied.
A:\INSTAID.CMD
A:\INSTAID.CNF
A:\INSTAID.LIB
A:\INSTAID.PRO
A:\INSTAIDE.EXE
        5 file(s) copied.
A:\SYSINST.CMD
A:\SYSINST.CNF
A:\SYSINST.LIB
A:\SYSINST.PIP
A:\SYSINST.RSP
        5 file(s) copied.
A:\ISPD.MSG
A:\ISPM.MSG
A:\OSO001.MSG
        3 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
A:\F80000.BIO
A:\F80100.BIO
A:\F80402.BIO
A:\F80902.BIO
A:\F80D00.BIO
A:\FC0400.BIO
A:\FC0403.BIO
A:\FC0500.BIO
        8 file(s) copied.
A:\KBD*.DCP
SYS0002: The system cannot find the file specified.
        0 file(s) copied.
A:\VTBL*.DCP
SYS0002: The system cannot find the file specified.
        0 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
```

```
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
COPYING IBM OS/2 DISKETTE 1
A:\ANSI.SY@
A:\ASSIGN.CO@
A:\BACKUP.CO@
A:\BASIC.CO@
A:\BASICA.CO@
A:\CHKDSK.COM
A:\COM01.SY@
A:\COM02.SY@
A:\COMMAND.CO@
A:\COMP.CO@
A:\DISKCOMP.CO@
A:\DISKCOPY.CO@
A:\EDLIN.CO@
A:\EGA.SY@
A:\EXTDSKDD.SY@
A:\GRAFTABL.CO@
A:\HELP.BA@
A:\HELP.CM@
A:\KEYB.CO@
A:\LABEL.CO@
A:\MODE.CO@
A:\MORE.CO@
A:\MORTGAGE.BA@
A:\MOUSEA00.SY@
A:\MOUSEA01.SY@
A:\MOUSEA02.SY@
A:\MOUSEA03.SY@
A:\MOUSEA04.SY@
A:\MOUSEA05.SY@
A:\MOUSEB00.SY@
A:\MOUSEB01.SY@
A:\MOUSEB02.SY@
A:\MOUSEB05.SY@
A:\POINTDD.SY@
A:\PRINT.CO@
A:\RECOVER.CO@
A:\RESTORE.CO@
A:\TREE.CO@
A:\UNPACK.EXE
A:\VDISK.SY@
        2 file(s) copied.
       38 file(s) unpacked.
        1 file(s) copied.
COPYING IBM OS/2 DISKETTE 2
A:\ANSI.EX@
A:\APPEND.EX@
A:\ATTRIB.EX@
A:\CONVERT.EX@
A:\CREATEDD.EX@
A:\DDINSTAL.EX@
A:\DOSCALLS.LI@
A:\E.DE@
A:\E.EX@
A:\E.MS@
A:\FIND.EX@
A:\HELPMSG.EX@
A:\JOIN.EX@
A:\KEYBOARD.DC@
A:\LINK.EX@
A:\NAMPIPES.DL@
A:\OSO001H.MS@
A:\OS2SMDUM.DL@
A:\PATCH.EX@
A:\PMWINDUM.DL@
A:\REPLACE.EX@
A:\SETCOM40.EX@
A:\SORT.EX@
A:\SPOOL.EX@
```

```
A:\STXTDMPC.DL@
A:\SUBST.EX@
A:\SYSLEVEL.EX@
A:\SYSLEVEL.OS@
A:\TRACE.EX@
A:\TRACEFMT.EX@
A:\UNP.MS@
A:\UNPH.MS@
A:\VIOTBL.DC@
A:\XCOPY.EX@
A:\4201.DC@
A:\5202.DC@
        0 file(s) copied.
       36 file(s) unpacked.
        1 file(s) copied.
        1 file(s) copied.
C:\OS2\UNP.MSG
C:\OS2\UNPH.MSG
        2 file(s) copied.
C:\OS2\NAMPIPES.DLL
C:\OS2\OS2SMDUM.DLL
C:\OS2\PMWINDUM.DLL
C:\OS2\STXTDMPC.DLL
        4 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
COPYING IBM OS/2 DISKETTE 3
A:\PMGPI.DL@
A:\PMSPL.DL@
A:\PMGRE.DL@
A:\PMWIN.DL@
A:\OS2SPLFS.DL@
A:\PMTKT.DL@
A:\PMTRACE.DL@
A:\SPL1B.DL@
A:\SPL2B.DL@
A:\SPLPRMAP.DL@
A:\SPOOLCP.DL@
        0 file(s) copied.
       11 file(s) unpacked.
A:\PMCPL.EX@
A:\PMEXEC.EX@
A:\PMFILE.EX@
A:\PMSHELL.EX@
A:\PMSPOOL.EX@
A:\SPLDVWRK.EX@
A:\SPLPRN.EX@
A:\VIOHELP.EX@
        0 file(s) copied.
        8 file(s) unpacked.
A:\PMDD.SY@
        0 file(s) copied.
        1 file(s) unpacked.
A:\COURIER.FO@
        0 file(s) copied.
```

```
        1 file(s) unpacked.
A:\PMPRINT.QP@
        0 file(s) copied.
        1 file(s) unpacked.
A:\INTRO\INTROE.EX@
A:\INTRO\INTROI.TU@
A:\INTRO\INTROS.TU@
        0 file(s) copied.
        3 file(s) unpacked.
        1 file(s) copied.
        1 file(s) copied.
COPYING IBM OS/2 DISKETTE 4
A:\PMAVIO.DL@
A:\PMSHAPI.DL@
A:\PMVIOP.DL@
A:\IBMBGA.DL@
A:\IBMEGA.DL@
A:\IBMVGA.DL@
A:\OS2SM.DL@
        0 file(s) copied.
        7 file(s) unpacked.
A:\PMC00000.TF@
A:\PMC10000.TF@
A:\PMC20000.TF@
A:\PMC30000.TF@
A:\PMC60000.TF@
A:\PMC70000.TF@
A:\PMC80000.TF@
A:\PMC90000.TF@
A:\PMC50000.TF@
        0 file(s) copied.
        9 file(s) unpacked.
A:\PMGRE.PD@
A:\PMWIN.PD@
A:\PMGPI.PD@
A:\PMSHAPI.PD@
A:\OS2SM.PD@
A:\PMSPL.PD@
A:\SPL1B.PD@
A:\SPL2B.PD@
A:\SPLPRMAP.PD@
        0 file(s) copied.
        9 file(s) unpacked.
A:\TRSEL.EX@
        0 file(s) copied.
        1 file(s) unpacked.
A:\IBM4201.DR@
        0 file(s) copied.
        1 file(s) unpacked.
A:\HELV.FO@
A:\TIMES.FO@
        0 file(s) copied.
        2 file(s) unpacked.
        1 file(s) copied.
COUNTRY=001  USADEF=YES
CREATE SPOOL DIRECTORY
```

Appendix F. OS/2 V1.3 Extended Edition Memory and Fixed Disk Requirements

The worksheets on the following pages, which have been copied from the ***IBM Operating System/2 Extended Edition Version 1.3 (290-690) Announcement Letter***, can be used to help determine the minimum required hardware-capacity requirements for a particular OS/2 configuration to achieve good performance. The first worksheet provides suggested steps to estimate the memory requirements for OS/2 functions. The second provides the steps necessary to estimate fixed-disk size for OS/2 functions.

Note: Memory and fixed-disk requirements vary according to the language compilers that a programmer uses. Refer to publications that accompany the language compilers for the memory requirement.

Migration of Database Manager Version 1.0 or 1.1 databases to Version 1.3 will increase the size of the database. Refer to ***IBM Operating System/2 Extended Edition Version 1.3 Database Manager Administrator's Guide (S01F-0267)*** for more information.

Table 14 (Page 1 of 2). Memory Estimating Worksheet for OS/2 EE V1.3

Functions	Recommended Memory (MB)	User Configuration
Operating System:		
Base (required)	1.5	
DOS compatibility[1]	0.5	
High Performance File System (HPFS)	0.3	
Active Spooling (while printing)	0.3	
System Performance Buffer[2]	0.5	
Communications Manager:		
Base (required for communications)	0.1	
3270 terminal emulation[3]	0.3	
3270 host print	0.3	
SNA gateway[4]	0.1	
5250 Work Station Feature[4]	0.2	
ASCII terminal emulation	0.3	
APPC	0.4	
LUA[4]	0.2	
File transfer[5]	0.3	
Database Manager:[6]		
Database Services (standalone)	1.3	
Database Requester[7]	0.5	
Database Server[7]	1.5	
For each additional concurrent process	0.15	
Query Manager	1.6	

Table 14 (Page 2 of 2). *Memory Estimating Worksheet for OS/2 EE V1.3*

Functions	Recommended Memory (MB)	User Configuration
LAN Requester[8]	0.4	
OS/2 LAN Server Version 1.3[9]	2.3	
OS/2 Applications[10]		
Applications data.[11]		
TOTAL		

Notes:

1. Automatically installed but removable by changing PROTECTONLY=NO statement in the CONFIG.SYS file to PROTECTONLY=YES. This memory is optionally available to OS/2 applications whenever the DOS mode is in the background.
2. This additional system memory allows better performance for transient conditions such as program loading, program termination and print spooling.
3. If the 3270 connection is LAN, SDLC, or X.25 (that is, not a DFT coaxial connection), also include APPC.
4. APPC is a prerequisite for this function. (Add the APPC recommended memory to the total only once.)
5. Concurrent file-transfer function requires 0.3MB for each session.
6. Additional memory for database buffer pool or sort heaps for each database may be required depending on the user's workload.
7. Includes the necessary Database Services code. Communications Manager base and APPC are also required for Remote Data Services.
8. Communications Manager base is a prerequisite for LAN Requester. If the LAN Requester full screen panels are not used (that is, only the command line interface is used) reduce the 0.4MB to 0.1MB.
9. This number depends on the installation parameters. Refer to ***IBM OS/2 LAN Server Version 1.3 Network Administrator's Guide (S33F-9428)*** for more information
10. Provided by the supplier of the application.
11. Dependent on the application and user.

Table 15 (Page 1 of 2). *Fixed-Disk Estimating Worksheet for OS/2 EE V1.3*

Functions	Required Fixed Disk (MB)	User Configuration
Operating System:		
Base (required)[1]	12.0	
Base (optional)[2]	3.5	
Segment swap data set[3]	2.5	
Communications Manager:		
Base (required for communications)	2.7	
3270 terminal emulation	0.6	
SNA gateway[4]	0.2	
5250 Work Station Feature[4]	0.3	
ASCII terminal emulation	0.4	
APPC	0.3	
LAN communications[5]	0.3	
X.25 communications	0.9	
Configuration services	2.9	
Terminal emulator keyboard remap	0.3	

Table 15 (Page 2 of 2). Fixed-Disk Estimating Worksheet for OS/2 EE V1.3

Functions	Required Fixed Disk (MB)	User Configuration
Programming interfaces (APIs)	1.3	
LUA[4]	0.2	
Database Manager:		
Database Services (standalone)	3.0	
Database Requester[6]	2.4	
Database Server (minimum)[6]	3.2	
For each additional database[7]	1.1	
Query Manager	3.2	
LAN Requester[8]	4.6	
OS/2 LAN Server Version 1.3 (minimum)	6.2	
OS/2 Applications[9]		
Applications data[10]		
TOTAL		

Notes:

1. Includes components unique to Extended Edition. For Standard Edition, reduce this amount by 2.1MB.
2. Approximate total of individually selectable options: SE Documentation = 0.5MB, EE Documentation = 0.5MB, Adobe fonts = 0.4MB, PM fonts = 0.5MB, HPFS = 0.3MB, Country support = 0,1MB, System utilities = 0.5MB, DOS utilities = 0.2MB, Picture utilities = 0.3MB, Diagnostic utilities = 0.2MB.
3. A 2MB to 4MB swap area is adequate for many environments. However, the swap area is dynamically allocated as applications require more memory than is available. A swap data set larger or smaller than shown may be needed.
4. APPC is a prerequisite for this function. (Include the APPC fixed-disk requirement in the calculations only once.)
5. Add this number for each type of LAN adapter that is installed.
6. Communications Manager and APPC are prerequisites for Remote Data Services. If DOS Database Requester is installed on Database Server, an additional 1.1MB is required.
7. This number is for table definitions and internal structures and should be added for each database. This number is subject to many variables, such as additional (secondary) or larger log files and temporary work files, and does not include user data. Actual fixed-disk requirements may differ.
8. Communications Manager base is a prerequisite for LAN Requester.
9. Provided by the supplier of the application.
10. Dependent on the application and user.

Appendix G. GATEWAY.DAT File

This section gives the listing of the ASCII file used by the Batch Configuration Utility to create the GATEWAY.CFG file.

Refer to the *IBM OS/2 Extended Edition Version 1.3 System Administrator's guide for Communications (SO1F-0302)* for detailed information on each statement.

```
*** Start of definition of GATEWAY.CFG
*
* File Name Record=Gateway
FNM GATEWAY
* Workstation Profile Record & comment
*
COM Gateway TR created by BCU from PROTO
*
* Change the Workstation profile record
* Portable Unit
* 3270 non-DFT not to be auto-started
* Communication Manager main menu display first
*
WSP 8573 061              2    1
*
* Update the 3270 TR connection profile record
* destination address = 400010000302
*
3ND      400010000302
*
* Update a TR Terminal session profile record
* session number = 01, 02
* session ID = SES1, SES2
*
3NT 01 SES1
3NT 02 SES2
*
* Update the SNA base profile record
* PU name = THT2A
* Network name = USIBMTH
* Node Id = A2001
*
SNA THT2A    USIBMTH  A2001
*
* Update the TR DLC profile record
* adapter number = 00
* DLC type = Token-Ring
* C&SM LAN id = NET1
* maximum RU size = 256
* maximum number of link stations = 29
*
LDL 00 1 NET1     00256     29
*
* Update the SNA Gateway Host connection profile record
* permanent connection = yes
* auto-logoff timeout = 999
* DLC type = Token-Ring
* adapter number = 00
* destination address = 400010000302
*
GWH 1 999 T 00 400010000302
*
* Add Pool to SNA Gateway Host connection
* Pool class = 0001
* LU address range (start) = 04
* LU address range (end) = 73
* Auto-Logoff = Yes
*
ADD GWH 0001 04 73 1
*
* Create SNA Gateway Workstation
```

```
* LU profile record (PU01 - LU02)
* model profile = M6 (default provided by CM)
* profile name = REQ0102
* pool class = 1
* PU name = PU01
* LU local address at workstation = 02
*
CRE GWW M6       REQ0102  0001    PU01     02
*
* Update of this record
* LU name at workstation = REQ0102
* DLC type = T
* adapter number = 00
* destination address = 400094AA0001
*
GWW REQ0102              T 00 400094AA0001
*
* Create SNA Gateway Workstation
* LU profile records (PU01 - LU03,LU04,LU05)
* model profile = REQ0102
* profile name = REQ0103, REQ0104, REQ0105
* pool class = 1
* PU name = PU01
* LU local address at workstation = 03, 04,05
*
CRE GWW REQ0102  REQ0103  0001    PU01     03
CRE GWW REQ0102  REQ0104  0001    PU01     04
CRE GWW REQ0102  REQ0105  0001    PU01     05
*
* Create SNA Gateway Workstation LU profile record (PU02 - LU02)
* model profile = REQ0102
* profile name = REQ0202
* pool class = 1
* PU name = PU02
* LU local address at workstation = 02
*
* update for destination address = 400094AA0002
*
* use it as model for others LU on same PU
*
CRE GWW REQ0102  REQ0202  0001    PU02     02
GWW REQ0202                 400094AA0002
CRE GWW REQ0202  REQ0203  0001    PU02     03
CRE GWW REQ0202  REQ0204  0001    PU02     04
CRE GWW REQ0202  REQ0205  0001    PU02     05
*
* idem for others PUs
*
CRE GWW REQ0102  REQ0302  0001    PU03     02
GWW REQ0302                 400094AA0003
CRE GWW REQ0302  REQ0303  0001    PU03     03
CRE GWW REQ0302  REQ0304  0001    PU03     04
CRE GWW REQ0302  REQ0305  0001    PU03     05
*
CRE GWW REQ0102  REQ0402  0001    PU04     02
GWW REQ0402                 400094AA0004
CRE GWW REQ0402  REQ0403  0001    PU04     03
CRE GWW REQ0402  REQ0404  0001    PU04     04
CRE GWW REQ0402  REQ0405  0001    PU04     05
*
CRE GWW REQ0102  REQ0502  0001    PU05     02
```

```
GWW REQ0502                        400094AA0005
CRE GWW REQ0502  REQ0503  0001     PU05      03
CRE GWW REQ0502  REQ0504  0001     PU05      04
CRE GWW REQ0502  REQ0505  0001     PU05      05
*
CRE GWW REQ0102  REQ0602  0001     PU06      02
GWW REQ0602                        400094AA0006
CRE GWW REQ0602  REQ0603  0001     PU06      03
CRE GWW REQ0602  REQ0604  0001     PU06      04
CRE GWW REQ0602  REQ0605  0001     PU06      05
*
CRE GWW REQ0102  REQ0702  0001     PU07      02
GWW REQ0702                        400094AA0007
CRE GWW REQ0702  REQ0703  0001     PU07      03
CRE GWW REQ0702  REQ0704  0001     PU07      04
CRE GWW REQ0702  REQ0705  0001     PU07      05
*
CRE GWW REQ0102  REQ0802  0001     PU08      02
GWW REQ0802                        400094AA0008
CRE GWW REQ0802  REQ0803  0001     PU08      03
CRE GWW REQ0802  REQ0804  0001     PU08      04
CRE GWW REQ0802  REQ0805  0001     PU08      05
*
CRE GWW REQ0102  REQ0902  0001     PU09      02
GWW REQ0902                        400094AA0009
CRE GWW REQ0902  REQ0903  0001     PU09      03
CRE GWW REQ0902  REQ0904  0001     PU09      04
CRE GWW REQ0902  REQ0905  0001     PU09      05
*
CRE GWW REQ0102  REQ1002  0001     PU10      02
GWW REQ1002                        400094AA0010
CRE GWW REQ1002  REQ1003  0001     PU10      03
CRE GWW REQ1002  REQ1004  0001     PU10      04
CRE GWW REQ1002  REQ1005  0001     PU10      05
*
CRE GWW REQ0102  REQ1102  0001     PU11      02
GWW REQ1102                        400094AA0011
CRE GWW REQ1102  REQ1103  0001     PU11      03
CRE GWW REQ1102  REQ1104  0001     PU11      04
CRE GWW REQ1102  REQ1105  0001     PU11      05
*
CRE GWW REQ0102  REQ1202  0001     PU12      02
GWW REQ1202                        400094AA0012
CRE GWW REQ1202  REQ1203  0001     PU12      03
CRE GWW REQ1202  REQ1204  0001     PU12      04
CRE GWW REQ1202  REQ1205  0001     PU12      05
*
CRE GWW REQ0102  REQ1302  0001     PU13      02
GWW REQ1302                        400094AA0013
CRE GWW REQ1302  REQ1303  0001     PU13      03
CRE GWW REQ1302  REQ1304  0001     PU13      04
CRE GWW REQ1302  REQ1305  0001     PU13      05
*
CRE GWW REQ0102  REQ1402  0001     PU14      02
GWW REQ1402                        400094AA0014
CRE GWW REQ1402  REQ1403  0001     PU14      03
CRE GWW REQ1402  REQ1404  0001     PU14      04
CRE GWW REQ1402  REQ1405  0001     PU14      05
*
CRE GWW REQ0102  REQ1502  0001     PU15      02
GWW REQ1502                        400094AA0015
CRE GWW REQ1502  REQ1503  0001     PU15      03
CRE GWW REQ1502  REQ1504  0001     PU15      04
CRE GWW REQ1502  REQ1505  0001     PU15      05
*
CRE GWW REQ0102  REQ1602  0001     PU16      02
GWW REQ1602                        400094AA0016
CRE GWW REQ1602  REQ1603  0001     PU16      03
CRE GWW REQ1602  REQ1604  0001     PU16      04
CRE GWW REQ1602  REQ1605  0001     PU16      05
*
CRE GWW REQ0102  REQ1702  0001     PU17      02
GWW REQ1702                        400094AA0017
CRE GWW REQ1702  REQ1703  0001     PU17      03
CRE GWW REQ1702  REQ1704  0001     PU17      04
```

```
CRE GWW REQ1702  REQ1705  0001     PU17      05
*
CRE GWW REQ0102  REQ1802  0001     PU18      02
GWW REQ1802                        400094AA0018
CRE GWW REQ1802  REQ1803  0001     PU18      03
CRE GWW REQ1802  REQ1804  0001     PU18      04
CRE GWW REQ1802  REQ1805  0001     PU18      05
*
CRE GWW REQ0102  REQ1902  0001     PU19      02
GWW REQ1902                        400094AA0019
CRE GWW REQ1902  REQ1903  0001     PU19      03
CRE GWW REQ1902  REQ1904  0001     PU19      04
CRE GWW REQ1902  REQ1905  0001     PU19      05
*
CRE GWW REQ0102  REQ2002  0001     PU20      02
GWW REQ2002                        400094AA0020
CRE GWW REQ2002  REQ2003  0001     PU20      03
CRE GWW REQ2002  REQ2004  0001     PU20      04
CRE GWW REQ2002  REQ2005  0001     PU20      05
*
CRE GWW REQ0102  REQ2102  0001     PU21      02
GWW REQ2102                        400094AA0021
CRE GWW REQ2102  REQ2103  0001     PU21      03
CRE GWW REQ2102  REQ2104  0001     PU21      04
CRE GWW REQ2102  REQ2105  0001     PU21      05
*
CRE GWW REQ0102  REQ2202  0001     PU22      02
GWW REQ2202                        400094AA0022
CRE GWW REQ2202  REQ2203  0001     PU22      03
CRE GWW REQ2202  REQ2204  0001     PU22      04
CRE GWW REQ2202  REQ2205  0001     PU22      05
*
CRE GWW REQ0102  REQ2302  0001     PU23      02
GWW REQ2302                        400094AA0023
CRE GWW REQ2302  REQ2303  0001     PU23      03
CRE GWW REQ2302  REQ2304  0001     PU23      04
CRE GWW REQ2302  REQ2305  0001     PU23      05
*
CRE GWW REQ0102  REQ2402  0001     PU24      02
GWW REQ2402                        400094AA0024
CRE GWW REQ2402  REQ2403  0001     PU24      03
CRE GWW REQ2402  REQ2404  0001     PU24      04
CRE GWW REQ2402  REQ2405  0001     PU24      05
*
CRE GWW REQ0102  REQ2502  0001     PU25      02
GWW REQ2502                        400094AA0025
CRE GWW REQ2502  REQ2503  0001     PU25      03
CRE GWW REQ2502  REQ2504  0001     PU25      04
CRE GWW REQ2502  REQ2505  0001     PU25      05
*
CRE GWW REQ0102  REQ2602  0001     PU26      02
GWW REQ2602                        400094AA0026
CRE GWW REQ2602  REQ2603  0001     PU26      03
CRE GWW REQ2602  REQ2604  0001     PU26      04
CRE GWW REQ2602  REQ2605  0001     PU26      05
*
CRE GWW REQ0102  REQ2702  0001     PU27      02
GWW REQ2702                        400094AA0027
CRE GWW REQ2702  REQ2703  0001     PU27      03
CRE GWW REQ2702  REQ2704  0001     PU27      04
CRE GWW REQ2702  REQ2705  0001     PU27      05
*
CRE GWW REQ0102  REQ2802  0001     PU28      02
GWW REQ2802                        400094AA0028
CRE GWW REQ2802  REQ2803  0001     PU28      03
CRE GWW REQ2802  REQ2804  0001     PU28      04
CRE GWW REQ2802  REQ2805  0001     PU28      05
*
* update IEEE 802.2 Adapter profile record
* adapter number = 0
* adapter address = 400094FF0001
* maximum link stations = 35
*
LN2 00 400094FF0001 035
*
```

List of Abbreviations

Note: These are some of the abbreviations that are found in this volume.

A

AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
APA	All Points Addressable
API	Application Programming Interface
APL	A Programming Language
ASCII	American (National) Standard Code for Information Interchange
ATM	Abode Type Manager

B

BASIC	Beginners All-Purpose Symbolic Instruction Code
BIOS	Basic Input Output System
BGA	Business Graphics Adapter (8514/A Adapter)
BMP	Bitmap (OS/2)

C

CCITT	Consultative Committee on International Telephone And Telegraph
CGA	Color Graphics Adapter
CGM	Computer Graphics Metafile
CMS	Conversational Monitor System
CP	Control Program
CPI	Common Programming Interface (SAA)
CPU	Computer Processor Unit
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube
CSD	Customer Service Diskette
CUA	Common User Access

D

DASD	Direct Access Storage Device
DCF	Document Composition Facility
DDE	Dynamic Data Exchange (OS/2)
DIA	Document Interchange Architecture
DLL	Dynamic Link Library
DM	Dialog Manager
DMA	Direct Memory Access
DOS	Disk Operating System
DPI	Dots Per Inch

E

EBCDIC	Extended Binary-Coded Decimal Interchange Code
EE	Extended Edition (OS/2)
EGA	Enhanced Graphics Adapter
EMEA	Europe, Middle East and Africa
EPS	Encapsulated PostScript
ESA	Enterprise System Architecture (MVS)

F

FAT	File Allocation Table
FAX	Facsimile
FIFO	First In First Out
FSD	File System Driver

G

Gb	Giga Bits
GB	Giga Bytes
GDT	Global Descriptor Table
GIF	Graphics Interchange Format
GML	Graphics Markup Language
GPI	Graphics Programming Interface
GPIB	General Purpose Interface Bus (IEEE 488)

H

HPFS	High Performance File System
HPGL	Hewlett-Packard Graphics Language (Plotter)

I

IEEE	Institute of Electrical and Electronical Engineers
IFF	Interchange File Format
IOCA	Image Object Content Architecture
IPF	Information Presentation Facility

IPFC	Information Presentation Facility Compiler
IPL	Initial Program Load
IRQ	Interrupt Request
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization

K

Kb	Kilo Bits
KB	Kilo Bytes

L

LAN	Local Area Network
LDT	Local Descriptor Table
LRU	Least Recently Used
LVB	Logical Video Buffer

M

Mb	Mega Bits
MB	Mega Bytes
MCA	Micro Channel Architecture
MCGA	Modified Color Graphics Adapter
MIPS	Millions Instructions Per Second
MVS	Multiple Virtual Storage

N

NLS	National Language Support
NVRAM	Non Violatable Random Access Memory

O

OEM	Other Equipment Manufacturer
OS	Operating System
OS/2	Operating System/2

P

PC	Personal Computer
PCL	Page Control Language (HP LaserJet)
PDP	Plasma Display Panel
PID	Process ID (OS/2)
PIF	Picture Interchange Format (IBM)
PM	Presentation Manager (OS/2)
PPAP	Personal Pageprinter Adapter Program (IBM)
PPDS	Personal Printer Data Stream (IBM)
PROFS	Professional Office System (IBM)
PS/2	Personal System/2
PWS	Programmable Workstation

Q

QM	Query Manager (OS/2)

R

RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
REXX	Restructured Extended Executor Language
RGB	Red, Green, Blue (Video Display Terminology)
ROM	Read Only Memory
R/W	Read / Write

S

SAA	System Application Architecture
SCSI	Small Computer Systems Interface
SDLC	Synchronous Data Link Control
SE	Standard Edition (OS/2)
SNA	Systems Network Architecture
SQL	Structured Query Language

T

Tb	Terra Bits
TB	Terra Bytes
TIFF	Tagged Image File Format

U

UK	United Kingdom

V

VM	Virtual Machine
VGA	Video Graphics Adapter
VTAM	Virtual Telecommunications Access Method

W

WAN	Wide Area Network
WORM	Write Once Read Many

WYSIWYG	What You See is What You Get

X

XA	Extended Architecture (MVS and VM)
XGA	Extended Graphics Adapter

Glossary

Notes:

> This glossary defines terms that are used with IBM Operating System/2 Version 1.3 (hereafter referred to as the OS/2 program).

A

abend. Synonym for *abnormal termination*.

abnormal termination. (1) A system failure, a program failure, or operator action that causes a program to end unsuccessfully. Contrast with *normal termination*. (2) In the OS/2 program, the unusual cessation of processing prior to termination. Synonymous with *abend*.

absolute path name. Synonym for *path*.

absolute priority. In the OS/2 program, pertains to the priority of a process that is not varied by the base operating system. Contrast with *dynamic priority*.

accelerator. A single keystroke that starts the processing of an application-defined function.

accented character. Synonym for *special character*.

access. (1) The manner in which files or data are referred to by a computer. (2) The controlled authorization to enter or to make use of objects. (3) In Database Manager, the several authority privileges such as Database privileges, Table privileges, or Plan privileges that can be granted to a user to allow use of the different systems. See also *authority*, *privilege*, and *privilege level*.

access control profile. A list of the access privileges assigned to users and groups for a particular network resource in a domain. There are two types of access profiles. See *discrete profile* and *generic profile*.

access mode. The manner in which an application gains access to a file it has opened: read-only, write-only, or read/write.

access path. In Database Manager, the path used to get to data specified in Structured Query Language (SQL) statements. An access path can involve an index or a sequential search or a combination of the two.

access plan. In Database Manager, a database object stored in the database that includes all of the information needed to process the Database Services statements of a single application program. An access plan is generated through processing of the SQLBIND program or through the precompile process if the bind option is used. Note also that applications can have multiple access plans. See *application plan*.

access priority. In the IBM Token-Ring Network, the maximum priority a token can have for the adapter to use for transmission.

address space. (1) The range of addresses available to a computer program. (2) The area of virtual storage available for a particular job.

access unit. See *multistation access unit*.

action. One of a set of predefined tasks a computer can perform.

action bar. The highlighted area at the top of a panel that contains the choices currently available in the application program that a user is running.

action point. The current position on the screen where the pointer is pointing. See also *hot spot*.

action text. The text added to panels and menus created in Query Manager that indicates the action a user has available to perform.

activate. To make an object, program, system, or device available for use.

active. Pertains to anything that is current or operational.

active program. A program currently running on the computer. Contrast with *inactive program*. See *interactive program*.

active session. The session in which a user is currently interacting with the computer. Synonymous with *foreground session*.

active window. The window with which the user is currently interacting.

adapter. (1) A piece of hardware that connects a computer and an external device. (2) See *communication adapter*.

adapter number. A specific number that identifies an adapter when more than one adapter is used in a workstation.

address. A value that identifies the location of a register, a particular part of storage, or a network node.

administrator. The person responsible for the designing, planning, installing, configuring, controlling, managing, and maintaining of a network, system, or database. See *system administrator, network administrator*, and *database administrator*.

alias. (1) An alternative name used to identify an object or a database. (2) A nickname set up by the network administrator for a file, printer, or serial device. (3) A name used to identify a network resource to a domain. Aliases are similar to network names but can be used only through the full-screen interface. See also *network name*.

all points addressable (APA). In computer graphics, pertaining to the ability to address and display or not display each picture element (pel) on a display surface.

alphanumeric. Pertaining to a character set that contains letters, digits, and usually other characters such as punctuation marks.

alternate (Alt) mode. A mode that is obtained by pressing and holding the Alt key. Pressing and holding the Alt key places the keyboard in a special or alternate mode, where pressing a second key gives predefined functions.

alternate shift. During keyboard remap, an operation that defines a different set of characters or functions for the keyboard when the Alt key is pressed. For example, the Clear function is defined as the alternative shift state of the Backspace key. The Clear function is started by pressing and holding the Alt key while pressing the Backspace key at the same time.

American National Standard Code for Information Interchange (ASCII). The standard code with a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems and data communication systems. The ASCII set consists of control characters and graphic characters.

American National Standards Institute (ANSI). An organization sponsored by the Computer and Business Equipment Manufacturers Association for establishing voluntary industry standards.

anchor. A window procedure that handles Presentation Manager message conversions between an icon procedure and an application.

anchor block. In Presentation Manager internal resources, an area allocated to a process or thread that initializes the Presentation Manager environment.

ANSI. See *American National Standards Institute.*

APA. All points addressable

API. See *application programming interface.*

append. To attach a record or file to the end of another record or file. Contrast with *replace*.

appendage. An application program routine provided to assist in handling a specific occurrence of an event.

application. A program or set of programs that perform a task; for example, a payroll application. For OS/2 LAN Server, see *private application* and *public application*. In Query Manager, see *customization tasks.*

application plan. The control structure produced during the bind process and used by Database Services to process Structure Query Language (SQL) statements encountered during application processing. See *access plan*. Synonymous with *plan*.

application program. (1) A collection of software components, such as Communications Manager and Database Manager that a user installs to perform particular types of work, or applications, on a computer. (2) A program written for or by a user to perform work on a computer.

application programming interface (API). A formally-defined programming language interface that is between an IBM system control program or a licensed program and the user of a program.

archive. A copy of one or more files or a copy of a database that is saved for future reference or for recovery purposes in case the original data is damaged or lost.

argument. A parameter passeo between a calling program and a called program. See also *value.*

arithmetic expression. An expression that contains arithmetic operations and operands that can be reduced to a single numeric value.

arithmetic function. A function that represents one of the basic arithmetic operations such as addition, subtraction, division, or multiplication. It can also be a mathematical operation such as finding the average, minimum, or maximum value of a set of values.

arithmetic operator. A symbol used to represent a mathematical operation such as the plus sign (+), which is used to represent addition, or the minus sign (-), which is used to represent subtraction.

arithmetic overflow. A condition in which the result of a mathematical operation exceeds the storage or register capacity.

ASCII. See *American National Standard Code for Information Interchange.*

ASCII terminal emulation. A feature of Communications Manager that emulates the function of an asynchronous terminal.

ASCIIZ. A string of ASCII characters where the string is ended with a byte containing the value (0).

asynchronous. Pertaining to the lack of regular time relationship; unpredictable with respect to the execution of program instructions. Contrast with *synchronous.*

asynchronous data transfer. A physical transfer of data to or from a device that occurs without a regular or predictable time relationship following the execution of an input/output (I/O) request.

asynchronous transmission. A type of data communication in which communication of a character or a block of characters can begin at any time, but in which the bits that represent the character or block have equal time duration. Contrast with *synchronous transmission.*

attribute. (1) A characteristic or property; for example, the color of a line or the length of a data field. (2) In Database Manager, a characteristic of data contained in a column or field, such as length, data type, or data format.

authority. A permission, privilege, or group of privileges granted to an individual or group of individuals.

authorization. (1) The right granted to a user or group of users to communicate with or make use of a computer system, network, or database. (2) In Database Manager, a collection of privileges which defines the right to access objects. For example, two levels of authorizations in Database Manager are SYSADM and DBADM.

authorization ID. A sequence of characters that uniquely identifies a user or group of users. See also *qualifier.*

auto-answer. A feature that enables a machine to respond without user action to a call it receives.

B

background. (1) In multiprogramming, the conditions under which low-priority programs are executed. Contrast with *foreground.* (2) An active session that is not currently displayed on the screen.

back up. (1) To copy information onto a diskette or fixed disk for record keeping or recovery purposes. (2) An OS/2 command (BACKUP) that saves files. (3) In Database Manager, a utility that saves a database.

backtab. A typing action that moves the cursor to the previous field.

backward. A scrolling action that displays information in the panel preceding the current panel. Contrast with *forward*.

base operating system. The component of OS/2 EE that manages system resources, excluding Database Manager, Communications Manager, and LAN Requester.

BASIC. See *beginner's all-purpose symbolic instruction code*.

batch file. (1) A file containing DOS commands organized for sequential processing while in DOS mode. (2) Files that are identified with a .BAT extension. For OS/2 mode, see *command file*.

beginner's all-purpose symbolic instruction code (BASIC). A high-level programming language with a small number of statements and a simple syntax that is designed to be easily learned. BASIC is widely used for interactive applications on personal computers.

beginning of data. A cursor-movement function that moves the selection cursor to the leftmost position on the topmost line of information.

beginning of field. A typing action that moves the cursor to the first entry position in the field.

binary. Pertaining to the base two system of numbers. The binary digits are 0 and 1.

binary string. A sequence of consecutive binary digits.

binary synchronous communication (BSC). A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences for binary synchronous transmission of binary-coded data between stations. Contrast with *Synchronous Data Link Control (SDLC)*.

bit. A binary digit (either 0 or 1).

bits per second (bps). In serial transmission, the speed at which bits are transmitted.

bit map. A representation of an image by an array of bits. The image can be displayed on an APA device, usually the screen.

block. (1) A string of data elements recorded or transmitted as a unit. (2) To wait, usually for an input/output (I/O) event to complete or for a resource to become available.

block device. A storage device that performs I/O operations on blocks of data called *sectors*. Data on block devices can be randomly accessed. Block devices include fixed disks, diskettes, and virtual disks. Block devices are designated by a drive letter (for example, **C:**).

blocking mode. A condition set by an application that determines when its threads may block. For example, an application may set the PipeMode parameter for the DosCreateNPipe function so that its threads perform I/O operations to the named pipe block when no data is available.

block size. The number of records, words, or characters in a block.

bps. See *bits per second.*

breakpoint. (1) An instruction in a program for halting execution. Breakpoints are usually established at positions in a program where halts, caused by external intervention, are convenient for restarting. (2) A place in a program, specified by a command or a condition, where the system halts execution and gives control to the workstation user or to a specified program.

broken pipe. A pipe that was created by a process that has since ended.

bridge. In a local area network, a device that connects an IBM Token Ring and a PC Network together. See also *gateway*.

BSC. See *binary synchronous communication*.

buffer. A memory area reserved for use in performing input/output (I/O) operations.

buffer pool. A memory area reserved to satisfy the buffering requirements for a function.

built-in function. (1) A function that is supplied by a programming language. (2) In Database Manager, a scalar function or column function.

button. A mechanism on a pointing device, such as a mouse, used to request or initiate an action. See *select button*.

byte. The amount of storage required to represent 1 character.

byte pipe. Pipes that handle data as byte streams. All unnamed pipes are byte pipes. Named pipes may be byte pipes or message pipes. See *byte stream*.

byte stream. Data that consists of an unbroken stream of bytes.

C

cache. (1) A special-purpose buffer storage, smaller and faster than main storage, used to hold a copy of instructions and data obtained from main storage and likely to be needed next by the processor. (2) A buffer used to manage the movement of data for greater performance (for example, reduced access time).

call. (1) The action of bringing a computer program, routine, or subroutine into effect, usually by specifying the entry conditions and an entry point. (2) A transmission for the purpose of identifying the transmitting station for which the signal is intended. (3) An attempt to reach a user whether or not the attempt is successful.

calling sequence. (1) A sequence of instructions together with any associated data necessary to execute a call. (2) The order in which function parameters are passed between an application and the operating system. For MASM applications, this sequence consists of a set of instructions pushing parameter values on the stack and invoking the call. For C applications, this sequence consists of a single instruction invoking the call (where the parameters are defined within a set of parentheses).

cancel. (1) To end a task before it is completed. (2) An action that removes the current panel or window without processing it and returns to a previous one. See also *Escape key*, *exit*, and *return*.

Caps Lock. A typing action resulting from pressing the Caps Lock key that causes the character keys to produce uppercase letters. The keyboard remains in Caps Lock mode until the user presses the Caps Lock key again.

capture. To gather data and save it in a file.

capture to file. A feature of ASCII terminal emulation that allows the user to save data received from a remote location into a file.

carriage return. An operation that prepares for the next character or cursor to be printed or displayed at the specified first position on the next line.

carrier return. An indication to continue printing at the left margin of the next line.

case-sensitive. A condition in which entries for an entry field must conform to a specific lowercase, uppercase, or mixed-case format in order to be valid.

character. (1) A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. (2) A column data type in Query Manager.

character box. In computer graphics, the rectangle or (for sheared characters) the parallelogram boundaries that govern the size, orientation, spacing, and italicizing of individual symbols or characters to be shown on a display screen or printer page. See also *character cell*.

character cell. The rectangular space in which any single character is displayed on a screen or printer device. The position is addressed by row and column coordinators. See also *character box*.

character constraint. A limitation placed by an information processing system on character formats.

character data. A type of data in the form of letters, digits, and special characters such as punctuation marks.

character device. A device that performs I/O operations on one character at a time. Because character devices view data as a stream of bytes, character-device data cannot be randomly accessed. Character devices include the keyboard, mouse, and printer, and are referred to by name.

character display. A display that uses a character generator to display predefined character boxes of images (characters) on the screen.

character set. A group of characters used for a specific reason; for example, the set of characters a printer can print or a keyboard can support.

character string. (1) A sequence of bytes or characters associated with a single-byte character set. (2) A sequence of printable characters. (3) A string of characters, such as a command and its parameters, used to communicate with the base operating system.

character string delimiter. (1) In Database Manager, the characters used to enclose character strings in delimited ASCII (DEL) files that are imported or exported. (2) In Query Manager, the default is a double quotation mark. See *delimiter*.

check box. A user-interface component that is a square box with associated text. It acts like a switch that users can toggle on and off. An **x** appears in the check box to show a choice is selected. Many check boxes are combined in related sets so users can choose one or more choices or no choices. Contrast with *radio button*.

check mark. A symbol used on a graphics screen to indicate that users have selected an item.

child process. A process that is started by another (parent) process. Contrast with *parent process*.

child window. A window that is positioned relative to another window (either a parent window or another child window). Child windows are dependent on their parent windows. See *sibling window*. Contrast with *parent window*.

choice. An item a user can select.

circuit switching. A process that, on demand, connects data terminal equipment (DTE) through telephone switching equipment and permits the exclusive use of a data circuit between them until the connection is released. Synonymous with *line switching*.

circular log. A file used to log data to a disk or diskette. Each record is timestamped. When the end of the file is reached, the file is not extended in size, but the file is reused from the beginning. See *database log*.

C/2* Language. See *IBM C/2 Language*.

click. To press and release the select button on a pointing device.

client. (1) A functional unit that receives shared services from a server. (2) A user, as in a client process that uses a named pipe or queue that is created and owned by a server process.

clip. To truncate information by removing those parts of a displayed image that lie outside a given boundary.

clipboard. An area of computer memory that temporarily holds data. Data in the clipboard is available to other applications. Data is placed into the clipboard by a Cut or Copy operation. Data is removed from the clipboard by a Paste operation. Data in the clipboard is lost when the system is turned off or restarted. See *Copy*, *Cut*, and *Paste*.

client area. In Presentation Interface, the area in the center of a window that contains the main information of the window.

clipboard. In Presentation Interface, an area of memory that holds data being passed from one program to another.

clock tick. The minimum unit of time that the system tracks. If the system timer currently counts at a rate of X Hz, the system tracks the time every 1/X of a second. Also known as *timer tick*.

CLOCK$. Character-device name reserved for the system clock.

close. (1) To end an activity and remove it from the screen. (2) To end the processing of a file or device.

CMS. See *Conversational Monitor System.*

COBOL/2*. See *IBM COBOL/2.*

code. (1) A set of instructions for the computer. (2) To write instructions for the computer. (3) A representation of a condition, such as an error code.

code page. (1) A table that defines a coded character set by assignment of a character meaning for a language or country to each code point in the table. (2) A mapping between characters and their internal (binary) representation. See *primary code page.*

code segment. An executable section of a program module; that is, a section of a program module that contains program instructions.

COM. A representation of one of the asynchronous serial communications ports, (COM1, COM2, and COM3), supported by the OS/2 program.

command. (1) A request from a terminal for performance of an operation or execution of a program.

command entry field. A field into which a user types commands.

command file. (1) A file containing OS/2 commands organized for sequential processing while in OS/2 mode. For DOS mode, see *batch file.* (2) Files that have a .CMD file name extension.

command interface. The method used to type commands at the OS/2 or DOS command prompt.

command name. The verb in a command that specifies the action to be performed.

command processor. A program that performs an operation specified by a command.

command prompt. A displayed symbol, such as C:\, where a user enters commands.

comment. Optional text that describes an object or statement.

commit. A process that causes data changed by an application or user to become part of a database.

commit point. Synonym for *point of consistency.*

Common User Access (CUA). A Computer interface that gives a series of rules that describe the way in which information appears on the screen, and the interaction techinques between people and computers.

communication. The transmission and reception of data. Synonymous with *data communication.*

communication adapter. (1) A circuit card with associated software that enables a processor, controller, or other device to be connected to a network. (2) See *Realtime Interface Coprocessor.*

communication port. (1) An access point for data entry or exit to or from a communication device such as a terminal. (2) On a personal computer, a synchronous or asynchronous serial port to which a modem can be attached. Synonymous with *port.*

communications line. The physical link (such as wire or a telephone circuit) that connects one or more workstations to a communications control unit or that connects one control unit to another. See also *data link.*

Communications Manager. A component of the OS/2 program that lets a workstation connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host. Communications Manager provides application programming interfaces (APIs) so that users can develop their own applications.

Communications Manager system administrator. See *system administrator.*

comparison operator. (1) A symbol, such as =, >, or <, used to specify a relationship between two values. (2) Comparison operators in the Query Manager prompted interface are expressed as words such as equal to.

compile. (1) To translate a program written in a high-level programming language into a machine language program. (2) The computer actions required to transform a source file into an executable object file.

component. A functional part of &os2ee. (Communications Manager, base operating system, Database Manager, and LAN Requester).

COM1, COM2, COM3. Character-device names reserved for serial ports 1 through 3.

CON. Character-device name reserved for the console keyboard and screen.

concatenation. The method of combining two strings into a single string by appending the second to the first.

concurrency. The shared use of resources by multiple interactive users or application programs at the same time.

concurrency control. In Database Manager, a feature that allows multiple users to run database transactions simultaneously without interfering with each other.

concurrent. Pertaining to the occurrence of two or more activities within a given interval of time.

CONFIG.SYS. A file that contains configuration options for an OS/2 program installed on a workstation. See also *configuration file*.

configuration. (1) The task of defining the devices, features, parameters, and programs for a system. Synonymous with *customization*. (2) The arrangement and relationship of the components in a system or network.

configuration file. (1) In Communications Manager, a file that describes the devices, optional features, communications parameters, and programs installed on a workstation. (2) For the base operating system, the CONFIG.SYS file that describes the devices, system parameters, and resource options of a workstation. See also *CONFIG.SYS*.

configure. (1) To prepare a workstation component or program for operational use. (2) To describe to a system the devices, optional features, and programs installed on the system.

confirmation. (1) A prompt in a menu window wherein a program questions the user when the consequences of a user action are significant. (2) Acknowledgement by a program that data has been received or a request completed.

connection. An association established between functional units for conveying information.

connectivity. The physical link that provides the ability to attach to another system.

consistency of data. In Database Manager, a feature that 'ensures that no transaction can see uncommitted updates for another transaction. See also *repeatable read*.

constant. A fixed value.

context-sensitive help. Synonym for *contextual help*.

contextual help. Information about a field or other display element on which the cursor is positioned that is provided to the user upon request. Synonymous with *context-sensitive help*. See *extended help*.

Control (Ctrl) mode. A mode that is obtained by pressing and holding the Ctrl key. Pressing and holding the Ctrl key places the keyboard in a special control mode, where pressing a second key gives predefined functions.

Control Panel. A function of the OS/2 program that is used to set up user preferences, such as color on a screen.

control privilege. In Database Services, the authority to completely control a Database Services object. This includes the authority to access, drop, or alter an object as well as the authority to extend or revoke privileges on the object to other users.

conventional LU application (LUA). An implementation of the SNA protocol that allows a workstation to communicate with host applications using LU0, LU1, LU2, or LU3 protocols.

Conversational Monitor System (CMS). In Advanced Program-to-Program Communications (APPC), two or more programs communicating using the services of logical units (LUs).

Copy. (1) A Presentation Manager* function that copies a marked section of text to the clipboard. (2) A choice that places onto the clipboard, a copy of what the user has selected. See also *Cut* and *Paste*.

correlation. In Presentation Interface, an action determining which element or object within a picture is at a given position on the display.

country code. (1) A three digit number specifying a country and that country's preferred formats for date and time values, currency, and numeric data. (2) In Asynchronous Communications Device Interface (ACDI), a number specifying a country that is used to enforce specific country requirements for connecting to telephone networks.

CPI. (1) See *callable programming interface*. (2) characters per inch.

critical extended attribute. An extended attribute that is necessary to the correct operation of the system or a particular application.

critical message. The information provided by the program to the user that describes a required action the user must take.

critical section. (1) In programming languages, a part of an asynchronous procedure that cannot be executed simultaneously with a certain part of another asynchronous procedure. **Note:** *That part of the other asynchronous procedure also is a critical section.* (2) A section of code that is not reentrant; that is, can be executed by only one thread at a time.

CUA. See *Common User Access*.

cursor. A movable, visible mark used to indicate the position at which the next operation will occur on a display screen. See also *pointer*.

cursor stability. An isolation level that provides more concurrency than repeatable read. With cursor stability, a unit of work holds locks only on its uncommitted changes and the current row of each of its cursors.

customization. Synonym for *configuration*.

customization tasks. A feature supported by Query Manager that includes a control language, Query Manager command language, customized panels, customized menus, and customized procedures.

customized processing. A method used to run a procedure in which the procedure is specified as a parameter when Query Manager is started.

Cut. (1) A Presentation Manager function that moves a marked section of text to the clipboard. (2) A choice in the Edit pull-down that removes a selected portion from whatever has been marked and copies it to the clipboard, compressing the space it occupied. See also *Copy* and *Paste*.

cycle. In Database Manager, a set of tables that can be ordered so that each table is a descendent of the one before it, and the first is a descendent of the last. Thus, each table is a descendent of itself.

cylinder. (1) The fixed disk or diskette tracks that can be read or written without moving the disk drive or diskette drive read and write mechanism. (2) The number of tracks for space allocation.

D

data. The coded representation of information for use in a computer. Data has certain attributes such as type and length.

data circuit-terminating equipment (DCE). The equipment installed at the user's premises that provides all the functions required to establish, maintain, and end a telephone connection for data transmission, and which does the signal conversion and coding between the data terminal equipment (DTE) and the line. See also *modem*.

data communication. Synonym for *communication*.

data file. A collection of user data that is not a program.

data frame. Synonym for *frame*.

data link. A physical link, such as a wire or a telephone circuit, that connects one or more devices or communication controllers. See also *communications line.*

data segment. A nonexecutable section of a program module; that is, a section of a program that contains data definitions.

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. See also *file.*

data station. Synonym for *station.*

data structure. A syntactic structure of symbolic expressions and their storage allocation characteristics.

data type. A category that identifies the mathematical qualities and internal representation of data.

database. (1) A systemized collection of data that can be accessed and operated upon by an information processing system.

decimal point character. (1) In Database Services, the character used to represent the decimal point in numeric data in delimited ASCII (DEL) files that are imported or exported. The default is a period (.). (2) In Query Manager, the character is also used in reports, panels, and data edit.

dedicate. To assign a system resource, such as an input/output (I/O) device, a program, or a whole system, to one application or purpose.

DBCS. Double-byte character set.

deadlock. (1) Unresolved contention for use of a resource. (2) An error condition in which processing cannot continue because each of two elements of the process is waiting for an action by, or a response from, the other. (3) An impasse that occurs when multiple processes are waiting for the availability of a resource that will not become available because it is being held by another process that is in a similar wait state.

debug. To detect, diagnose, and eliminate errors in programs.

descendant. A process or session that is loaded and started by a parent process or parent session.

default. A value, attribute, or option that is assumed when no option is explicitly specified.

default definition. A definition used in panels in Query Manager. A default definition can be used to create a customized panel and is based on one table.

default form. The form used by Query Manager to format reports for queries when a user-defined form is not specified.

default printer. A printer that is assigned to a system or user and accepts all the printed output from that system or user, when no other printer is specified.

default value. A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

DEL. A format used to export and import data formatted with delimited ASCII.

delete. To remove; for example, to delete a file.

delimiter. (1) A character or flag that groups or separates items of data. (2) In the IBM Token-Ring Network, a bit pattern that defines the limits of a frame or token on a ring network.

deselect. An action that causes a choice that was previously selected to no longer be selected.

device. (1) An input/output (I/O) unit such as a terminal, display, or printer. (2) In OS/2 LAN Server, a drive (for files resources) or port (for printers and serial devices) that is assigned when a resource is used.

device driver. (1) The executable code needed to attach and use a device such as a display, printer, plotter, or communications adapter. (2) Programming code that defines the interface between the operating system or its applications and a physical device. Device drivers send data to and receive data from physical devices; for example, from the keyboard or mouse and to the printer. They resolve device-independent I/O requests from the operating system and its applications with the device-dependent physical attributes of the device.

DFT. See *Distributed Function Terminal.*

diagnostic tool. One of the OS/2 program utilities designed to gather and process data to help identify the cause of a problem.

dialog. (1) The interaction between a computer and its user. (2) The interaction of the user with a predefined set of panels or window displays requiring navigation control through a structured hierarchy.

direct authorization. An authorization granted directly and specifically to an individual user ID or group ID.

direct manipulation. The action of using the mouse to move graphical representations (OS/2 icons) around the screen.

directory. (1) A list of the files that are stored on a disk or diskette. A directory also contains information about the file such as size and date of last change. (2) A named grouping of files in a file system. See also *files resource.*

discrete profile. An access control profile that protects a single resource such as a printer or serial device queue. See *access control profile.*

disk. A magnetic disk unit. See also *diskette.*

disk operating system (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

diskette. A disk enclosed in a protective container that is removable from the hardware. See also *disk.*

display. A visual presentation of data.

Distributed Function Mode. See *Distributed Function Terminal.*

Distributed Function Terminal (DFT). (1) An operational mode that allows multiple concurrent logical terminal sessions. (2) A hardware or software protocol used for communication between a terminal and an IBM 3274 or IBM 3174 control unit.

DLL. Dynamic-link library.

DLR. See *dynamic link routine.*

domain. (1) A set of servers that allocates shared network resources within a single logical system. (2) For database tables, the attribute or all possible valid values associated with a column.

domain controller. A server within the domain that provides details of the OS/2 LAN Server to all other servers and requesters on the domain. The domain controller is responsible for coordinating and maintaining activities on the domain.

DOS. See *disk operating system.*

DOS mode. The mode that allows the base operating system to run programs written for DOS.

double-byte character set (DBCS). A set of characters in which each character occupies two bytes. Languages, such as Japanese, Chinese, and Korean, that contain more symbols than can be represented by 256 code points, require double-byte character sets. Entering, displaying, and printing DBCS characters require special hardware and software support.

double-click. To press and release the select button on the pointing device twice in rapid succession.

doubleword. A contiguous sequence of bits or characters that comprises two computer words and is capable of being addressed as a unit.

download. To move data or programs from a host computer to a workstation. Contrast with *upload.*

dragging. Moving an object on the display screen as if it were attached to the pointer, performed by holding the select button and moving the pointer. See also *drop.*

drive. The device used to read and write data on disks or diskettes.

drop. (1) To fix the position of the object that is being dragged by releasing the select button on the pointing device or mouse. See also *drag-*

ging. (2) A cable that leads from a faceplate to the distribution panel in a wiring closet.

dual-boot function. A feature of OS/2 that allows the user to start DOS from within OS/2 or OS/2 from within DOS.

dump. To copy data from memory onto an external medium such as a diskette or printer.

dump diskette. (1) A diskette that contains a dump (data copied from memory) or that is prepared to receive data from memory. (2) In the OS/2 program, a diskette (created using the CREATEDD command) that contains the contents of storage at a specified point in time.

duplex. Pertaining to communication in which data can be sent and received at the same time. Synonymous with

duplicates. In Database Manager, (1) an option used in prompted query and prompted view definition to specify whether duplicate data rows in a query or view should be returned. (2) An option used when creating an index for a table to specify whether or not duplicate values are allowed for the set of columns that comprise the index.

duration. An interval of time.

dynamic link library. A module containing a dynamic link routine (DLR) that is linked at load or run time.

dynamic link routine (DLR). A program or routine that can be loaded by an application or as part of a program.

dynamic linking. In the OS/2 program, the delayed linking of a program to a routine so that the routine is not linked until load or run time.

dynamic priority. In the OS/2 program, pertaining to a priority of a process that is varied by the operating system. Contrast with *absolute priority*.

dynamic storage. (1) A device that stores data in a manner that permits the data to move or vary with time such that the specified data is not always available for recovery. (2) A storage in which the cells require repetitive application of control signals in order to retain stored data. Such repetitive application of the control signals is normally called a refresh operation. A dynamic storage may use static addressing or sensing circuits. (3) See also *static storage*.

dynamic-link library (DLL). An executable file containing subroutines that can be loaded into memory when needed. A DLL allows many applications to share routines. Sharing routines permits each application's executable file to be smaller, and allows applications to be updated without relinking subroutines.

dynamic-link module. A module that is linked at load time or run time. *full duplex*.

dynamic linking. The process of resolving external references in a program module at load time or run time rather than during linking.

E

EBCDIC. See *extended binary-coded decimal interchange code*.

edit. To add, change, delete, or rearrange data.

ellipsis. A symbol (...) used on a panel indicating that an additional menu panel follows.

emphasis. Highlighting, color changes, or other visual indicators that serve as visual cues to users. For example, a choice may be highlighted to show users a choice has been selected.

emulation. The imitation of all or part of one system by another so that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated computer system.

enable. (1) To make functional. (2) The state of a processing unit that allows the occurrence of certain types of interruptions. (3) In the OS/2 program, to initiate the operation of a circuit or device.

end rule. Used to designate what should happen when the end of the current panel set is reached in Change mode panel execution (as a result of Next, Change and Next, or Delete and Next), or when no rows are found by a search operation. Valid options are:

- Search - go to the nulled Search Mode Panel
- Extended Search - display the Search Conditions panel to allow the user to specify a query to determine the new panel set
- Exit the panel - terminate panel execution for the current panel.

engine. Synonym for *Database Services.*

enter. An action performed by pressing the Enter key or selecting the Enter push button. This action causes the computer to receive and process user input.

entry field. A panel element, usually highlighted in some manner and usually with its boundaries indicated, where users type information.

environment. In Database Manager, a collection of names of logical and physical resources used to support the performance of a function.

erase to end of field. A typing action that deletes the character at the cursor position and all characters in the field to the right of the cursor position.

environment segment. The list of environment variables and their values for a process.

environment strings. ASCII text strings that define the value of environment variables.

environment variables. Variables that describe the execution environment of a process. These variables are named by the operating system or by the application. Environment variables named by the operating system are PATH, DPATH, INCLUDE, INIT, LIB, PROMPT, and TEMP. The values of environment variables are defined by the user in the CONFIG.SYS file, or by using the SET command at the OS/2 command prompt.

error. An unexpected result from a program command or action.

error log. A file that stores error information for later access. See *log.*

Escape key. A key that removes the current panel, menu, or window without processing it and returns to a previous one. See also *cancel* and *return.*

even parity. A data transmission attribute in which the parity bit of a character frame is set so that the sum of the binary digits in the character and the parity bit is even.

event semaphore. A semaphore that enables a thread to signal a waiting thread or threads that an event has occurred or that a task has been completed. The waiting threads can then perform an action that is dependent on the completion of the signaled event.

exception. An abnormal condition, such as an I/O error encountered in processing a data set or a file.

exclusive system semaphore. A system semaphore that can be modified only by threads within the same process.

executable file. A collection of related data records that execute programs.

exit. An action that ends the current function and returns to a higher-level function. See also *Escape key, cancel,* and *return.*

expression. An operand or a combination of operands and operators yielding a single value.

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 8-bit coded characters used by host computers.

extended help. In Presentation Manager, an action available from Help panels that provides users with current information about the application panel rather than a particular item on the panel. See *contextual help.*

extended partition. The area beyond the primary partition. This area can be divided into smaller areas or partitions, each of which can be assigned a drive letter and be accessed as though it were a separate fixed disk.

extent. Continuous space on a disk or diskette that is occupied by or reserved for a particular data set, data space, or file.

F

fast path. A way to navigate through panels and menus using short cuts. Fast paths can include selections using numbers, mnemonics, and function keys. See also *mnemonic* and *function key*.

FAT. See *file allocation table*.

FCB. See *File Control Block*.

feature. A programming or hardware option.

field. (1) An area in a record or on a panel used to contain data. (2) In the IBM 3270 data stream, a group of consecutive positions on a presentation space with similar characteristics. These characteristics are defined by the field attribute byte at the beginning of the field.

field delimiters. Symbols, usually brackets, surrounding or within an entry field in a panel that indicate the limits of an entry field.

file. (1) To store current or stored objects. (2) A collection of information treated as a unit. See also *data set*.

file allocation table (FAT). In IBM personal computers, a table used to allocate space on a disk for a file. This can then be used to locate and chain together parts of the file that may be scattered on different sectors so that the file can be used in a random or sequential manner.

file attribute. Any of the attributes that describe the characteristics of a file.

file specification. The full identifier for a file. Includes its drive designation, path, file name, and extension.

File Control Block (FCB). A record that contains all of the information about a file, for example, its structure, length, and name.

file description. A part of a file where file and field attributes are described.

file name. (1) The name used by a program to identify a file. (2) The portion of the identifying name that precedes the extension.

file name extension. An optional three-letter code that may be used as the second part of an OS/2 file name and that is separated from the file name by a period (.). Extensions are used to more specifically identify the name of the file.

file specification. The full identifier for a file, which includes its drive, path, file name, and file name extension.

file system. The collection of files and file management structures on a physical or logical mass storage device such as a disk.

files resource. A directory or subdirectory on a server that contains programs or data files that can be made available to users. For the IBM PC LAN see *fileset*. See also *directory*.

filemode. The third field in the VM/CMS file identifier. The filemode indicates the mode letter currently assigned to the virtual disk in which the file resides.

fileset. An IBM PC LAN program term. See *files resource*.

filespec. The name and location of a file. The format is dependent on the storage medium of a file; for example, c:\path\filename.ext.

filetype. The second field in the VM/CMS file identification.

fixed disk. A flat, circular, nonremovable plate with a surface layer on which data can be stored by magnetic recording.

fixed-length. Pertaining to a characteristic of a file in which all of the records are the same length.

fixed-length string. A character or graphic string whose length is specified and cannot be changed.

flag. (1) An indicator or parameter that shows the setting of a switch. (2) A character that signals the occurrence of some condition, such as the end of a word.

flush. To delete, erase, or remove.

font. A particular style of typeface that contains definitions of character sets, marker sets, and pattern sets.

foreground. In multiprogramming, the environment in which high-priority programs are executed.

format. (1) A specific arrangement of a set of data. (2) In the base operating system (both DOS and OS/2 modes), a command used to prepare a diskette.

formatted diskette. A diskette on which track and sector control information has been written. Contrast with *unformatted diskette.*

FORTRAN/2*. See *IBM FORTRAN/2.*

forward. A scrolling action that displays information below the currently visible information. Contrast with *backward.*

full-screen mode. A form of screen presentation in which the contents of an entire screen can be displayed at once.

full-screen application. An application program that occupies the whole screen.

function. (1) In a programming language, a block, with or without formal parameters, whose execution is invoked by means of a call. (2) A set of related control statements that cause one or more programs to be performed.

function. A specific purpose of an entity or its characteristic action; for example, a column function, or scalar function. See *column function* and *built-in function.*

function key. A key that causes a specified sequence of operations to be performed when it is pressed. See also *fast path.*

function key area. The area at the bottom of a panel that identifies function key assignments that are available on that panel.

G

gateway. In communications, a functional unit that connects two computer networks of different network architectures. See also *bridge.*

GDDM. See *graphical data display manager.*

general protection fault. An exception condition that occurs when a process attempts to use storage or a module that has some level of protection assigned to it, such as I/O privilege level. See also *IOPL code segment.*

global dynamic-link module. A dynamic-link module that can be shared by all processes in the system that refer to the module name.

global file-name character. Either a question mark (?) or an asterisk used as a variable in a file name or file-name extension when referring to a particular file or group of files. For example, *.EXE* can be used to refer to a set of files with the extension, *EXE.*

generic profile. An access control profile that allows protection of a directory and its subdirectories and files. Only file resources can be protected by generic profiles. See *access control profile.*

grant. Gives authority to a user ID or group ID.

graphical data display manager (GDDM). A function of the host operating system that processes both text and graphics for output on a display, printer, or plotter.

graphics. A picture defined in terms of graphics primitives and graphics attributes.

graphics attributes. The attributes that apply to graphics primitives. Examples are color selection, line type, and shading pattern definition. See *segment attributes.*

graying. A form of emphasis that tells users which items are currently unavailable for selection because selection would violate some condition. Graying is accomplished by reducing the contrast between the choice and its background.

group. (1) In Presentation Interface, a collection of logically connected controls; for example, the buttons controlling paper size for a printer. See *program group*. (2) A logical organization of users that have IDs according to activity or resource access authority.

group access list. A list of groups and the associated access authorities for each group in the list.

group heading. The words identifying a group of related selection fields or entry fields.

group ID. An ID that relates to rights given to a group profile to access objects, resources, or functions.

H

handle. A unique identifier assigned to a system resource, such as a window or device.

hang. Synonym for *deadlock*.

hard error. An error condition on a network that requires that the network be reconfigured or that the source of the error be removed before the network can resume reliable operation.

header. (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message, such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

heap. A memory object containing memory blocks that are dynamically allocated and deallocated by the memory-management functions.

Help. A feature that provides assistance and information to the user.

help on help. General information about the types of help information available to assist users of an application. Help on help is accessed by selecting the function key or by selecting Help with the mouse.

help panel. The information displayed on a screen in response to a help request from a user.

hertz (Hz). A unit of frequency equal to 1 cycle per second. Note: In the United States, line frequency is 60 Hz or a change in voltage polarity 120 times per second; in Europe, line frequency is 50 Hz or a change in voltage polarity 100 times per second.

hex. See *hexadecimal*.

hexadecimal. Pertaining to a numbering system with a base of 16; valid numbers use the digits *0* through *9* and characters *A* through *F*, where *A* represents *10* and *F* represents *15*.

history file. A file providing a record of system installation.

host computer. (1) In a computer network, a computer providing services such as computation, database access and network control functions. (2) The primary or controlling computer in a multiple computer installation.

host processor. A processor that controls all or part of a network.

host support. The facilities a host processor makes available to attached terminals, processors, and other devices, such as problem determination aids and database facilities.

host system. The controlling or highest level system in a data communication configuration.

hot key. A key or combination of keys used to switch between sessions.

hot spot. The part of the pointer, usually the tip, that must touch an object before it can be selected. See also *action point*.

Hz. See *hertz*.

I

I/O operation. An input operation to, or output operation from a device attached to a computer

IBM C/2 Language. A language used to develop software applications in compact, efficient code that can be run on different types of computers with minimal change and was designed for use on an &ipc.* workstation.

IBM COBOL/2. Common business-oriented language. A high-level programming language, based on English, that is used primarily for business applications and was designed for use on an IBM Personal System/2* workstation.

IBM FORTRAN/2. A programming language primarily used to express computer programs by arithmetic formula and designed for use on an &ipc. workstation. Note: FORTRAN means (FORmula TRANslation).

IBM Macro Assembler/2* (MASM). This language translates assembler language source code into an object module containing machine language understood by the computer.

IBM Operating System/2 Extended Edition Version 1.3. A program that contains the features of OS/2 Standard Edition Version 1.3. In addition, this program contains an advanced relational Database Manager component, a Communications Manager, component and a LAN Requester component that provide intersystems communications, improved connectivity, terminal emulation, and access to shared network resources.

IBM Operating System/2 LAN Server. A program that allows resources to be shared with other computers on the network. See *server*.

IBM Pascal/2. A high-level general-purpose programming language, related to ALGOL. Programs written in IBM Pascal/2 are block structured, consisting of independent routines. They can run on different computers with little or no modification and were primarily designed for use on an &ipc. workstation.

IBM PC Network. IBM PC Network is a low-cost broadband network that allows attached IBM personal computers, such as IBM Personal System/2 workstations, IBM 5150 Personal Computers, IBM Personal Computer ATs*, IBM PC XTs*, and IBM Portable Personal Computers to communicate and to share resources.

IBM Support Center. The support center available to registered service coordinators to help resolve program defects on supported software. Synonymous with *Support Center*.

IBM Token-Ring Network. IBM Token-Ring Network is a high speed, star-wired local area network to which a variety of IBM products can be connected.

icon. A pictorial representation of a choice for the user to select. Icons can represent things (such as a document or file) the user wants to work on. Icons can also represent actions the user wants to perform.

icon area. In Presentation Manager, the area at the bottom of the screen that is used to display the icons for minimized windows.

identifier. A token used to form a name of a Database Manager object or column name.

idles. In the IBM Token-Ring Network, the signals sent when neither frames nor tokens are being transmitted.

IEEE 802.2 interface. An interface adhering to the 802.2 logical link control (LLC) Standard of the Institute of Electrical and Electronics Engineers (IEEE). This standard is one of several standards for local area networks approved by the IEEE.

inactive program. A program that has been started but is suspended so that it is no longer running. Contrast with *active program*.

increment. To increase by a constant value.

index help. General information listed by topic that is available to users who want more information for the current application. Index help is accessed by selecting the function key while within a help panel.

index key. In Database Manager, a column or an ordered collection of columns on which an index is defined. See *key*.

information processing system. A system including computer systems and associated personnel that in conjunction with data processing operations performs operations on information; for example, automated office operations.

initial field value. The information in an entry field provided by an application when the field is first presented.

initial procedure. A procedure that will run automatically before displaying each panel instance and that can access or modify variables for the current panel instance. One procedure can be specified for Add mode and one can be specified for Change mode.

initial program load (IPL). (1) The initialization procedure that starts an operating system. (2) The process of loading programs and preparing a system to run jobs.

initialize. (1) To set counters, switches, addresses, or contents of storage to 0 or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine. (2) To prepare an adapter for use by a program.

input. The information entered into a computer for processing or storage.

input device. A device such as a keyboard in a data processing system through which data may be entered into the system.

input field. A field that accepts a value.

input focus. The area of a window where user action is possible from either the keyboard or the mouse.

insert (Ins) mode. An entry mode, obtained by pressing the Insert (Ins) key, that is used for adding information between characters without deleting existing characters.

installation. The process of placing one or more OS/2 components on the fixed disk of a workstation.

instance. A single occurrence of an object class that has a particular behavior.

integer. A whole number; for example, one of the numbers 0, +1, -1.

integrity. The protection of systems, programs, and data from inadvertent or malicious destruction or alteration.

interactive. Pertaining to the exchange of information between people and computers.

interactive graphics. The graphic objects that can be moved or manipulated by a user at a workstation.

interactive processing. (1) A processing method in which each user action causes a response from a program or the system. (2) In Database Manager, a method of processing that allows users to interact with the Query Manager panels and menus while a procedure is running.

interactive program. A program that is running (active) and ready to receive (or is currently receiving) input from the user. See *active program*. Contrast with *non-interactive program.*

interactive session. The group of processes or tasks that currently own the keyboard, mouse, and display, or other interactive input devices.

interface. (1) A set of verbs used by a program to communicate with another program. (2) A shared boundary between two or more entities. An interface might be a hardware component to link two devices together or it might be a portion of memory or registers accessed by two or more computer programs. See *user interface.*

interrupt. A suspension of a process, such as the execution of a computer program caused by an event external to that process, performed in such a way that the process can be resumed.

interval timer. (1) A timer that provides program interruptions on a program-controlled basis. (2) An electronic counter that counts intervals of time under program control.

IPL. See *initial program load.*

isolation level. The degree to which a unit of work is isolated from the updating operations of other units of work. See also *cursor stability*, *repeatable read*, and *uncommitted read.*

IOCTL. A device-specific command that requests a function of a device driver through the DosDevIOCtl function.

IOPL. Input/output privilege level.

IOPL code segment. An IOPL executable section of programming code that enables an application to directly manipulate hardware interrupts and ports without replacing the device driver.

J

journal. A file used to record changes made in a system. See also *log*.

K

Kanji. A character set consisting of symbols used in Japanese ideographic alphabets.

KB.. See *kilobyte*.

KBD$. Character-device name reserved for the keyboard.

kernel. The part of an operating system that performs basic functions, such as allocating hardware resources.

key. (1) One or more characters or record fields used to identify the record and establish the order of the record within an indexed file.

keyboard. A systematic arrangement of keys by which a machine is operated or by which data is entered.

keyboard mapping. A table or profile containing the definitions assigned to keys on a keyboard for use in terminal emulation.

keyboard style. The manner in which the keyboard key assignments are changed to match a particular keyboard type or user-defined arrangement.

keys help. An action in help panels that gives users a listing of the key assignments for the current application.

keyword. One of the predefined words of a computer, command language, or an application.

kilobyte (KB). A term meaning 1024 bytes.

L

label. (1) The name in the disk or diskette directory that identifies a file. (2) In programming, the field of an instruction that assigns a symbolic name to the location at which the instruction begins.

LAN. See *local area network*.

LAN adapter. A card that is installed in a personal computer and that is used to attach a personal computer to a local area network.

LAN Requester. A component of the OS/2 program that allows users to access shared network resources made available by OS/2 LAN Servers. See *requester*.

LAN Server. See *IBM Operating System/2 LAN Server*.

leading zero. A 0 that is displayed beside the leftmost digit in a number.

leased line. See *non-switched line*.

line feed. A control character that causes an output device to move forward one line.

line speed. The rate at which data is transmitted from one point to another over a telecommunications line, expressed in bits per second (bps).

line switching. Synonym for *circuit switching*.

line type. A type of telecommunications link, either switched or non-switched.

link. (1) The physical medium of transmission, the protocol, and associated devices and programming used to communicate between computers. (2) To interconnect items of data or portions of one or more computer programs; for example, the linking of object programs by a linkage editor, or the linking of data items by pointers.

link-edit. To create a loadable computer program by means of a linkage editor.

list. An action requested while in an entry field to search for valid choices for that entry field.

The result is a panel from which the user can select a choice to be used in the entry field. See also *prompt.*

load. (1) To move data or programs into memory. (2) To place a diskette into a diskette drive.

load module. A computer program in a form suitable for loading into memory for execution.

load time. (1) The time during which a program is being loaded into memory for execution. (2) In programming, the time it takes to enter data into memory or working registers.

load-on-call. A function of a linkage editor that allows selected segments of the module to be disk resident while other segments are executing. Disk-resident segments are loaded for execution and given control when any entry point they contain is called.

local. Residing on the user's node or workstation.

local administrator. The second level of user type authority in User Profile Management. It assigns the user SYSADM (system administrator) authority for all Database Manager objects on the local workstation. A local administrator does not have the same User Profile Management authority as an administrator user type.

local area network (LAN). A data network located on the user's premises in which serial transmission or a ring is used for direct data communication among data stations.

Notes:

1. *Communication within a local area network is not subject to external regulation; however, communication across the LAN boundary may be subject to some form of regulation.*
2. *LAN does not use store-and-forward techniques.*

local descriptor table. A table that defines code and data segments specific to a single task.

lock. (1) To keep from being modified. (2) A mechanism that prevents multiple users from updating and checking out an object simultaneously.

logical storage device. A device that the user can map to a physical (actual) device.

local variable. (1) A parameter that is defined and used only within a specified portion of the program in which it is declared.

local workstation. A workstation that is connected directly to a system without the need of data transmission facilities. Contrast with *remote workstation.*

log. (1) To record; for example, to log all messages on the system printer. Synonymous with *journal*. See *message log* and *error log.*

logging on. See *logon.*

logical record. A set of related data or words considered to be a record from a logical viewpoint.

logical terminal. (1) A destination with a name that is related to one or more physical terminals. (2) The definition of a specific 3270 terminal emulation session.

logical unit (LU). In Systems Network Architecture (SNA), a port through which an end user accesses the SNA network in order to communicate with another end user and through which end users access the functions provided by system services control points (SSCPs).

logo. (1) A letter, combination of letters, or symbol used to represent an entire word or words. (2) The display that identifies the IBM Corporation and the program name on licensed programs.

logoff. The procedure by which a user ends a terminal session.

logon. The process that allows a user to access the system after determining user ID. This user ID is then used to determine privileges and authorization to access protected resources, such as a database or other resources on a network.

long file names. File names that do not conform to the 8.3 file-name format. They can consist of up to 255 characters.

long string. A fixed or variable-length string where the maximum length is greater than 254 bytes.

lookup table. In Query Manager panel definition, a table from which columns can be presented in the panel as output fields only.

LPT1, LPT2, LPT3. Character-device names reserved for parallel printers 1 through 3.

LU. See *logical unit*.

M

Main Frame Interactive (MFI) presentation space. The 3270 terminal emulation presentation interface.

main window. The window that is positioned relative to the desk top window.

manual connection. A communications connection made by manually dialing or answering a phone.

map. (1) A set of values having defined correspondence with the quantities or values of another set. (2) To establish a set of values having a defined correspondence with the quantities or values of another set.

Mark. A clipboard function that identifies a section of text to be copied or cut.

MASM. See *IBM Macro Assembler/2*.

MAU. See *multistation access unit*.

maximize. A window-sizing action that makes the window the largest size available.

memory. (1) The storage on electronic chips; for example, random access memory, read-only memory, or registers. (2) Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing. See also *storage*.

memory allocation. An operating system function that assigns memory areas to tasks. Synonymous with *storage allocation.*

memory management. A feature of the operating system for allocating, sharing, and freeing main storage.

menu. (1) A displayed list of available machine functions for selection by a user. (2) A displayed list of items from which a user can make a selection. (3) In Database Manager, a customized selection screen created with Query Manager.

merge. (1) To combine the items of two or more sets that are each in the same given order into one set in that order. (2) To combine overrides for a file from the first call level up to and including a greater call level, producing the override to be applied when the file is used. (3) To combine or unite two sets of data.

message. (1) The information not requested by users but presented to users by the computer in response to a user action or internal process. (2) In Presentation Manager, a packet of data used for communication between Presentation Interface and windowed applications.

message log. A file used to save or log certain types of messages and status information. See *log*.

message queue. A list of messages awaiting processing by the application.

message stream mode. A method of operation in which data is treated as a stream of messages. Contrast with *byte stream*.

metacharacter. See *global file-name character*.

metafile. In Presentation Interface, the generic name for the definition of the contents of a picture. Metafiles are used to allow pictures to be used by other applications.

MFI presentation space. See *Main Frame Interactive presentation space.*

migrate. (1) To move data from one hierarchy of storage to another. (2) To move to a changed environment, usually to a new release or version of a system.

migration. The process of converting an earlier released Database Manager database to a current Database Manager database. Migration allows you to acquire the capabilities of the current or new database without losing the data you created on the earlier released database.

minimize. A window-sizing action that makes the window the smallest size allowed. Minimized windows are represented by icons.

mixed-character string. A string containing a mixture of single-byte characters and double-byte characters.

mnemonic. A single character, within the text of a choice, identified by an underscore beneath the character. When users type a mnemonic of a choice, that choice is selected. See also *fast path.*

mode. A method of operation; for example, see *NumLock mode*. Synonymous with *transaction service mode.*

mode key. A key, such as the CapsLock, NumLock, or Alt key, that sets the keyboard in a special mode of operation different from its base state.

model profile. In Communications Manager, a supplied configuration profile with preconfigured options intended for use in the creation of a new profile.

modem. A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and that converts the received analog signal to digital data for the computer. Contrast with *digital data service adapter*. Synonymous with *modulator/demodulator*. See also *data circuit-terminating equipment.*

module definition file. A file that describes the code segments within a load module. For example, it indicates whether a code segment is loadable before module execution begins (PRELOAD), or loadable only when referred to at run time (load-on-call).

modulator/demodulator. Synonym for *modem.*

mouse. A device used to move a pointer on the screen.

move window. An action that allows the user to reposition a window.

multiple-choice selection field. A type of selection field that allows users to select any number of items. Contrast with *single-choice selection field.*

multiple-line entry field. A control window into which a user types more than one line of information. It also can be used to display text that cannot be edited. See also *entry field.*

multiprogramming. A mode of operation that provides for interleaved execution of two or more computer programs by a single processor.

multistation access unit (MAU). In the IBM Token-Ring Network, a wiring concentrator that can connect up to eight lobes to a ring network.

multitasking. A mode of operation that provides for concurrent performance or interleaved execution of two or more tasks.

mutex semaphore. (Mutual exclusion semaphore). A semaphore that enables threads to serialize their access to resources. Only the thread that currently owns the mutex semaphore can gain access to the resource, thus preventing one thread from interrupting operations being performed by another.

muxwait semaphore. (Multiple wait semaphore). A semaphore that enables a thread to wait either for multiple event semaphores to be posted or for multiple mutex semaphores to be released. Alternatively, a muxwait semaphore can be set to enable a thread to wait for any ONE of the event or mutex semaphores in the muxwait semaphore's list to be posted or released.

N

named pipe. A pipe that allows communications between unrelated processes running on the same or different computers. Contrast with *unnamed pipe.*

national language code page. A code page that contains the alphanumeric and special symbols used in a particular language.

NAU. See *network user address.*

navigation. The process of moving through panels and menus.

nest. (1) To incorporate a structure or structures into a structure of the same kind. For example, to nest one loop (the nested loop) within another loop (the nesting loop) or to nest one subroutine (the nested subroutine) within another subroutine (nesting subroutine).

NETBIOS. An application programming interface (API) between a local area network adapter and programs.

network. A configuration of data processing devices and software connected for information interchange.

network address. An address consisting of subarea and element fields that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached pairs local addresses with network addresses and vice versa.

network addressable unit (NAU). In Systems Network Architecture (SNA), a logical unit, physical unit, or system services control point (SSCP). An NAU is the origin or the destination of information transmitted through the path control network. See also *network name*.

network administrator. The person responsible for the installation, management, control, and configuration of a network. The network administrator defines the resources to be shared and user access to the shared resources, and determines the type of access those users can have.

non-8.3 file-name format.. A file-naming convention in which path names can consist of up to 255 characters. See also *8.3 file-name format.*

noncritical extended attribute. An extended attribute that is not necessary to the function of an application.

nondestructive read. A read process that does not erase the data in the source location.

no parity. A data transmission attribute in which the sender and receiver agree that no parity bit is transmitted in the character frame.

non-interactive program. An active program that is running but cannot receive input from the keyboard or other interactive input device. Contrast with *interactive program*.

normal stop. In Communications Manager, to stop communications after current functions (for example, file transfer) have completed. Contrast with *abnormal stop.*

normal termination. The state of a program that has ended successfully. Contrast with *abnormal termination*.

notification message. (1) A message that describes a condition for which a program requires a reply from its caller, or else a default reply will be sent to the program. (2) A type of message that describes the status of the user's work or some state of the system.

NUL. Character-device name reserved for a nonexistent (dummy) device.

null. A special value that indicates the absence of information.

nullable. The condition where a value for a field can have an absence of a value. For example, a field for a person's middle initial does not require a value.

null suppression. The bypassing of all null characters in order to reduce the amount of data to be transmitted.

number selection. A fast path selection method for a menu or pull-down where users type the number that precedes the choice.

numeric constant. A fixed value that is a number.

O

object window. A window that does not have a parent window but may have child windows. An object window cannot be presented on a device.

odd parity. A data transmission attribute in which the parity bit of a character frame is set so that the sum of the digits in the character with the parity bit is odd.

office. Synonym for *work area.*

OIA. See *operator information area.*

online. (1) Pertaining to a user's ability to interact with a computer. A description of a user's access to a computer by way of a screen. (2) Pertaining to the operation of a functional unit that is under the continuous control of a computer.

online information. The information stored in a computer system that can be displayed, used, and modified in an interactive manner without any need to obtain printed copy.

open. The function that connects a file, adapter, system resource, or database object to a program for processing.

open file. The status of a file when its contents are being accessed.

operand. (1) An entity on which an operation is performed. (2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

operating system. The software that controls the running of programs. An operating system may provide services such as resource allocation, scheduling, input/output (I/O) control, and data management.

Operating System/2 Extended Edition (OS/2) Version 1.3. See *IBM Operating System/2 Extended Edition Version 1.3.*

Operating System/2 LAN Server. See *IBM Operating System/2 LAN Server.*

operation. (1) A well-defined action that, when applied to any permissible combination of known entities, produces a new entity; for example, the process of addition. (2) A program step performed by a computer.

operational diskette. Synonym for *working diskette.*

operator. A symbol that represents an operation to be performed; for example, the plus sign (+).

operator information area (OIA). In 3270 terminal emulation, the bottom line of the screen where status about the communication session is displayed.

operator information line. In ASCII terminal emulation, the bottom line of the screen used to display messages and status information.

optimization. The determination of an efficient access strategy for satisfying a database access.

option. (1) A specification in a statement or command that can be used to influence the execution of the statement. (2) A selectable item on an action bar.

order. To place items in an arrangement in accordance with specified rules.

OS/2. See *IBM Operating System/2 Extended Edition.*

OS/2 Extended Edition. See *IBM Operating System/2 Extended Edition.*

OS/2 file system. A file system provided by the OS/2 program and accessible through the OS/2 file input/output (I/O) functions. A file system is restricted to a single medium, either an entire diskette or a single partition of one fixed disk.

OS/2 LAN Server. See *IBM Operating System/2 LAN Server.*

OS/2 program. See *IBM Operating System/2 Extended Edition.*

output area. The area of paper within which the picture is to be printed or plotted.

output device. A device such as a printer or screen in a data processing system.

owning process. The process that owns the resources that may be shared with other processes.

P

page. (1) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between memory and disk storage.

page footing. Words located at the bottom of a text area of each page of a report.

page heading. Words located at the top of a text area of each page of a report.

paint. To assign colors or other forms of emphasis to display elements.

panel. (1) A particular arrangement of information grouped together for presentation to the user.

panel body area. The part of a panel not occupied by the action bar or the function key area.

panel body area separator. A line or color boundary that provides users with a visual distinction between two adjacent areas of a panel. Synonymous with *separator*.

panel title. A panel element that identifies the panel to the user.

parallel port. On a personal computer, a port used to attach devices such as dot matrix printers and input/output (I/O) units; it transmits data 1 byte at a time. Contrast with *serial port*.

parameter. (1) The information supplied by a program or user to a command or function. (2) The data passed between programs or procedures.

parent directory. The directory one level above the current directory.

parent process. A process that loads and starts other processes. Contrast with *child process*.

parent row. A row of a parent table that has at least one dependent row.

parent table. The table in a relationship containing the primary key that defines the relationship with a dependent table. A table can be a parent in an arbitrary number of relationships.

parent window. A window that has one or more child windows positioned relative to it. Contrast with *child window*.

parity. The determination whether the number of ones (or zeros) in an array of binary digits is odd or even.

parity bit. A binary digit appended to a group of binary digits to make the sum of all the digits, including the appended binary digit, either odd or even as pre-established.

partition. (1) A fixed-size division of storage. (2) On an IBM personal computer fixed disk, one of four possible storage areas of variable size; one may be accessed by DOS, and each of the others may be assigned to another operating system.

Pascal/2. See *IBM Pascal/2*.

password. A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data.

Paste. A choice in the Edit pull-down that a user selects to move the contents of the clipboard into a preselected location. See also *Copy* and *Cut*.

path. (1) The route used to locate files on a disk or diskette. The route consists of a collection of drives and directories. (2) In the IBM Token-Ring Network, a route between any two nodes. (3) In Systems Network Architecture (SNA), the set of data links, data link control layers, and path control layers that a path information unit travels through when sent from transmission control of one half-session to transmission control of another half-session. Synonymous with *absolute path name*.

PC Network*. See *IBM PC Network*.

placeholder. A symbol that can assume any of a given set of values.

plan. Synonym for *application plan*.

plotter. An output device that uses pens to draw on paper or transparencies.

pipe. A named or unnamed buffer used to pass data between processes. A process writes to or reads from a pipe as if the pipe were a standard-input or standard-output file. See also *named pipe* and *unnamed pipe*.

point-to-point. Pertaining to data transmission between two locations without use of any intermediate terminal or computer.

pointer. The symbol displayed on the screen, that is moved by a pointing device such as a mouse.

POINTER$. Character-device name reserved for a pointer device (mouse screen support).

pointing device. An instrument such as a mouse, trackball, or joystick used to move a pointer on the screen.

pop-up. A bordered area of the screen containing information that supplements the interaction between a user and a computer.

port. Synonym for *communication port*.

preempt. To take control away from; for example, to interrupt the execution of a process to allow another process to execute.

Presentation Interface. An application programming interface (API) that allows users to write graphics applications.

Presentation Manager. The interface of the OS/2 program that presents, in windows, a graphics-based interface to applications and files installed and running on the OS/2 program.

presentation space. A data structure, maintained by the operating system, that describes the device-independent drawing environment for an application.

primary code page. (1) The code page that is active when the computer is started. (2) The first code page listed in the CODEPAGE= statement in the CONFIG.SYS file. See *code page*.

primary window. The window in which the main dialog between the user and the program takes place. Contrast with *secondary window*.

print job. The result of sending a file to be printed.

print options. Used in Query Manager to specify print attributes such as line width, printed lines per page, printer nickname, page number, date and time, and print type.

print queue. A file containing a list of the names of files waiting to be be printed.

printer queue. Synonym for *spooler queue*.

print spooler. A program that allows output to be temporarily stored until it can be printed.

printer nickname. In Query Manager, a file that specifies which printer ID, printer type, and page size to use when printing.

printer pool. A group of printers in a network servicing a single spooler queue.

private application. An application maintained by an individual user and not available across a network. Contrast with *public application*.

privilege. In Database Manager, the right or authority to access a specific database object in a specific way. These rights are controlled by users with SYSADM (system administrator) authority or DBADM (database administrator) authority or by creators of objects. Privileges include rights such as creating,· deleting, browsing tables, or connecting to a database. See *direct privilege* and *indirect privilege*.

privilege level. (1) In Database Manager, the degree or extent to which a user can access database objects. For example, a user with SYSADM (system administrator) authority may have more privileges than a database administrator. (2) This term is also used by the base operating system to indicate to what level of resources a piece of code has access to. See also *direct privilege* and *indirect privilege*. (3) A protection level imposed by the hardware architecture of the Personal System/2 computer. There are four privilege levels (0 through 3). Only certain types of programs are allowed to execute at each privilege level. See *IOPL code segment*.

procedure call. In programming languages, a language construct for starting a procedure.

process. A collection of code, data and system resources, such as files, shared memory, bit maps, and threads, that constitute a running application.

profile. (1) An object that contains information about the characteristics of a computer system or application. (2) In Communications Manager, a part of a configuration file.

program. A sequence of instructions that a computer can interpret and execute.

program group. In Presentation Manager, a set of several programs that can be acted upon as a single entity. For example, they can all be started together. See *group*.

program name. The full file specification of a program. Contrast with *program title.*

program title. The name of a program as it is listed in the Program Manager Window. Contrast with *program name.*

prompt. (1) A symbol or action that requests a user entry or selection; for example, on the display screen of an IBM personal computer, the DOS **A>** prompt or OS/2 [C\] prompt. The user must respond to the prompt in order to proceed. (2) To generate a list of available choices for an entry field.

prompted interface. An interface consisting of messages, menus, pull-downs, and panels that guides the user through the steps necessary to perform a task.

prompted query. In Query Manager, a series of prompts, menus, panels, messages, and helps used to define queries.

protected field. A display field in which a user cannot add, modify, or delete data. Contrast with *unprotected field.*

protect mode. A method of program operation that limits or prevents access to certain instructions or areas of storage. Contrast w:th *real mode.*

protocol. The set of rules governing the operation of functional units of a communication system that must be followed if communication is to take place.

pseudocode. An artificial language used to describe computer program algorithms without using the syntax of any particular programming language.

public application. An application maintained by the network administrator and shared with users on a network. Contrast with *private application.*

pull-down. An extension of the action bar that displays a list of one or more choices that are available for a selected action bar choice.

pushbutton. A rounded-corner rectangle with text inside used in graphics applications for actions that occur when the pushbutton is selected.

Q

QMF. See *Query Management Facility.*

Query Management Facility (QMF). An IBM database management tool that allows extensive interactive query and report writing support. It runs under the control of the Interactive System Productivity Facility (ISPF), which in turn runs under Virtual Machine (VM/CMS) or Time Sharing Option (TSO) on host computers.

Query Manager. The part of Database Manager that provides menus, panels, pull-downs, and messages to assist, for example, in creating databases, editing data, generating reports, and making changes to Database Services configuration files. Query Manager also provides customization tasks like panels, menus, and procedures.

queue. (1) A line or list formed by items waiting to be processed; for example, a list of print jobs waiting to be printed. (2) A named object that is used to pass packets of information (queue elements) between threads of the same process or different processes.

R

radio button. A round button on the screen that is used to indicate items in a list from which only one item can be selected. Contrast with *check box.*

RAM. See *random access memory.*

RAM semaphore. A semaphore used to serialize different threads of a single process. See *semaphore.* Contrast with *system semaphore.*

random access. An access mode in which records can be read from, written to, or removed from a file in any order.

random access memory (RAM). A memory device into which data is entered and from which data is retrieved in a nonsequential manner.

raster. (1) In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space. (2) The coordinate grid that divides the display area of a display device.

RDS. See *Remote Data Services.*

read-only. A type of access to data that allows it to be read but not copied, printed, or modified.

read-only file. A file that may be read from but not written to.

real mode. A method of program operation that does not limit or prevent access to any instructions or areas of storage. The operating system loads the entire program into storage and gives the program access to all system resources. Contrast with *protect mode.*

Realtime Interface Coprocessor (RIC). The generic name given to a family of IBM communication adapters.

receive. (1) To obtain a message or file from another computer. Contrast with *send.* (2) In Communications Manager, the command used to transfer a file from a host.

record. (1) A set of data treated as a unit.

record format. The definition of how data is structured in the records contained in a file. The definition includes record names, field names, and field descriptions such as length and data type.

recursive routine. A routine that can call itself or be called by another routine called by the recursive routine.

reentrant code. Code that can be executed by more than one thread at a time.

refresh. (1) To update a window or panel with information changed to its current status.

register. A storage device having a specified storage capacity such as a bit, byte, or computer word, and usually intended for a special purpose.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line.

remote equipment. The modem and controller that provide the communications connection between a communications line and remote device or system. This remote equipment is at the other end of a data link from the host system. See also *remote workstation.*

remote file system. A file-system driver that gains access to a remote system without a block device driver.

remote workstation. A workstation that is indirectly connected to the system and needs data transmission facilities. See also *remote equipment.* Contrast with *local workstation.*

replace. To substitute one record or file for another record or file. Contrast with *append.*

replace mode. An entry mode that replaces existing characters with new characters at the text cursor location.

replaceable parameter. A parameter whose value is supplied when a batch or command file is run.

report. In Query Manager, the displayed or printed data generated by a query and formatted by a form.

report file. The copy, upgrade, and report functions generate a list of profiles with names and comments that were copied, upgraded, or exist for a report.

restore. (1) In Presentation Manager, to return a window to its original size or position following a sizing or moving action. (2) An OS/2 command that restores files that have been backed up.

resource file. A file containing information used in the definition of a window. Definitions can be of fonts, templates, accelerators, and mnemonics.

return. A dialog control action that saves data, if appropriate, and causes a transition to an application-defined dialog state. See also *exit* and *cancel*.

return code. (1) A value returned to a program to indicate the results of an operation requested by that program. (2) A code used to influence the execution of succeeding instructions. (3) A value returned to a program to indicate the results of an operation requested by that program.

revoke. To revoke is to remove access or authority from a user or a group ID.

RIC. See *Realtime Interface Coprocessor*.

right. A scrolling action that displays information to the right of the currently visible panel body area.

root segment. In a hierarchical database, the highest segment in the tree structure.

round-robin scheduling. A process that allows each thread to run for a specified amount of time.

routine. A program or a sequence of instructions called by a program that may have some general or frequent use.

run. To cause a program, object, utility, or other machine function to be performed.

run time. (1) The elapsed time taken for the execution of a computer program. (2) The time during which a program is active. (3) Any instant at which a program is being executed. (4) The time during which an instruction in an instruction register is decoded and performed.

S

SAA*. See *Systems Application Architecture**.

sample tables. The tables that are shipped with Database Manager (STAFF and ORG).

save. (1) To retain a copy for future use while continuing operations. (2) Retains window size to its original size or position following a sizing or moving action. See also *restore*.

scheduler. A computer program designed to perform functions such as scheduling, initiation, and termination of jobs.

SCREEN$. Character-device name reserved for the display screen.

sector. On disk or diskette storage, an addressable subdivision of a track used to record one block of a program or data.

scientific notation. A set of symbols and the rules for their use for the representation of scientific data. See also *FLOAT*.

screen. The physical surface of a display device upon which information is presented to the user. See also *panel*.

screen group. An OS/2 or DOS session. The OS/2 program allows multiple applications to run concurrently; where each application can access the display screen. Synonymous with *session* in the OS/2 program.

scroll. To move all or part of a display image vertically or horizontally so new data is displayed at one edge as preceding data is no longer displayed at the opposite edge.

scroll bar. A panel element associated with a scrollable panel area and used in graphics mode only, which indicates the direction of additional panel information not visible in the panel.

scrollable entry field. An entry field larger than the visible field.

scrollable selection field. A selection field that contains more choices than are visible.

scrolling. To move a display image vertically or horizontally in a manner such that new data is displayed at one edge as preceding data is no longer displayed at the opposite edge.

scrolling arrows. A type of scrolling information used on character screens. Scrolling arrows consist of the word More followed by symbols indicating the direction in which more information is available.

SDLC. See *Synchronous Data Link Control*.

search. (1) The process of looking for a specific item. (2) In Database Manager, used to locate rows or sets of rows in a table that meet specific criteria. (3) To scan one or more data elements of a set in order to find elements that have a certain property.

segment. A portion of a computer program that may be executed without the entire computer program being resident in main storage.

search argument. See *search condition*.

secondary window. A window or panel associated with the primary window in an application. Contrast with *primary window*.

segment attributes. In Presentation Interface, the attributes that apply to the segment as an entity, as opposed to the individual graphics primitive within the segment; for example, the visibility, transformability, or detectability of a segment. See *graphics attributes*.

select. (1) To mark or choose an item in a panel, action bar, or menu. (2) For pointing devices, see *select button*.

select button. The button on a pointing device, such as a mouse, that is pressed to select a menu choice. Also known as button 1. See *button*.

selection cursor. A type of cursor whereby users indicate the choice they want to interact with. The selection cursor is represented by highlighting the choice on which the selection cursor is currently positioned. Synonym for *cursor*. See also *text cursor*.

selection field. (1) A field containing a list of choices from which the user can select one or more. (2) A set of related choices that are presented to users. See also *entry field*. (3) An area of a panel that cannot be scrolled and contains a fixed number of choices.

semantics. The relationships between symbols and their meanings.

semaphore. An indicator that an application can use to signal the beginning and ending of an operation and to control access to a shared resource by multiple processes or multiple threads within the same process. See *RAM semaphore* and *system semaphore*.

send. (1) To send a message or file to another computer. Contrast with *receive*. (2) For Communications Manager, the command used to transfer a file to a host.

separator. Synonym for *panel body area separator*.

serial device. In OS/2 LAN Server, a resource (such as a modem or plotter) attached to an LPT or COM port for direct input/output (I/O) use.

serial port. A connector on a serial device to which cables for other devices are attached. Contrast with *parallel port*.

serialization. The consecutive ordering of items.

serialize. To ensure one or more events occur in a specified sequence.

server. (1) On a local area network (LAN), a data station that provides facilities to other data stations. (2) A computer that shares its resources with other computers on the network. See *IBM Operating System/2 LAN Server*.

service coordinator. A person responsible for ensuring problem determination support for the OS/2 program.

session. (1) A logical connection between two stations or network addressable units (NAUs) that allows them to communicate. (2) The period of time during which a user can communicate with an interactive system. (3) For the OS/2 program, a synonym for *screen group*. (4) In OS/2 LAN Server, a logical connection between a server and a requester that begins with a successful request for a shared resource.

shared data. Data that is used by two or more programs.

shared memory. An operating-system feature that allows system memory to be shared among resources.

shared resource. A directory (files resource), printer, or serial device made available to users on a network. The shared resources are directly attached to servers that share them but are not attached to the requesters asking to use them.

shell. (1) A software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens and communicate them to the operating system. (2) Software that allows a kernel program to run under different operating-system environments. **Note:** *Shells simplify user interactions by eliminating the user's concern with operating-system requirements. A computer may have several layers of shells for various levels of user interaction.*

short name. In Communications Manager, the one-letter name (A through Z) of the host presentation space or terminal emulation session. Synonymous with *short-session ID*.

short-session ID. Synonym for *short name*.

shutdown. In Task Manager, the procedure required before the computer is switched off to ensure that data and configuration information is not lost. See also *Task Manager*.

sibling. A child element that has the same parent as another child element.

sibling window. In Presentation Manager, the child windows that have the same parent window. See *child window*.

single-choice selection field. A type of selection field that allows users to select only one choice. Contrast with *multiple-choice selection field*.

single session. In Systems Network Architecture (SNA), a session that is the only session connecting two logical units (LUs).

slider box. An area in the scroll bar that shows users the size and position of the visible information in a window in relation to the total amount of information available. See also *thumb mark*.

SNA. See *Systems Network Architecture*.

SNA gateway. A feature of the OS/2 program that allows an OS/2 workstation to act as a communications controller between a support workstation, such as a personal computer on a LAN, and an SNA host. To the individual workstation, the SNA gateway is transparent.

SNA network. The part of the user application network that conforms to the formats and protocols of Systems Network Architecture (SNA). It enables reliable transfer of data among users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

soft checkpoint. The process of writing some information to the log that is used to determine the starting point in the log in case database restart is required.

soft error. An intermittent error on a network that causes data to be transmitted more than once before it is received accurately.

sort. To arrange a set of items according to keys used as a basis for determining the sequence of the items; for example, to arrange the records of a personnel file in alphabetical sequence by using the employee names as sort keys.

source file. A file that contains source statements for such items as high-level language programs and data description specifications.

source statement. (1) A statement written in symbols of a programming language; for example, RPG, COBOL, BASIC, and PL/I specifications are source statements. (2) A statement written in a programming language.

specific dynamic-link module. A dynamic-link module created for the exclusive use of an application.

special character. A graphic character that is not a letter, digit, or blank character, and which is usually not an ideogram; for example, a punctuation mark, general currency symbol, or a percent sign.

Spool Queue Manager. In Presentation Manager, the part of the Spooler that manipulates print jobs.

Spooler. A program that intercepts the data going to a device driver and writes it to disk. The data is later printed or plotted when the required device is available. A Spooler prevents output from different sources from being intermixed.

spooler queue. An ordered list of print jobs waiting to access a printer. Synonymous with *printer queue.*

SQL. See *Structured Query Language.*

stack. A list constructed and maintained so that the next data element to be retrieved is the most recently stored. **Note:** *This method is characterized as last-in-first-out (LIFO).*

standalone. Pertaining to operations that are independent of another device, program, or system.

statement. A language construct that represents a step in a sequence of actions.

static storage. (1) A read/write storage unit in which data is retained in the absence of control signals. Static storage may use dynamic addressing or sensing circuits. (2) Storage other than dynamic storage.

station. (1) A workstation. (2) An input or output point in a system that uses telecommunications facilities. Synonymous with *data station.*

stop bits. In asynchronous communications, the bit or bits used to end the character frame transmission.

storage. A media used to save information, such as a fixed disk. See also *memory.*

storage allocation. Synonym for *memory allocation.*

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. See also *SQL statement.*

style. Synonym for *window style.*

subdirectory. A directory contained within another directory in a file system hierarchy.

suballocation. The allocation of a part of one extent for occupancy by elements of a component other than the one occupying the remainder of the extent. See also *extent.*

subdirectory. In an IBM personal computer, a file referred to in a root directory that contains the names of other files stored on the diskette or fixed disk.

Support Center. Synonym for *IBM Support Center.*

suspend. An action that causes an active program to become temporarily inactive.

swapping. (1) A process that interchanges the contents of an area of real storage with the contents of an area in auxiliary storage. (2) In a system with virtual storage, a paging technique that writes the active pages of a job to auxiliary storage and reads pages of another job from auxiliary storage into real storage. (3) The process of temporarily removing an active job from main storage, saving it on disk, and processing another job in the area of main storage formerly occupied by the first job.

swap file. A file that contains segments of a program or data temporarily moved out of main memory.

swapping. A process that moves segments between memory and storage.

switch. (1) An action that moves the input focus from one area to another. A switch can be within the same window or from one window to another. See also *toggle.* (2) In a computer program, a conditional instruction and an indicator to be interrogated by that instruction. (3) A device or programming technique for making a selection, for example, a toggle, a conditional jump. (4) An action that moves the input focus from one area to another. This can be within the same window or from one window to another.

switch list. In Task Manager, the list of programs that are active.

switched line. A telecommunication line in which the connection is established by dialing. Contrast with *non-switched line*.

sync point. Synonym for *point of consistency*.

synchronous. Pertaining to two or more processes that depend upon the occurrences of specific events such as a common timing signal. Contrast with *asynchronous*.

Synchronous Data Link Control (SDLC). A communications protocol for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges can be duplex or half-duplex, over switched or non-switched links. Contrast with *binary synchronous communication*.

synchronous transmission. In data communication, a method of transmission in which the sending and receiving of characters are controlled by timing signals. Contrast with *asynchronous transmission*.

SYSADM. See *system administrator*.

system. A computer and its associated devices and programs. See also *workstation*.

system administrator (SYSADM). (1) In Communications Manager, the person with responsibility for installing, configuring, and setting up local communications networks, and ensuring the proper use of Communications Manager on all supported hardware.

system diskette. A fixed disk or diskette that can be used to start a system or workstation.

System Editor. An ASCII text editor provided as part of the OS/2 program.

System menu. The pull-down in the top left-hand corner of all primary windows that allows windows to be moved and sized. It can also be used to close a program and to select Task Manager.

system trace formatter. In the OS/2 program, a diagnostic tool used to retrieve system trace data from the system trace buffer and format the data for a display, printer, or a file. See also *system trace utility*.

system trace. A historical record of specific events in the execution of the Extended Edition. The record is usually produced for debugging purposes. See also *trace buffer*.

system trace utility. In the OS/2 program, a diagnostic tool used to capture a sequence of system events, function calls, or data for analysis. See also *system trace formatter* and *trace buffer*.

Systems Application Architecture (SAA). A set of software interfaces, conventions, and protocols that provide a framework for designing and developing applications across multiple computing environments.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through the networks and also the operational sequences for controlling the configuration and operation of networks.

system semaphore. A semaphore providing control between threads of multiple processes and that can be used by processes that do not share memory. See *semaphore*. Contrast with *RAM semaphore*.

T

tab. A typing action that moves the selection cursor to the next entry field, selection field, or to the next tab position.

tag. One or more characters attached to a set of data that contain information about the set, including its identification.

task. A set of one or more sequences of instructions treated by a control program as an element of work to be accomplished.

Task Manager. In the OS/2 program, the function that controls the starting and stopping of programs and controls which program has the input focus. It also allows the user to shut down the system. See also *shutdown*.

template. (1) In Presentation Manager, a data structure that defines the size, position, and appearance of a collection of windows. Templates can be defined for dialog boxes and menus. (2) On an IBM personal computer, a line entered from the keyboard and stored in memory from which the line can be retrieved, used again, or modified.

temporary storage. In computer programming, the memory locations reserved for intermediate results. Synonymous with *working storage.*

terminal. In data communication, a device, usually equipped with a keyboard and display screen, capable of sending and receiving information.

Terminate. An option in Task Manager that allows the user to stop a program without saving data or configuration information.

text. A set of characters or symbols.

text cursor. A symbol, displayed in an entry field, that shows users where typed input will be displayed. Synonym for *cursor*. See also *selection cursor*.

text-windowed application. The environment in which the operating system performs advanced-video input and output operations.

thousands separator. A character with a value greater than 999 that separates on numeric fields.

thumb mark. The portion of the scroll bar that describes the range and properties of the data that is currently visible in a window. See also *slider box*.

thread. The smallest unit of operation to be performed within a process. Every process has at least one thread and can have several. Each thread has a unique identifier.

thread of execution. A dispatchable unit of programming code within a process. It uses the resources of the process.

time slice. The period of processing time allocated for running a program.

time-critical process. A process that must be performed within a specified time after an event has occurred.

timer tick. See *clock tick*.

title bar. The area at the top of the window that contains the window title.

toggle. (1) The ability to switch between two modes; for example, insert and replace. (2) To switch between two modes; for example, on a personal computer connected to a network, to switch between the data entry and command entry modes or between standalone operation and terminal emulation. See also *switch*.

token. (1) In a local area network (LAN), the symbol of authority passed among data stations to indicate the station temporarily in control of the transmission medium. It consists of a starting delimiter, a frame control field, and an ending delimiter. The frame control field contains a token indicator bit that indicates to a receiving station that the token is ready to accept information. If the station has data to send along the network, it appends the data to the token. The token then becomes a frame.

token-ring. A network with a ring topology that passes tokens from one attaching device to another.

Token-Ring Network. See *IBM Token-Ring Network*.

trace. (1) A record of data that provides a history of events that occurred in a system. (2) The process of recording the sequence in which the statements in a program are executed and, optionally, are the values of the program variables used in the statements.

trace buffer. An allocation of space on a system for trace information. See also *system trace*.

transaction. An exchange between a workstation and another device that accomplishes a particular action or result.

transfer file. To send a file from one computer to another.

transmit. To send information from one place for reception elsewhere.

Tree. The window in the File System program that shows the organization of drives and directories.

truncate. (1) To end a computational process in accordance with some rule. (2) To remove the beginning or ending elements of a string. (3) To drop data that cannot be printed or displayed in the line width specified or available. (4) To shorten a field or statement to a specified length.

U

unnamed pipe. A pipe that allows communication only between related processes (a parent process and a child process or sibling processes). See also *pipe* and *named pipe*.

unformatted diskette. A diskette that contains no data and no track or sector format information. Contrast with *formatted diskette*.

unlock. To release an object or system resource that was previously locked and return it to general availability.

unprotected field. A displayed field in which a user can add, modify, or delete data. Contrast with *protected field*.

upload. To move data or programs from a workstation to a host. Contrast with *download*.

user. A person who uses a resource on a computer.

user access list. A list that defines individual users and their access authorities.

user ID. A unique name that identifies a user to the network.

user interface. The hardware, software, or both that allows a user to interact with and perform operations on a system, program, or device. See *interface*.

user profile. In the OS/2 LAN Server, a command file containing commands that set environment values and run programs automatically when a user logs on.

User Profile Management. User Profile Management is automatically installed with the IBM Operating System/2 program. It provides user ID validation and use and group management facilities that are used by both Database Manager and Communications Manager. Each installation of User Profile Management is local to the particular workstation where it is installed and validates users accessing controlled data or using programs that reside on that particular workstation. It also provides the mechanism for users to LOGON to the system and LOGOFF from the system to identify and authenticate system users.

user types. Users and network administrators. A user is any person who uses a resource on a computer. See also *network administrator*.

utility. The capability of a system, program, or device to perform the functions for which it is designed.

utility program. (1) A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program. (2) A program designed to perform an everyday task, such as copying data from one storage device to another.

V

value. (1) A specific occurrence of an attribute; for example, blue for the attribute color. (2) A quantity assigned to a constant, a variable, parameter, or symbol. See also *argument*.

variable. An entity that can assume a value.

variable length string. A character or graphic string whose length is not fixed but variable within set limits.

verification. The act of determining whether an operation has been accomplished correctly.

video input/output (VIO). An application programming interface (API) used to write data directly to the screen.

view. An object maintained by Database Manager that is a logical table but does not exist in physical storage but consists of data generated by a query that can be obtained from one or more tables.

VIO. See *video input/output*.

virtual machine. A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system. See *Conversational Monitor System*.

virtual memory (VM). The addressable space that is apparent to the user as the processor storage space but not having a fixed physical location.

Virtual Machine/Conversational Monitoring System (VM/CMS). (1) A time sharing system control program (CP) that consists of a control program that manages the resources of an IBM System/370 computing system in such a way that multiple remote terminal users have a functional simulation of a computing system at their disposal. It also contains the Conversational Monitoring System (CMS) that provides general time sharing, program development, and problem solving facilities. (2) Synonym for *virtual storage*.

virtual storage. (1) The storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (2) Addressable space that is apparent to the user as the processor storage space, from which the instructions and the data are mapped into the processor storage locations. (3) Synonymous with *virtual memory*.

Virtual Telecommunications Access Method (VTAM*). A set of programs that control communications between nodes and application programs running on a host (System/370) system.

VM. See *virtual machine*.

VM/CMS. See *Virtual Machine/Conversational Monitoring System*.

VTAM. See *Virtual Telecommunications Access Method*.

VT100. A Digital Equipment Corporation (DEC**) ASCII terminal.

volume. A file-system driver that uses a block device driver for input and output operations to a local or remote device.

W

warning message. A message that provides information to users to alert them to a possible error condition in the system. Warning messages are also sent to warn users that the action they are attempting can cause an undesirable condition or consequence.

wildcard character. See *global file-name character*.

window. (1) An area of the screen with visible boundaries through which a panel or portion of a panel is displayed. (2) In data communication, the number of data packets a DTE or DCE can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets and frames.

window border. A visual indication of the boundaries of a window.

window style. In Presentation Interface, the set of properties that influence how events related to a particular window are processed. Synonymous with *style*.

window title. A title that identifies the window and associates it with an application.

window procedure. A function that controls the appearance and behavior of its associated windows. The procedure receives and processes all input (messages) to these windows.

work area. (1) An area in which terminal devices (such as displays, keyboards, and printers) are located. Synonymous with *office*. (2) An area reserved for temporary storage of data.

working directory. The directory in which an application program is found. The working directory becomes the current directory when the application is started unless otherwise set.

working diskette. A computer fixed disk or diskette to which files are copied from an original diskette for use in everyday operation. Synonymous with *operational diskette*.

working storage. Synonym for *temporary storage*.

workstation. A terminal or personal computer, usually one that is connected to a mainframe or within a network, at which a user can run applications. See also *system*.

workstation address. (1) A number used in a configuration file to identify a workstation attached to a computer port. (2) The address to which the switches on a workstation are set, or the internal address assumed by the system, if no address is specified.

wrap. A condition that occurs when a file or buffer is full and new data overlays existing data. In Query Manager, if the length of the value to be displayed or printed exceeds the specified width, and the data is numeric, the value is replaced with a row of asterisks (****). If the data is character data, it can be truncated or wrapped depending on the edit code specified. Column edit codes are used in specifying these values.

write protection. The restriction of writing into a file or memory area by a user or program not authorized to do so. See also *read-only*.

X

X.25. In data communication, a recommendation of the CCITT that defines the interface between data terminal equipment and packet switching networks. Recommendations X.25 (Geneva 1980) and X.25 (Malaga-Torremolinos 1984) have been published.

X.25 network. A service providing packet-switched data transmission that conforms to Recommendation X.25 adopted by the CCITT.

xmodem. An asynchronous communications data transfer protocol where data is transferred in 128-byte blocks.

Numerics

3101. An IBM ASCII terminal.

3270 terminal emulation. A feature of Communications Manager that emulates the function of a 3270 workstation.

8.3 file-name format.. A file-naming convention in which file names are limited to eight characters before and three characters after a single dot. Usually pronounced, *eight-dot-three*. See also *non-8.3 file-name format*.

Index

OS/2 V1.3 Volume 1 (GG24-3630-00)

READER'S COMMENT FORM

IBM Corporation, International Technical Support Center

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

NOTE: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Comments:

If you would like a reply, please provide your name and address.

____

____

____

____

Thank you for your cooperation. No postage stamp is necessary if mailed in the USA. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address on the back of this page.)

Reader's Comment Form

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM International Technical Support Center
Department 91J, Building 235-2
901 Northwest 51st Street
Boca Raton, Florida 33432

Fold and tape Please Do Not Staple Fold and tape

IBM®

Cut or Fold Along Line

GG24-3630-00

OS/2 Version 1.3 Volume 1: New Features

GG24-3630-00

Printed in the U.S.A.

GG24-3630-00