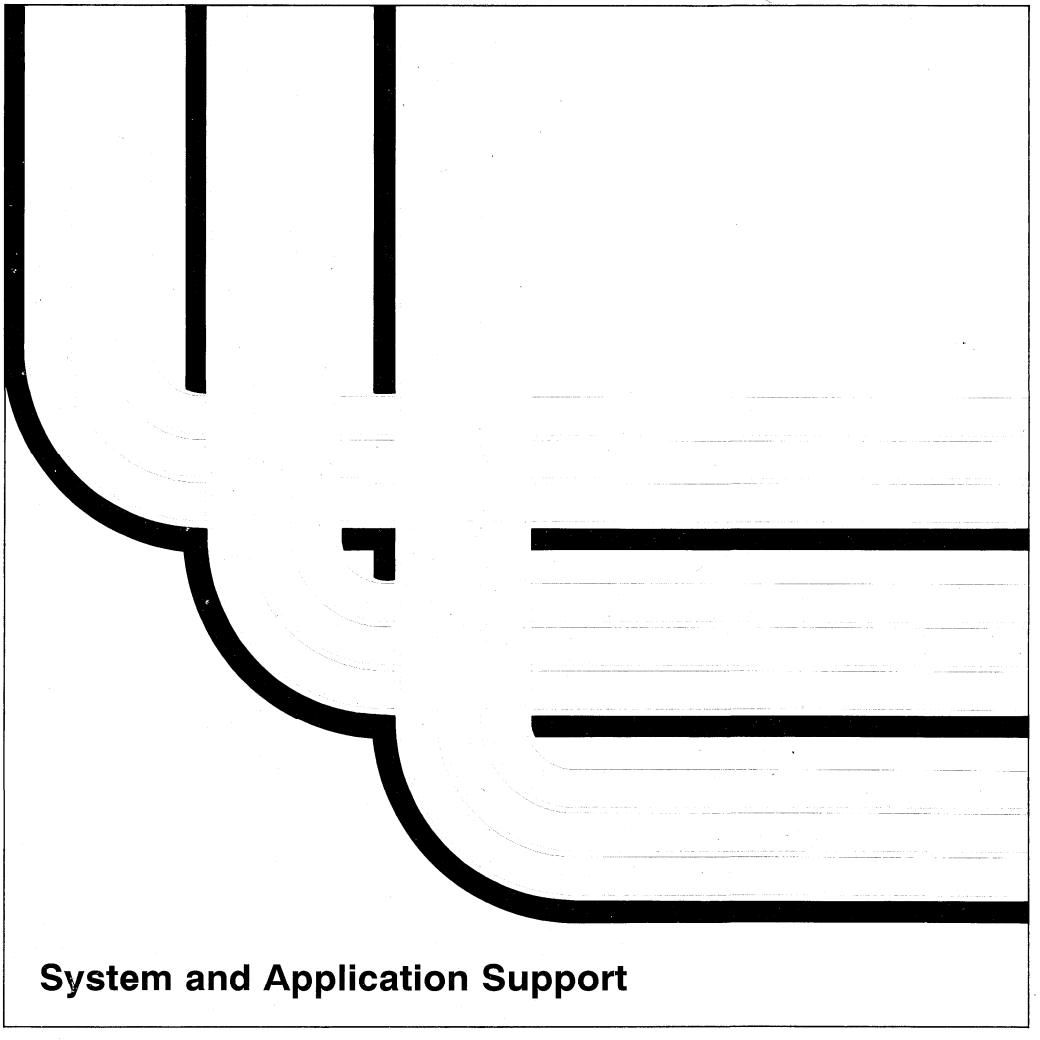
Application System/400

SC41-8091-02

PC Support/400: DOS and OS/2 Technical Reference

Version 2



System and Application Support

Application System/400

SC41-8091-02

PC Support/400: DOS and OS/2 Technical Reference

Version 2

Take Note!

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

Third Edition (November 1993)

This edition applies to the licensed programs

IBM PC Support/400, (Program 5738-PC1), Version 2 Release 3 Modification 0
IBM Operating System/400, (Program 5738-SS1), Version 2 Release 3 Modification 0

and to all subsequent releases and modifications until otherwise indicated in new editions. This major revision makes obsolete SC41-8091-01. Make sure you are using the proper edition for the level of the product.

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

A Customer Satisfaction Feedback form for readers' comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

Attn Department 245
IBM Corporation
3605 Highway 52 N
Rochester, MN 55901-7899 USA

or you can fax your comments to:

United States and Canada: 800+937-3430
Other countries: (+1)+507+253-5192

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 or restricting your use of it.

Contents

Figures

Notices

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 that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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, U.S.A.

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

Refer to the “Summary of Changes” on page xv for a summary of changes made to PC Support/400 and how they are described in this publication.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Programming Interface Information

This manual is intended to help programmers customize PC Support to meet their business requirements. For example, you can use the programs to implement additional security when using PC Support. This manual documents Product-Sensitive Programming Interface and Associate Guidance information provided by PC Support.

The Product-Sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-Sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

Application System/400
AS/400
AT
COBOL/2
ExecJet
| IBM
Operating System/2
Operating System/400
| OS/2
OS/400
Personal Computer AT
Personal Computer XT
Personal System/2
Proprinter
PS/2
Quickwriter
Quietwriter
RPG/400
SAA
Systems Application Architecture
XGA
XT
400

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies:

DIF	Lotus Development Corp.
Elite	AT&T
Epson	Epson America Inc
Helvetica	Linotype AG or its subsidiaries
HP	Hewlett-Packard Company
LaserJet	Hewlett-Packard Company
Microsoft	Microsoft Corp.
Multiplan	Microsoft Corp.
NEC	NEC Corporation
NetWare	Novell, Inc.
Novell	Novell, Inc.
PostScript	Adobe Systems Inc.
Windows	Microsoft Corp.
386	Intel Corp.

About This Manual

This manual contains technical information for advanced configuration and tailoring of PC Support/400 with the DOS or OS/2 operating system. This manual assumes that you have already installed and, if necessary, customized PC Support/400 for your personal computer.

You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

For a list of related publications, see the "Bibliography" on page H-1.

Who Should Use This Manual

This manual is intended primarily for programmers who have a personal computer attached to the AS/400 system and experienced personal computer users who want to understand PC Support/400 in more detail.

To work with the information in this manual, you should have knowledge of the personal computer, including either the DOS or OS/2 program, and the AS/400 system.

Before using this manual, you should be familiar with the following information:

- PC Support/400 installation information and memory requirements
- The use and operation of PC Support/400
- The use and operation of PC Support/400 on a IBM Personal System/55, if applicable

PC Support/400 Information

The following is a summary of the documentation available for PC Support/400. For a complete overview of the AS/400 system documentation, see the *Publications Guide*, GC41-9678.

Figure 0-1. PC Support/400 Tasks and Publications

Tasks	Environment	Look in
Planning, Installation, Administration, Problem Analysis, and Customization	DOS	*PC Support/400: DOS Installation and Administration Guide*, SC41-0006
	DOS (DBCS)	*PC Support/400: DOS Installation and Administration Guide (PS/55)*, SC41-0008
	OS/2	*PC Support/400: OS/2 Installation and Administration Guide*, SC41-0007
	OS/2 (DBCS)	*PC Support/400: OS/2 Installation and Administration Guide (PS/55)*, SC41-0009
Using PC Support Functions	DOS	*PC Support/400: DOS User's Guide*, SC41-8199
	DOS (DBCS)	*PC Support/400: DOS User's Guide (PS/55)*, SC41-2414
	OS/2	*PC Support/400: OS/2 User's Guide*, SC41-8200
	OS/2 (DBCS)	*PC Support/400: OS/2 User's Guide (PS/55)*, SC41-2415
Education	All	• Tutorial System Support[1] • PC Support Introduction (PCSINTRO)[2]
Problem Analysis	All	• Online message help (PCSHELP) and extended help[2] • The PC Support error log (PCSLOG)[2] • The PC Support Installation and Administration Guide for your environment
Technical Information and Programming	All	• *PC Support/400: DOS and OS/2 Technical Reference*, SC41-8091 • *PC Support/400: Application Program Interface Reference*, SC41-8254
Using RUMBA/400 and Programming	DOS Extenders Windows (SBCS) DOS Extenders Windows (DBCS) OS/2 2.0 (SBCS) OS/2 2.0 (DBCS)	• *PC Support/400: RUMBA/400 Guide and Reference*, SC41-0135 • *PC Support/400: RUMBA/400 Programmer's Tools Reference*, SC41-0136

Notes:

1. To start online education, enter STREDU at the AS/400 command line.
2. For information about using these PC Support help features, see "Getting Help When You Need It" in the *PC Support User's Guide*.

Summary of Changes

The changes to the product for this release are briefly described in this section. See references within each section for more details about each enhancement.

Changes to the Virtual Printer Function

Nine Virtual Printers in the OS/2 Environment

The Virtual Printer function supports nine virtual printers in the OS/2 environment. This is reflected throughout the OS/2 sections of Chapter 1, "Virtual Printer Function."

Installable File System for the OS/2 Environment

An installable file system allows users to redirect LPT1 through LPT9 as virtual printers, and provides an API in the form of a dynamic link library (DLL) that communicates with VPRT.EXE. See "Programs for OS/2 Users" on page 1-6.

Changes to the Shared Folders Function

Redirecting Shared Folders Drives

To redirect shared folders use the DOS network redirection interface rather than the DOS ASSIGN command. See "Using DOS Commands with the Shared Folders Function" on page 2-19.

Changes to the Transfer Function

Sort Sequence

When you transfer files from the AS/400 system to a personal computer, you can specify a sort sequence table that defines how the records are sorted. See "Sort Sequence Tables" on page 3-49.

Changes to Printer Emulation for Work Station Function

Host Print Transform

The print transform has been moved to the host, and sends the same ASCII data stream to an ASCII printer, regardless of how the printer is attached to the system.

Changes to User Exit Program Support

Data Queues Function

The data queues AS/400 server has additional support for the user exit program. See "Parameters for the Data Queues Function User Exit Program" on page 6-7.

Remote SQL Function

The remote SQL AS/400 server has additional support for the user exit program. See "Parameters for the Remote SQL Function User Exit Program" on page 6-8.

License Management Function

The license management AS/400 server has additional support for the user exit program. See "Parameters for the License Management Function User Exit Program" on page 6-11.

Chapter 1. Virtual Printer Function

This chapter describes the IBM* PC Support/400 virtual printer function when using the disk operating system (DOS), extended DOS, or the Operating System/2* (OS/2*) operating system. It discusses the programs, commands, translation tables, and output files used with the virtual printer function.

Unless otherwise noted, the information in this chapter applies to the PC Support/400 virtual printer function when using the DOS, extended DOS, or OS/2 operating system.

Virtual Printer Function Overview

The PC Support/400 virtual printer function allows use of both Application System/400* (AS/400*) and personal computer (PC) printers. You can take advantage of the faster speed and quality of the larger AS/400 printers, and share personal computer-attached printers with other users.

The virtual printer function exists as three versions for single-byte character set (SBCS) and three versions for double-byte character set (DBCS).

The SBCS versions are:

- A DOS version that runs on the DOS operating systems.
- An extended DOS version that runs on the DOS operating systems and takes advantage of the extended features of DOS.
- An OS/2 version that runs on the OS/2 operating systems.

The DBCS versions are:

- A DOS version that runs on the DOSJ operating systems.
- An extended DOS version that runs on the DOSV operating systems and takes advantage of the extended features of DOS.
- An OS/2 version that runs on the OS/2 J operating systems.

Note: The versions of virtual printer function that run on the DOS operating system will also run in the OS/2 DOS compatibility mode.

Virtual Printer Function Programs

This section describes the virtual printer function programs used by the DOS, extended DOS, and OS/2 operating systems.

Note: The EIMPCS.SYS program is required by the extended DOS version of the virtual printer function.

Programs for DOS and Extended DOS Users

Use the following programs in addition to the router when using the virtual printer function with the DOS operating system:

- CFGVPRT.COM
- SETVPRT.EXE
- VPRT.EXE

Use the following programs in addition to the router when using the virtual printer function with the extended version of the DOS operating system:

- CFGVPRT.COM
- SETVPRT.EXE
- VPRT.EXE
- EHNVPW.DLL
- SFNC.OVL

Note: Extended DOS support allows certain PC Support functions to use extended memory above 1MB (MB equals 1 048 576 bytes), thus making more of the conventional memory available for running other DOS programs. Extended DOS Support can be used on personal computers that have an 80286 processor or above and at least 384KB (KB equals 1024 bytes) of extended memory. Most 80286 and above systems with 1MB of memory have 384KB of extended memory available.

The VPRT.EXE program is required to run the other virtual printer function programs. When the VPRT.EXE program is run, the program becomes a resident portion of the DOS operating environment. The CFGVPRT.COM and SETVPRT.EXE programs communicate with the VPRT.EXE program.

CFGVPRT.COM, SETVPRT.EXE, and VPRT.EXE all communicate directly with the router, which manages communication to the AS/400 system. Interrupt X'68' is the interface to the router. You can change this interrupt to another value using the INTL entry in the CONFIG.PCS file.

The SETVPRT.EXE and CFGVPRT.COM programs run as personal computer application programs. You must have enough storage left after installing the router, the files and programs copied during the PC Support/400 installation, and the VPRT.EXE program to run at least one of these programs. For exact storage requirements, refer to the *PC Support/400 DOS Installation and Administration Guide*.

The EHNVPW.DLL file is an API program that communicates with the VPRT.EXE program in the Microsoft** Windows environment. It can be used by Windows** applications to control the VPRT.EXE program.

The SFNC.OVL program handles the interrupt X'21' interface to the VPRT.EXE program in the extended DOS environment.

You do not have to run both SETVPRT.EXE and CFGVPRT.COM. For example, if you plan to assign and release the same virtual printers each time you use your personal computer, you need to use only the CFGVPRT.COM program.

Figure 1-1 on page 1-3 shows the relationship among the programs required to use virtual printers with the DOS operating system.

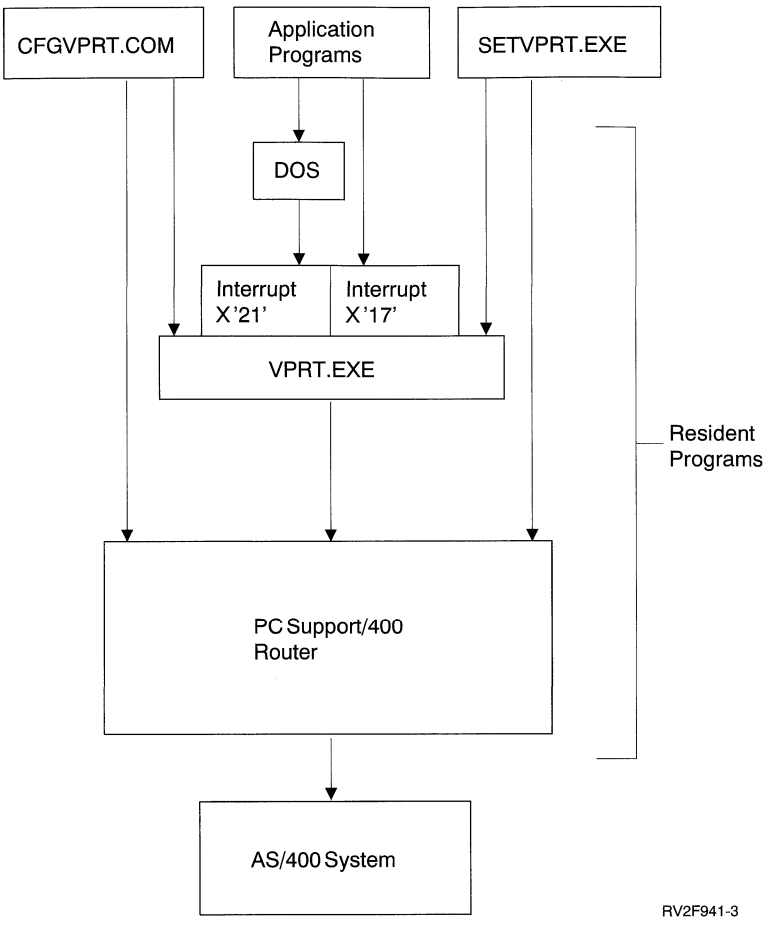


Figure 1-1. Relationship among Virtual Printer Function Programs (DOS)

Figure 1-2 shows the relationship among the programs required to use virtual printers with the DOS operating system on a PS/55 with a 5250PC program.

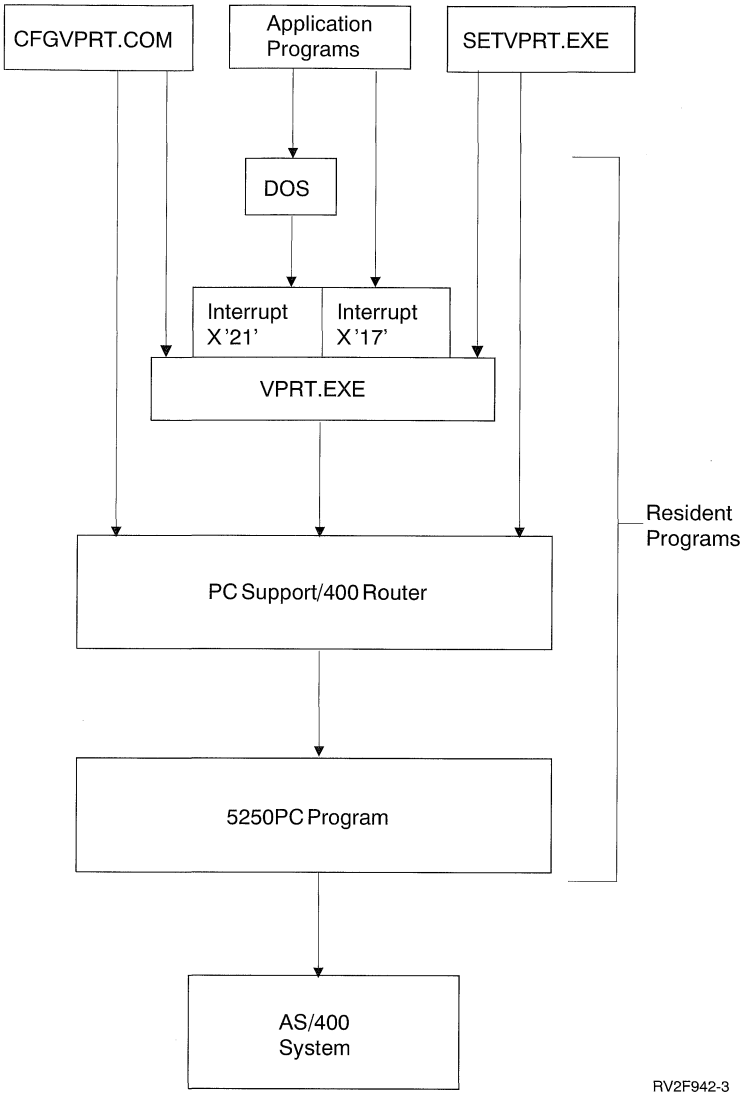


Figure 1-2. Relationship among Virtual Printer Function Programs Using DOS (DBCS)

Extended DOS users may use the PC Support program without the 5250PC program on a microchannel (MCA) PS/55. For more information about the 5250PC program, refer to Chapter 4, "Work Station Function" on page 4-1.

CFGVPRT.COM

CFGVPRT.COM is the batch interface to the virtual printer. It provides a quick way to assign and release virtual printers. CFGVPRT.COM uses PRNT identifiers in the CONFIG.PCS file or an alternative configuration file to assign or release virtual printers.

Batch Error Level Codes Set by CFGVPRT: If you are running a batch file, CFGVPRT sets a return code that can be checked to determine if errors have occurred. When all entries are processed successfully, the error level is 0. If an error occurs, but you choose to continue the operation anyway, the error level is set to 10 (X'0A'). If an unrecoverable error occurs, or you choose to end the CFGVPRT program, the error level is set to 20 (X'14').

SETVPRT.EXE

SETVPRT.EXE is the interactive interface to the virtual printer. It consists of a set of displays that allow you to:

- Refer to the currently assigned virtual printers and their parameters.
- Assign new virtual printers.
- Change the parameters of previously assigned virtual printers.
- Release existing virtual printers.
- Close output files that have been sent to virtual printers.

VPRT.EXE

VPRT.EXE is the resident portion of the virtual printer function. You must start VPRT.EXE before you can assign, change, or release the virtual printers, or close the output files. VPRT.EXE handles all print requests sent to Basic Input/Output System (BIOS) using interrupt X'17' and determines if the request is for a virtual printer or a personal computer printer. If the request is for a personal computer printer, VPRT passes the request to BIOS. If the request is for a virtual printer, then VPRT handles the request.

The extended DOS version of VPRT also intercepts DOS interrupt X'21' calls that are directed to an assigned printer. It uses these calls to open, close, and write data to spooled files.

VPRT.EXE uses AEPx entries in the CONFIG.PCS file or an alternative configuration file to change the ASCII-to-EBCDIC translation tables for the virtual printers. Refer to "Virtual Printer ASCII-to-EBCDIC Translation Tables" on page 1-27 for more information about translation tables.

Batch Error Level Codes Set by VPRT: If you are running a batch file, VPRT sets a return code that can be checked to determine if errors have occurred. When all entries are processed successfully, the error level is 0. If an error occurs, but you choose to continue the operation anyway, the error level is set to 10 (X'0A'). If an unrecoverable error occurs, or you choose to end the VPRT program, the error level is set to 20 (X'14').

EHNVPW.DLL

The EHNVPW.DLL program is an API that communicates with VPRT.EXE. It allows you to:

- Create Microsoft Windows customer applications.
- Assign and release virtual printers dynamically.
- Open and close spooled files.

See the *PC Support/400 API Reference* for more details.

SFNC.OVL

This program intercepts all interrupt X'21' requests issued by application programs and directs them to the virtual printer function programs for processing. The network connector (SFNC.OVL) allows the virtual printer to connect to other networks in a manner consistent with other applications.

Programs for OS/2 Users

Use the following programs in addition to the router and Communications Manager when using the virtual printer function with the OS/2 operating system:

- CFGVPRT.EXE
- EHNVPOS2.DLL
- SETVPRT.EXE
- VPRT.EXE
- EHNVPFS.IFS
- EHNVPRDR.SYS

The virtual printer function requires the router and Communications Manager.

Figure 1-3 shows the relationship among the programs required to use virtual printers with the OS/2 operating system.

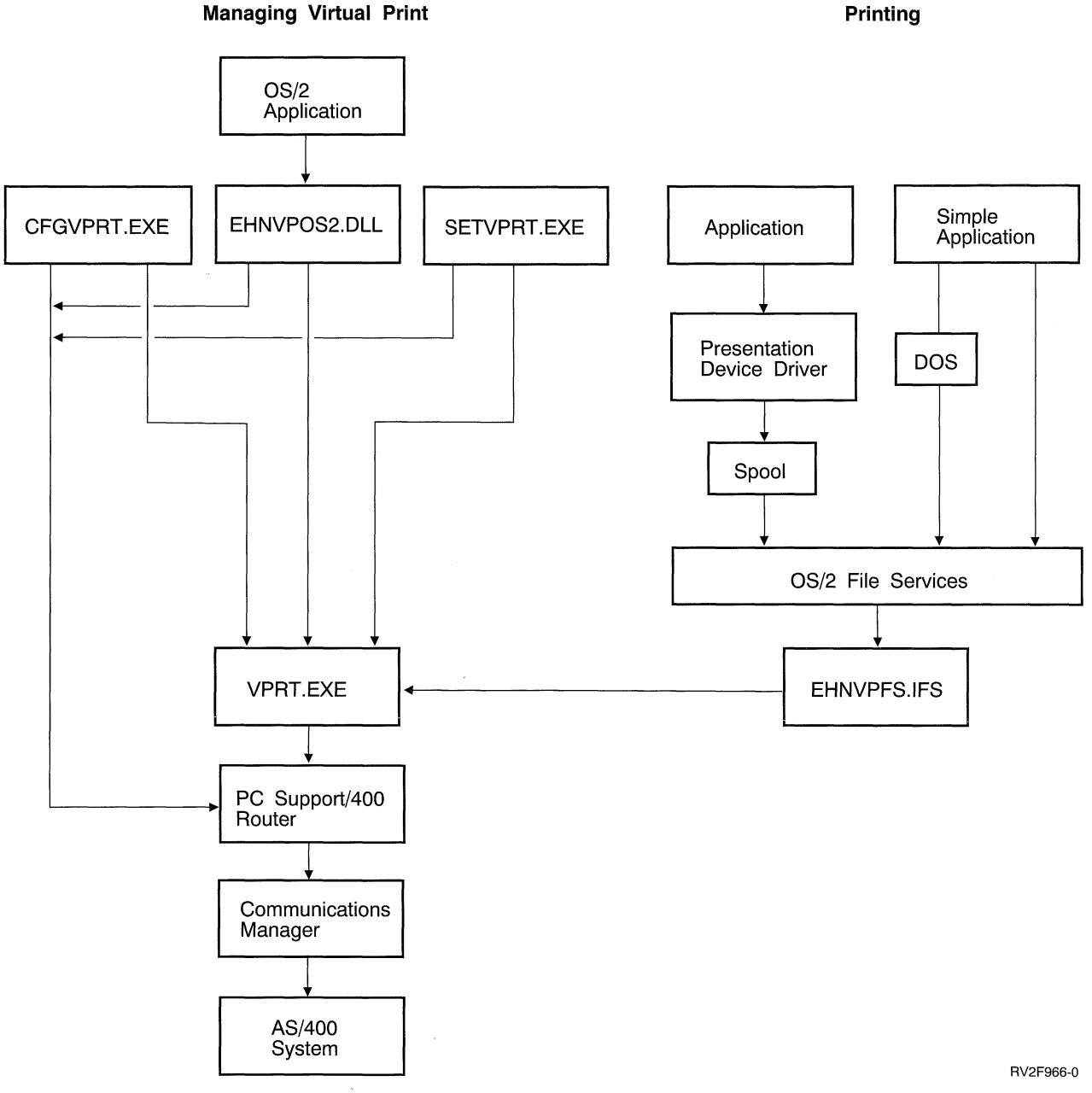


Figure 1-3. Relationship among Virtual Printer Function Programs (OS/2)

CFGVPRT.EXE

CFGVPRT.EXE program is the batch interface to the virtual printer. It provides a quick way to assign and release virtual printers. CFGVPRT.EXE uses PRNT entries to assign and release virtual printers and AEPx entries to change the ASCII to EBCDIC translation tables of assigned virtual printers in the CONFIG.PCS file or an alternative configuration file.

EHNVPOS2.DLL

The EHNVPOS2.DLL program is an API that communicates with VPRT.EXE. It allows you to:

- Create customer applications.
- Assign and release virtual printers dynamically.
- Open and close spooled files.
- Write data directly to the virtual printers for better performance.

See the *PC Support/400 API Reference* for more details.

SETVPRT.EXE

The SETVPRT.EXE program is the interactive interface to the virtual printer. It allows you to assign and release virtual printers, and change the ASCII-to-EBCDIC translation table for each assigned virtual printer. Refer to the *PC Support/400 OS/2 Installation and Administration Guide* for more information about assigning and releasing virtual printers. Refer to "Virtual Printer ASCII-to-EBCDIC Translation Tables" on page 1-27 for information on changing translation tables.

EHNVPFS.IFS

The EHNVPFS.IFS program is the installable file system (IFS) driver that reroutes print data sent to LPT1 through LPT9 if LPT1 through LPT9 are assigned as virtual printers. The operating system loads EHNVPFS.IFS when the system is started. To use this program, you should add a statement like the following to your CONFIG.SYS file:

```
IFS=D:\PCSOS2\EHNVPFS.IFS MON=1,2,3
```

This statement will load the IFS driver. The `MON=1,2,3` option disables the IFS redirection for LPT1, LPT2, and LPT3. If you add this option to the statement, the virtual printer function will get the data for these devices through the device monitor chain after they have gone through the spooling device.

VPRT.EXE

The CFGVPRT.EXE or SETVPRT.EXE programs load the VPRT.EXE program. You cannot start VPRT.EXE. If you try to start it, you will receive an error message.

When a virtual printer is assigned, the VPRT.EXE program begins processing printer data for LPT1 through LPT9. If an error occurs while VPRT.EXE is processing printer data, an error message is displayed. However, because VPRT.EXE accumulates blocks of printer requests, you may not receive an error message until the request is sent to the host system.

VPRT.EXE keeps track of the following for each virtual printer:

- Number of untranslatable characters (if the virtual printer is assigned data type 2)
- Number of pages for the last output file sent (if the virtual printer is assigned data type 2 and the request is the last job)
- Total number of files sent

Check this information by running SETVPRT and looking at the printer statistics section on the Current Virtual Printers display.

EHNVPRDR.SYS

The EHNVPRDR.SYS program is a device driver that defines LPT4 to LPT9 as devices, if they have not already been defined by some other device driver. The operating system loads EHNVPRDR.SYS when the system is started.

If LPT4 to LPT9 are already defined by another device driver, the EHNVPRDR.SYS program is not loaded. In this case, you can remove the DEVICE statement that loads this file from your CONFIG.SYS file.

Virtual Printer Commands When Using the DOS Operating System

This section describes the printer commands you can specify for virtual printers when using the DOS operating system. Use these commands in a programming language that has access to the personal computer registers and can start interrupt requests (IBM Assembler, for example). Personal computer application programs send printer commands either using a DOS function call or directly to the BIOS using interrupt X'17'.

SBCS Personal Computer Printer Requests

Virtual printers support the following types of personal computer printer requests:

- **Reset the printer**: This request is the same for a virtual printer as it is for a local printer. That is, register AH is equal to X'01' and register DX is equal to:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives a reset request, it closes the current output file, if one exists, and resets the virtual printer parameters to those you chose when you assigned the virtual printer.
- **Printer status request**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'02' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives a printer status request, it returns the current virtual printer status in the status byte (register AH). The virtual printer returns only status that applies to a virtual printer and, therefore, will not return all of the indicators supported by a locally attached personal computer printer.

  Figure 1-4 describes the indicators of the printer status byte and their meanings.

Figure 1-4 (Page 1 of 2). Printer Status Byte (SBCS)

Bit	Hex Value	PC Printer Status	Virtual Printer Status
0	01	Time-out	Not applicable
1	02	Unused	Unused
2	04	Unused	Unused

Figure 1-4 (Page 2 of 2). Printer Status Byte (SBCS)

Bit	Hex Value	PC Printer Status	Virtual Printer Status
3	08	I/O error	I/O error
4	10	Selected	Selected
5	20	Out of paper	Not applicable
6	40	Acknowledge	Not applicable
7	80	Not Busy	Not Busy

- **Print character**: The character sent to the printer is in register AL. This request is the same for a virtual printer as it is for a local personal computer printer. That is, register AH is equal to X'00' and register DX is equal to:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  This request prints most of the personal computer printer commands. Print character requests are sent 1 byte at a time. This byte can be a printer character, printer command, or part of a printer command string. These commands are described in "Personal Computer Printer Commands for SBCS" on page 1-12.

  When you use the DOS operating system or a programming language to print data, the print character request is handled for you. If you want to issue a reset or status request, use a programming language that allows you to set the registers and issue interrupts. A printer request issues with interrupt X'17'. Refer to the *DOS Technical Reference* manual for details.

DBCS Personal Computer Printer Requests

For DBCS, virtual printers support the following types of personal computer printer requests.

- **Reset the printer**: This request is the same for a virtual printer as it is for a local printer. That is, register AH is equal to X'01' and register DX is equal to:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives a reset request, it closes the current output file, if one exists, and resets the virtual printer parameters to those you chose when you assigned the virtual printer.
- **Printer status request1**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'02' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives a printer status request1, it returns the current virtual printer status in the status byte (register AH). The virtual and local personal computer printers use the same status. However, information that does not apply to the virtual printer is ignored.

Figure 1-5 describes the DBCS printer status byte for printer status request1 and their meanings.

Figure 1-5. Printer Status Byte (DBCS)

Bit	Hex Value	PC Printer Status	Virtual Printer Status
0	01	Time out	Not applicable
1	02	Unused	Unused
2	04	Unused	Unused
3	08	I/O error	I/O error
4	10	Selected	Selected
5	20	Out of paper	Not applicable
6	40	Acknowledge	Not applicable
7	80	Not Busy	Not Busy

- **Printer status request2**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'03' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives a printer status request2, it always returns X'0000' to the status byte (register AH).
- **Print an enlarged character**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'05' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives this request, it simulates enlarged printing by using the SCS CODE X'2BFD030220'.
- **Print a line with attributes**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'0B' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  Refer to the *DOS Technical Reference Manual* for more details. The virtual printer function supported for this request is described in "Personal Computer Printer Commands for SBCS" on page 1-12.
- **Set a printer control value**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'0C' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

Refer to the *DOS Technical Reference Manual* for more details. The virtual printer function supported for this request is described in "Personal Computer Printer Commands for SBCS" on page 1-12.

- **Get or release a printer**: This request is the same for the virtual printer as for a local personal computer printer. That is, register AH equals X'0D' and register DX equals:
  - 0000 for printer LPT1 (or PRN)
  - 0001 for printer LPT2
  - 0002 for printer LPT3

  When the virtual printer receives this request, it passes the request to BIOS.
- **Personal computer printer requests for LPT4 through LPT9**: To send print requests to printers LPT4 through LPT9 in the Extended DOS or OS/2 operating systems, use the following DOS function calls:
  - Open File
  - Write File
  - Close File

  These functions calls use interrupt X'21' to transfer characters to a file or device.

Personal Computer Printer Commands for SBCS

The virtual printer function supports most of the PC printer commands supported by the IBM 4201 Proprinter, but not all of them because each type of AS/400 printer in use as a virtual printer does not support all of the functions.

The virtual printer translates (or simulates) most personal computer printer commands (it ignores the personal computer printer commands that are unique to the personal computer) to an equivalent host system SNA character string (SCS) printer command. The translation is based on the Proprinter* personal computer data stream and occurs when you assign the virtual printer by specifying printer data type 2 (Convert ASCII to SCS) from the Assign or Change a Virtual Printer display. For more information on setting printer data type, refer to the *PC Support/400 User's Guide for DOS* and to the *PC Support/400 User's Guide for OS/2*.

The virtual printer does *not* support the following types of personal computer printer commands:

- Commands unique to personal computer printers but meaningless to system printers. For example, the Unidirectional Printing personal computer command causes a personal computer printer to print from left to right for all lines instead of printing bidirectionally. The host system printers do not need this unidirectional printing function.
- Commands that cannot be translated to or simulated by equivalent host system printer commands. For example, the 480 Bit Image Graphics Operation (ESC K) command for the personal computer graphics printer lets you print up to eight vertical dots at each horizontal position.

Figure 1-6 lists personal computer printer commands recognized by virtual printers assigned with printer data type 2.

Figure 1-6 (Page 1 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
NUL	00	Null Ends a list when you use it with ESC B and ESC D.	Simulate
BEL	07	Bell Sounds the printer buzzer for 1 second.	Ignore
BS	08	Backspace Spaces backward 1 character.	Ignore
HT	09	Horizontal Tab Tabs to the next horizontal stop. Set tab stops with ESC D. Proprinter printers set tab stops every eight columns when powered on.	Simulate
LF	0A	Line Feed Spaces the paper up one line.	Support
VT	0B	Vertical Tab Spaces the paper up to the next vertical tab position. Set tab stops with ESC B.	Simulate Virtual printer inserts the correct number of spaces.
FF	0C	Form Feed Advances the paper to the top of the next page and does a carriage return (CR).	Support
CR	0D	Carriage Return Moves the printer head to the first position of the same line. (No line feed operation occurs.) **Note:** IBM Personal Computer BASIC adds a line feed unless the command is X'8D' instead of X'0D'.	Support
SO	0E	Shift Out (Double Wide Printing by Line) Changes the printer to double-width print mode.	Support (if command override is Yes) If Shift In (SI) is active, the characters per inch value is changed to 10, otherwise, the characters per inch value is changed to 5. The characters per line value is changed so that the line length is the same as before the characters per inch number was changed. Simulate (if command override is No) Data is expanded by placing a blank after each character so that the line spaces properly. **Note:** A Carriage Return, Line Feed, DC4, Form Feed, ESC W, or Vertical Tab cancels double-width print mode.

Figure 1-6 (Page 2 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
SI	0F	Shift In (Condensed Printing) Causes the printer to begin condensed printing (17.1 cpi).	Simulate (if command override is Yes) If Shift Out (SO) or Continuous Double Wide Printing (ESC W) is active, the characters per inch number is changed to 10, otherwise, the characters per inch value is changed to 15. The characters per line value is changed so that the line length is the same as before the characters per inch value was changed. **Note:** DC2 cancels this mode, and returns spacing to 10 characters per inch. Ignore (if command override is No)
DC2	12	Device Control 2 (Print 10 cpi) Cancels condensed mode and 12 cpi and sets the printer to 10 cpi. Double-wide print is not canceled.	Support (if command override is Yes) If SO or ESC W is active, the characters per inch value is changed to 5, otherwise the characters per inch value is changed to 10. The characters per line value is changed so that the line length is the same as before the characters per line value was changed. Ignore (if command override is No)
DC4	14	Device Control 4 (Cancel Double-Width Printing by Line) Ends double-wide printing by line started by SO. Returns to the previous character spacing.	Support (if command override is Yes) The characters per inch value is changed to the previous characters per inch. The characters per line is changed so that the line length is the same as before the characters per inch value was changed. Simulate (if command override is No) The data is not expanded.
CAN	18	Cancel Clears the printer buffer.	Simulate If command override was set to Yes when the virtual printer was assigned, the printer buffer is cleared to the previous change of the characters per inch value. If the characters per inch was not changed or command override was set to No when the virtual printer was assigned, the printer buffer is cleared to the current print position.
ESC -	1B2D	Escape Minus (Underline) ESC - followed by a 1, prints all of the following data with an underline. ESC - followed by a 0, cancels the underline print mode.	Simulate Since some host system printers do not support this function, virtual printer simulates this function by placing a backspace and an underline character after each character.

Figure 1-6 (Page 3 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC 0	1B30	Escape Zero (1/8-inch line feeding) Changes paper feeding to 1/8 inch.	Simulate (if command override is Yes) **Note:** This command does not take effect for the current output file if virtual printer support receives it after a printable character. Ignore (if command override is No)
ESC 1	1B31	Escape One (7/72-inch line feeding) Changes paper feeding to 7/72 inch.	Ignore
ESC 2	1B32	Escape Two (Starts Variable Line Spacing) If no ESC A command has been given, line feeding returns to 1/6 inch. ESC 2 must be preceded by an ESC A. ESC 2 starts the variable line spacing set by the ESC A command.	Simulate This command is supported if the host attached printer supports variable line density commands.
ESC 3	1B33	Escape Three (Graphics Line Spacing) Format: ESC3;n Changes paper feeding to n/216 inch.	Simulate on host attached printers that support variable line spacing commands.
ESC 4	1B34	Escape Four (Set Top of Form) Format: ESC 4; Sets the current paper position as the top of form.	Ignore
ESC 5	1B35	Escape Five (Automatic Line Feed) Printer does a line feed each time it does a carriage return (CR).	Support
ESC 6	1B36	Escape Six (Select Character Set 2)	Simulate In character set 2, virtual printer treats any character between X'20' and X'FF' as a data character and translates it to an EBCDIC code according to the translation table you specify.
ESC 7	1B37	Escape Seven (Select Character Set 1)	Simulate In character set 1, virtual printer treats any character between X'00' and X'1F', and between X'80x' and X'9F' as a printer command. It treats other characters as data and translates them to an EBCDIC code according to the translation table you specify.
ESC 8	1B38	Escape Eight (Ignore Paper End)	Ignore
ESC 9	1B39	Escape Nine (Cancel Ignore Paper End)	Ignore

Figure 1-6 (Page 4 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC :	1B3A	Escape Colon (12 cpi) Format: ESC :; Sets the character spacing to 12 cpi.	Support (if command override is Yes) The characters per inch value is changed to 12. The characters per line value is changed so that the line length is the same as before the characters per inch value was changed. **Note:** DC2 resets to 10 cpi. Ignore (if command override is No)
ESC <	1B3C	Escape Less Than (Home Head) Format: ESC <;	Support Supported as a carriage return (CR) command.
ESC =	1B3D	Escape Equals (Character Font Image Transfer from AS/400 System) Format: ESC=;n1;n2; Starts a character font image transfer from the AS/400 system.	Ignore
ESC A	1B41	Escape A (Set Variable Line Spacing) Format: ESC A;n; ESC A sets the line feed value to n/72 inch.	Simulate on host-attached printers that support variable line spacing commands.
ESC B	1B42	Escape B (Set Vertical Tabs) Format: ESC B;n1;n2...;nk;NUL Sets vertical tab stop positions. The printer recognizes up to 64 vertical stop positions. The n's in the format above indicate tab stop positions. You must set tab stop numbers in ascending numeric order and end them with a NUL.	Simulate Since not all host system printers support this function, virtual printer simulates this function by placing the correct number of Line Feed commands.
ESC C	1B43	Escape C (Set Page Length) Format: ESC C;n; Sets the page length. The ESC C command must have a value following it to specify the length of page desired. Maximum form length is 255 lines. Format: ESC C;n;m; Sets the length of the pages in inches. It requires a value of 0 for n, and a value between 1 and 64 for m. **Note:** The total number of lines per page cannot exceed 255.	Simulate (if command override is Yes) **Note:** This command does not take effect for the current output file if the virtual printer support receives it after a printable character. Ignore (if command override is No)
ESC D	1B44	Escape D (Set Horizontal Tab Stops) Format: ESC D;n1;n2;...;nk;NUL; Sets the horizontal tab stop positions. Set a maximum of 112 tab stops.	Simulate

Figure 1-6 (Page 5 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC E	1B45	Escape E (Emphasized) Changes the printer to the emphasized print mode.	Simulate Not all host system printers support this function. The double-strike function is performed instead.
ESC F	1B46	Escape F (Emphasized Off) Stops printing in the emphasized print mode.	Support
ESC G	1B47	Escape G (Double Strike) Changes the printer to the double-strike print mode.	Simulate
ESC H	1B48	Escape H (Double Strike Off) Stops printing in the double-strike print mode.	Support
ESC I	1B49	Escape I (Select Print Mode) Format: ESC I;n Selects the printing quality.	Ignore
ESC J	1B4A	Escape J (Set Variable Line Feeding) Format: ESC J;n ESC J causes a line feed that is n/216 inch.	Simulate on host-attached printers which support variable line spacing commands.
ESC K	1B4B	Escape K (480-bit Image Graphics Mode) Format: ESC K;n1;n2;vl;...;vk; Changes from the text mode to the 480-bit image graphics mode.	Ignore
ESC L	1B4C	Escape L (960-bit Image Graphics Mode) Format: ESC L;n1;n2;vl;...;vk; Changes from the text mode to the 960-bit image graphics mode.	Ignore
ESC N	1B4E	Escape N (Set Skip Perforation) Format: ESC N;n; Sets the skip perforation function. The number following ESC N sets the value for the number of lines of skip perforation. The value of *n* must be between 1 and 255. You must reset ESC N anytime the page length (ESC C) is changed.	Simulate (if command override is Yes) **Note:** This command does not take effect for the current output file if virtual printer support receives it after a printable character. Ignore (if command override is No)
ESC O	1B4F	Escape O (Cancel Skip Perforation) Format: ESC O; Cancels the skip perforation function.	Simulate (if command override is Yes) **Note:** This command does not take effect for the current output file if virtual printer support receives it after a printable character. Ignore (if command override is No)

Figure 1-6 (Page 6 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC P	1B50	Escape P (Proportional Space Mode) Format: ESC P;n; Controls the proportional space mode. In this mode, the blank space around each character is unique.	Ignore
ESC Q 3	1B51	Escape Q (Deselect IBM Proprinter printer) Format: ESC Q 3; This sequence is intended for diagnostic purposes only.	Ignore
ESC R	1B52	Escape R (Set All Tabs to Power On Settings) Format: ESC R; Sets horizontal tabs at every eight positions starting at column 9 and clears all vertical tabs.	Support
ESC S 0	1B5300	Escape S (Superscript) Format: ESC S;0; Changes the printer to the superscript print mode.	Support (if command override is Yes) Ignore (if command override is No)
ESC S 1	1B5301	Escape S (Subscript) Format: ESC S;1; Changes the printer to the subscript print mode.	Support (if command override is Yes) Ignore (if command override is No)
ESC T	1B54	Escape T (Subscript/Superscript Off) Format: ESC T; Ends subscript or superscript printing started by ESC S. **Note:** If the printer receives an ESC T when the printer is not printing in subscript or superscript, the printer ignores it.	Support (if command override is Yes) Ignore (if command override is No)
ESC U	1B55	Escape U (Unidirectional Printing) Format: ESC U;n; The printer prints from left to right following the input of ESC U;1. The left-to-right printing operation is canceled when the printer receives ESC U;0.	Ignore

Figure 1-6 (Page 7 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC W 0	1B5700	Escape W (End Continuous Double-Wide Printing) Format: ESC W;0; Ends the double-wide printing started by ESC W 1. **Note:** ESC W resets double-wide printing mode previously set by SO.	Support (if command override is Yes) The characters per inch number is changed to the value it was before it was changed to 5. The characters per line is changed so that the line length is the same as before the characters per inch was changed. Simulate (if command override is No) The data is not expanded.
ESC W 1	1B5701	Escape W (Continuous Double-Wide Printing) Format: ESC W;1; Changes the printer to double-wide printing. **Note:** ESC W resets double-wide printing mode previously set by SO.	Support (if command override is Yes) If the Shift In (SI) is active, the characters per inch is changed to 10. If SI is not active, the characters per inch number is changed to 5. The characters per line is changed so that the line length is the same as before the characters per inch was changed. Simulate (if command override is No) Data is expanded by placing a blank after each character so that the line will space properly.
ESC X	1B58	Escape X (Set Horizontal Margins) Format: ESC X;n;m; Specify left and right margins in character positions, at the current pitch.	Ignore
ESC Y	1B59	Escape Y (960-bit Image Graphics Mode Normal Speed) Format: ESC Y;n1;n2;v1;v2;...;vk; Changes from the text mode to the 960-bit image graphics mode.	Ignore
ESC Z	1B5A	Escape Z (1920-bit Image Graphics Mode) Format: ESC Z;n1;n2;v1;v2;...;vk; Changes from the text mode to the 1920-bit image graphics mode.	Ignore
ESC [	1B5B40	Escape Left Bracket (Double High Printing) Format: ESC [;<@>;n1;n2;m1;m2;m3;m4; Specifies that subsequent text is to be highlighted as double wide, double high, or both. Line spacing can be controlled using this sequence.	Ignore The virtual printer ignores any command with the format 1B5B*xx*n1n2 (where *xx* is any 1-byte number and the total number of bytes ignored is (n1 + [256*n2]).
ESC \	1B5C	Escape Backslash (Print Continuously from all Characters on the Chart) Format: ESC \;n1;n2; Allows printing of all characters, including characters with an ASCII value below decimal 32 (normally control codes).	Ignore Characters with an ASCII value below decimal 32 are printed as the assigned virtual printer untranslatable character value.

Figure 1-6 (Page 8 of 8). PC Printer Commands Recognized by Virtual Printers Assigned with Printer Data Type 2

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC ^	1B5E	Escape ^ (Print Single Character from all Characters on Chart) Format: ESC ^ Allows printing of 1 character, including characters with an ASCII value below decimal 32 (normally control codes).	Ignore If the character has an ASCII value below decimal 32, it will be printed as the assigned virtual printer untranslatable character value.
ESC _	1B5F	Escape _ (Continuous Overscore) Format: ESC _; ESC _ followed by a 1, allows the printer to overscore all data and spaces. ESC _ followed by a 0, cancels overscoring.	Ignore
ESC o	1B6F	Escape o (End Document) Most personal computer printers do not use this function. Intended primarily for the 5218 Printer, this function tells the printer to make ready the time-out function, which was previously made ineffective by a Begin Document (ESC p) command.	Support (DOS operating system) When using the DOS operating system, the virtual printer supports this command. It closes the current output file (if any) and readies the printer time-out function previously made ineffective by a Begin Document (ESC p) command. Ignore (OS/2 operating system) When using the OS/2 operating system, the virtual printer does not support this command.
ESC p	1B70	Escape p (Begin Document) Most personal computer printers do not use this function. This function is intended primarily for the 5218 Printer. It tells the printer to make the time-out function temporarily ineffective until it receives an End Document (ESC o) command.	Support (DOS operating system) When using the DOS operating system, the virtual printer supports this command. It closes the current output file (if any). If the time-out function is made ready, this command temporarily makes the printer time-out function ineffective until the printer receives an End Document (ESC o) or Reset Printer command. Ignore (OS/2 operating system) When using the OS/2 operating system, the virtual printer does not support this command.

Changing Virtual Printer Output File Characteristics

While printing on a virtual printer, you can change some characteristics of an output file. These characteristics are:

- Characters per line
- Characters per inch
- Lines per inch
- Lines per page
- Page length

You can set these characteristics with printer commands within the data string or through SETVPRT or CFGVPRT. Figure 1-7 on page 1-21 describes the parame-

ters controlling the output file, the possible choices, the equivalent personal computer printer commands, and when they take effect.

Figure 1-7. Override Configuration Parameters

Parameter	Choices	PC Printer Command	In Effect
Characters per inch	5, 10, 12, or 15 cpi	SO ESC W DC2 ESC : SI	During the current print job only. Command override must be set to Yes.
Lines per inch	6 or 8 lpi	ESC 2 ESC 0	During the current print job only. Command override must be set to Yes. These commands only take effect if they are received by virtual printer before the first printable character.
Page length	1 to 255 lines	ESC C;n;	During the current print job only. Command override must be set to Yes. These commands only take effect if they are received by virtual printer before the first printable character.
Lines per page	1 to 255 lines, but not greater than the page length parameter. Set lines per page equal to the page length.	ESC N;n; ESC O;	During the current print job only. Command override must be set to Yes. These commands only take effect if they are received by virtual printer before the first printable character.

Using printer commands to change these characteristics affects only the current output file. The next output file opened for that printer uses the original values set by SETVPRT or CFGVPRT, unless you override them again by sending printer commands before the printable data.

Note: Using printer commands to change the characteristics of an output file does not affect the values displayed by the SETVPRT and CFGVPRT programs.

Personal Computer Printer Commands for DBCS

The virtual printer supports many PC printer commands, but not all of them because each type of AS/400 printer in use as a virtual printer will not support the functions.

Note: For DBCS, personal computer printers are PC CODE devices (prints in PC CODE).

The virtual printer translates (or simulates) most personal computer printer commands (it ignores the personal computer printer commands that are unique to the personal computer) to an equivalent host system SNA character string (SCS) printer command. The translation is based on the 5553 type printer personal computer data stream. The translation occurs when you assign the virtual printer by specifying printer data type 2 (Convert PC CODE to SCS) from the Assign or Change a Virtual Printer display. For more information on setting printer data type, refer to the *PC Support/400 User's Guide for DOS (PS/55)* and to the *PC Support/400 User's Guide for OS/2 (PS/55)*.

The virtual printer does *not* support the following types of personal computer printer commands:

- Commands unique to personal computer printers but meaningless to system printers. For example, the Unidirectional Printing (ESC%U) personal computer command causes a personal computer printer to print from left to right for all lines instead of printing bidirectionally. The host system printers do not need this unidirectional printing function.
- Commands that cannot be translated to or simulated by equivalent host system printer commands. For example, the Printing Image Data (ESC %1) command for the personal computer graphics printer lets you print up to 24 vertical dots at each horizontal position.

Figure 1-8 lists DBCS personal computer printer commands recognized by virtual printers assigned with printer data type 2.

Figure 1-8 (Page 1 of 2). DBCS PC Printer Commands Recognized by Virtual Printers

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
NUL	00	Null	Ignore
BEL	07	Bell	Ignore
BS	08	Back Space Spaces backward 1 character.	Support
HT	09	Horizontal Tab Tabs to the next horizontal stop.	Simulate
LF	0A	Line Feed Moves the paper up one line.	Support
VT	0B	Vertical Tab Spaces the paper up to the next vertical tab position.	Simulate Vertical printer inserts the correct number of spaces.
FF	0C	Form Feed Advances the paper to the top of the next page and does a carriage return (CR).	Support
CR	0D	Carriage Return Moves the printer head to the first position of the same line. (No line feed operation occurs.)	Support
DC1	11	Device Control 1	Ignore
DC3	13	Device Control 3	Ignore
CAN	18	Cancel Clears the printer buffer.	Support
FS	1C	Fixed Length Image Transmission	Ignore
SP	20	Space Spaces forward 1 character.	Support
ESC %1	1B2531	Image Transmission	Ignore
ESC %2	1B2532	Image Transmission and Enlarge	Ignore

Figure 1-8 (Page 2 of 2). DBCS PC Printer Commands Recognized by Virtual Printers

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESC %3	1B2533	Forward Horizontal Skip	Support
		Moves the specified number of dots.	Rounds up the specified dots to the nearest whole number, and moves that number of characters.
ESC %4	1B2534	Backward Horizontal Skip	Support
		Moves backward the specified number of dots.	Rounds up the specified dots to the nearest whole number, and spaces backward that number of characters.
ESC %5	1B2535	Forward Vertical Step Feed	Support
		Feeds paper at n/120 inch.	Rounds up the specified length to the nearest whole number of lines, and spaces that number of lines.
ESC %6	1B2536	CR to a Dot Position	Support
		When all data in the printer buffer is printed, tabs to the specified dot position with no line feed.	Rounds up the specified dots to the nearest whole number, and spaces that number of characters.
ESC %8	1B2538	Backward Vertical Step Feed	Ignore
ESC %9	1B2539	Define Line Spacing	Support (if command override is YES)
		Specifies the length of feed at n/120 inch.	Converts the specified length from n/120 inch into m/72 inch, and rounds up m to the nearest whole number of lines, and sets the length of feed at m/72 inch.
ESC %B	1B2542	Set Bidirectional Print Mode	Ignore
ESC %U	1B2555	Set Unidirectional Print Mode	Ignore
ESC (	1B28	Set 3 bytes Image Transmission Mode	Ignore
ESC)	1B29	Set 2 bytes Image Transmission Mode	Ignore
ESC F	1B46	Define Page Length	Support (if command override is YES)
		Defines page length at n/120 inch.	Rounds up the specified length to the nearest whole number of lines, and specifies the length of page.
ESC O	1B4F	Set High-speed Mode	Ignore
ESC P	1B50	Reset High-speed Mode	Ignore
ESC S	1B53	Cut Form Insert	Support
		Sets the printer position to the top of the sheet which is inserted from the Automatic Sheet Feeder, or Cut Form Feeder.	Feeds the paper to the top of the next page.
ESC V	1B56	Cut Form Eject	Ignore
ESC [	1B5B	Set Enlarge Mode	Support
		Characters after this code are printed in the double width.	
ESC]	1B5D	Reset Enlarge Mode	Support
		Characters after this code are printed normal width.	

Figure 1-9 lists DBCS personal computer printer extended escape commands (ESX) recognized by virtual printers assigned with printer data type 2. It does not list all extended escape commands. The commands that are not listed are ignored by the virtual printer.

Figure 1-9 (Page 1 of 2). DBCS PC Printer Extended Escape Commands Recognized by Virtual Printers

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESX 01	1B7E01 ..	Initialize Printer Initializes the parameters set by the control code.	Simulate After the page ejection, resets the following values: *Page Length* The value set by the SETVPRT and CFGVPRT commands *Character Pitch* 10 cpi *Line Pitch* The value set by the SETVPRT and CFGVPRT commands *Image Transmission Mode* 3 byte transmission mode *Other parameters* Values set by the control code
ESX 02	1B7E02 ..	Set Character Pitch	Support (if command override is YES)
ESX 03	1B7E03 ..	Set Line Pitch	Simulate (if command override is YES) Virtual printer uses the same line pitch. However, a personal computer printer uses the following character pitch. • ***Line pitch for the PC printer:*** 7.5 lpi 7.2 lpi • ***Line pitch for the AS/400 printer:*** 5 lpi 4.8 lpi
ESX 04	1B7E04 ..	Set Page Length	Simulate (if command override is YES) Page Length changes depending on the PC line pitch of 7.5 or 5 lpi.
ESX 0E .. 09	1B7E0E .. 09	Set Enlarge Mode Characters after this code are printed in double width.	Support
ESX 0E .. 0A	1B7E0E .. 0A	Reset Enlarge Mode Characters after this code are printed in normal width.	Support

Figure 1-9 (Page 2 of 2). DBCS PC Printer Extended Escape Commands Recognized by Virtual Printers

PC Printer Code	Hex Value	PC Printer Command/Function	Virtual Printer Support
ESX 0E .. 19	1B7E0E .. 19	Set Double Strike Mode Characters after this code are printed using double strike.	Support
ESX 0E .. 1A	1B7E0E .. 1A	Reset Double Strike Mode Characters after this code are printed using normal strike.	Support
ESX 11	1B7E11 ..	Set/Reset Underscore Mode	Simulate The horizontal grids are printed on the next line as an underscore. The maximum length of the grid is the same as the line length.
ESX 16	1B7E16 ..	Attribute Grid	Simulate The horizontal and vertical grids print as a grid. The maximum length of the grid is the same as the line length.
ESX 1C	1B7E1C ..	Horizontal Column Skip Skips the print position horizontally.	Simulate Virtual printer uses the absolute position that is decided by the current position and horizontal column skip.
ESX 1D	1B7E1D ..	Vertical Column Skip Skips the print position vertically.	Simulate Feeds the paper the number of vertical column skips.

Figure 1-10 shows the DBCS personal computer printer functions and the virtual printer supports for the attribute buffer print (INT 17 with AH=0B).

Figure 1-10 (Page 1 of 2). DBCS PC Printer Attribute

PC Printer Attribute	PC Printer Function	Virtual Printer Support
40	Draws an underscore.	Support
20	Prints bold characters.	Support Prints twice.
0C	Draws a vertical broken line.	Support Draws a vertical solid line.
08	Draws a vertical double width solid line.	Support Draws a vertical solid line.
04	Draws a vertical solid line.	Support
03	Prints a horizontal broken line.	Support Prints a horizontal solid line.
02	Prints a horizontal double width solid line.	Support Prints a horizontal solid line.

Figure 1-10 (Page 2 of 2). DBCS PC Printer Attribute

PC Printer Attribute	PC Printer Function	Virtual Printer Support
01	Prints a horizontal solid line.	Support

Figure 1-11 shows the DBCS personal computer printer functions and the virtual printer supports for the setup of the printer control value (INT 17 with AH=0C).

Figure 1-11. DBCS PC Printer Control Value

AL	BH	Printer Function	Virtual Printer Support
00		Default (5/10 cpi, 6 lpi, half speed)	Support (if command override is YES) Sets the value specified in SETVPRT or CFGVPRT for the lines per inch and characters per inch.
01	60 6C 78 Others	7.5/15 cpi 6.7/13.3 cpi 6/12 cpi 5/10 cpi	Support (if command override is YES)
02	00 01-10 11-14 15-18 19-1E 1F-FF	Ignore 7.5 lpi 6 lpi 5 lpi 4 lpi Ignore	Support (if command override is YES)
03	*nn* (BX)	Lines per page	Support (if command override is YES)
04	00 01	Full speed Half speed	Ignore
05	00 01	Unidirectional printing file Bidirectional printing	Ignore

Changing Virtual Printer Output File Characteristics

While printing on a locally attached personal computer printer, you can change some characteristics of an output file. These characteristics are:

- Characters per inch
- Lines per inch
- Page length

Figure 1-12 describes the parameters controlling the output file, the possible choices, the equivalent personal computer printer commands, and when they take effect.

Figure 1-12 (Page 1 of 2). DBCS Override Configuration Parameters

Parameter	Choices	PC Printer Command	In Effect
Characters per inch	5, 6, 6.7, or 7.5 cpi	INT 17 AX:0C01 (DOS only) ESX 02	During the current print job only. Command override must be set to Yes. These commands take effect from the next line after the virtual printer receives them.

Figure 1-12 (Page 2 of 2). DBCS Override Configuration Parameters

Parameter	Choices	PC Printer Command	In Effect
Lines per inch	2, 3, 4, 5, 6, 7.5, or 8 lpi	INT 17 AX:0C02 (DOS only) ESX 03	During the current print job only. Command override must be set to Yes.
Page length	Refer to your printer's manual.	INT 17 AX:0C03 (DOS only) ESX 04 ESC F	During the current print job only. Command override must be set to yes.

Using printer commands to change these characteristics affects only the current output file. The next output file opened for that printer uses the original values set by SETVPRT or CFGVPRT, unless you override them again by sending printer commands before the printable data.

Note: Using printer commands to change the characteristics of an output file does not affect the values displayed by the SETVPRT and CFGVPRT programs.

Virtual Printer ASCII-to-EBCDIC Translation Tables

This section describes what happens and what is required when the ASCII-to-EBCDIC translation tables are used with virtual printers. General information about virtual printer translation tables is also provided.

General Information

When a virtual printer is assigned with printer data type 2 specified, the virtual printer must translate the ASCII data to EBCDIC data. The ASCII-to-EBCDIC translation table is used for this translation.

Refer to the *PC Support/400 DOS Installation and Administration Guide* and to the *PC Support/400 OS/2 Installation and Administration Guide* for more information on the AEP1 through AEP9 entries. To create a new translation table or change the values in the default table, use the translation table utility.

Changing the translation table values from X'00' to X'1F' normally has no effect on the output from a virtual printer. If you specify one of these values, it is ignored because these values are reserved for printer commands.

The following ASCII values are reserved for printer commands:

character set 1 X'00' to X'1F'
X'80' to X'9F'

character set 2 X'00' to X'1F'

If the value of any entry in the translation table is below X'40', the value is changed to X'40'. This is the default character that is printed whenever an untranslatable character is found. You can specify a different value for this default by changing the *Untranslatable character* field when you assign the virtual printer. The value you choose must be greater than X'40' and less than X'FF'. Otherwise, you will get an error message.

Using the DOS Operating System

When the extended DOS version of the VPRT program is run, it searches for the AEP1 through AEP9 entries in the CONFIG.PCS file or the specified alternative configuration file. If an entry for a device exists, the ASCII-to-EBCDIC translation table is found and loaded into the device's memory. If the entry for a device is not found, the system default translation table is used.

All of the translation tables may be changed each time the VPRT program is run.

Translation tables for all virtual printers can be changed without the virtual printers having been assigned.

Using the OS/2 Operating System

When CFGVPRT is run, it searches for PRNT identifiers in CONFIG.PCS or in a specified alternative configuration file. The system default translation table is used when a virtual printer is assigned with a PRNT entry. Then, CFGVPRT searches for AEPx entries in the configuration file. You can only change the ASCII-to-EBCDIC translation table of an assigned virtual printer.

You can use SETVPRT to interactively specify your own translation tables by selecting the *Translation table* option from the Application Action Bar. This action is only displayed if there is a virtual printer assigned. You can change only the translation table of an assigned virtual printer. The PC Support default translation table is used when a virtual printer is assigned. If you use SETVPRT to change other values of a virtual printer that is already assigned, the translation table will not be changed.

Virtual Printer Translation Tables (DBCS)

This section describes what happens and what is required for DBCS when VPRT is used in the DOS operating system, and when CFGVPRT and SETVPRT are used in the OS/2 operating system. General information about virtual printer translation tables is also provided.

General Information

Refer to the *PC Support/400 DOS Installation and Administration Guide (PS/55)* and to the *PC Support/400 OS/2 Installation and Administration Guide (PS/55)* manuals for more information on the AEP1 through AEP9 entries. To create a new translation table or change the values in the default table, use the translation table utility.

Changing the translation table values from X'00' to X'1F' normally has no effect on the output from a virtual printer.

The following PC CODE values are reserved for printer commands:

X'00' to X'1F'

If you specify a value between X'00' and X'1F', it is ignored.

Blank fields in the table translate to the character you specify to replace the untranslatable character by using the SETVPRT or CFGVPRT program.

Using the DOS Operating System

- When the extended DOS version of the VRPT program is run, it searches for the AEP1 through AEP9 entries in CONFIG.PCS or the specified alternative configuration file. If an entry for a device exists, the translation table is found and loaded into the device's memory. If the entry for a device is not found, the PC Support/400 default translation tables are used.
- Virtual printer translation tables are loaded only when the virtual printer is assigned first, or when you change the table.
- All of the translation tables may be changed each time the VPRT program is run.
- Translation tables for all LPTs can be changed without the virtual printer being assigned.
- If the value of any entry in the translation table is below X'40', the value is changed to X'40'. The character prints as the value you assigned to the untranslatable character.

Using the OS/2 Operating System

- When CFGVPRT is run, it searches for PRNT entries in CONFIG.PCS or in a specified alternative configuration file. The PC Support/400 default translation tables are used when a virtual printer is assigned with a PRNT entry. Then, CFGVPRT searches for AEP*x* entries in the configuration file. You can only change the translation tables of assigned virtual printers.
- You can use SETVPRT to interactively specify your own translation tables by selecting the *Transaction table* option from the Application Action Bar. However, if you use SETVPRT to change a virtual printer that is already assigned, the translation table will not be changed.
- If the value of any entry in the translation table is below X'40', the value is changed to X'40'. The character prints as the value you assigned to the untranslatable character.

Chapter 2. Shared Folders Function

This chapter discusses the PC Support/400 shared folders function when using the disk operating system (DOS), extended DOS, and the Operating System/2 (OS/2) programs. It describes the programs associated with the shared folders function and discusses the DOS and OS/2 commands and function calls you use with folders on the AS/400 system. It also discusses caching, file sharing, and maximizing performance through better file management.

For information on how to access and use folders from your personal computer, refer to the following publications:

- *PC Support/400 User's Guide for DOS*
- *PC Support/400 User's Guide for DOS (PS/55)*
- *PC Support/400 User's Guide for OS/2*
- *PC Support/400 User's Guide for OS/2 (PS/55)*

Using Folders

The AS/400 system uses folders to store information such as text documents, mail, and data. PC Support/400 shared folders function allows you to access and share members in folders on the AS/400 system. It also allows you to store personal computer (PC) files in folders. The documents in a folder on the AS/400 system can be used by both AS/400 system users and personal computer users. A folder can be accessed by more than one AS/400 system user or personal computer user at a time.

Shared Folders Function Using the DOS Operating System

This section discusses the PC Support/400 shared folders function when you are using the DOS operating system. DOS classifies drives as either local or remote drives. When you assign a drive using the shared folders function, the drive is identified to DOS as a remote drive.

If you intend to use a DOS application, such as a spreadsheet or database application to store and access data on a folder on the AS/400 system, you should specify in the application that you are using a remote drive.

We recommend that you use the extended DOS version of the shared folders function. For information on using the shared folders function with extended DOS, see "Shared Folders Function Using Extended DOS" on page 2-16.

Shared Folders Function Programs for DOS Users

The shared folders function consists of the following three sets of programs:

shared folders function type 0

Use this set of programs only when your personal computer does not have enough memory to run shared folders function type 1 or type 2 together with your applications, or if you do not want to upgrade your personal computer to use shared folders function type 1 or type 2. Shared folders function type 0 uses less personal computer memory than shared folders function type 1 or type 2, but it may not perform as quickly.

shared folders function type 1
Use this set of programs when your personal computer has enough memory to run it together with your applications, and you want improved performance and the capability to remove shared folders function type 1 from memory when it is not being used.

shared folders function type 2
Use this set of programs when your personal computer has enough memory to run it together with your applications and you want improved performance over shared folders function type 1. To run this set of programs, your AS/400 system needs to be running Version 1 Release 3.0 or later.

Refer to the *PC Support/400 DOS Installation and Administration Guide* or to the *PC Support/400 DOS Installation and Administration Guide (PS/55)* for information on how to change the shared folders type that you are using and for the memory requirements of your personal computer.

Shared Folders Function Type 0

Shared folders function type 0 consists of the following programs used in addition to the router and the memory manager:

- FSPC.EXE
- CFGFLR.EXE
- ECYDDX.SYS
- FSDD.SYS

These programs communicate with the router and the memory manager (EIMPCS.SYS).

Note: The router manages communications with the AS/400 system, and the memory manager manages the personal computer memory.

Figure 2-1 on page 2-3 shows the SBCS relationship between these programs.

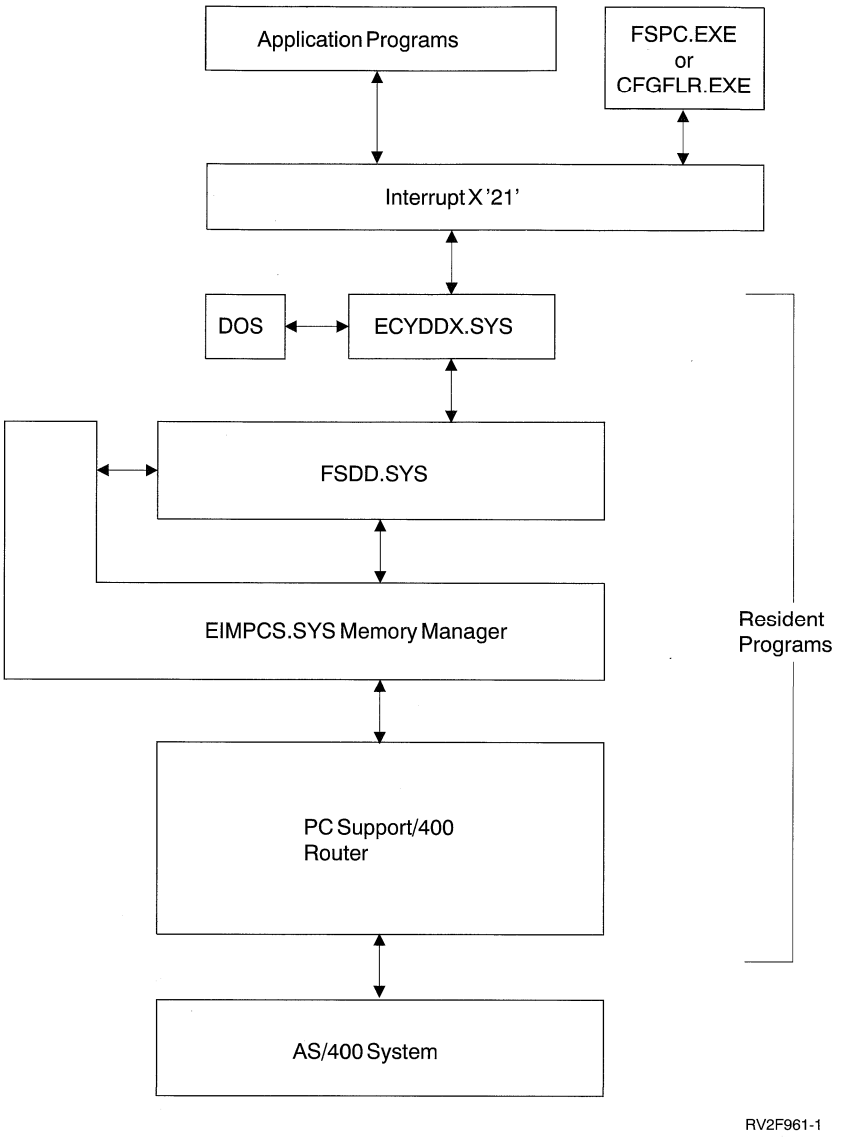


Figure 2-1. Shared Folders Function Type 0 Programs (SBCS)

Figure 2-2 on page 2-4 shows the DBCS relationship between the shared folders function type 0 programs.

Extended DOS users may use the PC Support program without the 5250PC program on a microchannel (MCA) PS/55. For more information about the 5250PC program, refer to Chapter 4, "Work Station Function" on page 4-1.

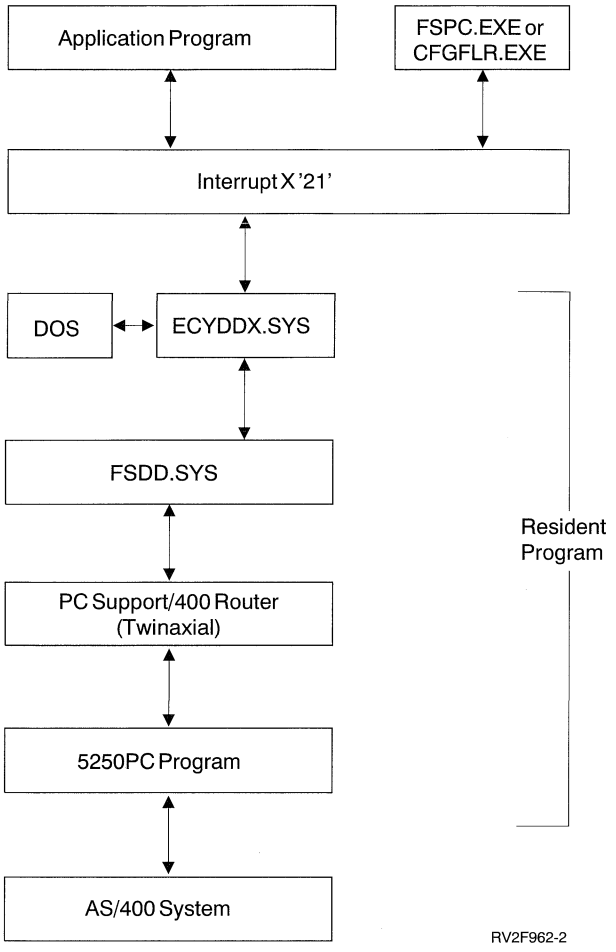


Figure 2-2. Shared Folders Function Type 0 Programs (DBCS)

FSPC.EXE Program: This program runs as a PC application program. It works with the other programs to provide the user interface for shared folders function type 0. This program allows you to assign or release drives through a batch command interface or an interactive display interface.

Note: FSPC.EXE provides the same user interface for shared folders function type 1 and type 2.

CFGFLR.EXE Program: This program runs as a PC application program. It can be used instead of FSPC.EXE to assign and release shared folder drives. This program allows you to assign or release drives through a batch command interface.

ECYDDX.SYS Program: This program is a device driver which is loaded and becomes resident when the personal computer is started. It intercepts all interrupt 21 (hex) requests issued by application programs and directs them to the DOS operating system or to the shared folders function type 0 programs for processing. The ECYDDX.SYS program cannot be removed from PC memory.

Note: ECYDDX.SYS is also used with shared folders function type 1 and type 2.

FSDD.SYS Program: This program is a device driver which is loaded and becomes resident when the personal computer is started. It receives requests from ECYDDX.SYS and passes these requests to the AS/400 system through the router.

FSDD.SYS returns control information and data back to ECYDDX.SYS. The FSDD.SYS program cannot be removed from PC memory.

Shared Folders Function Type 1

Shared folders function type 1 consists of the following programs used in addition to the router, the memory manager, and the Remove PC Support command:

- FSPC.EXE
- CFGFLR.EXE
- ECYDDX.SYS
- STARTFLR.EXE
- FLRREQ1.COM
- FLRMCAC1.COM

These programs communicate with the router, the memory manager (EIMPCS.SYS), and the Remove PC Support (RMVPCS.EXE) command.

Notes:

1. The router manages communications with the AS/400 system, and the memory manager manages the personal computer memory.
2. The Remove PC Support command removes all of the resident shared folders type 1 programs listed above except for ECYDDX.SYS.

Figure 2-3 shows the SBCS relationship between these programs.

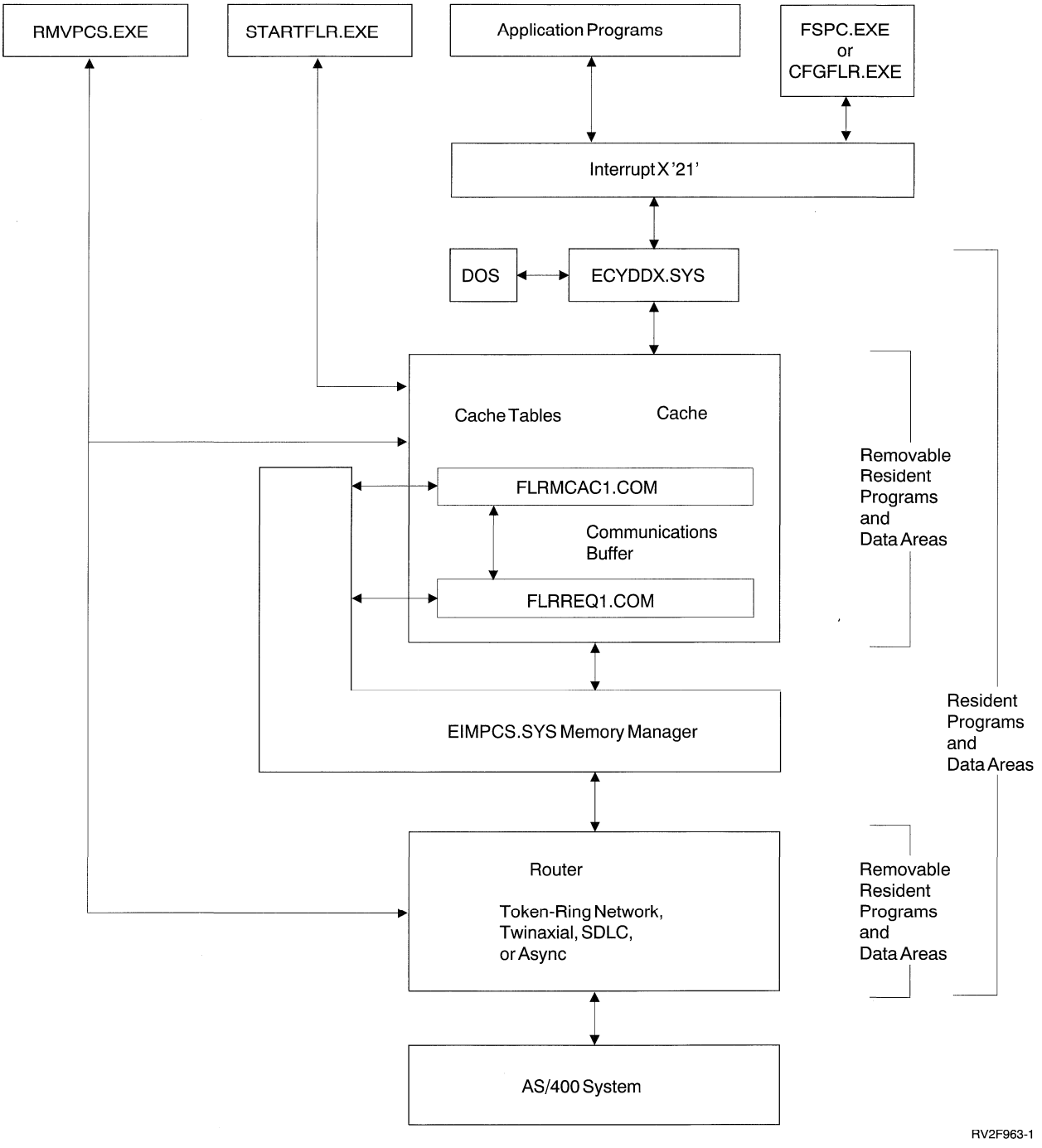


Figure 2-3. Shared Folders Function Type 1 Programs (SBCS)

Figure 2-4 shows the DBCS relationship between the shared folders function type 1 programs.

Extended DOS users may use the PC Support program without the 5250PC program on a microchannel (MCA) PS/55. For more information about the 5250PC program, refer to Chapter 4, "Work Station Function" on page 4-1.

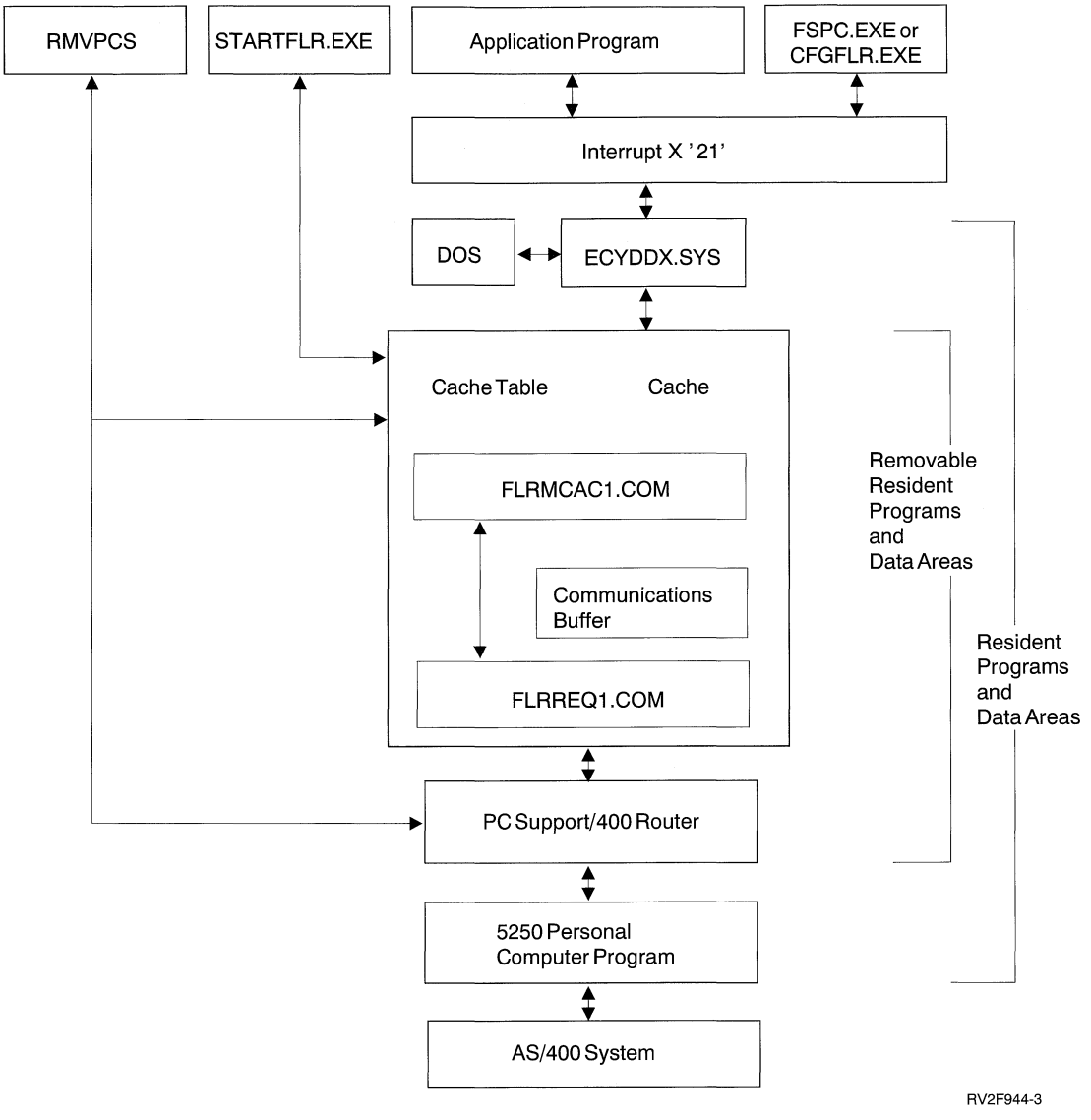


Figure 2-4. Shared Folders Function Type 1 Programs (DBCS)

FSPC.EXE Program: This program runs as a personal computer application program. It works with the other programs to allow the user to work with shared folders type 1.

Note: FSPC.EXE provides the same user interface for shared folders type 0 and type 2.

CFGFLR.EXE Program: This program runs as a PC application program. It can be used instead of FSPC.EXE to assign and release shared folder drives.

ECYDDX.SYS Program: This program is a device driver which is loaded and becomes resident when the personal computer is started. It intercepts all interrupt 21 (hex) requests issued by application programs and directs them to the DOS operating system or to the shared folders type 1 programs for processing. The ECYDDX.SYS program cannot be removed from personal computer memory.

Note: ECYDDX.SYS is also used with shared folders type 0 and type 2.

STARTFLR.EXE Program: This program runs as a personal computer application program. It starts the shared folders type 1 function by doing the following:

- Processing the following identifiers in the CONFIG.PCS or alternative configuration file specified after the STARTFLR command:
  - MCAC (memory cache in conventional memory)
  - MCAE (memory cache in expanded memory)
  - FEMU (shared folders expanded memory usage)
  - FTYP (shared folders function type)
  - CBSZ (communications buffer size)

  Refer to the *PC Support/400 DOS Installation and Administration Guide* for information on these identifiers.
- Loading the FLRREQ1.COM program in conventional or expanded memory as specified by the FEMU identifier.
- Creating a communications buffer of the size specified by the CBSZ identifier and locating it in conventional or expanded memory as specified by the FEMU identifier.

If caching is to be used (specified by the MCAC or MCAE identifiers), STARTFLR.EXE also does the following:

- Loads the FLRMCAC1.COM program in conventional or expanded memory as specified by the FEMU identifier.
- Creates cache tables and a cache according to the cache size specified by the MCAC or MCAE identifiers, and locates them in conventional or expanded memory as specified by the FEMU identifier.

The STARTFLR.EXE program is normally run from the AUTOEXEC.BAT file or the STARTPCS.BAT file. It must be run before the first folder can be assigned.

EIMPCS.SYS Program: This program is used by STARTFLR to allocate and initialize the shared folders programs and memory cache. The FLRREQ and FLRMCAC programs use EIMPCS.SYS to transfer operation control between conventional segments of code and expanded memory segments of code and to access expanded memory segments of data (application buffers, cache tables and buffers, and the communications buffer).

FLRREQ1.COM and FLRMCAC1.COM Programs: FLRREQ1.COM receives requests for file input and output from ECYDDX.SYS and passes these requests to the AS/400 system through the router. FLRREQ1.COM returns control information and data back to ECYDDX.SYS.

When using caching, FLRREQ1.COM works with FLRMCAC1.COM to cache data that is sent to and received from the AS/400 system. FLRMCAC1.COM controls the cache and maintains the cache tables.

RMVPCS.EXE Program: This program runs as a personal computer application program. It ends the shared folders type 1 or type 2 function and removes the FLRREQ1.COM or FLRREQ2.COM programs and the FLRMCAC1.COM or FLRMCAC2.COM programs, the communications buffer, cache tables, and cache from PC memory.

Shared Folders Function Type 2

Shared folders type 2 consists of the following programs used in addition to the router, the memory manager, and the Remove PC Support command:

- ECYDDX.SYS
- STARTFLR.EXE
- FLRREQ2.COM
- FLRMCAC2.COM
- FSPC.EXE
- CFGFLR.EXE

These programs communicate with the router, the memory manager (EIMPCS.SYS), and the Remove PC Support (RMVPCS.EXE) command.

Notes:

1. The router manages communications with the AS/400 system, and the memory manager manages the PC memory.
2. The Remove PC Support command removes all of the resident shared folders type 2 programs listed above except for ECYDDX.SYS.

Figure 2-5 on page 2-10 shows the SBCS relationship between the programs that communicate with the router which manages the communications with the AS/400 system.

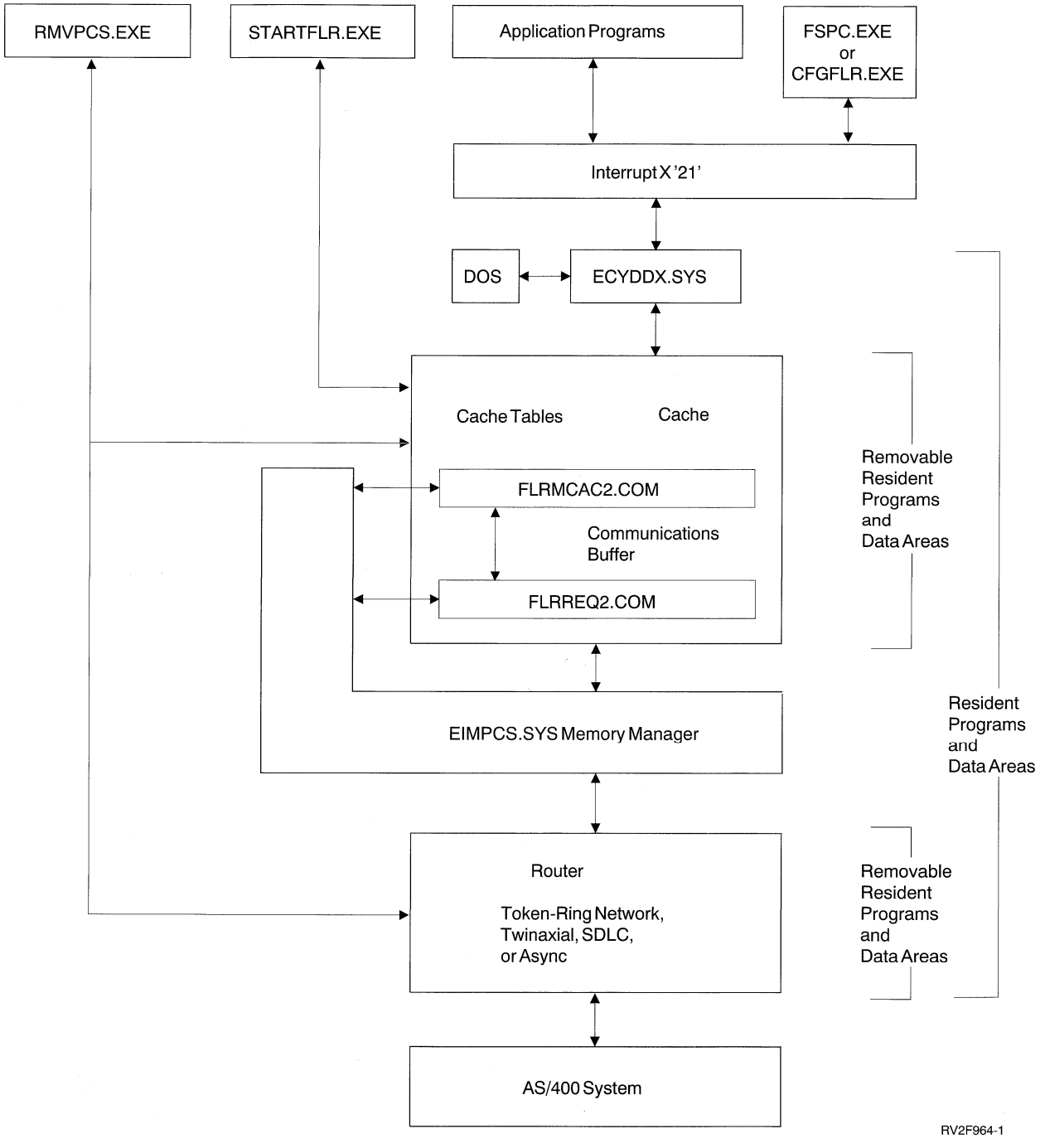


Figure 2-5. Shared Folders Function Type 2 Programs (SBCS)

Figure 2-6 shows the DBCS relationship between the programs that communicate with the router which manages the communications with the AS/400 system.

Extended DOS users may use the PC Support program without the 5250PC program on a microchannel (MCA) PS/55. For more information about the 5250PC program, refer to Chapter 4, "Work Station Function" on page 4-1.

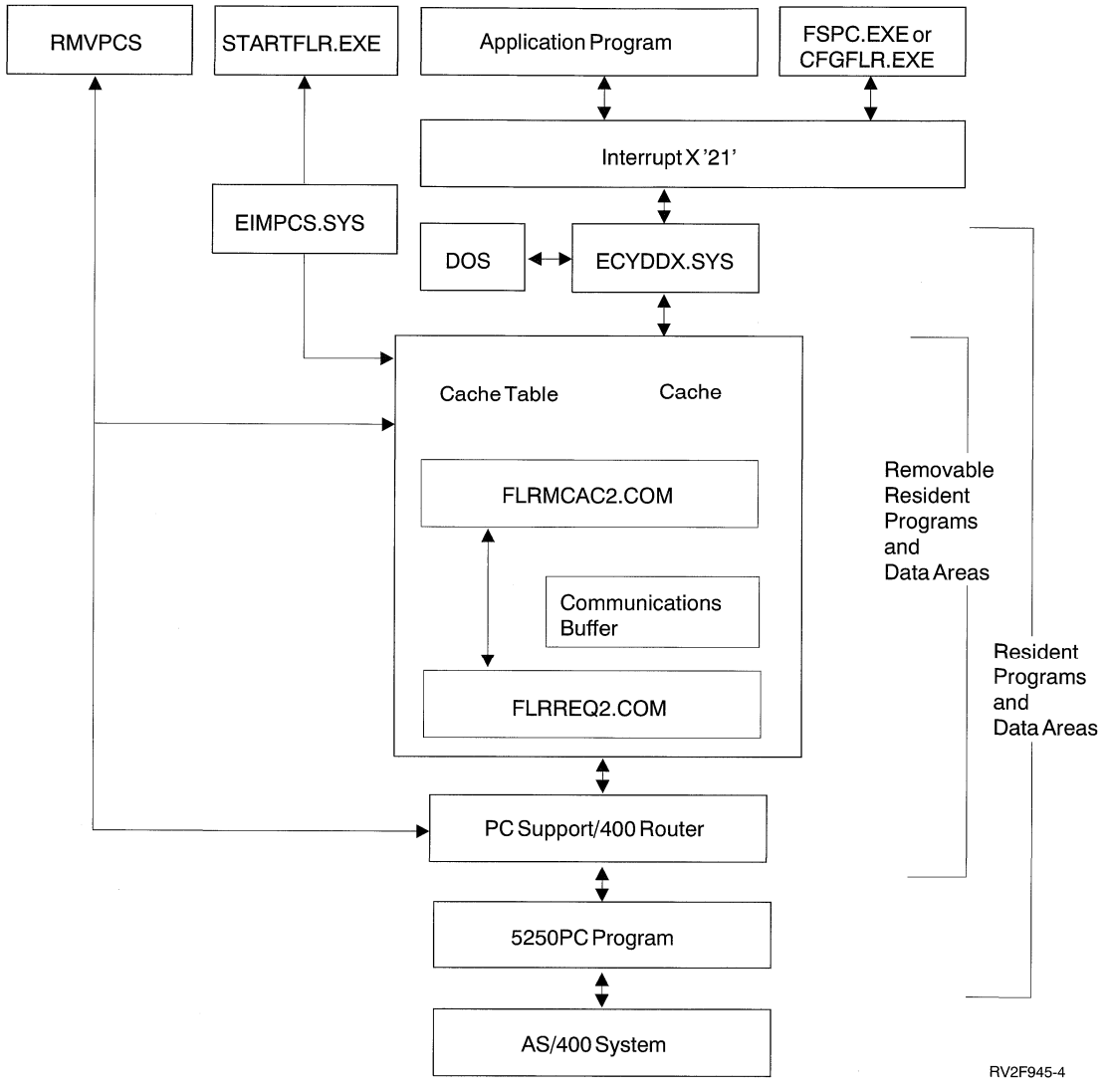


Figure 2-6. Shared Folders Function Type 2 Programs (DBCS)

ECYDDX.SYS Program: This program is a device driver which is loaded and becomes resident when the personal computer is started. It intercepts all interrupt 21 (hex) requests issued by application programs and directs them to the DOS operating system or to the shared folders type 2 programs for processing. The ECYDDX.SYS program cannot be removed from PC memory.

Note: ECYDDX.SYS is also used with shared folders type 0 and shared folders type 1.

STARTFLR.EXE Program: This program runs as a personal computer application program. It starts the shared folders function by doing the following:

- Processes the following identifiers in the CONFIG.PCS or alternative configuration file specified after the STARTFLR command:
  - MCAC (memory cache in conventional memory)
  - MCAE (memory cache in expanded memory)
  - FEMU (shared folders expanded memory usage)
  - FTYP (shared folders function type)
  - CBSZ (communications buffer size)

  Refer to the *PC Support/400 DOS Installation and Administration Guide* for information on these identifiers.
- Loads the FLRREQ2.COM program in conventional or expanded memory as specified by the FEMU identifier.
- Creates a communications buffer of the size specified by the CBSZ identifier and locates it in conventional or expanded memory as specified by the FEMU identifier.

If caching is to be used (specified by the MCAC or MCAE identifiers), STARTFLR.EXE also does the following:

- Loads the FLRMCAC2.COM program in conventional or expanded memory as specified by the FEMU identifier.
- Creates cache tables and a cache according to the cache size specified by the MCAC or MCAE identifiers, and locates them in conventional or expanded memory as specified by the FEMU identifier.

The STARTFLR.EXE program is normally run from the AUTOEXEC.BAT file or the STARTPCS.BAT file. It must be run before the first folder can be assigned.

EIMPCS.SYS Program: This program is used by STARTFLR to allocate and initialize the shared folders programs and memory cache. The FLRREQ and FLRMCAC programs use EIMPCS.SYS to transfer operation control between conventional segments of code and expanded memory segments of code and to access expanded memory segments of data (application buffers, cache tables and buffers, and the communications buffer).

FLRREQ2.COM and FLRMCAC2.COM Programs: FLRREQ2.COM receives requests for file input and output from ECYDDX.SYS and passes these requests to the AS/400 system through the router. FLRREQ2.COM returns control information and data back to ECYDDX.SYS.

When using caching, FLRREQ2.COM works with FLRMCAC2.COM to cache data that is sent and received from the AS/400 system. FLRMCAC2.COM controls the cache and maintains the cache tables.

FSPC.EXE Program: This program runs as a personal computer application program. It works with the other programs to provide the user interface for shared folders type 2.

Note: FSPC.EXE provides the same user interface for shared folders type 0 and shared folders type 1.

CFGFLR.EXE Program: This program runs as a PC application program. It can be used instead of FSPC.EXE to assign and release shared folder drives.

RMVPCS.EXE Program: This program runs as a personal computer application program. It ends the shared folders type 1 or shared folders type 2 function and removes the FLRREQ1.COM or FLRREQ2.COM programs and the FLRMCAC1.COM or FLRMCAC2.COM programs, the communications buffer, cache tables, and cache from personal computer memory.

Processing Information about Shared Folders Type 2 Requests

The purpose of this section is to provide information to help you when shared folders type 2 ends unexpectedly. Typically, the message 5054 (DOS) or the message 5990 (OS/2) is displayed.

The AS/400 system handles shared folders type 2 requests from the personal computer differently from the shared folders type 0 and 1 requests. Shared folders type 0 and 1 requests are handled by the OS/400 program. Type 2 requests are handled by the PC Support/400 program. The following paragraphs discuss shared folders type 2 processing on the AS/400 system and error conditions that may occur with shared folders type 2.

Shared folders type 2 jobs run in their own subsystem. The subsystem, QXFPCS, is activated either manually with the Start Subsystem (STRSBS) command or automatically by the QXFINIT module when the first allocate request is received from a personal computer. QXFAUTO is an autostart job for subsystem QXFPCS that must be active with the subsystem for successful initiation of shared folders jobs. Jobs are routed to the QXFPCS subsystem through the PGMEVOKE routing entry in the QCMN routing table, or in another, user-defined, communications subsystem routing table.

The major modules in shared folders type 2 on the AS/400 system are:

Module	Action
QXFINIT	Handles startup.
QXFSERV	Performs server processing tasks.
QXFLSTFA	Handles personal computer requests to the AS/400 system for file attributes, such as the DIR command.

General Problem-Solving Suggestions for Shared Folders Type 2

Ensure that all parts of shared folders type 2 have started successfully, including subsystem QXFPCS for user QSYS, job QXFAUTO for user QSYSOPR, and QXFSERV for your user ID.

Stopping and starting the QXFPCS subsystem again may resolve the initialization problem. Such a problem may result from a PTF being applied when the subsystem was still active.

The version and release levels of the various programs on the system are found by selecting option 10 on the Work with Licensed Programs (LICPGM) menu. The same release should be shown for the OS/400 program and for the other licensed programs. If a program is not installed correctly, *ERR is shown in the *Installed Release* column.

When an unusual condition is encountered by the AS/400 system, at least one message is sent to both the job log and the history log. The message indicates the cause of a host system error. Finding any message sent to a log is most important in isolating the cause of the problem. Error messages in the job log on the AS/400 system include module names. When you find a shared folders type 2 module name, read the section below that discusses the module.

QXFINIT

The XPF server initialization program (QXFINIT) is started whenever a personal computer sends an allocate request to the AS/400 system. First, it starts subsystem QXFPCS if that subsystem is not already active. Then, the initialization program either starts the XPF server job QXFSERV or sends conversation-related information to a previously started XPF server job. After the server job is active and allocates the conversation, the initialization program ends.

The QXFINIT module is started through work management when a personal computer sends an allocate request to an APPC network. A message is enqueued to this program's machine interface (MI) response queue when the system has successfully started the conversation. If the XPF server does not start successfully, then this initialization program starts a time-out when it leaves the queue and ends.

When initialization processing fails, the message 5054 (DOS) or the message 5990 (OS/2) is displayed. The exact cause of the error can only be found on the AS/400 system, in either the job log or the history log. The following messages are logged by QXFINIT. If you have considered the "General Problem-Solving Suggestions for Shared Folders Type 2" on page 2-13 and the problem still cannot be solved, you should contact your IBM service representative or remarketer.

Message Number	Text
CPD0032	User not authorized to command STRSBS.
CPF1144	Cannot allocate job queue. Job queue may not exist.
CPF2182	User not authorized to library QIWS.
MCH1001	User not authorized to system object QXFPCS or QXFAUTO.
MCH5801	Dequeue operation timeout. Maximum jobs on job queue reached or job queue held.
IWS6047	Could not start subsystem QXFPCS. See related CPFxxxx message.
IWS6048	Dequeue operation time-out. Maximum jobs on job queue reached or job queue held. Could not start QXFAUTO job.
IWS6013	Look for another message indicating the specific problem.

QXFSERV

The QXFSERV module is started through the submit job command issued by the XPF server initialization program, QXFINIT, when the personal computer allocates its first APPC conversation. The XPF server first allocates the APPC conversation. It then continues to receive data from APPC/FM until a change direction indication is received. This indicates that APPC/FM has no more data for that conversation in its buffer. The data received from APPC/FM is the exchange server attributes request, chained to a list file attributes request.

When this data is received, the server initializes its data area, performs processing required to support user exit programs, and processes the exchange server attributes request and the list file attributes request. Then, it waits on the request queue for more work to do. When all conversations are deallocated, the job ends. When an error is detected, diagnostic messages are sent to the job log and an escape message is shown.

The following messages are logged by QXFSERV. If you have considered the "General Problem-Solving Suggestions for Shared Folders Type 2" on page 2-13 and the problem still cannot be solved, you should contact your IBM service representative or remarketer.

Message Number	Text
IWS6012	Server job ended abnormally.
IWS6041	Internal error occurred in the server job.
IWS6046	Incorrect code point sent by DDM. Possible incompatible releases on personal computer and AS/400 system.
IWS6049	Allocate of APPC/FM conversation failed.
IWS6051	Communication failure.
IWS6052	Internal error. The contents were sent to main or auxiliary storage.
IWS6053	Error returned by server.

The following messages are caused by problems in, or associated with, a user exit program:

Message Number	Text
IWS9801	Requested function not allowed by user exit program.
IWS9802	Exit program could not be found.
IWS9803	User not authorized to the exit program or the library of the exit program.
IWS9804	Exit program detected an error.
IWS9806	Host system does not allow requested function (PCSACC = *REJECT).

QXFLSTFA

The QXFLSTFA module is called by the QXFSERV module to process the list file attributes request received from the PC requester.

There are four types of errors that can occur:

- The first type includes errors such as `Storage limit exceeded` and `Folder not found`. QXFLSTFA is notified of these errors by signalling an exception. An error reply indicating the error is built and sent to the PC requester.
- The second type of error is just like the first type, except that the text of the error message is sent to the job log. These errors are more severe than the first type, and the message text is sent to the job log to help you find the reason that the PC request failed. The recovery may involve taking some action on the host system. Examples of these errors are: `Not authorized to folder`, `Damaged folder`, and `Folder needs to be reclaimed`.

- The third type of error occurs when closing the folder. This error is not reported back to the PC requester by an error message because the error occurs after the file attribute list objects have already been sent to the PC requester, indicating the request was successfully processed. The errors that could occur when closing the folder are severe errors, and most often occur when the shared folders XPF server communicates to folder management services (FMS) again on the next request from the PC requester (if there is one). The message text associated with the error may be sent to the job log.
- The fourth type of error is the function check. If the function check occurred while the user's exit program was running, the USEXFCHK exception handler receives control. The message text of the exception is sent to the job log. An error reply containing the `Not authorized to command` reply message is sent to the PC requester. Then, an escape message is sent to the caller indicating that a function check occurred in the user exit program. Function checks should not occur at any other time. The message text for the error is sent to the job log. Then, escape message IWS60FF is sent to the caller indicating an internal error occurred during list file attributes processing.

The following messages are logged by the QXFLSTFA module. If you have considered the "General Problem-Solving Suggestions for Shared Folders Type 2" on page 2-13 and the problem still exists, you should call your IBM service representative or remarketer.

Message Number	Text
IWS6053	Unexpected error occurred while communicating to the Licensed Internal Code server.
IWS60FF	Internal error Reason code &1 occurred.
IWS6041	Internal error occurred in the server job.
IWS9801	User exit program does not allow the requested function.
IWS9804	An error was detected in the user exit program.

Shared Folders Function Using Extended DOS

This section discusses the extended DOS shared folders function.

Use this set of programs when you want to run PC Support in extended DOS mode. Extended DOS shared folders function can be used only on personal computers having a 286 processor or above with extended memory, and only with an AS/400 system that is using Version 2 Release 1.0 or later. It requires less conventional memory than any of the DOS shared folders types, while keeping the memory cache and performance benefits of shared folders type 2 and enabling additional functions, such as Check Out/Check In.

The EIMPCS program is not required for the extended DOS version of the shared folders function.

Note: Extended DOS support allows certain PC Support functions to use extended memory above 1MB (MB equals 1 048 576 bytes), thus making more of the conventional memory available for running other DOS programs. Extended DOS support can be used on personal computers that have an 80286 processor or above and at least 384KB (KB equals 1024 bytes) of extended memory. Most 80286 and above systems with 1MB of memory have 384KB of extended memory available.

Refer to the *PC Support/400 DOS Installation and Administration Guide* for information on installing the shared folders function program in extended DOS, for information on how to configure shared folders, and for the memory requirements of your personal computer.

Shared Folders Function Programs

Shared folders function in extended DOS consists of the following programs used in addition to the router, the extended memory interface, and the Remove PC Support command:

- SFNC.OVL
- STARTFLR.EXE
- FLRREQ.OVL
- FSPC.EXE
- CFGFLR.EXE

These programs communicate with the router, the extended memory manager (PCSXI.EXE), and the Remove PC Support (RMVPCS.EXE) command.

SFNC.OVL Program: This program replaces the ECYDDX.SYS device driver. It intercepts all interrupt 21 (hex) requests issued by application programs and directs them to the DOS operating system or to the shared folders function programs for processing. The shared folders network connector (SFNC.OVL) allows shared folders to connect to other networks in a manner consistent with other applications.

As a DOS application, rather than a device driver like the ECYDDX.SYS, the SFNC.OVL program has these advantages:

- SFNC.OVL can be removed from memory.
- Drive letters do not need to be consecutive.
- Up to 26 drives can be assigned.
- The drives are not interpreted by DOS or Microsoft Windows as local drives.
- DOS redirection calls are supported.

The SFNC.OVL program supports the DOS redirection calls to allow network applications to assign, release, and query shared folders drives like any other network drive. It allows any application to redirect any drive, including local drives (for example, A:, B:, C:) to shared folders. The SFNC.OVL program can support all of the existing shared folders IOCTL APIs; however, this will continue the incompatibilities that currently exist between shared folders and other products that support network drives such as Novell** NetWare** 386.

Current PC Support/400 applications that use the shared folders function continue to work as they did before. The FSPC.EXE and CFGFLR.EXE programs allow you to assign only eight consecutive drives. However, the DOSREDIR.EXE program in the PC Support/400 tools folder (QIWSTOOL) allows you to use the advantages of the SFNC.OVL program.

Some other products, such as CD-ROM, use the ECYDDX.SYS device driver. Since the SFNC.OVL program is intended to be used with PC Support/400 shared folders function specifically, it does not handle other products. If you are running another product that uses ECYDDX.SYS, you should leave the line in your CONFIG.SYS file, but remove the parameters specific to the shared folders function.

STARTFLR.EXE Program: This program runs as a personal computer application program. It starts the shared folders function by doing the following:

- Processes the following identifiers in the CONFIG.PCS or alternative configuration file specified after the STARTFLR command:
  - CBSZ (communications buffer size)
  - MCAX (memory cache in extended memory)

  Refer to the *PC Support/400 DOS Installation and Administration Guide* for information on these identifiers.
- Loads the FLRREQ.OVL program in extended memory.
- Loads the SFNC.OVL program in extended memory.
- Creates the communications buffer. The communications buffer is created in extended memory when the SDLC or ASYNC router is being used. If the token-ring or twinaxial router is being used, or if there is not enough extended memory, the communications buffer is created in conventional memory.

If caching is to be used (specified by the MCAX identifier), STARTFLR.EXE allocates cache tables and a cache according to the cache size specified by the MCAX identifier and locates them in extended memory. If there is not enough extended memory for the cache and cache tables, no caching is performed.

The STARTFLR.EXE program is normally run from the AUTOEXEC.BAT file or the STARTPCS.BAT file. It must be run before the first folder can be assigned.

Note: The PCSXI.EXE program must be started prior to running STARTFLR.EXE. PCSXI.EXE provides the extended memory management function for PC Support.

FLRREQ.OVL Program: FLRREQ.OVL receives requests for file input and output from SFNC.OVL and passes these requests to the AS/400 system through the router. FLRREQ.OVL returns control information and data back to SFNC.OVL.

When using caching, FLRREQ.OVL caches data that is sent and received from the AS/400 system. FLRREQ.OVL controls the cache and maintains the cache tables.

FSPC.EXE Program: This program runs as a personal computer application program. It works with the other programs to provide the user interface for shared folders function. This program allows you to assign or release drives through a batch command interface or an interactive display interface.

CFGFLR.EXE Program: This program runs as a personal computer application program. It can be used instead of FSPC.EXE to assign and release shared folder drives.

RMVPCS.EXE Program: This program runs as a personal computer application program. It ends the shared folders function and removes the FLRREQ.OVL and SFNC.OVL programs, the communications buffer, cache tables, and cache from personal computer memory.

Using DOS Commands with the Shared Folders Function

Shared folders function supports the DOS commands that exclusively use the file management function calls listed in Figure 2-8 on page 2-22 to perform file-oriented functions.

ASSIGN The ASSIGN command reroutes requests from one drive to another drive. For example, you can use the ASSIGN command to reroute requests for your A diskette drive to the E drive. Continue to use this command if you are **not** using the extended DOS version of the shared folders function.

If you are using the extended DOS version of the shared folders function, the ASSIGN command cannot be used. Instead, use the DOSREDIR program to assign any of the 26 drive letters as shared folders drives. The format for the DOSREDIR.EXE program is:

```
DOSREDIR ASSIGN d:  //hostname
```

where `d` is the drive to be redirected and `hostname` is the AS/400 system to which you want to redirect read and write operations. Information on how to get the DOSREDIR.EXE program can be found in "Shared Folders Function Programs" on page 2-17.

For example, if you want to reroute requests for the A diskette drive to a shared folders drive you would do one of the following:

- If you are **not** using the extended DOS version of the shared folders function, assign the shared folders drive using the FSPC.EXE or CFGFLR.EXE programs. Then, use the DOS ASSIGN command to reroute requests to that shared folders drive.
- If you are using the extended DOS version of the shared folders function, assign the A diskette drive as the shared folders drive using the DOSREDIR program.

BREAK The BREAK command, either interactive or installed in the CONFIG.SYS, enables the DOS operating system's extended CTRL-BREAK checking. With BREAK=OFF (the default condition), the DOS operating system only checks for CTRL-BREAK during standard device operations. However, because shared folders does not perform DOS standard I/O operations, BREAK must be enabled in order for CTRL-BREAK to work properly with a shared folders operation.

If BREAK is not enabled when CTRL-BREAK is issued during a shared folders operation, unpredictable results may occur. The *DOS 4.00 Command Reference* manual contains additional information about the BREAK command.

COMP The COMP command compares the contents of the first set of specified files with the contents of the second set of specified files. Because the DOS operating system leaves an open copy of each file being compared, the COMP command ends after comparing a certain number of files. (This is the number of open files specified in CONFIG.SYS ECYDDX FILES=entry.) The SFNC.OVL program uses the value for the FILES option on the DOS ECYDDX entry in the CONFIG.SYS file.

To use COMP with a shared folders drive, specify the shared folders drive as the first drive to compare. For example:

```
COMP I:\*.* C:\PCS\*.*
```

Another consideration when using the COMP command is that if COMP encounters a file to which it does not have authority while you are comparing files in a shared folders drive, COMP ends prematurely and displays the error message `Sharing violation`, and prompts the user to compare more files. This error does not occur when using DOS 5.0 operating system.

PRINT The PRINT command prints a queue of files on an output device while the user is starting other tasks on the system.

TREE If, when using DOS 4.00, you use the TREE command to display a directory path that contains a subdirectory to which you do not have authority, TREE ends prematurely and displays the error message `Invalid path`. This does not occur when using DOS 3.30.

Files The DOS operating system uses two different methods to open files. One is the standard file OPEN, the other is to open a file as a file control block (FCB). The difference between the open file methods is that the DOS operating system leaves open FCBs behind after completing an operation. The DOS operating system later closes FCBs if required, but there is no limit to the number of open FCBs that can exist at any given time.

However, shared folders treats open FCBs and open files in the same manner, and both are counted when determining how many files shared folders has open. Since the DOS operating system uses FCBs when running DOS functions, it is possible that these functions may prematurely end when performed against a shared folder because the allowable number of open files has been exceeded.

Paths When using paths to access subdirectories on a shared folders drive, the path length should not exceed 64 characters with the trailing backslash added on (63 characters without the backslash). The DOS operating system normally allows 64 characters in a path length, but when accessing a shared folders drive, an extra backslash is added at the beginning of the path, thereby making the limit 63 characters.

DOS Commands Not Supported by the Shared Folders Function

The shared folders function does not support DOS commands that:

- Perform disk- or drive-oriented functions.
- Perform direct sector I/O to a DOS formatted disk.
- Do not exclusively use interrupt 21 function calls.
- Are not supported by the shared folders function.
- Are not supported by the ECYDDX.SYS or SFNC.OVL programs. Shared folders function drives are defined as network drives. Some DOS commands are not supported using network drives.

You cannot use the DOS commands described in Figure 2-7 with shared folders function.

Figure 2-7. DOS Commands Not Supported by Shared Folders Function

Command	Description	Reasons Not Supported
CHKDSK (Check Disk)	Analyzes the directories, files, and the file allocation table on a drive.	CHKDSK cannot analyze directories, files, and file allocation tables on drives assigned to folders because directories and files are not stored in DOS format on the AS/400 system, and file allocation tables do not exist as known to DOS. Also, CHKDSK does not exclusively use DOS interrupt 21H function calls and it performs sector reads to the diskette drive.
DISKCOMP (Compare Diskettes Only)	Compares the contents of two diskettes.	DISKCOMP does not work on PC diskette drives or drives assigned to folders. Also, it does not exclusively use DOS interrupt 21H function calls, and it performs sector reads to the diskette drive.
DISKCOPY (Copy Diskette Only)	Copies the contents of one diskette to another diskette.	DISKCOPY does not work on PC diskette drives or drives assigned to folders. Also, it does not exclusively use DOS interrupt 21H function calls, and it performs sector reads to the diskette drive.
FDISK (Hard Disk Setup)	Prepares a PC hard disk for use by DOS.	FDISK does not access remote drives.
FORMAT	Initializes the disk in a specified drive.	FORMAT is not appropriate for drives assigned to folders because disk I/O is controlled by the AS/400 system.
JOIN	Logically connects a drive to a directory on another drive to produce a single directory structure.	JOIN is not supported by the ECYDDX device driver or the SFNC.OVL program.
LABEL (Volume Label)	Allows you to create, change, or delete a volume label on a disk.	LABEL is not supported for folder drives, since the volume labels of a drive assigned to a folder are constant. System drive labels are the system name or the alias of the host system. Directory drive labels are always the lowest level subdirectory name entered on the shared folder assign command. For example, if the assign command is `FSPC ASSIGN L: ABLE/BAKER/CHARLIE` the volume label appears as `CHARLIE`.
RECOVER	Recovers files from disks with defective sectors.	RECOVER does not exclusively use DOS interrupt X'21' function calls, and it performs sector reads to the diskette drive.
SUBST (Substitute)	Allows you to use a different drive specifier to refer to another drive or path.	SUBST is not supported by the ECYDDX device driver or the SFNC.OVL program.
SYS (System)	Transfers the operating system files IBMDOS.COM and IBMBIO.COM from one drive to another.	Because of DOS restrictions on where IBMDOS.COM and IBMBIO.COM files may be placed on a disk, SYS must use sector I/O on the drive. Since a personal computer cannot be started from a drive assigned to a folder, you cannot place the operating system files on these drives.

Using DOS Interrupt 21 Function Calls

The shared folders function works only with those function calls that deal with file management. Therefore, device driver extensions (ECYDDX.SYS) and the SFNC.OVL program supply file level redirection capability. These programs allow DOS applications to continue to use the DOS interrupt X'21' functions set, while the file requests are serviced without using the local DOS file system. File level redirection bypasses the logical and physical parts of the local DOS file system. In contrast, device drivers bypass only the physical parts.

File Management Function Calls for DOS Users

The function calls listed in Figure 2-8 help you manage files.

Refer to the *DOS Technical Reference* manual for more information about the function calls listed in Figure 2-8.

Figure 2-8 (Page 1 of 3). DOS Shared Folders Function File Management Function Calls

21H Function	Function Name	Description
01H[1]	Keyboard Input	Reads one character from the file when the standard input device is redirected to come from a file.
02H[1]	Display Output	Writes one character to the file when the standard input device is redirected to go to a file.
06H[1]	Direct Console I/O	Reads one character from the file when the standard input device is redirected to come from a file.
07H[1]	Direct Console Input Without Echo	Reads one character from the file when the standard input device is redirected to come from a file.
08H[1]	Console Input Without Echo	Reads one character from the file when the standard input device is redirected to come from a file.
09H[1]	Print String	Writes a string of characters to the file when the standard output device is redirected to go to a file.
0AH[1]	Buffered Keyboard Input	Reads a specified number of characters from the file when the standard input device is redirected to come from a file.
0BH[1]	Check Standard Input Status	Checks if the end of the file has been reached when the standard input device is redirected to come from a file.
0CH[1]	Clear Keyboard Buffer and Invoke a Keyboard Function	Starts function call 01, 06, 07, 08, or 0A when the standard input device is redirected to come from a file.
0DH	Disk Reset	Writes file buffers that have been changed to the disk.
0EH[2]	Select Disk	Selects the drive specified as the default drive.
0FH	Open File	Opens an existing file whose file name is specified in the Function Control Block (FCB) in compatibility sharing mode. The existing file can only be opened by a single process.
10H	Close File	Closes a file opened by the 0FH function call.
11H	Search for First Entry	Searches the current directory for the first file name matching the generic file name specified in the FCB.
12H	Search for Next Entry	Searches the current directory for the next file name matching the generic file name specified in the FCB on the 11H function call.
13H	Delete File	Deletes files in the current directory whose names match the generic file name specified in the FCB.
14H	Sequential Read	Reads the next record from the file, using location information specified in the FCB.
15H	Sequential Write	Writes a record to the next location in the file, using information in the FCB.
16H	Create File	Creates an empty file, whose file name is specified in the FCB, and opens it using function call 0FH.

1 This function call performs file input and output when input or output is redirected to or from a file.
2 This function call requires little processing time because it does not require AS/400 processing.
3 This function call is not supported by the ECYDDX.SYS program.

Figure 2-8 (Page 2 of 3). DOS Shared Folders Function File Management Function Calls

21H Function	Function Name	Description
17H	Rename File	Renames files in the current directory whose names match the generic file name specified in the FCB.
19H[2]	Current Disk	Determines the current default drive.
1AH[2]	Set Disk Transfer Address	Sets the disk transfer address to registers DS:DX. DDX and DOS record the user's supplied data.
1BH[2]	Allocation Table Information	Returns information about the allocation table for the default drive.
1CH	Allocation Table Information for Specific Device	Returns allocation table information for a specific device.
21H	Random Read	Reads the record addressed by the FCB's current block and the current record fields into memory at the current disk transfer address.
22H	Random Write	Writes the record addressed by the FCB's current block and the current record fields from the current disk transfer address.
23H	File Size	Searches the disk directory for an entry matching the specified file and sets the FCB's random record field to the number of records in the file.
24H[2]	Set Relative Record Field	Sets the FCB's random record field to the same file address as the current block and record fields.
27H	Random Block Read	Reads the specified number of records (in terms of the record size field) from the file address specified by the random record field into the disk transfer address.
28H	Random Block Write	Writes the specified number of records from the disk transfer address into the file address specified by the FCB's random record field.
2EH[2]	Set/Reset Verify Switch	Sets the verify switch.
2FH[2]	Get Disk Transfer Address	Returns the current disk transfer address.
36H	Get Disk Free Space	Returns the disk free space (available clusters, clusters-per-drive, and bytes-per-sector).
39H	Create Subdirectory (MKDIR)	Creates the specified directory.
3AH	Remove Subdirectory (RMDIR)	Removes the specified directory.
3BH	Change the Current Directory (CHGDIR)	Changes the current directory to the specified directory.
3CH	Create a File (CREAT)	Creates a new file or truncates an old file to zero length in preparation for writing.
3DH	Open a File	Opens the specified file.
3EH	Close a File Handle	Closes the specified file handle.
3FH	Read from a File or Device	Transfers the specified number of bytes from a file into a buffer.
40H	Write to a File or Device	Transfers the specified number of bytes from a buffer into a specified file.

1 This function call performs file input and output when input or output is redirected to or from a file.
2 This function call requires little processing time because it does not require AS/400 processing.
3 This function call is not supported by the ECYDDX.SYS program.

Figure 2-8 (Page 3 of 3). DOS Shared Folders Function File Management Function Calls

21H Function	Function Name	Description
41H	Delete a File from a Specified Directory (UNLINK)	Removes a directory entry associated with a file name.
42H[2]	Move File Read Write Pointer (LSEEK)	Moves the read/write pointer according to the specified method.
43H	Change File Mode (CHMOD)	Changes the file mode of the specified mode.
44H	I/O Control for Devices (IOCTL)	Sets or gets device information associated with open device handlers, or sends or receives control strings to or from the device handle.
45H[2]	Duplicate a File Handle (DUP)	Returns a new file handle for an open file that refers to the same file at the same position.
46H[2]	Force a Duplicate of a Handle (FORCEDUP)	Forces the handle in register CX to refer to the same file at the same position as the handle in register BX.
47H[2]	Get Current Directory	Places the full path name (starting from the root directory) of the current directory for the specified drive in the area pointed to by registers DS:SI.
4BH	Load or Start a Program (EXEC)	Allows a program to load another program into memory and optionally begins running it.
4EH	Find First Matching File (FIND FIRST)	Finds the first file name matching the specified file specifications.
4FH	Find Next Matching File (FIND NEXT)	Finds the next directory entry matching the name specified on the previous Find First or Find Next function call.
54H[2]	Get Verify Setting	Returns the value of the verify flag.
56H	Rename a File	Renames the specified file.
57H[2]	Get/Set a File's Data and Time	Gets or sets a file's data and time.
5AH	Create Unique File	Generates a unique file name, and creates that file in the specified directory.
5BH	Create New File	Creates a new file.
5C00H	Lock File Access	Locks a range of bytes in an opened file.
5C01H	Unlock File Access	Unlocks a range of bytes in an opened file.
5F02	Query	Queries a network device.
5F03 [3]	Assign	Assigns a network device.
5F04 [3]	Release	Releases a network device.
68H	Commit File	Causes all buffered data for a file to be written to the device. You can use this function instead of the close-open sequence.
6CH	Extended Open	Optionally opens and creates a file.

1 This function call performs file input and output when input or output is redirected to or from a file.
2 This function call requires little processing time because it does not require AS/400 processing.
3 This function call is not supported by the ECYDDX.SYS program.

Improving File Management When Using the DOS Operating System

File management for applications that use the shared folders function should be written to maximize performance. Almost all file requests for folders (open, read, write, and so on) are sent to the AS/400 system for processing, which requires time and resources. These same file requests may not require disk accesses when using a diskette or a hard file.

Improve file management performance by using the following suggestions:

- Use an appropriate create or open operation when creating or opening a file, instead of searching for the file first to verify its existence. A file search requires significant time and resources relative to other file operations.
- Design an application to open a file once, perform all necessary operations, and then close the file.
- Do not write with verification unless absolutely necessary. Write buffering is not used in the program when verification is on.
- Use the change file mode function call instead of a search first or find first to get file attributes after a file is open. This reduces the necessity of an AS/400 request.
- Read data from a file from the beginning to end, and not vice-versa. Reading from the beginning to the end uses the program buffer storage more efficiently.
- Perform reads and writes sequentially, rather than randomly. Sequential reads and writes use the program buffer storage more efficiently.
- Read small files (such as a profile) entirely into storage. Storage accesses are faster than disk or AS/400 accesses.

Shared Folders Function Type 0 Caching

The shared folders function type 0 device driver (FSDD.SYS) contains a variable length read/write cache buffer from 0 to 63 920 bytes. Specify the buffer size in the CONFIG.SYS file using the B=nn parameter on the DEVICE=FSDD.SYS statement. The values allowed for nn are 0 to 34. The default value is B=2, or 3760 bytes. To change the cache buffer size, specify the number of 1880-byte blocks of storage to be allocated to caching. Specify a value of 0 to disable caching.

The cache buffer reduces the number of read and write operations that go to the AS/400 system. This improves the performance of any personal computer application that accesses files in a folder. The cache buffer holds file data that can satisfy a future file read operation if the file is processed in a serial manner. It also temporarily holds data written to a file for values of B that are greater than one. Two files can use the cache buffer at the same time. When two files use the buffer, the effective cache size for each file is a portion of the buffer size. The cache buffer is only used in file-sharing environments that allow buffering on the personal computer. These environments are:

- Reading from or writing to a file using a compatibility file-sharing mode
- Reading from or writing to a file using a deny read/write (exclusive) file-sharing mode
- Reading from a file using a deny-write file-sharing mode with a read-access intent

You can use one of the following methods for determining an appropriate buffer size:

- Use a buffer of about 13KB (KB equals 1024 bytes) (B=7) as a starting point. You can then vary the cache buffer size based on the amount of remaining memory available on the personal computer. As a general rule, a larger buffer provides better performance. However, for some applications, additional data moved to the personal computer on read operations cannot be used, and performance can be adversely affected by a large buffer.
- Start with a cache buffer size that is at least four times the size of the most prevalent read/write operations performed by your personal computer applications, if you can determine this amount. Any increases should be incremented by even multiples of that amount.

Shared Folders Function Type 1 and Type 2 Caching

The shared folders type 1 and type 2 functions use caching techniques that improve performance over the shared folders type 0 caching technique.

The objective of the shared folders function type 1 and type 2 caching support is to reduce the number of requests made to the AS/400 system to read and write data. This is accomplished by using personal computer memory to provide efficient buffering of read/write data and to store that data for reuse at a later time.

Reads to files being accessed sequentially are buffered. More data is read from the AS/400 system than is actually requested by the personal computer application. This extra data is stored in the cache so that when the application requests it, it can be read directly from the cache, rather than having to request it from the AS/400 system.

Writes to files that are being accessed sequentially are buffered. This means that data written by the personal computer application is collected in the cache and sent to the AS/400 as a single group.

Reads to files being accessed randomly are stored in the cache to improve the chances that subsequent reads and writes can refer to it.

Many files (both open and closed) can have data stored in the cache for buffering and reuse purposes. Large caches can support larger buffers, store more data for reuse, and generally provide greater performance benefit than smaller caches. The data in the cache is maintained using a least-recently-used technique. Data that has not been used for a while is discarded from the cache to make room for data that is being used.

File Sharing Using the DOS Operating System

When opening a file, you must inform the DOS operating system what operations other personal computers may perform on the file being opened (sharing operation). The DOS default (compatibility operation) denies all other computers in a network access to the file. Shared folders allow other read accesses if that file is currently being opened only with read access.

Similarly, you must specify what operations your personal computer performs (access intent). The allowable access intents are read, write, and read/write. The DOS operating system default access intent (read/write) causes the open request

to fail if another personal computer has the file opened with any sharing operation other than deny-none. However, if you only intend to read from the file, you cannot open the file unless all other personal computers have specified deny-none or deny-write (therefore increasing access to the file). File sharing requires cooperation that is communicated through the sharing operation and access intent.

Compatibility Operation

Under the DOS operating system and shared folders type 2, a file can be opened any number of times in a compatibility operation by a single personal computer, if the file is not currently open under any of the other four file sharing operations.

Note: Compatibility operation is a sharing mode that is meant to be compatible with the default sharing mode used by files opened with function control blocks (FCBs), as opposed to file handles. Unlike handle opens, FCB opens do not allow the sharing mode to be specified. Compatibility operation is an exclusive sharing mode.

Under shared folders type 0 and type 1, a file can be opened any number of times using any sharing operation by a single personal computer.

Deny Read/Write Operation

Under the DOS operating system and shared folders type 2, other access to a file that is opened in a deny read/write operation is rejected. Access to the file is exclusive. A file currently open in this operation cannot be opened again in any sharing operation by any personal computer (including the same personal computer) until the file is closed.

Under shared folders type 0 and type 1, accesses by the same personal computer are allowed, much like the compatibility operation. Accesses by other personal computers are not allowed.

Deny-Write Operation

Under the DOS operating system and shared folders type 2, a file opened in a deny-write sharing operation prevents any other write access to the file (write or read/write) until the file is closed. An attempt to open a file in a deny-write operation is not successful unless the file is currently open with a read access.

Under shared folders type 0 and type 1, a file opened in a deny-write sharing operation works in the same manner, except that a compatibility operation open with a read intent and other opens by the same personal computer are also allowed.

Deny-Read Operation

Under the DOS operating system and shared folders type 2, a file opened in a deny-read sharing operation prevents any other read sharing access opens to the file (read or read/write) until the file is closed. An attempt to open the file in a deny-read sharing operation is unsuccessful if the file is currently open in compatibility operation or with a read access.

Under shared folders type 0 and type 1, a file opened in a deny-read sharing operation is treated the same as the deny-none sharing operation. Again, a compatibility operation open with a read intent and other opens by the same personal computer are allowed.

Deny-None Operation

Under the DOS operating system and shared folders type 2, a file opened in a deny-none operation places no restrictions on the read/write accessibility of the file. An attempt to open a file in a deny-none operation is not successful if the file has already been opened in compatibility mode.

Under shared folders type 0 and type 1, this access operation is handled in the same manner as the DOS operating system, except that shared folders allows another compatibility operation with a read intent and also allows other opens by the same personal computer.

Specifying Access Intent

On the following figures, the left margin indicates how the file was initially opened. To determine what the result will be on a subsequent open of the same file by an application on the same personal computer (single personal computer figures) or a different personal computer (multiple personal computers figures), select the appropriate column for the subsequent open from the top margin and find the intersection with the way the file was initially opened from the left margin. The figures list all combinations of access intents and sharing operations.

Refer to the *DOS Technical Reference* manual for a similar table when the files are being opened using the DOS operating system.

Type 0 and Type 1 Single Personal Computer — Read-Only File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	1	A	A	1	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DRW	I	1	A	A	1	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DW	I	1	A	A	1	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DR	I	1	A	A	1	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DN	I	1	A	A	1	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 2 Single Personal Computer — Read-Only File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DRW	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DW	I	S	S	S	A	A	A	1	A	A	A	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DR	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DN	I	S	S	S	A	A	A	1	A	A	A	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 0 and Type 1 Single Personal Computer — Read/Write File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		O	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		IO	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	DRW	I	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		O	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		IO	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	DW	I	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		O	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		IO	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	DR	I	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		O	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		IO	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	DN	I	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		O	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		IO	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 2 Single Personal Computer — Read/Write File

Currently Open File		File to Be Opened														
		COMP			DRW			DW			DR			DN		
		I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
COMP	I	2	2	2	A	A	A	A	A	A	A	A	A	A	A	A
	O	2	2	2	A	A	A	A	A	A	A	A	A	A	A	A
	IO	2	2	2	A	A	A	A	A	A	A	A	A	A	A	A
DRW	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
	O	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
	IO	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
DW	I	S	S	S	A	A	A	2	A	A	A	A	A	2	A	A
	O	S	S	S	A	A	A	A	A	A	2	A	A	2	A	A
	IO	S	S	S	A	A	A	A	A	A	A	A	A	2	A	A
DR	I	S	S	S	A	A	A	A	2	A	A	A	A	A	2	A
	O	S	S	S	A	A	A	A	A	A	A	2	A	A	2	A
	IO	S	S	S	A	A	A	A	A	A	A	A	A	A	2	A
DN	I	S	S	S	A	A	A	2	2	2	A	A	A	2	2	2
	O	S	S	S	A	A	A	A	A	A	2	2	2	2	2	2
	IO	S	S	S	A	A	A	A	A	A	A	A	A	2	2	2

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 0 and Type 1 Multiple Personal Computers — Read-Only File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	1	A	A	A	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DRW	I	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DW	I	1	A	A	A	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DR	I	1	A	A	A	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DN	I	1	A	A	A	A	A	1	A	A	1	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 2 Multiple Personal Computers — Read-Only File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	1	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DRW	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DW	I	S	S	S	A	A	A	1	A	A	A	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DR	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
	DN	I	S	S	S	A	A	A	1	A	A	A	A	A	1	A	A
		O	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
		IO	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 0 and Type 1 Multiple Personal Computers — Read/Write File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	2	A	A	A	A	A	2	A	A	2	A	A	2	A	A
		O	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		IO	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	DRW	I	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		O	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		IO	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
	DW	I	2	A	A	A	A	A	2	A	A	2	A	A	2	A	A
		O	A	A	A	A	A	A	A	A	A	2	A	A	2	A	A
		IO	A	A	A	A	A	A	A	A	A	2	A	A	2	A	A
	DR	I	2	A	A	A	A	A	2	2	2	2	2	2	2	2	2
		O	A	A	A	A	A	A	A	A	A	2	2	2	2	2	2
		IO	A	A	A	A	A	A	A	A	A	2	2	2	2	2	2
	DN	I	2	A	A	A	A	A	2	2	2	2	2	2	2	2	2
		O	A	A	A	A	A	A	A	A	A	2	2	2	2	2	2
		IO	A	A	A	A	A	A	A	A	A	2	2	2	2	2	2

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

Type 2 Multiple Personal Computers — Read/Write File

			File to Be Opened														
			COMP			DRW			DW			DR			DN		
			I	O	IO	I	O	IO	I	O	IO	I	O	IO	I	O	IO
Currently Open File	COMP	I	2	S	S	A	A	A	2	A	A	A	A	A	2	A	A
		O	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		IO	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
	DRW	I	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		O	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
		IO	S	S	S	A	A	A	A	A	A	A	A	A	A	A	A
	DW	I	2	S	S	A	A	A	2	A	A	A	A	A	2	A	A
		O	S	S	S	A	A	A	A	A	A	2	A	A	2	A	A
		IO	S	S	S	A	A	A	A	A	A	A	A	A	2	A	A
	DR	I	S	S	S	A	A	A	A	2	A	A	A	A	A	2	A
		O	S	S	S	A	A	A	A	A	A	A	2	A	A	2	A
		IO	S	S	S	A	A	A	A	A	A	A	A	A	A	2	A
	DN	I	2	S	S	A	A	A	2	2	2	A	A	A	2	2	2
		O	S	S	S	A	A	A	A	A	A	2	2	2	2	2	2
		IO	S	S	S	A	A	A	A	A	A	A	A	A	2	2	2

1 2nd, 3rd,... open is allowed when the file is created with the read-only attribute
2 2nd, 3rd,... open is allowed when the file is created with read/write attribute allowed
A 2nd, 3rd open is denied — access denied
S Sharing violation error — access denied on 2nd, 3rd open
N Not applicable
DRW Deny read/write operation (exclusive)
DW Deny write operation
DR Deny read operation
DN Deny none operation
I Read-only access intent
O Write-only access intent
IO Read/write access intent

DOS Directory Attributes

A DOS attribute-byte in a files-directory entry contains the following information about the file:

Archive candidate bit Indicates alteration of the file since it was last backed up.

Subdirectory bit Indicates a subdirectory. It is excluded from directory searches that do not specifically ask to have sub-directories included.

System bit Indicates a system file. The file is excluded from normal directory searches.

Hidden bit Indicates a hidden file. The file is excluded from normal directory searches.

Read-only bit Indicates that the write access to a file is not allowed. The file cannot be deleted.

Shared folders function handles the archive, subdirectory, and read-only bits in the same manner as the DOS operating system. Shared folders function ignores the system and hidden bits. Files with the system or hidden attributes set in the directory of a DOS file that resides on a personal computer disk or diskette will not have them set when they are transferred to an AS/400 folder. Once they are transferred, the files are visible on normal directory searches on the folder.

Shared Folders Function Using the OS/2 Operating System

This section discusses the OS/2 version of the PC Support/400 shared folders function.

Shared Folders Programs for OS/2 Users

The OS/2 version of the shared folders function consists of the following programs in addition to the OS/2 Communications Manager.

- FSPC.EXE
- EHNSFL0.DLL
- EHNSFL3.EXE
- UEHNSFL0.DLL
- STARTFLR.EXE
- CFGFLR.EXE
- STOPFLR.EXE

These programs communicate with the STARTRTR.EXE program and the OS/2 Communications Manager, which manages communications with the AS/400 system.

FSPC.EXE

The FSPC.EXE program is a program that runs as a personal computer application program. This program allows you to assign or release drives through a batch command interface or an interactive display interface.

EHNSFL0.DLL

The EHNSFL0.DLL program is the file system driver.

CFGFLR.EXE

This program runs as a PC application program. It can be used instead of FSPC.EXE to assign and release shared folders drives.

UEHNSFL0.DLL

The UEHNSFL0.DLL program is called when FORMAT, CHKDISK, SYS, or RECOVER is run against a shared folder drive. These commands are not supported by shared folders, so an error message is returned.

EHNSFL3.EXE

The EHNSFL3.EXE program receives requests from EHNSFL0.DLL and sends the requests to the OS/2 Communications Manager. Figure 2-9 shows the relationship between these programs.

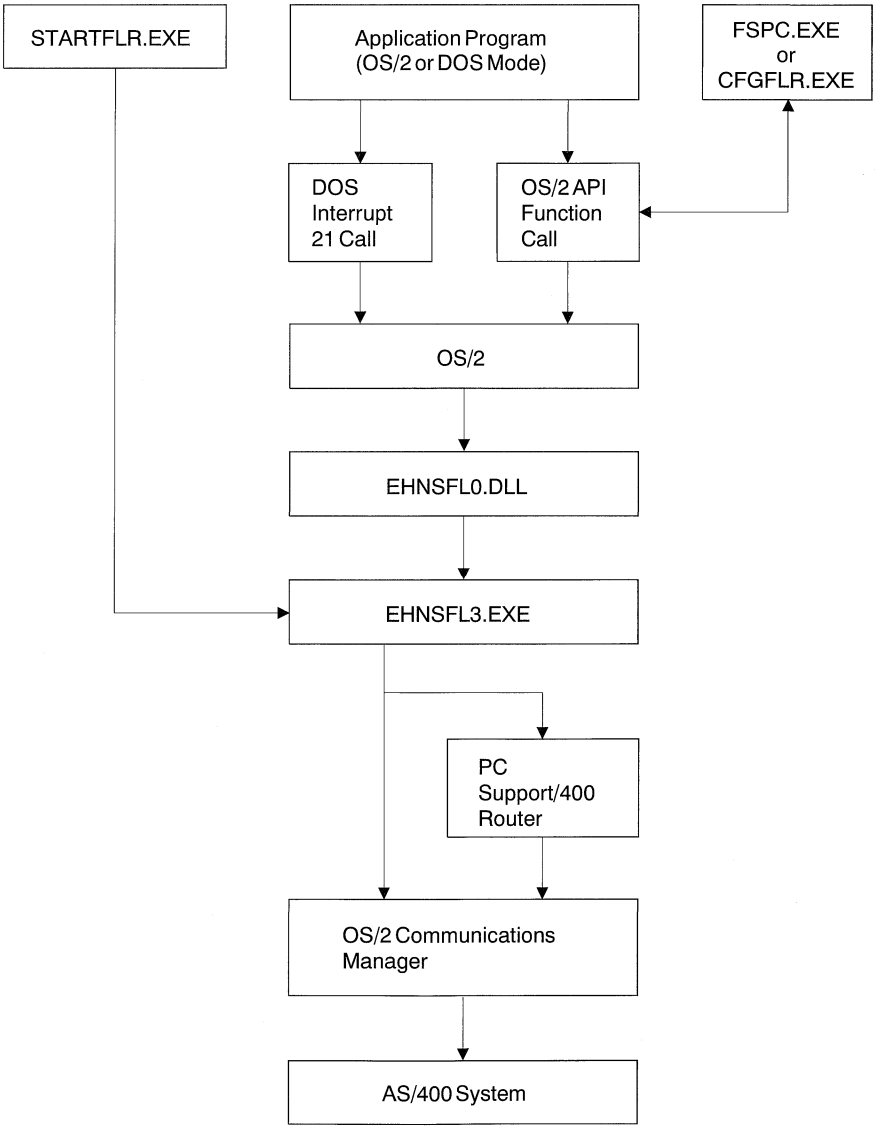


Figure 2-9. Shared Folders Function with EHNSFL3.EXE Program

STARTFLR.EXE

The STARTFLR.EXE program starts the EHNSFL3.EXE program.

For double-byte character set (DBCS), STARTFLR.EXE is placed in STARTPCS.CMD and run from there. However, if STARTFLR.EXE. is not located there, the user can detach the program using the DETACH STARTFLR command.

STOPFLR.EXE

The STOPFLR.EXE program removes shared folders from memory.

Using OS/2 Commands with the Shared Folders Function

Shared folders function supports the OS/2 commands that exclusively use the file management function calls listed in Figure 2-11 on page 2-40 to perform file-oriented functions.

Considerations

FSPC.EXE, CFGFLR.EXE, STARTFLR.EXE, and STOPFLR.EXE run only in OS/2 mode. They cannot be run in DOS mode.

OS/2 Commands Not Supported by Shared Folders Function

Shared folders function does not support OS/2 commands that:

- Perform disk- or drive-oriented functions
- Perform direct sector I/O to an OS/2 formatted disk

You cannot use the OS/2 commands shown in Figure 2-10 with shared folders function.

Figure 2-10 (Page 1 of 2). OS/2 Commands Not Supported by Shared Folders Function

Command	Description	Reasons Not Supported
BACKUP (to root of System/36 system drive)	Saves files for later retrieval.	The BACKUP command creates a directory as part of its processing. The System/36 does not support creating directories at the root level of a system drive. Because of this, the BACKUP command should not be run to the root level of a System/36 system drive.
CHKDSK (Check Disk)	Analyzes the directories, files, and the file allocation table on a drive.	CHKDSK cannot analyze directories, files, and file allocation tables on drives assigned to folders because directories and files are not stored in OS/2 format on the AS/400 system, and file allocation tables do not exist as known to the OS/2 operating system. Also, CHKDSK performs sector reads on the diskette drive.
DISKCOMP (Compare Diskettes Only)	Compares the contents of two diskettes.	DISKCOMP does not work on drives assigned to folders. Also, DISKCOMP performs sector reads on the diskette drive.
DISKCOPY (Copy Diskette Only)	Copies the contents of one diskette to another diskette.	DISKCOPY does not work on drives assigned to folders. Also, DISKCOPY performs sector reads on the diskette drive.

[1] DBCS data only.

Figure 2-10 (Page 2 of 2). OS/2 Commands Not Supported by Shared Folders Function

Command	Description	Reasons Not Supported
FDISK [1] (Hard Disk Setup)	Prepares a PC hard disk for use by the OS/2 operating system.	FDISK does not access remote drives.
FDISKPM (Hard Disk Setup)	Prepares a PC hard disk for use by the OS/2 operating system.	FDISKPM does not access remote drives.
FORMAT	Initializes the disk in a specified drive.	FORMAT is not appropriate for drives assigned to folders since disk I/O is controlled by the AS/400 system.
JOIN	Logically connects a drive to a directory on another drive to produce a single directory structure.	JOIN is not directly supported by the OS/2 operating system, but is supported in the OS/2 DOS compatibility mode.
RECOVER	Recovers files from disks with defective sectors.	RECOVER performs sector reads on the diskette drive.
RENAME (Renaming Folders)	Renaming folders when connected to System/36 is not supported.	System/36 does not support the renaming of folders. Files can be renamed.
SUBST (Substitute)	Allows you to use a different drive specifier to refer to another drive or path.	SUBST is not directly supported by the OS/2 operating system, but is supported in the OS/2 DOS compatibility mode.
SYS (System)	Transfers the OS/2 hidden system files from one drive to another.	Because of OS/2 restrictions on where operating system files may be placed on a disk, SYS must use sector I/O on the drive. Since a personal computer cannot be started from a drive assigned to a folder, you cannot place the operating system files on these drives.

[1] DBCS data only.

Using OS/2 Application Program Interface Function Calls

Shared folders function processes function calls that deal with file management. It uses the OS/2 Installable File System (IFS) to supply file level redirection. Every application call comes through the OS/2 operating system and, if appropriate, is redirected to the file system that services the request. The OS/2 operating system then keeps the file system transparent to the application program.

File Management Function Calls for OS/2 Users

The OS/2 file management function calls listed in Figure 2-11 help you manage files. The table notes whether the function call is supported by the shared folders function.

The *OS/2 Programming Guide* contains additional information about the function calls listed in Figure 2-11.

Figure 2-11 (Page 1 of 2). OS/2 Shared Folders File Management Function Calls

Function Call	Description
DOSBufReset	Flushes file buffers.
DOSChDir	Changes the current directory.
DOSChgFilePtr	Moves the file read/write pointer.
DOSClose	Closes the file handle.
DOSCopy	Copies the file.
DOSDelete	Deletes a file.
DOSDevIOCTL [1]	IO Control for devices.
DOSDupHandle	Duplicates a file handle.
DOSExit	Starts the exit process.
DOSFileIO [4]	Used for a multifunction call.
DOSFileLocks	Locks or unlocks a range of bytes in an opened file.
DOSFindClose	Ends directory search handle usage.
DOSFindFirst [2]	Finds the first matching file.
DOSFindNext [2]	Finds the next matching file.
DOSFSAttach	Attaches a drive to a file system driver (FSD).
DOSFsCtl [3]	File control system.
DOSMkDir	Makes a subdirectory.
DOSMove	Moves or renames a file or subdirectory.
DOSNewSize	Changes the size of a file.
DOSOpen	Opens or creates a file.
DOSQCurDir	Queries the current directory.
DOSQCurDisk	Queries the current default drive.
DOSQFHandstate	Queries the file handle state information.
DOSQFileInfo [2]	Queries the file handle information.
DOSQFileMode	Queries the file mode.
DOSQFSAttach	Queries the attached FSD information.
DOSQFSInfo	Queries the file system information.
DOSQHandtype	Queries the file handle type information.
DOSQPathInfo [2]	Queries the path information.
DOSQVerify	Queries the verify setting.
DOSRead	Reads from a file.
DOSReadAsync	Reads from a file asynchronously.
DOSRmDir	Removes a subdirectory.

1 Only category 8, Function 20 DOSDevIOCTL calls are supported.

2 File date and time of creation as well as file date and time of last access are not supported and are returned as zeros.

3 Only DOSFsCtl calls defined by the shared folders function are supported.

4 The DOSfileIO function call is not fully supported. See "Support of OS/2 File Management Function Call DOSfileIO" on page 2-45 for more information.

Figure 2-11 (Page 2 of 2). OS/2 Shared Folders File Management Function Calls

Function Call	Description
DOSScanEnv	Scans the environment.
DOSSearchPath	Searches the path.
DOSSelectDisk	Selects the disk.
DOSSetFHandstate	Sets the file handle state.
DOSSetFileInfo	Sets the file information.
DOSSetFileMode	Sets the file mode.
DOSSetFsInfo	Sets the file system information.
DOSSetMaxFH	Defines the new maximum file handle.
DOSSetPathInfo	Sets the path information.
DOSSetVerify	Sets the verify setting.
DOSWrite	Writes to a file or device.
DOSWriteAsync	Writes to a file or device asynchronously.

1 Only category 8, Function 20 DOSDevIOCTL calls are supported.

2 File date and time of creation as well as file date and time of last access are not supported and are returned as zeros.

3 Only DOSFsCtl calls defined by the shared folders function are supported.

4 The DOSfileIO function call is not fully supported. See "Support of OS/2 File Management Function Call DOSfileIO" on page 2-45 for more information.

Improving File Management When Using the OS/2 Operating System

File management for applications that use the shared folders function should be written to maximize performance. Almost all file requests for folders (open, read, write, and so on) are sent to the AS/400 system for processing, which requires time and resources. These same file requests may not require disk accesses when using a diskette or a hard file.

Improve file management performance by using the following suggestions:

- Use an appropriate create or open operation when creating or opening a file, instead of searching for the file first to verify its existence. A file search requires significant time and resources relative to other file operations.
- Design an application to open a file once, perform all necessary operations, and then close the file.
- Do not write with verification unless absolutely necessary. Write buffering is not used in the program when verification is on.
- Read data from a file from the beginning to end, and not vice-versa. Reading from the beginning to the end uses the program buffer storage more efficiently.
- Perform reads and writes sequentially, rather than randomly. Sequential reads and writes use the program buffer storage more efficiently.
- Read small files (such as a profile) entirely into storage. Storage accesses are faster than disk or AS/400 accesses.

File Sharing Using the OS/2 Operating System

When you open a file, you must specify what operations other processes perform on that file by specifying the sharing mode. Shared folders function supports all sharing options supported by the OS/2 operating system.

You must also specify the access intent of a file when you open it. The access intent specifies which operations your processes can perform on the file. Allowable access intents are read, write, and read/write. Shared folders function supports all access intent options supported by the OS/2 operating system.

Note: A process running in the OS/2 DOS compatibility mode opens a file in compatibility mode. The shared folders function supports this open as if it is a request to open the file with read/write access that allows no other processes to access the file.

Deny Read/Write Sharing Mode

Under the OS/2 operating system, if a process opens a file with a sharing mode of deny read/write, no other processes are allowed to open the file. Access to the file is exclusive to the process that opens the file until that process closes the file. Shared folders function supports this sharing mode.

Deny Read Sharing Mode

Under the OS/2 operating system, if a process opens a file with a sharing mode of deny read, no other process that requests read access to the file is allowed to open the file. Other processes are allowed to open the file with write-only access intent, however. Shared folders function supports this sharing mode as a deny none sharing mode.

Deny Write Sharing Mode

Under the OS/2 operating system, if a process opens a file with a sharing mode of deny write, no other process that requests write access to the file is allowed to open the file. Other processes are allowed to open the file with read-only access intent, however. Shared folders function supports this sharing mode.

Deny None Sharing Mode

Under the OS/2 operating system, if a process opens a file with a sharing mode of deny none, another process will be allowed to open the file. Shared folders supports this sharing mode.

File Access Summary

If a process opens a file with a sharing option of deny read/write, all other attempts to open the file fail.

If a process opens a file with a sharing option of deny write, and another process attempts to open the file, the following occurs:

- If the second process attempts to open the file with a sharing mode of deny read/write, the request fails.
- If the second process attempts to open the file with a sharing mode of deny write, the request is successful only if both processes open the file for read-only access.
- If the second process attempts to open the file with a sharing mode of deny none, the open is successful only if the process that originally opened the file

did the open with an access method of read only. The second process can open the file for any access.

If a process opens a file with a sharing option of deny none, and another process attempts to open the file, the following occurs:

- If the second process is deny read/write, the request fails.
- If the second process is deny write, the open is successful only if the process that originally opened the file did the open with an access method of read only. The second process can open the file for any access.
- If the second process is deny none, the open will be successful for any access method.

Specifying Access Intent

The following figure shows the results of opening a file and then attempting to reopen the same file from a process in another personal computer. The figure lists all combinations of access intents and sharing mode.

			File to Be Opened								
			DRW			DW			DN/DR		
			I	IO	O	I	IO	O	I	IO	O
Currently Open File	DRW	I	A	A	A	A	A	A	A	A	A
		IO	A	A	A	A	A	A	A	A	A
		O	A	A	A	A	A	A	A	A	A
	DW	I	A	A	A	2	A	A	2	A	A
		IO	A	A	A	A	A	A	2	A	A
		O	A	A	A	A	A	A	2	A	A
	DN/DR	I	A	A	A	2	1	1	2	1	1
		IO	A	A	A	A	A	A	2	2	2
		O	A	A	A	A	A	A	2	2	2

2 2nd, 3rd,...open is allowed
1 2nd, 3rd,...open is allowed if the file does not have the read-only directory attribute set on
A 2nd, 3rd,...open is denied
DRW Deny read/write mode (exclusive)
DW Deny write mode
DR Deny read mode
DN Deny none mode
I Read-only access intent
O Write-only access intent
IO Read/write access intent

OS/2 Directory Attributes

An OS/2 attribute-byte in a files-directory entry contains the following information about the file:

Archive candidate bit Indicates alteration of the file since it was last backed up.

Subdirectory bit Indicates a subdirectory. It is excluded from directory searches that do not specifically ask to have sub-directories included.

System bit Indicates a system file. The file is excluded from normal directory searches.

Hidden bit Indicates a hidden file. The file is excluded from normal directory searches.

Read-only bit Indicates that the write access to a file is not allowed. The file cannot be deleted.

Read-only, Hidden, and System Files: Shared folders function supports the archive, read-only, hidden, and system file types in the same manner as the OS/2 operating system with the following exception. Hidden and system files can be changed and replaced if the application knows the name of the file. A hidden or system file is excluded from normal directory searches, but can be changed if an application knows the name of the file. See the *OS/2 Programming Guide* for more information on read-only, hidden, and system files.

Support of OS/2 File Management Function Call DOSfileIO: The shared folders function will support the timeout parameter of the lock command of the DOSFILEIO function call. That is, if a lock timeout is specified on the lock command of DOSFILEIO, shared folders will try the lock request again if it failed, because another process had that area of the file locked. However, the shared folders function does not support the infinite lock timeout (this timeout specifies that the application will wait indefinitely). The shared folders function will try the lock operation again only a finite number of times. If the lock cannot be granted, the shared folders function returns an error to the application.

Extended Attributes

Extended file attributes are supported by OS/2 shared folders. A length limitation of 32KB (KB equals 1024 bytes) of extended attribute data per file, including operating system overhead, is imposed. For more information on extended attributes, see the *PC Support/400 OS/2 Installation and Administration Guide*.

Communications Conversations

The shared folders function uses communications conversations to communicate with the host system. It uses one conversation for each shared folder drive you have assigned. Starting with Version 2 Release 2, the shared folders function uses one conversation for each drive and one conversation for each host system for which a drive is assigned. This extra conversation is used for data caching. If your host system has a limited number of conversations, or you do not want to use shared folders to use this extra conversation with your host system, turn off caching. You can do this by changing your CONFIG.PCS entry, MCAO, to a cache size of 0 (zero).

Chapter 3. Transfer Function

This chapter describes the PC Support/400 transfer function when using the DOS, extended DOS, and OS/2 operating system. It provides information on the transfer function programs, personal computer file description files, and data conversions.

Unless otherwise noted, the information in this chapter applies to the PC Support/400 transfer function when using the DOS, extended DOS, or OS/2 operating system.

Transfer Function Overview

The AS/400 transfer function transfers data to or from the AS/400 system and a personal computer. PC Support/400 provides interactive, automatic, and API programs for the transfer functions. This support gives you flexibility in sharing AS/400 system data with your personal computer.

The transfer function consists of the following:

- AS/400 transfer function
- Personal computer transfer function, which consists of two parts:
  - Interactive or automatic transfer function programs
  - API program

The AS/400 transfer function program runs on the AS/400 system. The personal computer transfer function program (the API and interactive or automatic transfer function programs) runs on the personal computer.

The interactive or automatic transfer function programs pass requests to the AS/400 transfer function program through the appropriate router and communications support, depending on whether you use the DOS, extended DOS, or OS/2 operating system.

The AS/400 transfer function programs use the transfer request to send data or to retrieve data from the specified AS/400 database files.

Figure 3-1 shows how the interactive and automatic programs relate to each other when you use the DOS operating system.

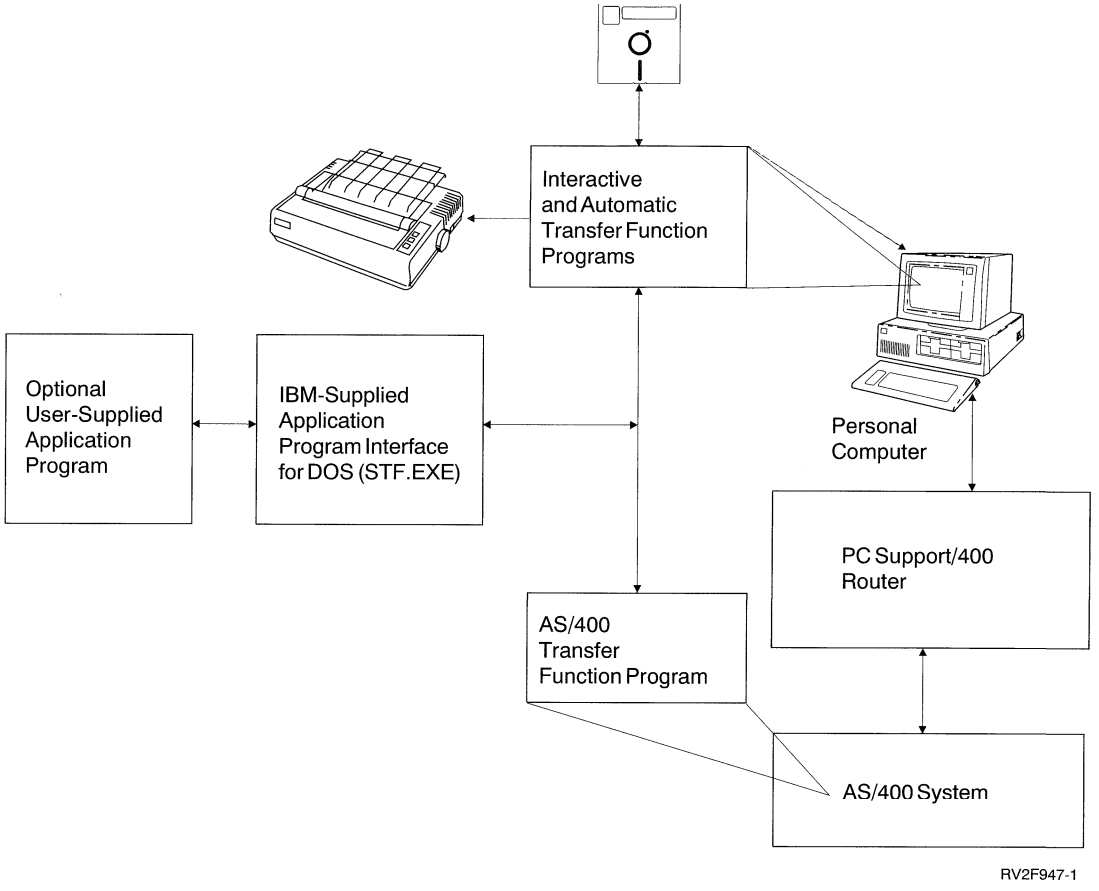


Figure 3-1. Transfer Function Overview Using the DOS Operating System

Figure 3-2 shows how the interactive and automatic programs relate to each other when you use the OS/2 operating system.

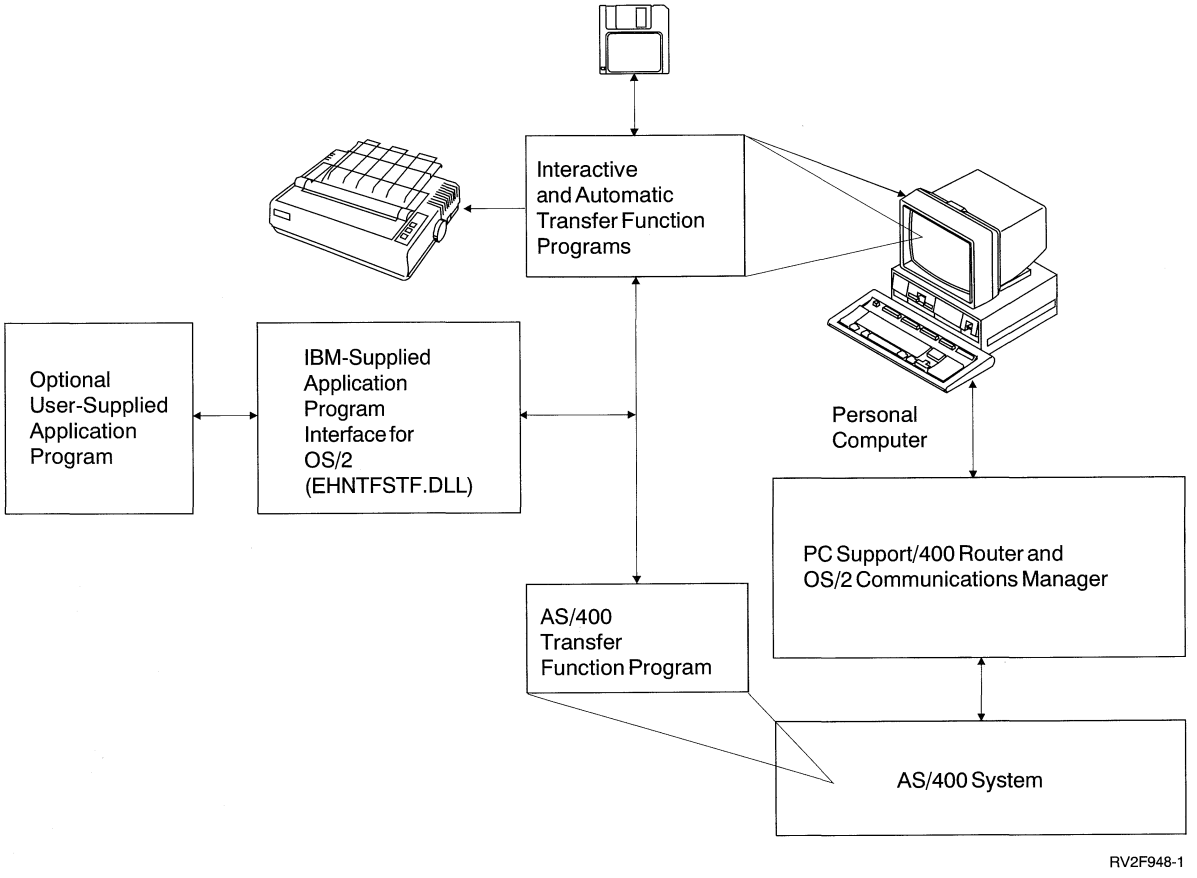


Figure 3-2. Transfer Function Overview Using the OS/2 Operating System

For an AS/400 system-to-personal computer transfer request, the interactive or automatic transfer function program retrieves the data, one record at a time, and changes the system data records to the format needed by the personal computer. The interactive or automatic transfer function program then does one of the following:

- Displays the data on the personal computer display
- Prints the data on the personal computer printer
- Writes the data to a personal computer disk or diskette file

For a personal computer-to-AS/400 transfer request, the interactive or automatic transfer function program sends data records, one at a time, to the AS/400 transfer function program. The AS/400 transfer function program writes the data to a physical database file.

An API is supplied which allows you to write personal computer application programs to create transfer requests instead of using the interactive or automatic transfer function programs. This allows the personal computer application programs to send data to or receive data from a database file. Refer to the *PC Support/400 API Reference* manual for information on the personal computer API.

Transferring Files from the AS/400 System to the Personal Computer

When using a personal computer, you can retrieve and use data from the following file types on the AS/400 system:

- Physical database
- Logical database
- DDM

When retrieving files, you can:

- Control which records (and which fields within a record) are retrieved.
- Control the ordering of records and the ordering of fields within the record.
- Specify the sequence in which you want the records sorted.
- Select a subset of the records.
- Group records into summary records.
- Join two or more files.
- Specify formats and separators of date and time fields.
- Correct or ignore decimal data errors.
- Specify decimal separator character.

Transferring Files from the Personal Computer to the AS/400 System

You can transfer data from a file on a personal computer to an AS/400 database file member. When transferring data to the AS/400 system, you can optionally create the file or the file member, or both. Personal computer data transferred to the AS/400 file member replaces data in the file member.

Note: Because existing data in a file member is replaced when you send data to that file member, it might be a safer practice to send your data to a work file on the AS/400 system and then have an AS/400 application merge the contents of this work file into the primary file.

For the AS/400 system, field definitions are contained in the file or in a data dictionary.

Database physical files are transferred by individual field. That is, numeric and character data fields are translated from personal computer format to AS/400 format as part of the sending process. The translation is specific to each field.

If the AS/400 file has fields not in the personal computer file you are sending, default values are used for the fields. If a field of the AS/400 file has a default value specified through the DFT DDS keyword, the default value is used. If no default value is specified, the field default values are as follows:

Field Type	Default Value
Character	EBCDIC blanks
Numeric	Zeros

You can also transfer data to an AS/400 source file. A record length is required to transfer data to an AS/400 source file. A source sequence number and date are stored with each record of the data.

Transfer Function Programs

This section contains information on transfer function programs used by the DOS and OS/2 operating systems.

Programs for DOS Users

The transfer function consists of the following programs when using the DOS operating system:

RFROMPC.EXE

This program is the interactive transfer program used to build requests to send data to the AS/400 system. It has options to run or save the request, or to recall a previously saved request.

RFROMPCB.EXE

This program is the automatic transfer program used to run a previously saved (by RFROMPC) send request without interaction.

RTOPC.EXE

This program is the interactive transfer program used to build requests to retrieve data from the AS/400 system. It has options to run or save the request, or to recall a previously saved request.

RTOPCB.EXE

This program is the automatic transfer program used to run a previously saved (by RTOPC) retrieve request without interaction.

STF.EXE

This program is the resident portion of the transfer function. It supplies the API for user applications and communicates with the AS/400 transfer function (through the router) to perform the sending and retrieving of data. STF.EXE must be installed for user applications, but does not need to be installed for the other transfer function programs.

TFR.EXE

This program is used by the transfer function interactive and automatic programs to communicate with the AS/400 transfer function program. It uses the router to transfer data between the AS/400 system and the personal computer. The transfer function interactive and automatic programs load TFR.EXE from the same drive and path from which the interactive or automatic transfer function program was loaded. You receive an error message if you try to run TFR.EXE from the command line.

Figure 3-3 shows the relationship among the DOS transfer function programs.

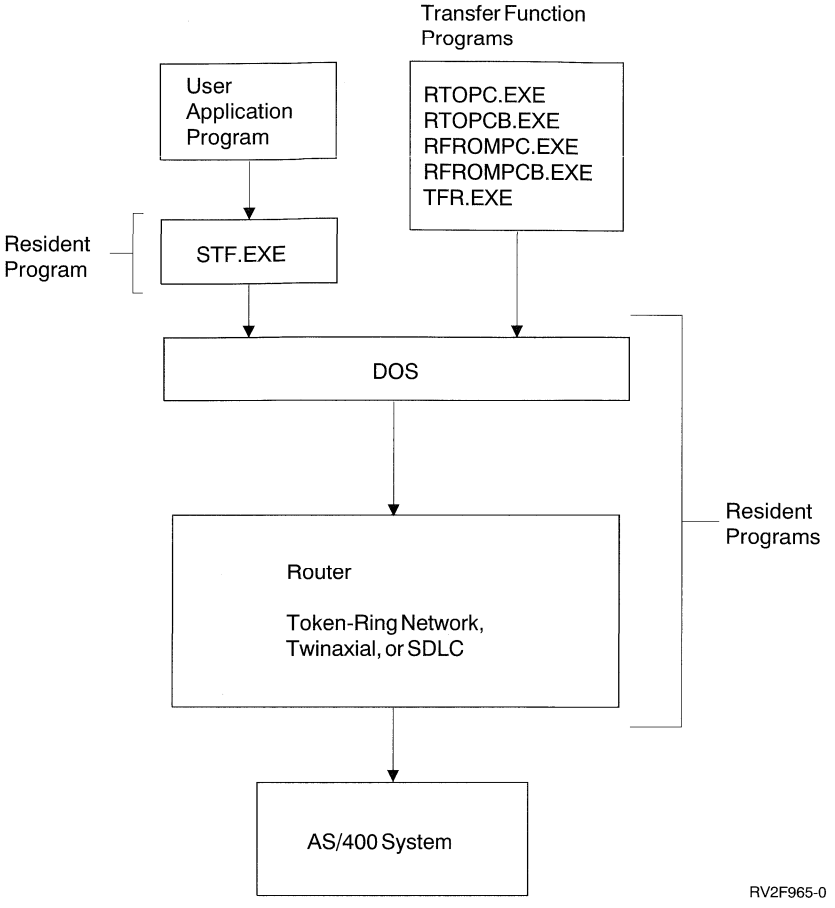


Figure 3-3. Relationship among Transfer Function Programs-DOS (for DBCS there is no SDLC support)

All transfer request files, data files, and file description files created by DOS are compatible with the OS/2 version of the transfer function. All transfer request files, data files, and file description files created by OS/2 are compatible with the transfer function when using DOS.

Instead of using the IBM-supplied transfer function programs RTOPC, RTOPCB, RFROMPC, and RFROMPCB, you can write your own application program to transfer data to and from the AS/400 system. Your program must use the application program interface supplied through STF.EXE to communicate with the AS/400 system. Refer to the *PC Support/400 API Reference* manual for details.

Programs for OS/2 Users

The transfer function consists of the following programs when using the OS/2 operating system:

RFROMPC.EXE
This program is the interactive transfer program used to build requests to send data to the AS/400 system. It has options to run or save the request, or to recall a previously saved request.

RFROMPCB.EXE
This program is the automatic transfer program used to run a previously saved (by RFROMPC) send request without interaction.

RTOPC.EXE
This program is the interactive transfer program used to build requests to retrieve data from the AS/400 system. It has options to run or save the request, or to recall a previously saved request.

RTOPCB.EXE
This program is the automatic transfer program used to run a previously saved (by RTOPC) retrieve request without interaction.

EHNTFSTF.DLL
This program is a dynamic link library. It is used by the transfer function interactive and automatic programs to communicate with the AS/400 transfer function program. It also supplies the API for user applications. It uses the PC Support/400 router and OS/2 Communications Manager to transfer data between the AS/400 system and the personal computer.

You receive an error message if you try to do any of the following with the DOS and OS/2 transfer function programs:

- Run EHNTFSTF.DLL in the DOS compatibility mode or in the DOS operating system.
- Run STF.EXE or TFR.EXE in the DOS compatibility mode or in the OS/2 operating system.
- Run the interactive or automatic programs meant for DOS users in the OS/2 operating system.
- Run the interactive or automatic programs meant for OS/2 users in the DOS compatibility mode or in the DOS operating system.

Note: You can run the interactive or automatic programs meant for DOS users in DOS compatibility mode, but they cannot communicate with the AS/400 system.

Figure 3-4 on page 3-8 shows the relationship among the transfer function programs when you use the OS/2 operating system.

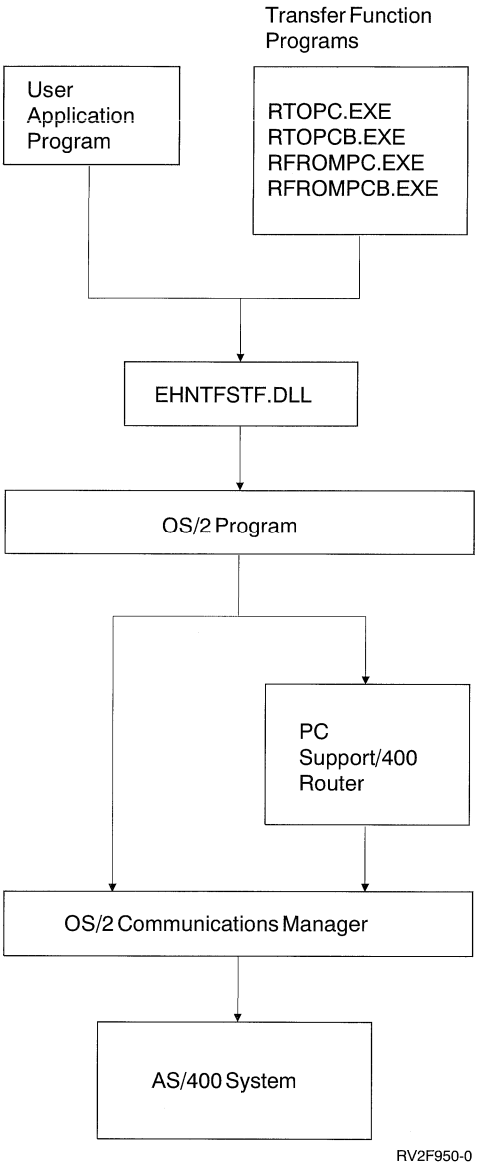


Figure 3-4. Relationship among Transfer Function Programs (OS/2)

All transfer request files, data files, and file description files created by the OS/2 operating system are compatible with the DOS version of the transfer function. All transfer request files, data files, and file description files created by the DOS operating system are compatible with the transfer function when using the OS/2 operating system.

Instead of using the IBM-supplied transfer function programs RTOPC, RTOPCB, RFROMPC, and RFROMPCB, you can write your own application program to transfer data to and from the AS/400 system. Your program must use the API supplied through EHNTFSTF.DLL to communicate with the AS/400 system. Refer to the *PC Support/400 API Reference* manual for details.

Transfer Function Batch or Command Files

You can include the transfer function programs in your own DOS batch or OS/2 command files. This section describes how to start those programs and provides an example.

Note: For the OS/2 operating system, batch files are called command files and have a file extension of .CMD instead of .BAT. They are functionally equivalent. When a DOS batch file is renamed with an extension of .CMD, it will run successfully under the OS/2 operating system (assuming, of course, that the programs called in the batch file run in both the DOS and the OS/2 operating system). For purposes of this discussion, we will refer to these files as batch files.

If you are using the DOS operating system, you must first start the PC Support/400 router to run the transfer function batch file.

For double-byte character set (DBCS), if you are using the DOS operating system and you are using the twinaxial PC Support/400 router, you must first start the 5250 personal computer program and the PC Support/400 router to run the transfer function batch file.

If you are using the OS/2 operating system, you must start the OS/2 Communications Manager and the PC Support/400 router to run the transfer function batch file. For more information on the PC Support/400 router, refer to the *PC Support/400 DOS Installation and Administration Guide* or the *PC Support/400 OS/2 Installation and Administration Guide*.

Batch File Example

The following example shows how you might write a batch file to run various transfer function programs. For this example, assume that the transfer requests were previously created and saved using the interactive AS/400-to-personal computer transfer function. (For details on how to create and save transfer requests, refer to the *PC Support/400 User's Guide for DOS* and to the *PC Support/400 User's Guide for OS/2*.)

Note: See "Creating Complicated Transfer Requests" in the *PC Support/400 User's Guide for DOS* for information on batch error codes.

The batch file in this example runs four AS/400-to-personal computer transfer requests: CUST01.TTO, CUST02.TTO, CUST03.TTO, and CUST04.TTO. If any transfer request is not successful, a message appears and the batch file ends.

Not all programming considerations or techniques are shown in the following batch file example. Review the example before you begin application design and coding.

```
ECHO OFF
RTOPCB.EXE CUST01.TTO
IF ERRORLEVEL 20 GOTO EXIT1
RTOPCB.EXE CUST02.TTO
IF ERRORLEVEL 20 GOTO EXIT2
RTOPCB.EXE CUST03.TTO
IF ERRORLEVEL 20 GOTO EXIT3
RTOPCB.EXE CUST04.TTO
IF NOT ERRORLEVEL 20 GOTO EXIT
GOTO EXIT4
:EXIT1
ECHO CUST01 DID NOT WORK
GOTO EXIT
:EXIT2
ECHO CUST02 DID NOT WORK
GOTO EXIT
:EXIT3
ECHO CUST03 DID NOT WORK
GOTO EXIT
:EXIT4
ECHO CUST04 DID NOT WORK
:EXIT
```

Personal Computer File Description File

The file description file is a PC file that contains all field descriptions of the data in the corresponding personal computer data file. Each field descriptor contains the field name, data type, and field length. There is one for each field in the PC file.

The personal computer files require field definitions when the files are transferred. The field definitions describe the file as it exists on the personal computer. These definitions contain data that is similar to the field definitions (DDS) required by AS/400 system files. The data must be defined for both the AS/400 system and the personal computer files because the field names from each file are needed to send the data to the AS/400 system and the data in each file may be in different formats.

A file description file is created on request during the transfer process of data from an AS/400 file to a PC file. Therefore, you do not usually need to worry about the file description file, its format, or its contents. However, if you have data that has not been previously transferred to the system, and you want to transfer it to the system, you must create a file description file.

Creating a File Description File

You can create a file description file using a personal computer text editor. The file description file must be an ASCII (PC CODE for DBCS) text file. Therefore each record must end with a carriage return (CR) character (X'0D') followed by a line feed (LF) character (X'0A'). All tab characters (X'09') are treated as ASCII (PC CODE for DBCS) spaces. The last byte of the file must contain an end-of-file (EOF) character (X'1A'). PC editors that create ASCII (PC CODE for DBCS) text files usually use these special character designators, so normally you do not need to be concerned about them.

The DOS 5.0 editor does not place an end-of-file (EOF) character (X'1A') at the end of the file when a file is saved. Because an EOF character must be the last character in the file, you need to correct this problem. To correct it, use the DOS COPY command with the /A option to add an EOF character (X'1A') to the end of the file.

For example, type the following DOS commands at the DOS prompt to correct a file named FILE.FDF.

```
COPY FILE.FDF TEMP /A      adds EOF character to temporary file
COPY TEMP FILE.FDF         copies contents back to correct file name, FILE.FDF
ERASE TEMP                 erases the temporary file
```

File Description File Format

The format of the file description file is as follows:

```
PCFDF [comment]
PCFT file-type-indicator [comment]
PCFO time-format,time-separator, date-format, date-separator, decimal-separator [comment]
PCFL field-name-1 data-type-1 length-1[/decimal-position-1][comment]
    .
    .
    .
PCFL field-name-n data-type-n length-n[/decimal-position-n][comment]
[* comment]
```

Items within brackets are optional. Use either uppercase or lowercase characters anywhere in the file.

PCFDF Entries: PCFDF is a keyword that identifies this file as a personal computer file description file. It must appear in the first line of the file, starting in column one. A comment is the only other entry allowed on the first line. If you type a comment, it must be separated from the PCFDF keyword by a space.

PCFT Entries: PCFT is a keyword that identifies this record as containing the file type indicator. It is followed by an indicator identifying the type of PC file in which the data is stored. It must appear only once, and must start in column one, after the PCFDF record and before any PCFL records. An optional comment can follow this file type indicator if separated from the indicator by at least one space.

Following is an example of a PCFT entry:

```
PCFT 4 BASIC RANDOM FILE
```

Figure 3-5 displays the valid PC file type indicators.

Figure 3-5 (Page 1 of 2). PC File Type Indicators

Indicator	SBCS File Type	DBCS File Type
1	ASCII text	PC CODE text
2	DOS random	DOS random
3	BASIC sequential	BASIC sequential
4	BASIC random	BASIC random
5	Data interchange format (DIF**)	Reserved
6	No-conversion file	No-conversion file

Figure 3-5 (Page 2 of 2). PC File Type Indicators

Indicator	SBCS File Type	DBCS File Type
7	Reserved	Symbolic Link Format (SYLK)
8	DOS random type 2	DOS random type 2

PCFO Entry: The PCFO entry is optional. PCFO is a keyword that identifies this record as containing information about the date and time formats, timestamp, and separator characters for applicable formats. It must appear only once and must start in column one, after the PCFT record and before any PCFL records. If there is no PCFO entry, the information or characters assigned as defaults for the host system are used.

Figure 3-6 displays the valid PC time formats.

Figure 3-6. PC Time Formats

Indicator	Format Name	Time Format
1	HMS	hh:mm:ss
2	ISO - International Standards Organization	hh.mm.ss
3	USA - USA standard	hh:mm AM or PM
4	EUR - European	hh.mm.ss
5	JIS - Japanese Industrial Standard Christian Era	hh:mm:ss
6	DDS	format given by AS/400 file attribute
7	DFT	host job default is used
*	Unspecified	host job default is used

Figure 3-7 displays the valid PC time separators.

Figure 3-7. PC Time Separators

Indicator	Separator
1	Colon (:)
2	Period (.)
3	Comma (,)
4	Blank ( )
5	Null (N)
6	Default (D) (host job default)
*	Unspecified (host job default)

Figure 3-8 displays the valid PC date formats.

Figure 3-8 (Page 1 of 2). PC Date Formats

Indicator	Format Name	Date Format
1	MDY	mm/dd/yy

Figure 3-8 (Page 2 of 2). PC Date Formats

Indicator	Format Name	Date Format
2	DMY	dd/mm/yy
3	YMD	yy/mm/dd
4	Julian	yy/ddd
5	ISO	yyyy-mm-dd
6	USA	mm/dd/yyyy
7	EUR	dd.mm.yyyy
8	JIS	yyyy-mm-dd
9	DDS	format given by AS/400 file attribute
10	DFT	host job default is used
*	Unspecified	host job default is used

Figure 3-9 displays the valid PC date separators.

Figure 3-9. PC Date Separators

Indicator	Separator
1	Slash (/)
2	Dash (–)
3	Period (.)
4	Comma (,)
5	Blank ( )
6	Null (N)
7	Default (D) (host job default)
*	Unspecified (host job default used)

Figure 3-10 displays the valid PC decimal separators.

Figure 3-10. PC Decimal Separators

Indicator	Separator
1	Period (.)
2	Comma (,)
3	Default (D) (PC country default)
*	Unspecified (PC default used)

The following is an example of a PCFO entry:

```
PCFO 1,1,1,1,1 OPTIONS SETTINGS
```

PCFL Entries: PCFL identifies a definition for a field. Enter a PCFL entry in the file description file for each field in the PC data file. The PCFL records must be in the same order as the fields they define in the data file.

Define as many as 256 PCFL records in the file description file and start PCFL records in column one. If you enter more than 256 PCFL records, you receive an error message. You cannot continue a record on more than one line, and only the first 80 characters of a record are used.

Following is an example of a PCFL entry:

```
PCFL CUSTNAME 1 20  CUSTOMER NAME
```

Each PCFL entry contains the following:

- The keyword, PCFL, starting in column one and followed by a space. This identifies the record as a field description.
- The field name, followed by a space. This must match the name that exists in the field definitions on the AS/400 system and can be from 1 to 10 characters long.
- The indicator for the data type. Figure 3-11 shows the indicators that represent the personal computer data type of the data that exists in the field. You must follow the specified indicator with a space.
- The size of the field (in bytes) as it is stored in the personal computer file. The length specification can be from 1 to 4 characters long.

Figure 3-11. PC Data Type Indicators

Indicator	SBCS Data Type	DBCS Data Type
1	ASCII [1]	PC CODE [1]
2	ASCII numeric	PC CODE numeric
3	Hexadecimal	Hexadecimal
4	Binary	Binary
5	Zoned	Zoned
6	Packed	Packed
7	BASIC integer	BASIC integer
8	BASIC single-precision floating point	BASIC single-precision floating point
9	BASIC double-precision floating point	BASIC double-precision floating point
10	EBCDIC	EBCDIC [1]
11	EBCDIC zoned	EBCDIC zoned
12	EBCDIC packed	EBCDIC packed
13		DBCS-open
14		DBCS-only
15		DBCS-either
16		DBCS-graphic

1 Includes date, time, and timestamp except for files that are not converted.

The data type indicator you enter must be valid for the file type entered earlier. Any other data types are not valid and are diagnosed as errors during a data transfer to the AS/400 system.

Figure 3-12 contains the valid SBCS data types for each file.

Figure 3-12. Valid SBCS Data Types for File Types

File Type	Valid Data Type
ASCII text	ASCII ASCII numeric
DOS random	ASCII Binary Hexadecimal Packed Zoned
BASIC sequential	ASCII ASCII numeric
BASIC random	ASCII BASIC double-precision floating point BASIC integer BASIC single-precision floating point Hexadecimal
DIF	ASCII ASCII numeric
No-conversion	Binary EBCDIC EBCDIC packed EBCDIC zoned Hexadecimal
DOS random type 2	ASCII Binary Hexadecimal Packed Zoned

Note: ASCII (SBCS) and PC CODE (DBCS) includes date, time, and timestamp types if converted. EBCDIC includes date, time, and timestamp if not converted.

Figure 3-13 contains the valid DBCS data types for each file.

Figure 3-13 (Page 1 of 2). Valid DBCS Data Types for File Types

File Type	Valid Data Type
PC CODE text	PC CODE PC CODE numeric
DOS random	PC CODE Binary Hexadecimal Packed Zoned
BASIC sequential	PC CODE PC CODE numeric
BASIC random	PC CODE BASIC double-precision floating point BASIC integer BASIC single-precision floating point Hexadecimal

Figure 3-13 (Page 2 of 2). Valid DBCS Data Types for File Types

File Type	Valid Data Type
No-conversion	Binary EBCDIC EBCDIC packed EBCDIC zoned DBCS-either DBCS-only DBCS-open DBCS-graphic Hexadecimal
SYLK	PC CODE PC CODE numeric
DOS random type 2	PC CODE Binary Hexadecimal Packed Zoned

Note: ASCII (SBCS) and PC CODE (DBCS) includes date, time, and timestamp types if converted. EBCDIC includes date, time, and timestamp if not converted.

For numeric fields in BASIC sequential and DIF files, a size specification must be present. However, because the data in these fields is of variable length, the transfer function assumes a maximum length of 65 characters. This length more than covers the largest possible exponential ASCII (PC CODE for DBCS) numeric value. The size specifications for character fields must be the maximum size of any data item in that field.

Figure 3-14 shows the allowed data length limits for each SBCS personal computer data type. These are the maximum lengths you can specify for size in the PCFL entry.

Figure 3-14 (Page 1 of 2). Allowable Data Length Limits for SBCS PC Data Types

Personal Computer Data Type	Data Length Limit (in Bytes)
ASCII	4096
ASCII numeric	33 (65 for DIF and BASIC sequential)
BASIC double-precision	8 (only allowed length)
BASIC integer	2 (only allowed length)
BASIC single-precision	4 (only allowed length)
Binary	4
EBCDIC	4096
Hexadecimal	2048
Packed decimal (ASCII and EBCDIC)	16
Zoned decimal (ASCII and EBCDIC)	31

Figure 3-14 (Page 2 of 2). Allowable Data Length Limits for SBCS PC Data Types

Personal Computer Data Type	Data Length Limit (in Bytes)
Time	
HMS (see note 1)	8
USA	8
ISO, EUR, and JIS (see note 1)	8
DDS, DFT	8 or 10 (see note 2)
Date	
MDY, DMY, YMD	8
Julian	6 (only allowed length)
ISO, EUR, JIS, USA (see note 1)	10
DDS, DFT	6, 8, or 10 (see note 2)
Timestamp	26

Notes:

1 These abbreviations appear in the time and date parameter sections.

Abbreviation	Description
HMS	Hours Minutes Seconds
EUR	IBM European Standard
JIS	Japanese Industrial Standard Christian Era
ISO	International Standards Organization

2 Additional data is included in each record. This data indicates whether the field contains a null value. The following restrictions apply to the maximum field lengths:

- The limit is 4093 bytes when using Version 2 Release 2 (or later) interactive or automatic transfer programs to transfer data to or from a Version 2 Release 2 (or later) AS/400 system.
- The limit is 4096 bytes when using Version 2 Release 2 programs with an AS/400 system from an earlier release for all PC file types except BASIC sequential and DIF(SBCS) or SYLK(DBCS) file types. These file types are limited to 4093 bytes.
- The limit is 4093 bytes when using interactive or automatic transfer programs from an earlier release to transfer data to or from a Version 2 Release 2 AS/400 system.
- The limit is 4096 bytes when using interactive or automatic transfer programs from an earlier release to transfer data to or from an AS/400 system with an earlier release.

3 The length is determined by the format defined in the host file for DDS, or from the AS/400 job default (DFT keyword).

Figure 3-15 shows the allowed data length limits for each DBCS personal computer data type. These are the maximum lengths you can specify for size in the PCFL entry.

Figure 3-15 (Page 1 of 2). Allowable Data Length Limits for DBCS PC Data Types

Personal Computer Data Type	Data Length Limit (in Bytes)
PC CODE	4096 (see note 1)
PC CODE numeric	33 (65 for SYLK and BASIC sequential)
BASIC double-precision	8 (only allowed length)
BASIC integer	2 (only allowed length)

Figure 3-15 (Page 2 of 2). Allowable Data Length Limits for DBCS PC Data Types

Personal Computer Data Type	Data Length Limit (in Bytes)
BASIC single-precision	4 (only allowed length)
Binary	4
EBCDIC	4096 (see note 1)
DBCS-either	4096 (see note 1)
DBCS-only	4096 (see note 1)
DBCS-open	4096 (see note 1)
DBCS-graphic	4096 (see note 1)
Hexadecimal	2048 (see note 1)
Packed decimal (PC CODE and EBCDIC)	16
Zoned decimal (PC CODE and EBCDIC)	31
Time	
HMS	6
USA	8
ISO, EUR, and JIS	8
DDS, DFT	8 or 10 (see note 2)
Date	
MDY, DMY, YMD	8
Julian	6 (only allowed length)
ISO, EUR, JIS, USA	10
DDS, DFT	6, 8, or 10 (see note 2)
Timestamp	26

Notes:

1 These abbreviations appear in the time and date parameter sections.

Abbreviation	Description
HMS	Hours Minutes Seconds
EUR	IBM European Standard
JIS	Japanese Industrial Standard Christian Era
ISO	International Standards Organization

2 Additional data is included in each record. This data indicates whether the field contains a null value. The following restrictions apply to the maximum field lengths:

- The limit is 4093 bytes when using Version 2 Release 2 (or later) interactive or automatic transfer programs to transfer data to or from a Version 2 Release 2 (or later) AS/400 system.
- The limit is 4096 bytes when using Version 2 Release 2 programs with an AS/400 system from an earlier release for all PC file types except BASIC sequential and DIF(SBCS) or SYLK(DBCS) file types. These file types are limited to 4093 bytes.
- The limit is 4093 bytes when using interactive or automatic transfer programs from an earlier release to transfer data to or from a Version 2 Release 2 AS/400 system.
- The limit is 4096 bytes when using interactive or automatic transfer programs from an earlier release to transfer data to or from an AS/400 system with an earlier release.

3 The length is determined by the format defined in the host file for DDS, or from the AS/400 job default (DFT Keyword).

Figure 3-16 on page 3-19 shows the allowed data length limits for each AS/400 data type.

Figure 3-16. Allowable Data Length Limits for AS/400 Data Types

AS/400 Data Type	Data Length Limit in Bytes (see note 1)
Binary	2 or 4 (only allowed lengths)
EBCDIC	4096
Hexadecimal	2048
Packed decimal (EBCDIC)	16
Zoned decimal (EBCDIC)	31
Time	
HMS	8
USA	8
ISO, EUR, and JIS	8
DDS, DFT	8 or 10 (see note 2)
Date	
MDY, DMY, YMD	8
Julian	6 (only allowed length)
ISO, EUR, JIS, USA	10
DDS, DFT	6, 8, or 10 (see note 2)
Timestamp	26

Notes:

1 The data length limits for the personal computer and the system data fields are different in some cases. For these cases, the transfer function attempts to fit the personal computer data into the system field. If the data does not fit into the field, a message displays. Refer to "Data Conversions" on page 3-20 for more details.

2 The length is determined by the format defined in the host file for DDS, or from the AS/400 job default (DFT Keyword).

If there is a decimal position associated with the data in that field, you must place a forward slash (/) and then the number of decimal positions after the length specification. There are no spaces between the length, slash, and decimal position specifications.

The decimal position specification refers to the number of positions from the right-hand byte of the resulting decimal number. Do not specify a decimal position for floating-point numbers unless the data type is one of the following:

- ASCII (PC CODE for DBCS) numeric
- Binary
- Packed
- Zoned

Following the data length of the PCFL entry, you can have a description of the field.

Note: The number of decimal positions in a field ranges from 0 to 9 or the maximum number of decimal digits in this number, whichever is smaller. The transfer function may round the number in order to fit it into the field. Refer to "Data Conversions" on page 3-20 for more details.

Comment Entries: Enter comment lines anywhere in the file description file, observing the following restrictions:

- The last element of the field descriptor entry specification is a comment. This is an optional entry for your information only, and must be separated from the size entry by a space. PCFL entries created by the transfer function (RTOPC) do not contain a comment field.
- Precede the comment with an asterisk (*) as the first character in the line that is not a space.
- Do not exceed 80 characters in length.
- Do not make the comment the first record in the file description file.

Following is an example of a comment:

```
* This is a comment
```

Example

Following is an example of a file description file for an inventory file:

```
PCFDF
PCFT 3 BASIC SEQUENTIAL FILE
* ITEM INVENTORY FILE
PCFO 1,1,1,2,1 OPTIONS SETTINGS
PCFL ITEMNO 2 8  ITEM NUMBER
PCFL ITEMDESC 1 20  DESCRIPTION OF ITEM
PCFL COLOR 1 8  COLOR
PCFL WEIGHT 2 7/2  ITEM WEIGHT
PCFL PRICE 2 7/2  PRICE PER ITEM
PCFL INSTOCK 2 6  ITEMS IN STOCK
```

Data Conversions

The transfer function needs data conversions for transferring data from the system to the personal computer, and vice versa. For both types of transfers, the necessary conversion depends on the record size, type of data being transferred, the type of personal computer file being used, the system data type, and, in some cases, the data length.

Record Size

Each transferred record contains data indicating whether or not each field contains a null value. There is a restriction on the maximum data record that can be sent or received from the AS/400 system because of this data.

The following formula determines the maximum record length that can be transferred.

```
4096 - (number of fields in the record + 2) = (maximum record length)
```

Data Types

The transfer function supports the following system data types:

- Binary data
- Character data
- Date
- Hexadecimal data
- Packed decimal data
- Time
- Timestamp
- Zoned decimal data

The transfer function supports the following PC data types:

- ASCII (PC CODE for DBCS) numeric data
- BASIC numeric data, including:
  - Double-precision data
  - Integer data
  - Single-precision data
- Binary data
- Character data, including:
  - ASCII (PC CODE for DBCS)
  - EBCDIC
- Hexadecimal data
- Packed decimal data
- Zoned decimal data

ASCII (PC CODE for DBCS) Numeric Data

The transfer function defines ASCII (PC CODE for DBCS) numeric data to represent any numeric value stored in ASCII (PC CODE for DBCS) format. This is not a valid AS/400 system data type. For example, the number –123.45 in ASCII (PC CODE for DBCS) format is:

2D 31 32 33 2E 34 35

The decimal point and sign are stored explicitly for ASCII numeric (PC CODE for DBCS) data. The character on the left displays the sign (space or plus (+) for positive, minus (–) for negative). Leading zeros to the left of the decimal point change to spaces. The decimal point, if any, is added in the correct position.

BASIC sequential and DIF file types also support another form of ASCII numeric data called exponential numbers. For DBCS, exponential numbers are supported in PC CODE by BASIC sequential and SYLK file types.

An exponential number is a decimal number followed by the letter E or D and a signed integer of two or three digits. E represents a single-precision number and D represents a double-precision number. The exponent portion (E or D and the integer) represents "times 10 to the power of the integer specified."

For example, the number –1.0E+03 (representing -1.0×10^3 in ASCII (PC CODE for DBCS) numeric format) is:

2D 31 2E 30 45 2B 30 33

For example, the number 9.5D–15 (representing 9.5 x 10^{-15} in ASCII (PC CODE for DBCS) numeric format) is:

39 2E 35 44 2D 31 35

Binary Data

This data represents signed or unsigned numbers in two complementary forms. Binary numbers of 1, 2, 3, or 4 bytes in length are allowed on the personal computer, but the AS/400 system allows only 2 or 4 bytes in length. The bit on the left side of the high order bit determines the sign of the number (0 for positive, 1 for negative). The system stores the data with the high-order byte on the left side of the field while the personal computer stores the data with the high-order byte in the right-hand position of the field.

The decimal position, if specified by the file description, represents the number of decimal digits to the right of the decimal point. The file description specifies the presence of a decimal position.

For example, the binary number 0011 1011 1111 0101 (X'3BF5') is equivalent to the decimal number 15349, and the binary number 1111 1111 1011 0100 (X'FFB4') is equivalent to the decimal number –76.

Character Data for SBCS

You can think of this data as a string of bits that represents particular characters and symbols.

The tables used to translate characters from ASCII to EBCDIC and from EBCDIC to ASCII contain the following kinds of values:

- Values where the personal computer ASCII characters and AS/400 EBCDIC characters match exactly
- Values where a substitute character is chosen for a character that cannot be translated

The transfer function uses tables to translate data from ASCII to EBCDIC and EBCDIC to ASCII. You can change these default tables using the translation table utility (TRTABLE).

Note: ASCII (SBCS) and PC CODE (DBCS) data include date, time, and timestamp types if converted. EBCDIC data includes date, time, and timestamp if not converted.

Character Data for DBCS

The single-byte character set (SBCS) and double-byte character set (DBCS) exist on the personal computer and the AS/400 system. On the personal computer, a half space character is represented by a single byte and a full space character is represented by a double byte. The two types of characters are identified by the first byte value.

The exact translation from PC CODE DBCS data to EBCDIC DBCS data and from EBCDIC DBCS data to PC CODE DBCS data is based on the corresponding translation tables.

Note: ASCII (SBCS) and PC CODE (DBCS) data include date, time, and timestamp types if converted. EBCDIC data includes date, time, and timestamp if not converted. Bracketed DBCS is a character string in which each character is represented by 2 bytes. The character string starts with a shift-out (SO) character and ends with a shift-in (SI) character. Bracketed DBCS includes the DBCS-open(O), -only(J), and -either(E) data types.

On the AS/400 system, the following DBCS character data types are allowed:

DBCS-Open Data: In DBCS-open data, SBCS and bracketed DBCS characters can be mixed.

DBCS-Only Data: In DBCS-only data, only bracketed DBCS characters exist.

DBCS-Either Data: In DBCS-either data, either bracketed DBCS or SBCS characters exist, but not both. The data type is identified by the first character.

DBCS-Graphic Data: DBCS-graphic data contains a sequence of double-byte characters that does not include the shift-out or shift-in characters. Graphic data can only be compared with graphic data.

DBCS-graphic data gives you the ability to use double byte characters without shift-out and shift-in characters.

Double-Precision Data

Double-precision data is defined only for the personal computer. The AS/400 system does not support this data type. BASIC applications use double-precision data. This data type is a number in the range of $2.938735877055719 \times 10^{-39}$ to $1.701411834604692 \times 10^{38}$ positive or negative. Double-precision numbers are stored in 8 bytes, with 7 bytes representing the mantissa, and 1 byte representing the exponent.

Date, Time, and Timestamp Data Types

Date, time, and timestamp values can be used in certain arithmetic and character operations and are compatible with certain character constants, but they are neither characters nor numbers.

A date is a three-part value (year, month, and day) designating a point in time on the calendar. The range of the year is 0001 to 9999. The range of the year for a non-SAA format is 1940 to 9999. The range of the month is 1 to 12. The range of the day is 1 to *x*, where *x* depends on the month.

A time is a three-part value (hour, minute, and second) designating a time of day under a 24-hour clock. The range of the hour is 0 to 24 and the range of the other values is 0 to 59.

A timestamp is a seven-part value (year, month, day, hour, minute, second, and microsecond) that represents a date and time including the specified microseconds. The maximum length of the timestamp is a character string of 26.

Dates, times, and timestamps can be assigned to result fields. A valid character string representation of a date can be compared with a date field, or a valid character string representation of a time can be compared with a time field.

Hexadecimal Data

You can think of this data as a string of bits representing base 16 numbers. For example, you can represent X'3D' with the following string of bits:

0011 1101

Integer Data

Integer data is defined only for the personal computer. BASIC applications use integer data. Integer data is stored in 2 bytes and represents a whole number in the range –32768 to 32767.

Packed Decimal Data

For both the AS/400 system and the personal computer, each half-byte represents a value from 0 through 9. The hexadecimal value in the half-byte on the right side of the right-hand byte specifies the sign.

For the AS/400 system, a value of X'B' or X'D' in this half-byte represents a negative number.

For DOS random files, only the last half-byte (the half-byte that contains the sign) is changed. For the sign half-byte, the personal computer uses X'3' to indicate a positive number or X'B' to indicate a negative number.

For example, X'0865431F' appears as X'08654313'.

For DOS random type 2 files, the last half-byte (the half-byte that contains the sign) is not changed. The sign convention used on the personal computer and on the host system is the same.

For example, X'0865431C' appears as X'0865431C'.

The decimal position, if specified, represents the number of decimal digits to the right of the decimal point. The presence of a decimal position is specified in the file description.

Single-Precision Data

Single-precision data is defined only for the personal computer. The AS/400 system does not support this data type. BASIC applications use single-precision data. This data type is a number in the range of 2.938736×10^{-39} to 1.701412×10^{38} positive or negative. Single-precision numbers are stored in 4 bytes, with 3 bytes representing the mantissa, and 1 byte representing the exponent and sign.

Zoned Decimal Data

This data is represented in a form in which each byte corresponds to one decimal digit. Each of these bytes is stored in character form. For example, the digit 7 is stored on the AS/400 system as F7, which is the EBCDIC representation, and is stored on the personal computer as 37, which is the ASCII (PC CODE for DBCS) representation.

The size of each digit is determined by its half-byte on the right side. Valid values for the half-bytes are decimal 0 through 9.

The sign in both the AS/400 system and personal computer zoned decimal fields is specified by the hexadecimal value in the left half-byte of the right byte of the field. For the AS/400 system, an X'B' or X'D' in this half-byte represents a negative number (for example, X'F6D2' represents -62).

For DOS random files, zoned decimal fields from the system change from EBCDIC to ASCII (EBCDIC to PC CODE for DBCS), as do character fields, except that the sign half-byte in the PC field is changed to an X'3' to indicate a positive number or an X'B' to indicate a negative number.

For DOS random type 2 files, zoned decimal fields from the system change from EBCDIC to ASCII (EBCDIC to PC CODE for DBCS), as do character fields, except that the sign half-byte in the PC field is changed to an X'3' to indicate a positive number or an X'7' to indicate a negative number.

The decimal position, if specified, represents the number of decimal digits to the right of the decimal point and is specified by the file description.

Personal Computer File Types

The following PC file types are supported:

- ASCII (PC CODE for DBCS) text files
- BASIC random files
- BASIC sequential files
- DIF files
- SYLK files (DBCS only)
- DOS random files
- DOS random type 2 files
- No-conversion files

ASCII (PC CODE for DBCS) Text Files

ASCII (PC CODE for DBCS) text files are normally used with programs that work with text (such as editors and print routines). The characteristics of an ASCII (PC CODE for DBCS) text file are the following:

- Records consist of ASCII (PC CODE for DBCS) characters.
- A carriage return character (X'0D') and a line feed character (X'0A') delimits each record from the next. The character for end-of-file is X'1A'.
- PC Records in an ASCII (PC CODE for DBCS) file may be variable in length due to truncation of trailing blanks at the end of an AS/400 record.

Transferring Data to ASCII (PC CODE for DBCS) Text Files: When creating an ASCII (PC CODE for DBCS) text file, the data coming from the AS/400 system changes as follows:

- Hexadecimal fields change to equivalent ASCII (PC CODE for DBCS) characters for each half-byte. For example, X'D3' expands to ASCII (PC CODE for DBCS) 4433 and writes to the file. When displayed by an editor or printed, the string appears as D3.
- EBCDIC character fields change byte by byte and are mapped into ASCII (PC CODE for DBCS) characters as defined by the translation tables.
- Date, time, and timestamp data are mapped into ASCII (PC CODE for DBCS) characters as defined by the translation tables.

- Variable length and null fields are converted to fixed lengths, and trailing blanks (for character, hexadecimal, date, time, and timestamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- EBCDIC single-byte characters, for which there are no equivalent PC CODE characters for DBCS, are converted to X'00'. EBCDIC double-byte characters, for which there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages). Double-byte characters that are not DBCS-graphic type characters and are not surrounded with shift-out and shift-in characters are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).

  Note: Some EBCDIC nondisplayable characters are translated into ASCII (PC CODE for DBCS) control characters on the personal computer. If EBCDIC character fields contain nondisplayable data, you may get unexpected results and your ASCII (PC CODE for DBCS) text file may appear to be corrupted.

  For example, X'05' in an EBCDIC field is translated to an ASCII (PC CODE for DBCS) X'09', which is an ASCII (PC CODE for DBCS) control character for horizontal tab. Most PC text editors process this tab character so that the data in your PC text file appears to be shifted to the right when viewed. If one of the EBCDIC characters is X'3F', it translates into an ASCII (PC CODE for DBCS) X'1A', which is interpreted as end-of-file by the PC text editor. You may not be able to view any of the remaining records that were transferred to your ASCII (PC CODE for DBCS) text file.

  One possible solution to this problem is to define these fields on the host system as hexadecimal fields instead of character fields.
- Binary fields change to ASCII (PC CODE for DBCS) numeric. For example, X'FFD3' with no decimal position expands to ASCII (PC CODE for DBCS) 20202020202020202D3435. When displayed by an editor or printed, the string appears as -45.

  Note: The length of the ASCII (PC CODE for DBCS) field depends on the length of the binary field.

  A binary field on the system is either 2 or 4 bytes long. The resulting ASCII (PC CODE for DBCS) field length is from 6 to 11 bytes, including the sign. Another byte is added for a decimal point.

  Figure 3-17 shows the mapping between binary field lengths and their ASCII (PC CODE for DBCS) lengths.

Figure 3-17. Binary-to-ASCII (Binary-to-PC CODE for DBCS) Field Length Mapping

Binary Length	ASCII (PC CODE) Length	Value Range
2	6	-32768 to 32767
4	11	-2147483648 to 2147483647

- Zoned decimal fields are changed to ASCII (PC CODE for DBCS) numeric. For example, EBCDIC F0F0F9F5F2D6 with a field length that indicates two digits to the right of the decimal point expands to ASCII (PC CODE for DBCS) 20202D39352E3236. When displayed by an editor or printed, the string

appears as -95.26. The resulting PC field length is equal to the length of the system field plus one for the sign and one for the decimal point, if specified.

- Packed decimal fields change to ASCII (PC CODE for DBCS) numeric. For example, X'871D' (no decimal point) changes to ASCII (PC CODE for DBCS) 2D383731. When displayed by an editor or printed, the string appears as -871.

  Since two decimal digits are packed into 1 byte, the length of the resulting PC field is equal to two times the length of the AS/400 field, plus one for the decimal point (if specified). This length always includes the sign. A minus sign (-) indicates negative, and a space indicates positive.

Transferring Data from ASCII (PC CODE for DBCS) Text Files: When transferring data from ASCII (PC CODE for DBCS) text files to system files, the data changes as follows:

- ASCII (PC CODE for DBCS) character data changes to EBCDIC character, date, time, or timestamp data (based on the AS/400 field type) on a byte-to-byte basis, or to hexadecimal data by changing two ASCII (PC CODE for DBCS) bytes into one hexadecimal byte.
- ASCII (PC CODE for DBCS) numeric data changes to AS/400 binary, zoned decimal, or packed decimal data, depending on the specified data type.

  The field lengths on the AS/400 system and the personal computer are different because of the explicit way minus signs and decimal points are stored in ASCII (PC CODE for DBCS) numeric fields. Each field changes individually, to ensure that the resulting field length matches the specifications for that field. The transfer function tries to fit the PC data into the system field.
- For null-capable AS/400 fields, null values (except date, time, and timestamp) cannot be reliably detected and are not uploaded. For variable-length AS/400 fields, trailing blanks are removed and the field is converted to the variable length format.

Errors When Transferring Data from ASCII (PC CODE for DBCS) Text Files: When transferring data from a PC ASCII (PC CODE for DBCS) text file to an AS/400 file, the following errors can occur:

- A data field in the ASCII (PC CODE for DBCS) text file is too long for a field in the AS/400-defined file. In this case, the data is truncated. This occurs when the PC description file defines the character data as longer than the field length specified for the system file.

  If the PC CODE text file includes bracketed DBCS characters, the shift-out and shift-in codes are inserted to the transferring data. As a result, the character data is longer than the field length.

  If the data transfers to an EBCDIC field, this error occurs only if the extra bytes are not spaces.

  If the data transfers to a hexadecimal field, this error occurs only if the extra bytes are not zeros. These extra bytes are truncated so the data fits into the specified field.
- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:

- Numeric data in the PC field does not fit into the specified number of bytes for the field.
- The decimal value of a numeric field contains more digits than were specified for the field.

The value of the field is set to the maximum value possible for the number of bytes and digits specified by the AS/400 system.

- Data in this field has too many decimal positions. The number is rounded. This error occurs when the number of decimal positions in the PC field is greater than the number of decimal positions specified on the AS/400 system. These extra bytes are significant because the data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. This error occurs when:
  - Nonnumeric data is found in a field that the file descriptions defined as numeric. The transfer request ends to prevent transferring incorrect data to the file.
  - ASCII (PC CODE for DBCS) numeric data is found which does not match the format the file description specified. An incorrectly-positioned decimal point position within the field could cause this error.
  - A value other than X'30' through X'39', minus, plus, or decimal point is found. A duplicated decimal point or minus is found. The transfer request ends to prevent transferring incorrect data to the file.
- Data for this field is missing. The default values are used. This error occurs when a data field is defined, but the data is not in the file. This means that the end of the record is reached before all of the defined data is found.

  The field or fields for which data has been defined but not found then fill with default values and transfer to the file. The default values are EBCDIC spaces for character fields, or zeros for numeric and hexadecimal fields.

  To supply your own default values, use the default (DFT) keyword in the data description specifications (DDS) for the file.
- Extra data is found at the end of this record. The extra data does not transfer. Data found at the end of this record and not defined by the system data definitions or PC file description file does not transfer to the system file because no definitions exist to define the data and how it should change.

When transferring data from an ASCII (PC CODE for DBCS) text file to an AS/400 file without using a file description file, any extra data found past the record length specified for the file does not transfer.

BASIC Random Files

BASIC random files are the most general-purpose BASIC file type. They contain fixed-length records with the following:

- No delimiters between fields or records.
- No end-of-file marks.

Transferring Data to BASIC Random Files: When you create a BASIC random file, system data changes as follows:

- Hexadecimal fields do not change.

- Change from a system binary field depends on the field length:
  - Fields of 2 bytes, with no decimal positions to the right of the decimal point, change to 2-byte BASIC integer values. The only change is that the order of the bytes reverses.
  - Fields of 2 bytes, with decimal positions to the right of the decimal point, change to BASIC single-precision numbers.
  - Fields of 4 bytes change to BASIC double-precision numbers.
- EBCDIC character, date, time, and timestamp fields change byte by byte and are mapped into ASCII characters as defined by the translation tables.
- Variable length and null fields are converted to fixed lengths, and trailing blanks (for character, hexadecimal, date, time, and timestamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- For DBCS, EBCDIC character fields change byte by byte into PC CODE equivalents.
  - EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.
  - EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).
  - Double-byte characters, that are not DBCS-graphic type characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).
- Zoned decimal fields change into one of the following BASIC variables depending on the field length and the number of decimal positions:
  - Zoned decimal fields of 4 bytes or less with no positions to the right of the decimal point change to a BASIC integer of an equivalent value. A zoned decimal field of 4 bytes or less, but with a decimal point, falls into the following category.
  - Zoned decimal fields up to 7 bytes (including those that did not fall into the previous category) change to a BASIC single-precision number of an equivalent value.
  - Zoned decimal fields greater than 7 bytes change to a BASIC double-precision number of an equivalent value.
- Packed decimal fields change into one of the following BASIC variables depending on the length of the field:
  - Packed decimal fields of 2 bytes or less with no positions to the right of the decimal point change to a BASIC integer of an equivalent value. A packed decimal field of 2 bytes or less, but with a decimal point, falls into the following category (up to 4 bytes).
  - Packed decimal fields of up to 4 bytes (including those that did not fall into the previous category) change to a BASIC single-precision number of an equivalent value.
  - Packed decimal fields greater than 4 bytes change to a BASIC double-precision number of an equivalent value.

  Note: Changes between binary, packed decimal, and zoned decimal numbers with decimal points are not the equivalent to their BASIC number

counterparts because BASIC uses a binary number format that does not always change into exact decimal fractions.

Transferring Data from BASIC Random Files: When transferring data from BASIC random files to system files, the data changes as follows:

- Hexadecimal fields transfer to the system file as unchanged hexadecimal data. The field lengths as stored on the personal computer should be the same as the field lengths as stored on the system.
- ASCII (PC CODE for DBCS) character, date, time, and timestamp data changes to EBCDIC character data on a byte-by-byte basis.
- For null-capable AS/400 fields, null values (except date, time, and timestamp) cannot be reliably detected and are not uploaded. For variable-length AS/400 fields, trailing blanks are removed and the field is converted to the variable length format.
- Numeric fields from BASIC random files (BASIC integers, single-precision floating-point numbers, and double-precision floating-point numbers) change to system binary data, zoned decimal data in EBCDIC format, or packed decimal data in EBCDIC format.

Note: Because the change of floating-point numbers into decimal fractions is not always exact, each number automatically changes into the most precise number possible with respect to the system field length. If you want more precision, specify a larger system field size.

Errors When Transferring Data from BASIC Random Files: When transferring data from a PC BASIC random file to a system file, the following errors can occur:

- Data in this field is too short for the system field. The data is padded. This error occurs when the file contains character or hexadecimal data shorter than the field length specified on the system. This error can occur if the PC field is defined as shorter than the system, or if the data in the last record of the file is too short. Character fields are padded on the right with EBCDIC spaces, and hexadecimal fields are padded with zeros.
- Data in this field is too long for the system field. The data is truncated. This error occurs when the personal computer file description file defines character or hexadecimal data as longer than the field length specified on the system.

  If the PC CODE text file includes bracketed DBCS characters, the shift-out code (X'0E') and the shift-in code (X'0F') are inserted in the transferred data. As a result, the character data is longer than the field length.

  For character data, this error occurs only if the extra bytes are not spaces. For hexadecimal data, this error occurs only if the extra bytes are not zeros. These extra bytes are then truncated so that the data fits into the specified AS/400 field.
- The value of numeric data is too large for the system field. The maximum number is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the system field.
  - The decimal value of a numeric field contains more digits than are specified for the system field.

- Data in this field has too many decimal positions. The number is rounded down to zero. In BASIC random processing, this error occurs if the value of the number is too small to fit into the specified field.

  For example, the number 0.00001 does not fit into a system zoned field specified as being 2 bytes in length and 2 decimal positions to the right of the decimal point. In this example, the resulting value is zero.

- Data for this field is missing. The default values are used. This error occurs when a data field is defined, but the data is not in the file. This means that the end of the file is reached before all of the defined data is found. For BASIC random files, this error occurs only on the last record in the file, since there are no explicit record delimiters.

  When this error occurs, the field or fields for which data is defined, but not found, fill with default values and transfer to the AS/400 file. These default values are EBCDIC spaces for character fields and zeros for numeric fields.

  To supply your own default values, use the Default (DFT) keyword in the DDS for the file.

When transferring data from a BASIC random file to an AS/400 file, any data shorter than the record length defined for the system file is padded with EBCDIC spaces.

Because there are no record delimiters in BASIC random files, this error can occur only on the last record of the file. This probably indicates that the record length of the system file does not match the record length of the personal computer file.

For DBCS, EBCDIC character fields change byte by byte into PC CODE equivalents.

- EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.
- EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).
- Double-byte characters, that are not DBCS-graphic type characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).

BASIC Sequential Files

BASIC uses BASIC sequential files for sequential processing (for example, INPUT and WRITE statements). The fields written are considered either character or numeric. Characteristics of BASIC sequential files are as follows:

- Both numeric and character fields are written as displayable characters. However, character strings are distinguished from numeric strings by the ASCII (PC CODE for DBCS) double quotation marks (X'22') that surround them.

  Therefore, character data in BASIC sequential files cannot contain ASCII (PC CODE for DBCS) double quotation marks, because they are interpreted as the end of the character string.

- Fields are delimited by ASCII (PC CODE for DBCS) commas (X'2C'). Therefore, commas are not allowed as date, time, or decimal separators.
- Each record is delimited from the next by a carriage return character (X'0D') and a line feed character (X'0A'). The end-of-file character is X'1A'.

- Records and fields are variable length.

Transferring Data to BASIC Sequential Files: The following list describes how AS/400 data created by a BASIC-sequential-file-defined data definition changes:

- Hexadecimal fields change to equivalent ASCII (PC CODE for DBCS) characters for each half-byte. Double quotation marks surround them.

  For example, X'F3' expands to ASCII (PC CODE for DBCS) 22443322 and writes to the file.

- EBCDIC character, date, time, and timestamp fields change byte by byte and are mapped into ASCII (PC CODE for DBCS) characters as defined by the translation tables. ASCII (PC CODE for DBCS) double quotation marks are added before and after the character string.

- Null fields are represented by the absence of the field (comma comma, or by a single comma if the null field is the last field of the record).

- For null fields, successive commas in the file will result in a null value being sent to the AS/400 system if the field is null capable.

- In variable length fields, if the AS/400 field is variable length, the field will be converted to the AS/400 variable length format.

- For DBCS, EBCDIC character fields change byte by byte into PC CODE equivalents.

  – EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.

  – EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).

  – Double-byte characters, that are not DBCS-graphic type characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).

- Binary fields change to ASCII (PC CODE for DBCS) numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, X'FFD3' appears as ASCII (PC CODE for DBCS) 2D3435. When displayed on an ASCII (PC CODE for DBCS) device, the string appears as -45.

- Zoned decimal fields change to ASCII (PC CODE for DBCS) numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, EBCDIC F0F0F9F5F2D6 with a field length that indicates two digits to the right of the decimal point expands to ASCII (PC CODE for DBCS) 2D39352E3236. The string appears as -95.26 when an editor displays it or it prints.

- Packed decimal fields change to ASCII (PC CODE for DBCS) numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, X'871F' (no decimal point) changes to ASCII (PC CODE for DBCS) 383731. The string appears as 871 when an editor displays it or it prints.

Transferring Data from BASIC Sequential Files: When transferring data from BASIC sequential files to AS/400 files, the data changes as follows:

- ASCII (PC CODE for DBCS) character, date, time, and timestamp data changes to EBCDIC character data on a byte-by-byte basis and to hexadecimal by changing 2 ASCII (PC CODE for DBCS) bytes into 1 hexadecimal byte.
- ASCII (PC CODE for DBCS) numeric data translates to system binary, zoned decimal, or packed decimal data, depending on the specified data type. The lengths of the system data and the PC data may be different because the minus signs and decimal points are stored in ASCII (PC CODE for DBCS) numeric fields, and leading and trailing spaces are stripped away.

  In addition, BASIC may create exponential numbers in these files. The transfer function also changes these numbers.

  Each translated field is individually verified to ensure that the resulting field length matches the specifications for that field. The transfer function tries to fit the PC data into the system field.

Errors When Transferring Data from BASIC Sequential Files: When transferring data from a PC BASIC sequential file to a AS/400-defined file, the following errors can occur:

- Data in this field is too long for the AS/400 field. The data is truncated. The PC file description file defines character data as longer than the field length specified for the file.

  If the PC CODE text file includes bracketed DBCS characters, the shift-out and shift-in codes are inserted in the transferred data. As a result, the character data is longer than the field length.

  If the data transfers to an EBCDIC field, this error occurs only if the extra bytes are not spaces. If the data transfers to a hexadecimal field, this error occurs only if the extra bytes are not zeros. These extra bytes are truncated so that the data fits into the specified AS/400 field.
- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the system field.
  - The decimal value of a numeric field contains more digits than were specified for the system field.

  The value of the field is set to the maximum value possible for the number of bytes and digits specified by the AS/400 system.
- Data in this field has too many decimal positions. The number is rounded. This error occurs when the number of decimal positions in the PC field is greater than the number of decimal positions specified on the system. The extra bytes are significant since the data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. This error occurs when a field defined as numeric by the file descriptions contains nonnumeric data. This could also result if a character or hexadecimal field contains a numeric field, or if a numeric (zoned, packed, or binary) field contains a character field.

When this error occurs, the transfer request ends to prevent transferring incorrect data to the system file.

- Data for this field is missing. The default values are used. This error occurs when a data field is defined, but the data is not in the file. This means that the end of the record is reached before all of the defined data is found.

  When this error occurs, the field or fields for which data has been defined, but not found, fill with default values and transfer to the AS/400 file. These default values are EBCDIC spaces for character fields, or zeros for numeric fields.

  To supply your own default values, use the default (DFT) keyword in the DDS for the file.

- Data in this field exceeds PC field size. The data is lost. This error occurs when extra data, not defined by the PC file description file, is found at the end of a character field. The extra bytes truncate and do not transfer to the system file.

- Extra data found at the end of the record. The extra data does not transfer. This error occurs when extra data is found at the end of the record, and has not been defined by the system data definitions or PC file description file. This extra data does not transfer to the system because no definitions exist to define the data and describe how it should change.

Data Interchange Format Files

Data Interchange Format (DIF) files represent data in rows and columns. DIF files contain character and numeric data (positive and negative decimal numbers).

The DIF file format is used for data interchange between spreadsheet programs and other application programs.

The transfer function supports only the following two data types within DIF files:

- Character data: The data in a character cell (think of a **cell** as one field in one record) must be enclosed in double quotation marks if there is an embedded space in the string. However, if the string begins with a quotation mark, it must also end with a quotation mark.
- Numeric data: The numeric data supported by the transfer function consists of a decimal number that may contain a minus sign and a decimal point. The transfer function also supports exponential numeric data.

Transferring Data to DIF Files: When creating a DIF file, system data changes as follows:

- Hexadecimal fields change to equivalent ASCII characters for each half-byte. Double quotation marks surround them.
- EBCDIC character, date, time, and timestamp data changes byte by byte and is mapped into ASCII characters as defined by the translation tables.

  No modifications are made to character data when it is transferred from the AS/400 system to the personal computer file. However, the translation from EBCDIC to ASCII does occur. ASCII double quotation marks are added before and after the character string.

- Binary fields change to ASCII numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

- Zoned decimal fields change to ASCII numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, EBCDIC F0F0F9F5F2D6 with a field length that indicates two digits to the right of the decimal point expand to ASCII 2D39352E3236. When displayed or printed, the string appears as -95.26.

- Packed decimal fields change to ASCII numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, X'871D' (no decimal point) changes to ASCII 2D383731. When displayed or printed, the string appears as -871.

- If untranslatable data is found, the entire field becomes an error cell. An **error cell** results when untranslatable data is found when a DIF file is created or when a not valid calculation is done using the DIF file with a spreadsheet program.

Transferring Data from DIF Files: If an error cell is found when data transfers from a DIF file to the AS/400 system, one of the following can occur, depending on the type of data in the file:

- If the system field is a character (EBCDIC) field, it fills with untranslatable characters (hexadecimal zeros) and transfers to the system. A message appears, telling you how many bytes of untranslatable data have transferred.
- If the system field is a hexadecimal, zoned, packed, or binary field, you receive an error message telling you that the data in this cell is incorrect, and that the data did not transfer to the system.

When transferring data from a system file to a DIF file, the field names are placed in the first record and you can consider them column headings. When transferring DIF files back to the system, the first row must either be these field names (exactly as they are defined on the system) or data. If the first row does not consist of field names, the file processes as if it contains only data.

No DIF header information is used when sending the file to the AS/400 system. To correctly transfer a DIF file to the system, the file must be in the correct format (row and column). It is essential that the field names, if present, make up the first row of data. The subsequent records make up the remaining rows of data. Therefore, when attempting to transfer the data to the AS/400 system, the file must be saved in the same format as originally created by the transfer function.

When transferring data from DIF files to AS/400 files, the data changes as follows:

- ASCII character, date, time, and timestamp data is changed to EBCDIC character data or to hexadecimal data. ASCII-to-EBCDIC change is done on a byte-by-byte basis. ASCII-to-hexadecimal change is done by changing 2 ASCII bytes to 1 hexadecimal byte.
- ASCII numeric data changes to system binary, zoned decimal, or packed decimal data, depending on the data type the system specifies.

  The lengths of the fields on the system and the personal computer may be different because of the explicit way minus signs and decimal points store in ASCII numeric fields. This means that each field changes individually, to ensure that the resulting field length matches the system specifications for that field. The transfer function tries to fit the PC data into the system field.

- In null fields, a NULL DIF character field will result in a null value being sent to the AS/400 field if the field is null-capable.
- If the AS/400 field is variable length, the field will be converted to the AS/400 variable length format.

Errors When Transferring Data from DIF Files: When transferring data from a PC DIF file to a system file with data definitions, the following errors can occur:

- DIF file is incomplete or incorrect. The DIF file does not follow the standard DIF format. Processing ends, and no more records transfer.
- Data in this field is too long for the AS/400 field. The data is truncated. The PC file description file defines character or numeric data as longer than the field length specified for the system file.

  For character data, this error occurs only if the extra bytes are not spaces. For hexadecimal data, this error occurs only if the extra bytes are not zeros. The extra bytes truncate so that the data fits into the specified AS/400 field.
- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the AS/400 field.
  - The decimal value of a numeric field contains more digits than are specified for the system field.

  The value of the field is set to the maximum value possible for the number of bytes and digits the system specifies.
- Data in this field has too many decimal positions. The number is rounded. The number of decimal positions in the PC field is greater than the number of decimal positions specified on the system. The data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. One of the following has occurred:
  - A numeric field contains nonnumeric data.
  - A character or hexadecimal field contains a numeric field or a numeric (zoned, packed, or binary) field contains a character field.
  - An AS/400 hexadecimal or numeric (zoned, packed, or binary) field contains a DIF error cell.

  When this error occurs, the transfer request ends to prevent the transfer of incorrect data to the system file.
- Data for this field is missing. This occurs when a data field is defined, but the data is not in the file. This means that the end of the record is reached before all of the defined data is found. If the host field is null-capable then a null is inserted; otherwise, the default values are used.

  When this error occurs, the field or fields for which data is defined, but not found, are filled with default values and are transferred to the system file. These default values are EBCDIC spaces for character fields, or zeros for numeric fields.

  To supply your own default values, use the Default (DFT) keyword in the DDS for the file.

- Data in this field exceeds the PC field size. The data is lost. This error occurs when extra data, not defined by the PC file description file, is found at the end of a character field. The extra bytes truncate and do not transfer to the system file.
- Extra data is found at the end of this record. The extra data does not transfer. This error occurs when there is extra data at the end of the record, and the AS/400 data definitions or PC file description file have not defined it. This extra data does not transfer to the system because no definitions exist to define the data and how it should change.

Symbolic Link Format Files

Symbolic Link Format (SYLK) files represent data in rows and columns. SYLK files contain character and numeric data (positive and negative decimal numbers).

The SYLK file format is used by the Multiplan** licensed program.

The transfer function supports only the following two data types within SYLK files:

- Character data: The data in a character cell (think of a **cell** as one field in one record) must be enclosed in double quotation marks (X'22') if there is an embedded space in the string. However, if the string begins with a quotation mark, it must also end with a quotation mark.
- Numeric data: The numeric data supported by the transfer function consists of a decimal number that may contain a minus sign or a decimal point. The transfer function also supports exponential numeric data.

Transferring Data to SYLK Files: In data transfer, SYLK files consist of the following parts:

- Header
- Cell width information
- Field names
- Actual data
- Trailer information

A description of each part follows.

- **Header** consists of the following three lines:

  ID;PTF
  F;DG0G10
  B;Y*n*;X*m*

  The first line represents the file that is made by TF (transfer function). The second line represents the default format when displayed in the Multiplan program. The third line represents the actual data area range. The characters *n* and *m* are PC CODE numeric data, indicating the number of rows and the number of columns, respectively.

- **Cell width information** consists of a single line:

  F;W*n1 n2 n3*

  n1, *n2*, and *n3* are PC CODE numeric data, meaning that columns from *n1* to *n2* have the width of *n3*. If the width is ten, *n3* can be omitted. If all columns have a width of ten, the whole cell width information can be omitted.

Cell width depends on the DDS field size definition. It is always 10 to 32 and follows the rules described in the following table:

Figure 3-18. Cell Width Mapping

Data type	Field Length on the AS/400 system	Cell Width
Character, date, time, timestamp	N	N
Hexadecimal data	N	N x 2
Binary data	2 4 (without a decimal point) 4 (with a decimal point)	10 12 13
Zoned or packed	N (without a decimal point) N (with a decimal point)	N+2 N+3

- **Field names** are displayed on the first row. This information is represented by the following format in the SYLK file.

  C;Y1;X1;K"*fname1*"
  C;X2;K"*fname2*"
  C;X3;K"*fname3*"
  ⋮

  "*fname...*" is the field name. These names are the same as defined by DDS. They are arranged in the same sequence as specified in the SELECT clause in the Download Transfer Request. The first field corresponds to the first column.

  If data is downloaded, field names are created and the actual data begins from the first row.

- **Actual data** consists of a single line:

  C;Y*n*;X*m*;K*data*

  Y*n*;X*m* means the data position, for row and column. The characters *n* and *m* are PC CODE numeric data, representing the row number and the column number, respectively. Y*n*; or X*m*; can be omitted when the cell position in a row or column is the same as the previous cell.

  The actual field data is indicated by *data*. It is PC CODE numeric or character data. The PC CODE character is distinguished from numeric by the surrounding double quotation marks.

  Data length can be longer than the cell width because they are independent of each other in Multiplan. The maximum length is 254 bytes.

- **Trailer information** consists of a single line:

  E

  This line indicates the end-of-SYLK file.

When creating a SYLK file, system data changes as follows:

- Hexadecimal fields change to equivalent PC CODE characters for each half-byte. Double quotation marks surround them.
- EBCDIC character, date, time, and timestamp data changes byte by byte into PC CODE equivalents.

- EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.
- EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).
- Double-byte characters, that are not DBCS-graphic characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).

PC CODE double quotation marks are added before and after the character string.

- Binary fields change to PC CODE numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, X'FFD3' appears as PC CODE 2D3435. When displayed or printed, the string appears as -45.

- Zoned decimal fields change to PC CODE numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, EBCDIC F0F0F9F5F2D6 with a field length that indicates two digits to the right of the decimal point expands to PC CODE 2D39352E3236. When displayed or printed, the string appears as -95.26.

- Packed decimal fields change to PC CODE numeric. Leading zeros to the left of the decimal point and trailing zeros to the right of the decimal point are removed.

  For example, X'871D' (no decimal point) changes to PC CODE 2D383731. When displayed or printed, the string appears as -871.

- If untranslatable data is found, the entire field becomes an error cell. An **error cell** results when untranslatable data is found when a SYLK file is created or when a calculation that is not valid is done using the SYLK file with a spreadsheet program.
- Null fields are represented by the absence of the cell.
- Fields in this file type are all variable length. The data portion of the field is written to the file (specified by the length field).

Transferring Data from SYLK Files: If an error cell is found when data transfers from a SYLK file to the AS/400 system, one of the following can occur, depending on the type of data in the file:

- If the system field is a character (EBCDIC) field, it fills with untranslatable characters (hexadecimal zeros) and transfers to the system. A message appears, telling you how many bytes of untranslatable data have transferred.
- If the system field is a hexadecimal, zoned, packed, or binary field, you receive an error message telling you that the data in this cell is incorrect, and that the data did not transfer to the system.

When transferring data from a system file to a SYLK file, the field names are placed in the first record and you can consider them as column headings. When transferring SYLK files back to the system, the first row must either be these field names (exactly as they are defined on the system) or data. If the first row does not consist of field names, the file processes as if it contains only data.

No SYLK header information is used when sending the file to the AS/400 system. To correctly transfer a SYLK file to the system, the file must be in the correct format (row and column). It is essential that the field names, if present, make up the first row of data. The subsequent records make up the remaining rows of data. Therefore, when attempting to transfer the data to the AS/400 system, the file must be saved in the same format as originally created by the transfer function.

When transferring data from SYLK files to AS/400 files, the data changes as follows:

- PC CODE character, date, time, and timestamp data is changed to EBCDIC character data or to hexadecimal data. PC CODE-to-EBCDIC change is done on a byte-by-byte basis. PC CODE-to-hexadecimal change is done by changing 2 PC CODE bytes to 1 hexadecimal byte.
- PC CODE numeric data changes to system binary, zoned decimal, or packed decimal data, depending on the data type the system specifies.

  The lengths of the fields on the system and the personal computer may be different because of the explicit way minus signs and decimal points are stored in PC CODE numeric fields. This means that each field changes individually, to ensure that the resulting field length matches the system specifications for that field. The transfer function tries to fit the personal computer data into the system field.
- SYLK cell errors can occur when a field is marked with the SYLK special error designators (string starting with '#') instead of valid data. If this happens and the AS/400 field is character or hexadecimal, the default untranslatable characters are written to the file (X'00' for both EBCDIC ANK and AS/400 hexadecimal, and X'4040' for EBCDIC DBCS character). If the SYLK cell error occurs and the AS/400 field is a numeric data type (zoned, packed, or binary), processing stops and the remainder of the file, including that record, is not sent to the AS/400 system.
- In NULL fields, missing cells result in a null value being sent to the AS/400 system if the field is null capable.
- If the AS/400 field is variable length, the field is converted to the AS/400 variable length format.

When transferring data from SYLK files to the AS/400 system, the following unique processing takes place:

- A SYLK file is constructed of various kinds of information items instead of key characters. However, PC Support/400 picks up only a few particular items when transferring to the AS/400 system. These are the cell values and their positions in rows and columns. Cell value, column number, and row number always follow the K, X, and Y, respectively. Other items, such as width information, expression, and so on, are ignored.
- When transferring data to a SYLK file, the name of the field being transferred is placed in the first record and is considered a column heading. When transferring SYLK files back to the AS/400 system, this same first row must either be the same field names (exactly as they are defined on the AS/400 system) or data. In other words, if the first row does not contain character fields that match exactly with the AS/400 system, it is assumed that the file contains no column headings, and only the data is processed.

- No header information is used when transferring the SYLK file to the AS/400 system.
- If a row that has no data or only blank fields appears, PC Support/400 interprets it as the end of actual data and processing ends normally.
- To properly transfer a SYLK file to the AS/400 system, the file must be in the correct format (row and column). It is essential that each row of data corresponds to one record of the AS/400 file. Thus, when the file is saved on the PC from Multiplan or any other spreadsheet, it must be saved in the same format as it was originally transferred from the AS/400 system.

Errors When Transferring Data from SYLK Files: When transferring data from a personal computer SYLK file to a system file with data definitions, the following errors can occur:

- SYLK file is incomplete or incorrect. The SYLK file does not follow the standard SYLK format. Processing ends, and no more records transfer.
- Data in this field is too long for the AS/400 field. The data is truncated. The PC file description file defines character or numeric data as longer than the field length specified for the system file.
  - For PC CODE-to-EBCDIC transfer, this error occurs if the PC CODE data on the personal computer is defined by the PC file description file to be longer than the AS/400 specified field length or PC CODE data includes DBCS characters.

    For PC CODE-to-hexadecimal transfer, this error occurs if the PC CODE data on the personal computer is defined by the PC file description file to be twice the AS/400 specified field length. This is because two PC CODE bytes are converted to one hexadecimal character.
  - A truncation error occurs only if the extra bytes are not blank (X'20'/X'8140') for PC CODE-to-EBCDIC conversion, or zeros (X'30') for PC CODE-to-hexadecimal conversion. These extra bytes are truncated so that the data will fit into the AS/400 specified field.
- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the AS/400 field.
  - The decimal value of a numeric field contains more digits than are specified for the system field.

  The value of the field is set to the maximum value possible for the number of bytes and digits the system specifies.
- Data in this field has too many decimal positions. The number is rounded. The number of decimal positions in the PC field is greater than the number of decimal positions specified on the system. The data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. One of the following has occurred:
  - A numeric field contains nonnumeric data.
  - A character or hexadecimal field contains a numeric field or a numeric (zoned, packed, or binary) field contains a character field.

– An AS/400 hexadecimal or numeric (zoned, packed, or binary) field contains an SYLK error cell.

When this error occurs, the transfer request ends to prevent the transfer of incorrect data to the system file.

- Data for this field is missing. This occurs when a data field is defined, but the data is not in the file. This means that the end of the record is reached before all of the defined data is found. For null-capable fields, a null is inserted; otherwise, the default values are used.

  When this error occurs, the field or fields for which data is defined, but not found, are filled with default values and are transferred to the system file. These default values are EBCDIC spaces for character fields, or zeros for numeric fields.

  To supply your own default values, use the Default (DFT) keyword in the DDS for the file.

- Data in this field exceeds the PC field size. The data is lost. This error occurs when extra data is found at the end of a field, and the PC file description file has not defined that data. If the data is character data, the extra bytes truncate and do not transfer to the system file. If the data is numeric data, the entire field changes to zeros and transfers to the system file.

- Extra data is found at the end of this record. The extra data does not transfer. This error occurs when there is extra data at the end of the record, and the AS/400 data definitions or PC file description file have not defined it. This extra data does not transfer to the system because no definitions exist to define the data and how it should change.

DOS Random Files

DOS random files are fixed-length files used by the DOS random read and write routines. The characteristics of DOS random files are as follows:

- There are no end-of-record or end-of-file markers.
- Records are delimited by their constant length, relative positions in the file, and the total length of the file.

Note: DOS random and DOS random type 2 files are identical, except for the way in which the signs are represented for packed decimal and zoned decimal numbers.

Transferring Data to DOS Random Files: When creating DOS random file data definitions, system data changes as follows:

- Binary fields on the AS/400 system and the personal computer are represented as twos complement numbers, so it is unnecessary to change individual bytes. The personal computer uses the convention of storing numeric values with the least significant byte in the left-hand byte position. The transfer function then reverses the order of the bytes in binary fields.

  For example, X'CEF3', coming from the system as a 2-byte binary number (representing the value –12557), appears as X'F3CE'.

- EBCDIC character, date, time, and timestamp data changes byte by byte and is mapped into ASCII characters as defined by the translation tables.

- Variable length and null fields are converted to fixed lengths, and trailing blanks (for character, hexadecimal, date, time, and timestamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- For DBCS, EBCDIC character fields change byte by byte into PC CODE equivalents.
  - EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.
  - EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).
  - Double-byte characters, that are not DBCS-graphic type characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).
- Hexadecimal fields do not change.
- Packed decimal fields do not change except for the last half-byte, which contains the sign. The personal computer uses X'3' to indicate a positive number and X'B' to indicate a negative number in the sign half-byte.

  For example, X'0865431F' appears as X'08654313'.
- Zoned decimal fields from the system change from EBCDIC to ASCII (PC CODE for DBCS), as do character fields, except that the sign half-byte in the personal computer changed field is X'3' to indicate a positive number and X'B' to indicate a negative number.

  For example, EBCDIC X'F0F1F2F5F2D6' appears as ASCII X'3031323532B6'.

Transferring Data from DOS Random Files: When transferring data from DOS random files to AS/400 files, the data changes as follows:

- ASCII (PC CODE for DBCS) character, date, time, and timestamp data changes to EBCDIC character data on a byte-by-byte basis.
- Binary fields in the PC file are stored in an order reversed from what the system file expects. These bytes reverse and transfer to the system file.
- Hexadecimal fields do not change. The field length on the system should be the same as the field length on the personal computer.
- For packed decimal fields, only the last half-byte (the byte that contains the sign) is changed. The host system uses X'F' to indicate a positive number and X'D' to indicate a negative number for the sign half-byte.

  For example, X'08654313' appears as X'0865431F'.
- Zoned decimal fields on the personal computer change from ASCII to EBCDIC (PC CODE to EBCDIC for DBCS), as do character fields. The last half-byte (the half-byte that contains the sign) in the personal computer field is changed to X'F' to indicate a positive number and X'D' to indicate a negative number.

  For example, ASCII X'3031323532B6' appears as EBCDIC X'F0F1F2F5F2D6'.
- For null-capable AS/400 fields, null values (except date, time, and timestamp) cannot be reliably detected and are not uploaded. For variable-length AS/400

fields, trailing blanks are removed and the field is converted to the variable length format.

Errors When Transferring Data from DOS Random Files: When transferring data from a DOS random file to an AS/400 file, the following errors can occur:

- Data in this field is too short for the system field. The data is padded. This error occurs when the PC file contains character or hexadecimal data shorter than the specified field length. It also occurs if the length of the personal computer field is defined as less than the system field, or if the data in the last record of the file is too short. Character fields are padded on the right with EBCDIC spaces. Hexadecimal fields are padded on the right with zeros.
- Data in this field is too long for the system field. The data is truncated. This error occurs when the PC file description file defines character or hexadecimal data as longer than the field length specified for the system file.

  If the PC CODE text file includes bracketed DBCS characters, the shift-out and shift-in codes are inserted in the transferred data. As a result, the character data is longer than the field length.

  For character data, this error occurs only if the extra bytes are not spaces. For hexadecimal data, this error occurs only if the extra bytes are not zeros. These extra bytes are truncated so that the data fits into the specified field.
- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the AS/400 field.
  - The decimal value of a numeric field contains more digits than were specified for the AS/400 field.

  The value of the field is set to the maximum value possible for the number of bytes and digits specified by the system.
- Data in this field has too many decimal positions. The number is rounded. This occurs when the number of decimal positions in the PC field is greater than the number of decimal positions specified on the system. The extra bytes are significant since the data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. This error occurs when nonnumeric data appears in a field defined as numeric by the file descriptions. When this occurs, the transfer request ends to prevent transferring incorrect data to the system file.
- Data for this field is missing. The default values are used. This error occurs when a data field is defined, but the data is not in the file. This means that the end of the file is reached before all the defined data is found.

  When this error occurs, the field or fields for which data has been defined, but not found, fill with default values and transfer to the system file. Default values are EBCDIC spaces for character fields, or zeros for numeric fields.

  To supply your own default values, use the default (DFT) keyword in the DDS for the file.

When transferring data from a DOS random file to a system file without data definitions, any data shorter than the record length defined for the system file is padded with EBCDIC spaces.

Because DOS random files have no record delimiters, this error occurs only on the last record and probably indicates that the record length of the system file does not match that of the personal computer file.

DOS Random Type 2 Files

DOS random type 2 files are fixed-length files used by the DOS random read and write routines. The characteristics of DOS random type 2 files are as follows:

- There are no end-of-record or end-of-file markers.
- Records are delimited by their constant length, relative positions in the file, and the total length of the file.

Note: This PC file type is identical to the DOS random file type, except that the internal sign representation for packed decimal and zoned decimal data types follow Systems Application Architecture* (SAA*) standards. Some PC applications, such as applications written in IBM COBOL/2* programming language, need to have the signs for packed decimal and zoned decimal data types represented this way. Use the DOS random type 2 file type for those PC applications.

Transferring Data to DOS Random Type 2 Files: When creating DOS random type 2 file data definitions, system data changes as follows:

- Binary fields on the AS/400 system and the personal computer are represented as twos complement numbers, so it is unnecessary to change individual bytes. The personal computer uses the convention of storing numeric values with the least significant byte in the left-hand byte position. The transfer function then reverses the order of the bytes in binary fields.

  For example, X'CEF3', coming from the system as a 2-byte binary number (representing the value -12557), appears as X'F3CE'.

- EBCDIC character, date, time, and timestamp fields change byte by byte and are mapped into ASCII characters as defined by the translation tables.
- Variable length and null fields are converted to fixed lengths, and trailing blanks (for character, hexadecimal, date, time, and timestamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- For DBCS, EBCDIC character fields change byte by byte into PC CODE equivalents.
  - EBCDIC single-byte characters, if there are no equivalent PC CODE characters, are converted to X'00'.
  - EBCDIC double-byte characters, if there are no equivalent PC CODE characters, are converted to X'BFFB' (in Korean) or X'FCFB' (in other languages).
  - Double-byte characters, that are not DBCS-graphic type characters and are not surrounded by shift-out and shift-in characters, are converted to X'BFFC' (in Korean) or X'FCFC' (in other languages).
- Hexadecimal fields do not change.

- Packed decimal fields do not change. The sign convention used on the personal computer and on the host system is the same.

  For example, X'0865431C' appears as X'0865431C'.

- Zoned decimal fields from the system change from EBCDIC to ASCII, as do character fields. However, the sign half-byte is changed to a 3 to indicate a positive number or a 7 to indicate a negative number when the data is sent to the personal computer.

  For example, EBCDIC X'F0F1F2F5F2D6' appears as ASCII X'303132353276'.

Transferring Data from DOS Random Type 2 Files: When transferring data from DOS random type 2 files to AS/400 files, the data changes as follows:

- ASCII character data, date, time, and timestamp data change to EBCDIC character data on a byte-by-byte basis.
- Binary fields in the PC file are stored in an order reversed from what the system file expects. These bytes reverse and transfer to the system file.
- Hexadecimal fields do not change. The field length on the system should be the same as the field length on the personal computer.
- For packed decimal fields, the last half-byte (the half-byte that contains the sign) is not changed unless the sign half-byte is less than X'A' (represented by values 0 through 9). If the sign half-byte is less than X'A', it is changed to X'F' on the host system.

  For example, X'865431D' appears as X'0865431D', but X'08654318' appears as X'0865431F'.

- Zoned decimal fields on the personal computer change from ASCII to EBCDIC, as do character fields. However, the sign half-byte is changed to a F to indicate a positive number or a D to indicate a negative number when the data is sent to the host system.

  For example, ASCII X'303132353276' appears as EBCDIC X'F0F1F2F5F2D6'.

- For null-capable AS/400 fields, null values (except date, time, and timestamp) cannot be reliably detected and are not uploaded. For variable-length AS/400 fields, trailing blanks are removed and the field is converted to the variable length format.

Errors When Transferring Data from DOS Random Type 2 Files: When transferring data from a DOS random type 2 file to an AS/400 file, the following errors can occur:

- Data in this field is too short for the system field. The data is padded. This error occurs when the PC file contains character or hexadecimal data shorter than the specified field length. It also occurs if the length of the personal computer field is defined as less than the system field, or if the data in the last record of the file is too short. Character fields are padded on the right with EBCDIC spaces. Hexadecimal fields are padded on the right with zeros.
- Data in this field is too long for the system field. The data is truncated. This error occurs when the PC file description file defines character or hexadecimal data as longer than the field length specified for the system file.

If the PC CODE text file includes bracketed DBCS characters, the shift out and shift-in codes are inserted in the transferred data. As a result, the character data is longer than the field length.

For character data, this error occurs only if the extra bytes are not spaces. For hexadecimal data, this error occurs only if the extra bytes are not zeros. These extra bytes are truncated so that the data fits into the specified field.

- The value of numeric data is too large for the system field. The maximum value is used. This error occurs when:
  - Numeric data in the PC field does not fit into the specified number of bytes for the AS/400 field.
  - The decimal value of a numeric field contains more digits than were specified for the AS/400 field.

  The value of the field is set to the maximum value possible for the number of bytes and digits specified by the system.
- Data in this field has too many decimal positions. The number is rounded. This occurs when the number of decimal positions in the PC field is greater than the number of decimal positions specified on the system. The extra bytes are significant since the data rounds up if the first extraneous digit is 5 or greater, and rounds down if it is less than 5.
- Data in this field is incorrect or does not match the PC data type. This error occurs when nonnumeric data appears in a field defined as numeric by the file descriptions. When this occurs, the transfer request ends to prevent transferring incorrect data to the system file.
- Data for this field is missing. The default values are used. This error occurs when a data field is defined, but the data is not in the file. This means that the end of the file is reached before all the defined data is found.

  When this error occurs, the field or fields for which data has been defined, but not found, fill with default values and transfer to the system file. Default values are EBCDIC spaces for character fields, or zeros for numeric fields.

  To supply your own default values, use the default (DFT) keyword in the DDS for the file.

When transferring data from a DOS random type 2 file to a system file without data definitions, any data shorter than the record length defined for the system file is padded with EBCDIC spaces.

Because DOS random type 2 files have no record delimiters, this error occurs only on the last record and probably indicates that the record length of the system file does not match that of the personal computer file.

No-Conversion Files

No-conversion files, defined by the transfer function, consist of data that has not changed. For example, when data transfers from the system to a personal computer no-conversion file, the data transfers exactly as it is stored on the AS/400 system. Date, time, and timestamp data transfers to EBCDIC character data on the personal computer.

Transferring Data to No-Conversion Files: When transferring data from the AS/400 system to a no-conversion file, the data transfers exactly as it is stored on the system.

Variable length AS/400 fields are converted to fixed-length fields, and trailing EBCDIC blanks are added to the maximum length of the field.

Date, time, and timestamp data is converted to EBCDIC character data.

Variable length and null fields are converted to fixed lengths, and trailing EBCDIC blanks (for character, hexadecimal, date, time, and timestamp) or EBCDIC zeros (for binary, zoned, and packed) are added to the maximum length of the field.

Transferring Data from No-Conversion Files: The data types that exist in a no-conversion file are EBCDIC system data types only. When a no-conversion file transfers to the system, the transfer function performs no data change or translation. Date, time, and timestamp data transfers to EBCDIC character data on the personal computer.

However, the transfer function verifies that all numeric data is in the correct EBCDIC format. If any numeric data is found that is not in the correct EBCDIC format, that data and any remaining data does not transfer.

In variable length fields, trailing blanks are removed and the field is converted to the variable length format.

Errors When Transferring Data from No-Conversion Files: When transferring data from a personal computer no-conversion file to a system file, the following errors can occur:

- Data sizes are not equal. When transferring no-conversion files, the length and decimal position specifications for the system and the personal computer must match exactly. If not, no records transfer.
- Data in this field is too short for system field. The data is padded. This error occurs when the personal computer file contains character or hexadecimal data shorter than the field length specified for the system file. This could occur if the data in the last record of the file is too short. Character fields are padded on the right with EBCDIC spaces. Hexadecimal fields are padded with zeros.
- Data in this field is incorrect or does not match the personal computer data type. The transfer request ends to prevent transferring incorrect data to the system file. This error occurs when a field defined by the file descriptions as numeric contains nonnumeric data.

  Note: The data is verified assuming that the data is in EBCDIC format. If you want to transfer data in another format, do not use data definitions or file descriptions, and specify the record lengths defined on the system and the personal computer in the same way.
- Data for this field is missing. The default values are used. This error occurs when a data field has been defined, but the data is not in the file. This error can occur only in the last record of the file, since no-conversion files have no explicit record delimiters.

  When this error occurs, the field or fields for which data has been defined but not found fill with default values and transfer to the system file. These default values are EBCDIC spaces for character fields, or zeros for numeric fields.

To supply your own default values, use the default (DFT) keyword in the DDS for the file.

Sort Sequence Tables

When transferring files from the AS/400 system to a personal computer, you can specify a sort sequence table that defines how the records are to be sorted. The sort sequence affects not only the order in which the records appear, but also other string comparison operations, such as the WHERE option.

The sort sequence table is accessed at run time. The transfer function saves the name of the sort sequence table that was specified, not the contents of the table. If the contents of the table change between transfer requests, you may get different results.

See the *National Language Support Planning Guide* for a list of the language IDs you can specify when you select a shared- or unique-weight sort sequence table and a user-specified language. See the *PC Support/400 User's Guide for DOS* or the *PC Support/400 User's Guide for OS/2* for information on how to specify the sort sequence table.

AS/400 System-to-Personal Computer Performance Considerations

Transferring data from the AS/400 system to the personal computer depends on the following performance considerations:

- The system work load.
- How many records have to be looked at to complete the transfer.
- If more than two files are joined. You need extra AS/400 resources to join records from more than one file.
- If GROUP BY fields are specified.
- If complicated WHERE or HAVING comparisons are specified.
- The relationship between the sort sequence table and the organization of the database.

These factors and others influence the time needed to determine which data should be transferred. For example, the time needed to receive the first record of a transfer in which all the records are chosen is less than the time needed to start transferring a smaller group of records based on complicated WHERE or HAVING values. However, transferring all the records in a large file is sometimes impractical or unnecessary.

The AS/400-to-personal computer transfer function may perform significantly better when transferring data if the *Ignore decimal data errors* option is set to No, particularly in situations where existing indexes can be used.

The AS/400-to-personal computer transfer function uses many functions within the AS/400 system to determine the fastest method of selectively retrieving records. When it selects a smaller group of records to transfer, AS/400-to-personal computer transfer function uses the existing access paths whenever possible to improve performance.

For the AS/400-to-personal computer transfer function to consider using an existing access path (logical file), the access path must meet the following conditions:

- It must be defined to the data that transfers.
- It must have either *DELAY or *IMMED maintenance.

When you meet these conditions, you must then match the transfer request to the access path. The following considerations might be helpful when you define your transfer request:

- The time it takes to select records based on WHERE clause values is less when the following are true of the WHERE field:
  - It is compared to a constant.
  - It is the first key field in an existing access path defined to the data to be transferred.
- A transfer request containing a GROUP BY or ORDER BY clause or both can work better as long as the key fields in the access path are in the same order as specified on the GROUP BY or ORDER BY clauses.
- A transfer request containing a JOIN BY clause can work better as long as:
  - An access path exists over the file that you are joining to.
  - The field you are joining to is a primary key field in the access path.
  - You are not returning records with missing fields.

Transfer Function Translation Tables

The transfer function uses the default PC Support/400 translation tables to translate data from ASCII to EBCDIC and EBCDIC to ASCII (PC CODE to EBCDIC and EBCDIC to PC CODE for DBCS). To create a new translation table or change the values in the default table, use the translation table utility. Refer to "General Information" on page 1-27 for more information.

Chapter 4. Work Station Function

This chapter details characteristics of the work station function and how it works using the DOS and extended DOS operating system. It supplies technical information about the work station function display characteristics and specifications, graphics function handling, and a description of the graphics orders supported by the work station function.

This chapter does **not** discuss the OS/2 Communications Manager 5250 work station feature. For information on using the 5250 work station feature of the OS/2 Communications Manager, refer to the OS/2 operating system manuals.

The 5250WS function of PC Support/400 allows the 5250 data stream to be processed by a non-IBM data stream optimizer program. Using this function may improve response time for 5250 and 3270 remote systems respectively.

The AS/400 3270 remote-attach program currently translates the 5250 data stream for its attached 3x74 remote controller and 3270 displays. The translation process from 5250 to 3270 doubles the amount of data in the data stream, which makes the response times for 3270 remotes understandably longer than for 5250 remotes. The 5250 optimization would reduce the amount of data in the 5250 data stream and thereby also in the 3270 data stream.

Display Characteristics

When you use a personal computer as a display station on the AS/400 system, its functions and display appearance are not identical to the display station it imitates. This section describes these differences.

Attribute Appearances

When you use a personal computer as a 3196 display station, some attributes look different. Figure 4-1 on page 4-2 gives the hexadecimal code and description for each attribute, and a description of how the attribute appears on the 3196 display station and on the personal computer you are using as a 3196 display station. Figure 4-3 on page 4-7 gives the hexadecimal code and description for each attribute, and a description of how the attribute appears on an IBM 5292 Model 2 (5292-2) display station and on the personal computer you are using as a 5292-2 display station.

Some display adapters are more capable of making your personal computer look like an AS/400 display. This capability is know as mainframe interactive (MFI) text mode. MFI text mode can be configured for work station function display sessions by selecting the `Host Field Attributes` option in the session profile menu of the CFGWSF program. When a session is configured for MFI text mode and the display adapter is MFI-capable, the display will look very similar to a 3477 AS/400 display. Use Figure 4-2 on page 4-4 or Figure 4-4 on page 4-9 to configure your work station function display sessions using the CFGWSF program. These tables identify the hexadecimal code and description for each display attribute.

For more information on MFI text mode, see the XGA Function section of the *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces*.

This manual describes the layout of the individual bits within an MFI attribute byte. You can use the CFGWSF program to customize these MFI attribute bytes.

The form of translation for the AS/400 attributes to personal computer attribute styles is controlled by the work station function configuration program (CFGWSF). You can also use the WSFCOLOR function to change the attributes.

Monochrome Display

Figure 4-1 shows the display attributes you can select if you are using a personal computer monochrome display with a monochrome display I/O card installed. Following is a description of the table columns:

- The first column lists the hexadecimal code used by the AS/400 system for the attribute.
- The second column describes the attribute.
- The third column describes the display appearance of a 3196 display station when you specify the attribute.
- The fourth column describes the display appearance of your personal computer display when you specify the attribute while you are using your personal computer as a 3196 display station.

Figure 4-1 (Page 1 of 3). Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display
20	Normal	Green characters on black field	Green characters on black field
21	Reverse	Black characters on green field	Black characters on green field
22	High Intensity	Bright green characters on black field	Bright green characters on black field
23	High Intensity and Reverse	Black characters on bright green field	Black characters on green field
24	Underline	Underline	Green characters on black field, green underline
25	Underline and Reverse	Black characters on green field, black underline	Black characters on green field, no underline
26	Underline and High Intensity	Bright green characters on black field, bright green underline	Bright green characters on black field, bright green underline
27	Blank field	Solid black field, no characters	Solid black field, no characters
28	Blink	Blinking green characters on black field	Blinking green characters on black field
29	Blink and Reverse	Blinking black characters on green field	Blinking black characters on green field
2A	Blink and High Intensity	Blinking bright green characters on black field	Blinking bright green characters on black field
2B	Blink, High Intensity, and Reverse	Blinking black characters on bright green field	Blinking black characters on green field
2C	Blink and Underline	Blinking green characters on black field, green underline	Blinking green characters on black field, green underline

Figure 4-1 (Page 2 of 3). Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display
2D	Blink, Underline, and Reverse	Blinking black characters on green field, black underline	Blinking black characters on green field, no underline
2E	Blink, Underline, and High Intensity	Blinking bright green characters on black field, green underline	Blinking bright green characters on black field, green underline
2F	Blank field	Solid black field, no characters	Solid black field, no characters
30	Column Separators	Vertical green bars between green characters on black field	Green rectangles in null positions on black field
31	Column Separators and Reverse	Vertical black bars between black characters on green field	Black rectangles with black characters on green field
32	Column Separators and High Intensity	Bright green bars between bright green characters on black field	Bright green rectangle for nulls with bright green characters on black field
33	Column Separators, High Intensity, and Reverse	Black bars between black characters on bright green field	Black rectangle for nulls with black characters on green field
34	Column Separators and Underline	Green bars between green underlined characters on black field	Green rectangle for nulls with green underlined characters on black field
35	Column Separators, Underline, and Reverse	Black bars between black characters on green field, black underline	Black rectangle for nulls with black on green field
36	Column Separators, Underline, and High Intensity	Bright green bars between bright green characters on black field	Bright green rectangles for nulls with bright green characters on black field, bright green underline
37	Column Separators and no characters	Green bars on black field, no characters	Green rectangles for nulls on black field, no characters
38	Column Separators and Blink	Green bars between blinking green characters on black field	Blinking green rectangles for nulls with blinking green characters on black field
39	Column Separators, Blink, and Reverse	Black bars between blinking black characters on green field	Blinking black rectangle for nulls with blinking black characters on green field
3A	Column Separators, Blink, and High Intensity	Bright green bars between blinking bright green characters on black field	Bright green rectangle for nulls with blinking bright green characters on black field
3B	Column Separators, Blink, High Intensity, and Reverse	Black bars between blinking black characters on bright green field	Blinking black rectangle for nulls with blinking black characters on green field
3C	Column Separators, Blink, and Underline	Green bars between blinking characters on black field, green underline	Blinking green rectangles for nulls with blinking green characters on black field, green underline
3D	Column Separators, Blink, Underline, and Reverse	Black bars between blinking black characters on green field, black underline	Blinking black rectangles for nulls with blinking black characters on green field, no underline
3E	Column Separators, Blink, Underline, and High Intensity	Bright green bars between blinking bright green characters on black field, bright green underline	Blinking bright green rectangles for nulls with blinking bright green characters on black field, blinking bright green underline

Figure 4-1 (Page 3 of 3). Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display
3F	Column Separators and no characters	Green bars on black field, no characters	Blinking green rectangles for nulls on black field, no characters

MFI Monochrome Display

Figure 4-2 shows the display attributes you can select if you are using a personal computer monochrome display with a display adapter that supports the mainframe interactive (MFI) text mode. Following is a description of the table columns:

- The first column lists the hexadecimal code used by the AS/400 system for the attribute.
- The second column describes the attribute.
- The third column describes the display appearance of a 3196 display station when you specify the attribute.
- The fourth column describes the display appearance of your personal computer display when you specify the attribute while you are using your personal computer as a 3196 display station.

Figure 4-2 (Page 1 of 4). MFI Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display[1]
20	Normal	Green characters on a black field	Green characters on a black field
21	Reverse	Black characters on a green field	Black characters on a green field
22	High Intensity	Bright green characters on a black field	Bright green characters on a black field
23	High Intensity and Reverse	Black characters on a bright green field	Black characters on a bright green field
24	Underline	Green characters on a black field, green underline	Green characters on a black field, green underline
25	Underline and Reverse	Black characters on a green field, black underline	Black characters on a green field, black underline
26	Underline and High Intensity	Bright green characters on a black field, bright green underline	Bright green characters on a black field, bright green underline
27	Blank field	Solid black field, no characters	Solid black field, no characters
28	Blink	Blinking green characters on a black field	Blinking green characters on a black field
29	Blink and Reverse	Blinking black characters on a green field	Blinking black characters on a green field
2A	Blink and High Intensity	Blinking bright green characters on a black field	Blinking bright green characters on a black field
2B	Blink, High Intensity, and Reverse	Blinking black characters on a bright green field	Blinking black characters on a bright green field

Figure 4-2 (Page 2 of 4). MFI Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display[1]
2C	Blink and Underline	Blinking green characters on a black field, green underline	Blinking green characters on a black field, green underline
2D	Blink, Underline, and Reverse	Blinking black characters on a green field, black underline	Blinking black characters on a green field, black underline
2E	Blink, Underline, and High Intensity	Blinking bright green characters on a black field, green underline	Blinking bright green characters on a black field, green underline
2F	Blank field	Solid black field, no characters	Solid black field, no characters
30	Column Separators	Green vertical bars between green characters on a black field	Green dots between green characters on a black field
31	Column Separators and Reverse	Black vertical bars between black characters on a green field	Black dots between black characters on a green field
32	Column Separators and High Intensity	Bright green vertical bars between bright green characters on a black field	Bright green dots between bright green characters on a black field
33	Column Separators, High Intensity, and Reverse	Black vertical bars between black characters on a bright green field	Black dots between black characters on a bright green field
34	Column Separators and Underline	Green vertical bars between green characters on a black field, green underline	Green dots between green characters on a black field, green underline
35	Column Separators, Underline, and Reverse	Black vertical bars between black characters on a green field, black underline	Black dots between black characters on a green field, black underline
36	Column Separators, Underline, and High Intensity	Bright green vertical bars between bright green characters on a black field, bright green underline	Black dots between bright green characters on a black field, bright green underline
37	Column Separators and no characters	Green vertical bars on a black field, no characters	Green dots on a black field, no characters
38	Column Separators and Blink	Green vertical bars between blinking green characters on a black field	Green dots between blinking green characters on a black field
39	Column Separators, Blink, and Reverse	Black vertical bars between blinking black characters on a green field	Black dots between blinking black characters on a green field
3A	Column Separators, Blink, and High Intensity	Bright green vertical bars between blinking bright green characters on a black field	Bright green dots between blinking bright green characters on a black field
3B	Column Separators, Blink, High Intensity, and Reverse	Black vertical bars between blinking black characters on a bright green field	Black dots between blinking black characters on a bright green field
3C	Column Separators, Blink, and Underline	Green vertical bars between blinking green characters on a black field, green underline	Black dots between blinking green characters on a black field, green underline

Figure 4-2 (Page 3 of 4). MFI Monochrome Display Attributes

Hex Code	Attribute	3196 Display Station	Personal Computer Monochrome Display[1]
3D	Column Separators, Blink, Underline, and Reverse	Black vertical bars between blinking black characters on a green field, black underline	Green dots between blinking black characters on a green field, black underline
3E	Column Separators, Blink, Underline, and High Intensity	Bright green vertical bars between blinking bright green characters on a black field, bright green underline	Black dots between blinking bright green characters on a black field, bright green underline
3F	Column Separators and no characters	Green vertical bars on a black field, no characters	Green dots on a black field, no characters

Note:

[1] In the Personal Computer Monochrome Display column, when the display adapter is in MFI text mode, column separators are not represented by vertical bars but by the first and last pels (picture elements) of the underline scan line of the character. The column separators will appear as dots to the lower left and right of each character.

Color Display

Figure 4-3 on page 4-7 shows the display attributes you can select if you are using a personal computer color display with any of the following installed:

- Color Graphics Adapter (CGA) card
- Enhanced Graphics Adapter (EGA) card
- Multi-Color Graphics Array (MCGA) on Personal System/2* (PS/2*) Models 25 and 30 system board
- Video Graphics Array (VGA) on PS/2 Models 30-286, 50, 55, 60, 65, 70, and 80 system board
- Personal System/2 Display Adapter
- Personal System/2 Display Adapter 8514/A
- Personal System/2 Extended Graphics Array (XGA)

Note: You can use the WSFCOLOR.EXE utility to change the colors of each display session. By using WSFCOLOR.EXE, you do not have to select the display attributes shown in Figure 4-3 on page 4-7.

Following is a description of the table columns:

- The first column lists the hexadecimal code used by the AS/400 system for the attribute.
- The second column describes the attribute.
- The third column describes the display appearance of an IBM 5292 Model 2 (5292-2) display station when you specify the attribute.
- The fourth column describes the display appearance of your personal computer color display when you specify the attribute while you are using your personal computer as a 5292-2 display station.

Figure 4-3 (Page 1 of 2). Color Display Attributes

Hex Code	Attribute	5292-2 Display Station	Personal Computer Color Display[1]
20	Normal	Green characters on black field	Green characters on black field
21	Reverse	Black characters on green field	Black characters on green field
22	High Intensity	White characters on black field	White characters on black field
23	High Intensity and Reverse	Black characters on white field	Black characters on white field
24	Underline	Green characters on black field with blue underline	Green characters on black field with green underline characters for nulls
25	Underline and Reverse	Black characters on green field with blue underline	Black characters on green field with black underline characters for nulls
26	Underline and High Intensity	White characters on black field with blue underline	White characters on black field with white underline characters for nulls
27	Blank field	Solid black field, no characters	Solid black field, no characters
28	Blink	Red characters on black field	Red characters on black field
29	Blink and Reverse	Black characters on red field	Black characters on red field
2A	Blink and High Intensity	Blinking red characters on black field	Blinking red characters on black field
2B	Blink, High Intensity, and Reverse	Blinking black characters on a red field	Blinking black characters on a red field
2C	Blink and Underline	Red characters on black field with blue underline	Red characters on black field with red underline characters for nulls
2D	Blink, Underline, and Reverse	Black characters on red field with blue underline	Black characters on red field with black underline characters for nulls
2E	Blink, Underline, and High Intensity	Blinking red characters on black field with blue underline	Blinking red characters on black field with blinking red underline characters for nulls
2F	Blank field	Solid black field, no characters	Solid black field, no characters
30	Column Separators	Blue dots in lower corners with turquoise characters on black field	Cyan rectangle characters for nulls with cyan characters on black field

[1] In the Personal Computer Color Display column, the word "nulls" indicates that underline characters (_) are shown on the screen if nothing is typed over them.

Figure 4-3 (Page 2 of 2). Color Display Attributes

Hex Code	Attribute	5292-2 Display Station	Personal Computer Color Display[1]
31	Column Separators and Reverse	Blue dots in lower corners with black characters on turquoise field	Black rectangle characters for nulls with black characters on cyan field
32	Column Separators and High Intensity	Blue dots in lower corners with yellow characters on black field	Yellow rectangle characters for nulls with yellow characters on black field
33	Column Separators and High Intensity	Blue dots in lower corners with black characters on yellow field	Black rectangle characters for nulls with black characters on yellow field
34	Column Separators and Underline	Segmented blue underline with turquoise	Cyan rectangle characters for nulls with cyan characters on black field, no underlines
35	Column Separators, Underline, and Reverse	Segmented blue underline with black characters on turquoise field	Black rectangle characters for nulls with black characters on cyan field, no underline
36	Column Separators, Underline, and High Intensity	Segmented blue underline with yellow characters on black field	Yellow rectangle characters for nulls with yellow characters on black field, no underline
37	Column Separators and no characters	No characters and no column separators	Red rectangle characters for nulls, no other characters
38	Column Separators and Blink	Pink characters on black field, no column separators	Magenta rectangle characters for nulls with magenta characters on black field
39	Column Separators, Blink, and Reverse	Black characters on pink field, no column separators	Black rectangle characters for nulls with black characters on magenta field
3A	Column Separators, Blink, and High Intensity	Blue characters on black field, no column separators	Light blue rectangle characters for nulls with light blue characters on black field
3B	Column Separators, Blink, High Intensity, and Reverse	Black characters on blue field, no column separators	Black rectangle characters for nulls with black characters on blue field
3C	Column Separators, Blink, and Underline	Blue underline with pink characters on black field, no column separators	Magenta rectangle characters for nulls with magenta characters on black field
3D	Column Separators, Blink, Underline, and Reverse	Blue underline with black characters on pink field, no column separators	Black rectangle characters for nulls with black characters on magenta field
3E	Column Separators, Blink, Underline, and High Intensity	Blue underline with blue characters on black field, no column separators	Light blue rectangle characters for nulls with light blue characters on black field
3F	Column Separators and no characters	Black field, no characters	Red rectangle characters for nulls on black field, no other characters

[1] In the Personal Computer Color Display column, the word "nulls" indicates that underline characters (_) are shown on the screen if nothing is typed over them.

MFI Color Display

Figure 4-4 shows the display attributes you can select if you are using a personal computer color display with a display adapter that supports the mainframe interactive (MFI) text mode. Following is a description of the table columns:

- The first column lists the hexadecimal code used by the AS/400 system for the attribute.
- The second column describes the attribute.
- The third column describes the display appearance of an IBM 5292 Model 2 (5292-2) display station when you specify the attribute.
- The fourth column describes the display appearance of your personal computer color display when you specify the attribute while you are using your personal computer as a 5292-2 display station.

Figure 4-4 (Page 1 of 2). MFI Color Display Attributes

Hex Code	Attribute	5292-2 Display Station	Personal Computer Color Display[1]
20	Normal	Green characters on a black field	Green characters on a black field
21	Reverse	Black characters on a green field	Black characters on a green field
22	High Intensity	White characters on a black field	White characters on a black field
23	High Intensity and Reverse	Black characters on a white field	Black characters on a white field
24	Underline	Green characters on a black field with blue underline	Green characters on a black field with green underline
25	Underline and Reverse	Black characters on a green field with blue underline	Black characters on a green field with black underline
26	Underline and High Intensity	White characters on a black field with blue underline	White characters on a black field with white underline
27	Blank field	Solid black field, no characters	Solid black field, no characters
28	Blink	Red characters on a black field	Red characters on a black field
29	Blink and Reverse	Black characters on a red field	Black characters on a red field
2A	Blink and High Intensity	Blinking red characters on a black field	Blinking red characters on a black field
2B	Blink, High Intensity, and Reverse	Blinking black characters on a red field	Blinking black characters on a red field
2C	Blink and Underline	Red characters on a black field with blue underline	Red characters on a black field with red underline
2D	Blink, Underline, and Reverse	Black characters on a red field with blue underline	Black characters on a red field with black underline
2E	Blink, Underline, and High Intensity	Blinking red characters on a black field with blue underline	Blinking red characters on a black field with red underline
2F	Blank field	Solid black field, no characters	Solid black field, no characters
30	Column Separators	Blue dots in lower corners with turquoise characters on a black field	Cyan dots in lower corners with cyan characters on a black field

Figure 4-4 (Page 2 of 2). MFI Color Display Attributes

Hex Code	Attribute	5292-2 Display Station	Personal Computer Color Display[1]
31	Column Separators and Reverse	Blue dots in lower corners with black characters on a turquoise field	Black dots in lower corners with black characters on a cyan field
32	Column Separators and High Intensity	Blue dots in lower corners with yellow characters on a black field	Yellow dots in lower corners with yellow characters on a black field
33	Column Separators, High Intensity, and Reverse	Blue dots in lower corners with black characters on a yellow field	Black dots in lower corners with black characters on a yellow field
34	Column Separators and Underline	Turquoise characters on a black field with segmented blue underline	Cyan characters on a black field with segmented cyan underline
35	Column Separators, Underline, and Reverse	Black characters on a turquoise field with segmented blue underline	Black characters on a cyan field with segmented black underline
36	Column Separators, Underline, and High Intensity	Yellow characters on a black field with segmented blue underline	Yellow characters on a black field with segmented yellow underline
37	Column Separators and no characters	Solid black field, no characters or column separators	Green dots in lower corners with no characters
38	Column Separators and Blink	Pink characters on a black field with no column separators	Magenta characters on a black field with no column separators
39	Column Separators, Blink, and Reverse	Black characters on a pink field with no column separators	Black characters on a magenta field with no column separators
3A	Column Separators, Blink, and High Intensity	Blue characters on a black field with no column separators	Blue characters on a black field with no column separators
3B	Column Separators, Blink, High Intensity, and Reverse	Black characters on a blue field with no column separators	Black characters on a blue field with no column separators
3C	Column Separators, Blink, and Underline	Pink characters on a black field with blue underline and no column separators	Magenta characters on a black field with magenta underline and no column separators
3D	Column Separators, Blink, Underline, and Reverse	Black characters on a pink field with blue underline with no column separators	Black characters on a magenta field with black underline with no column separators
3E	Column Separators, Blink, Underline, and High Intensity	Blue characters on a black field with blue underline and no column separators	Blue characters on a black field with blue underline and no column separators
3F	Column Separators and no characters	Solid black field, no characters or column separators	Green dots in lower corners with no characters

Note:

[1] In the Personal Computer Color Display column when the display adapter is in MFI text mode, column separators are not represented by vertical bars but by the first and last pels (picture elements) of the underline scan line of the character. The column separators will appear as dots to the lower left and right of each character.

Data Stream Size Restriction

The host controls the operation of the keyboard and display with commands and data located within the data stream sent from the host to the display station.

In the case of display stations that are not personal computers, the length of the data stream has no limit from a work station controller's point of view. When you use a personal computer as a display station, the design of work station function limits the size of the data stream. In the case of the DOS real mode version of the work station function, the limit is 5780 characters. In the case of the DOS extended version of the work station function, the default limit is 6144 characters. You can change this default limit to any value up to 65536 characters using the CFGWSF program.

To ease possible concern about this restriction, typical data streams sent to the personal computer from the host are 2000 characters or fewer. The information is provided to inform you that there is a restriction. The restriction may have an effect on a screen that has many fields defined.

If the length of a data stream sent to a personal computer exceeds the limit, you will see the following symptoms:

- The work station function session loses system availability.
- The work station function resets the session.
- The display station's display is cleared of data except for the cursor status line and a message.
- The cursor appears in the upper-right corner.
- The status line appears with the system available indicator off and the system name blank.
- The message indicates that work station function received more data than will fit into its buffer.

To recover if you are using the DOS real mode version of work station function:

- Redesign the display defined by the data.

  You can do this by decreasing the number of fields on the display. This will decrease the size of the data required to define the display.

To recover if you are using the DOS extended version of work station function:

- Redesign the display defined by the data as mentioned in the steps for the DOS real mode version of work station function.

  Note: You can use the CFGWSF program to increase the buffer size of the host data stream, instead of the steps described above.

Graphics Function Handling

Figure 4-5 lists the differences in the ways the IBM 5292 Model 2 (5292-2) and a personal computer using the work station function handle graphics functions. The first column lists the function. The second and third columns describe how IBM 5292-2 work stations and personal computers acting as graphics work stations handle the function.

Figure 4-5 (Page 1 of 2). Display Differences When Using Functions

Function	IBM 5292-2	Personal Computer
Control of graphics displays	Turned on and off using local-select operation functions.	May be turned on and off by entering local select operation and pressing the top row 1 key. Simulated by saving and rebuilding the display. Rebuild time depends on the complexity of the graphics and orders used. This may be 30 seconds or longer. Order stream can overflow the internal buffer causing loss of part of a chart that has been displayed again.
Graphics coordinates	Determined by hardware and application.	Scaled to fit current display width, height, and aspect ratio. Text may be shifted with respect to graphics when alphanumeric data overlays graphics data.
Line style	Determined by hardware and application.	Supplied by virtual device interface (VDI) and may not match.
Line style mapping	Determined by hardware and application.	Specified during configuration.
Color mapping	Determined by hardware and application.	Specified during configuration for personal computer displays with fewer than 8 colors available.
Polygon fill	Reference line: vertical; follow polygon edge; follow +45° from vertical; follow –45° from vertical. Fill mode: solid boundary and style specified in the Set Style order; solid boundary only; fill with style specified in the Set Style order; boundary only with style specified in the Set Style order.	Reference line: not supported by VDI. Fill mode: solid boundary and style specified in the Set Style order and current line style; solid boundary only; fill with style specified in the Set Style order and current line style; boundary only with style specified in the Set Style order and current line style.
Set line weight order	Determined by hardware and application.	Not supported.
Status line control	Always displayed.	May be turned on and off by entering local-select operation 2.
Styled line	Does not write missing or offline segment (area between dashes not painted).	Missing line segments may be painted with background color.
Text attributes and colors	Determined by hardware and application.	May appear different when in graphics operation due to color characteristics of the personal computer hardware.
Write background	Uses current line style and writing mode.	Clears and then paints display with solid line style and replaces writing mode.
Write polymarker	Determined by hardware and application.	Always uses replace function.
XOR and OR writing modes	Determined by hardware and application.	Boolean logic is performed on the color indexes. VDI uses different index values for color selection from the 5292-2. This can result in different output colors.
Blink, underline, and column separators	Determined by hardware and application.	Not supported in graphics operation.

Figure 4-5 (Page 2 of 2). Display Differences When Using Functions

Function	IBM 5292-2	Personal Computer
Vertical scaling factor	Determined by hardware and application.	Can be altered by local-select operation 3.
Select modes	Determined by hardware and application.	Supported for Erase Graphics display and End Graphics processing.

Display Specifications

Figure 4-6 compares the specifications of the IBM 5292 Model 2 (5292-2) color display station with the combinations of displays and adapters available for personal computers.

Figure 4-6 (Page 1 of 2). Display Specifications

Display/Display Adapter	Resolution	Colors	Characters per Line
5292-2	480 by 288	8 of 512	80
5151 EGA	640 by 350	4 pel styles	80
5153 CGA	640 by 200 320 by 200	2 4 of 16 (see note 1)	80 40
5153 EGA	320 by 200 640 by 200	16 (see note 2) 16 (see note 2)	40 80
5154 EGA	640 by 350 640 by 350	4 of 64 16 of 64 (see notes 2 and 3)	80 80
Monochrome Display 8503 with Personal System/2 Display Adapter (see note 4)	320 by 200 640 by 480	64 gray shades (see note 2) 2 gray shades	80 80
Color Displays 8512, 8513, 8514, or 8515 with Personal System/2 Display Adapter (see note 4)	320 by 200 640 by 480	256 (see note 2) 16 (see note 2)	80 80
Monochrome Display 8503 with MCGA	320 by 200 640 by 480	64 gray shades (see note 2) 2 gray shades	80 80
Color Displays 8512, 8513, 8514, or 8515 with MCGA	320 by 200 640 by 480	256 (see note 2) 2	80 80
Monochrome Display 8503 with VGA or XGA	320 by 200 640 by 480	64 gray shades (see note 2) 16 gray shades (see note 2)	80 80
Color Displays 8512, 8513, 8514, or 8515 with VGA or XGA	320 by 200 640 by 480	256 (see note 2) 16 (see note 2)	80 80

Figure 4-6 (Page 2 of 2). Display Specifications

Display/Display Adapter	Resolution	Colors	Characters per Line
Monochrome Display 8503 with Display Adapter 8514/A or XGA	640 by 480	16 gray shades (see note 2)	80
	640 by 480	256 gray shades (see notes 2 and 5)	80
	1024 by 768	16 gray shades (see note 2)	80
	1064 by 768	256 gray shades (see notes 2 and 5)	80
Color Displays 8512, 8513, 8514, or 8515 with Display Adapter 8514/A or XGA	640 by 480	16 (see note 2)	80
	640 by 480	256 (see notes 2 and 5)	80
	1024 by 768	16 (see note 2)	80
	1064 by 768	256 (see notes 2 and 5)	80

Notes:

1. This combination has four palettes of three colors plus background. The colors cannot be selected individually.
2. Data stream limits display up to eight colors on any display.
3. EGA storage expansion option and modules (up to 128KB) are required for additional color support.
4. Valid only in XT*, XT 286, or AT*.
5. IBM 8514 Memory Expansion Kit must be installed.

Graphics Support

This section lists both fully supported and no-op (ignored) orders used to build the graphics data stream. The work station function verifies all no-op orders for correct 5292-2 format and context and posts a message if it detects an error.

Although all the orders are listed in this section, only the orders that the work station function uses differently from the 5292-2 are described.

For more information on the orders used by the AS/400 system to communicate with graphics work stations, such as the IBM 5292 Model 2 (5292-2) Color Graphics Display Station, refer to the *IBM 5250 Information Display System Functions Reference*. For information on VDI drivers, refer to the *PC Support/400 DOS Installation and Administration Guide* or to the *PC Support/400 OS/2 Installation and Administration Guide*.

Figure 4-7 shows the graphics orders you find in a graphics data stream. You can divide them into the following groups:

- Read orders
- Graphics control orders
- Graphics draw orders
- Graphics set orders
- Printer orders

- IEEE-488 interface orders

Figure 4-7 (Page 1 of 2). Graphics Orders

Graphic Byte (Hex)	Order Name	Program Action
80	Read Status	Changed
81	Read Attributes	New order
90	End Graphics Block	Fully supported
91	More Data to Come	Fully supported
92	End of Data	Fully supported
93	Graphics Display On	Changed
94	Graphics Display Off	Changed
95	End Graphics	Fully supported
96	Suppress Pacing Response	Fully supported
A0	Draw Polyline	Fully supported
A1	Draw Scanline	Fully supported
A3	Write Background	Changed
A4	Write Polymarker	Fully supported
A5	Fill Polygon	Fully supported
A6	Define Shield Area	No-op order
B0	Set Color	Changed
B1	Set Style	Changed
B2	Set Style Offset	No-op order
B3	Set Function	Changed
B4	Set Color Table	Changed
B5	Set Marker	Fully supported
B6	Set Line Weight	No-op order
B7	Set Fill Mode	Changed
B8	Set Attributes	New order
C0	Printer Data Follows	P1 error (printer not attached)
C1	Screen Copy	P1 error (printer not attached)
C2	Load Printer A/N Color Mix Table	No-op order
C3	Load Printer Graphics Mix Table	No-op order
C4	Set Printer Time-Out	No-op order
D1	Set Address	Changed
D2	Set Talker Time-Out	Changed
D3	Set EOI Mode	No-op order
E0	IEEE-488 Data Follows	Changed
E1	Take Control Asynchronously	Changed

Figure 4-7 (Page 2 of 2). Graphics Orders

Graphic Byte (Hex)	Order Name	Program Action
E2	Go to Standby	Changed
E3	Remote Enable	No-op order
E4	Remote Disable	No-op order
E5	Interface Clear	Changed

Read Orders

Figure 4-8 lists the read orders.

Figure 4-8. Read Orders

Graphics Byte (Hex)	Order Name	Described
80	Read Status	Yes
81	Read Attributes	Yes

The description of the changed graphics read order follows. The hexadecimal value of the graphics read order is in parentheses after the name of the order.

Read Status (80)

The Read Status order prepares graphics status information for the AS/400 system to read. The information is in the alphanumeric display buffer at the address specified by the 2 bytes that follow the order.

Format: Figure 4-9 shows the format of the Read Status order.

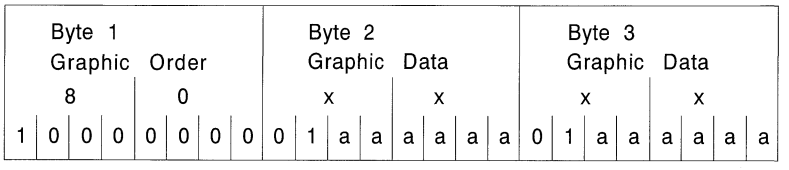


Figure 4-9. Read Status Order Format

Figure 4-10 displays the contents of the bytes in this format.

Figure 4-10. Read Status Order Field Contents and Description

Byte	Field Content	Description
1	X'80'	Graphics order
2, 3	Graphics data	The last 6 bits of the 2 data bytes form an offset between 0 and 1919 (0 corresponds to row 1, column 1), where the status information is written in the alphanumeric display buffer.

Read Status Data Format: Figure 4-11 displays the format of the 20 bytes of read status data.

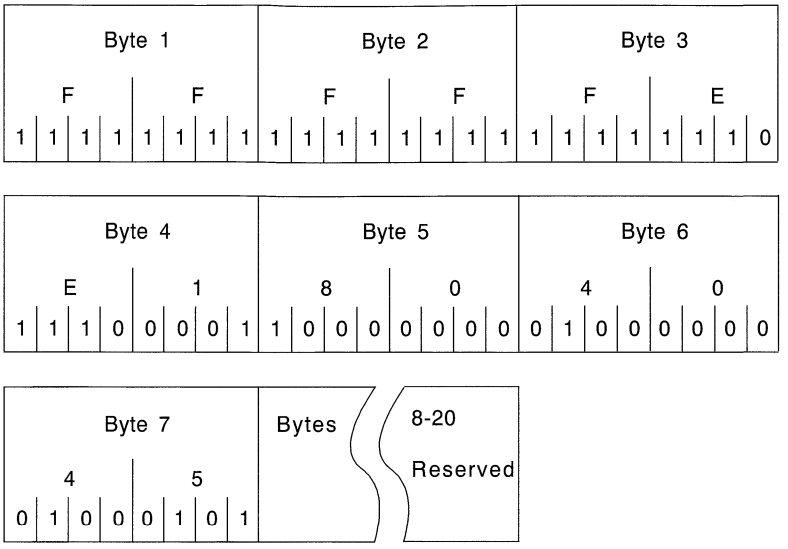


Figure 4-11. Read Status Data Format

Figure 4-12 displays the contents of the bytes in this format.

Figure 4-12 (Page 1 of 2). Read Status Bytes

Byte	Status Data Description
1, 2	2-byte error code of the last graphics error detected. These codes are described in the *PC Support/400 User's Guide for DOS*.
3	Model Identification: *Hex Value* — *Model* X'FF' — 5292-2 or the work station function
4	Graphics Level Number: *Hex Value* — *Description* X'F1' — 5292-2 Level 1 X'F2' — 5292-2 Level 2 X'E1' — Work Station Function Level 1 X'E2' — Work Station Function Level 2
5	Attached Printer Type: *Hex Value* — *Description* X'80' — Not able to identify printer. There is no printer support.
6, 7	When an error is detected, this 2-byte value represents a 1-byte offset into the graphics buffer to the last byte processed. *Hex Value* — *Error Status* X'FFFF' — No error pending. X'4A4B' — Error pending. Data stream error offset (for example, X'4A4B' is byte 139).

Figure 4-12 (Page 2 of 2). Read Status Bytes

Byte	Status Data Description
8–20	Reserved (X'40').

Notes:

1. The complete 20 bytes of status data may not always be written. Data written to the display buffer for a Read Status order ends when all 20 bytes are written, when the end of the display is reached, or when a field attribute is reached.
2. All data written to the display buffer as a result of a Read Status order are coded above X'3F'.
3. All pending errors are cleared when a Read Status order is received.

Code each field in the Read Status order to a value greater than X'40'. For example, if you place X'37' into a 2-byte coded field, the resulting format appears as in Figure 4-13.

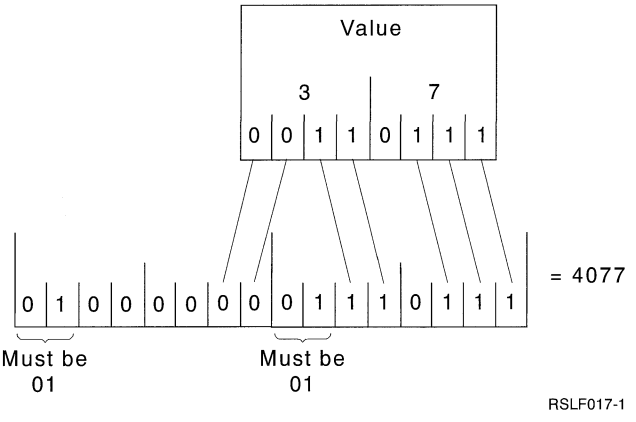


Figure 4-13. Read Status Order Example

Read Attributes (81)

The Read Attributes order gathers information about the characteristics of the graphics display and adapter currently used. The information stores in the alphanumeric buffer at the offset specified by the 2 data bytes that follow the order.

Format: Figure 4-14 shows the format of the Read Attributes order:

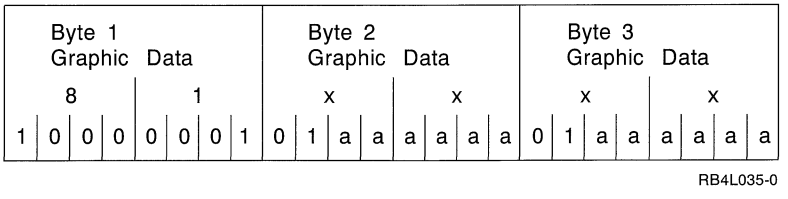


Figure 4-14. Read Attributes Format

Figure 4-15 displays the contents of the bytes in this format.

Figure 4-15. Read Attributes Order Field Contents and Description

Byte	Field Content	Field Description
1	X'81'	Order code.
2, 3	Graphics data	The last 6 bits of the 2 data bytes form an offset between 0 and 1919 (0 corresponds to row 1, column 1), where the attribute information is written in the alphanumeric display buffer.

The following information is stored in the alphanumeric buffer:

Bytes	Attribute Data Description
1 to 4	Maximum horizontal pel count
5 to 8	Maximum vertical pel count
9 to 12	Number of screen colors
13 to 16	Color palette size
17 to 18	Number of characters per row
19 to 20	Number of characters per column
21 to 22	Horizontal pel size in microns
23 to 24	Vertical pel size in microns
25 to 26	Number of line widths supported
27 to 40	Reserved

Notes:

1. The complete 40 bytes of attribute data may not always be written. Data written to the display buffer for a Read Attributes order ends when all 40 bytes are written, when the end of the display screen is reached, or when a field attribute is reached.
2. Code each field to a value greater than X'40'. For example, if you place X'37' in a 2-byte coded field, the result is as shown in Figure 4-16.

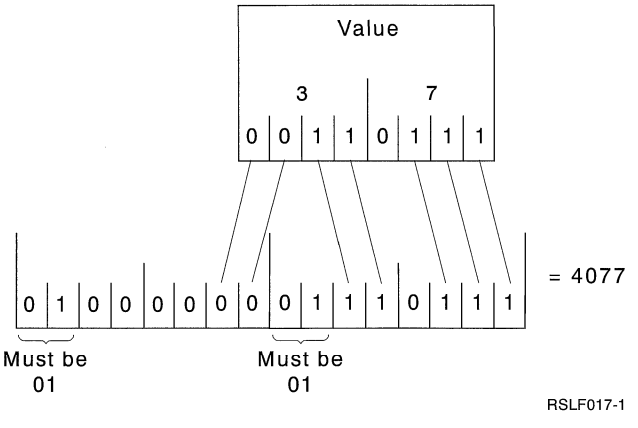


Figure 4-16. Read Attributes Order Display Buffer Example

Example: If the horizontal resolution of the graphics display is 640, then the maximum horizontal pel count is 639 (639 = 640 – 1), which is X'27F'. Bytes 1 through 4 of the attribute information appear as in Figure 4-17 on page 4-20.

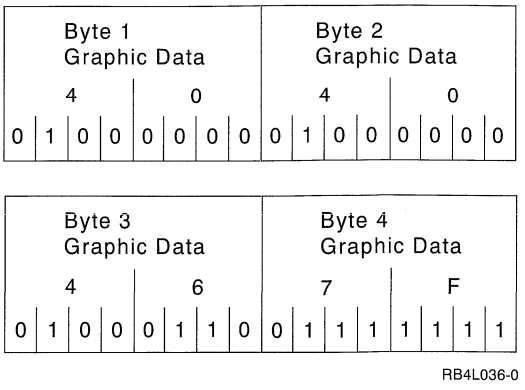


Figure 4-17. Example Read Attributes Order

Graphics Control Orders

Figure 4-18 lists the graphics control orders. Although all the graphics control orders are listed, only the orders that the work station function uses differently from the 5292-2 are described.

Figure 4-18. Graphics Control Orders

Graphics Byte (Hex)	Order Name	Described
90	End Graphics Block	No
91	More Data to Come	No
92	End of Data	No
93	Graphics Display On	Yes
94	Graphics Display Off	Yes
95	End Graphics	No
96	Suppress Pacing Response	No

Warning: The personal computer operates at speeds unrelated to the 5292-2, making operation with the Suppress Pacing Response order not predictable.

The description of each changed graphics control order follows. The hexadecimal value of each graphics control order is in parentheses after the name of the order.

Graphics Display On (93)

The Graphics Display On order turns on the graphics display. This allows the display station to display data from the graphics buffer. Graphics display is on when a G (in reverse image) appears on the status line.

The work station function enters the requested graphics operation, sets the resolution, clears the display, and redraws any data in the graphics buffer by using the orders in the graphics buffer. Any text on the display at the time the order was received is then added to the display.

Notes:

1. Shifting in and out of graphics operation (with either the local-select mode 1 key or a hot key sequence) can cause the size and exact location of text to change in relationship to graphics data.
2. When the display is operating in 320 x 200 pel mode, text displays 40 characters per row. Text beyond column 40 does not display.

Format: Figure 4-19 shows the format of the Graphics Display On order.

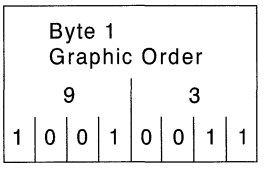


Figure 4-19. Graphics Display On Order Format

Graphics Display Off (94)

The Graphics Display Off order turns off the graphics display. Graphics display is off when a G appears (in normal image) on the status line.

The graphics operation resets and the program returns to the previous text operation. Any data in the text buffer displays using the current display size and attributes. All graphics data in the VDI buffer is saved until the next Write Background order processes or until you press Local Select 9 (erase graphics).

Format: Figure 4-20 shows the format of the Graphics Display Off order.

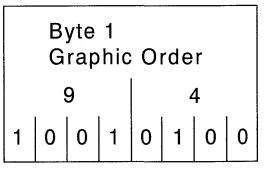


Figure 4-20. Graphics Display Off Order Format

Graphics Draw Orders

Figure 4-21 lists the graphics draw orders. Although all the graphics draw orders are listed, only the orders that the work station function uses differently from the 5292-2 are described.

Figure 4-21 (Page 1 of 2). Graphics Draw Orders

Graphics Byte (Hex)	Order Name	Described
A0	Draw Polyline	No
A1	Draw Scanline	No
A3	Write Background	Yes
A4	Write Polymarker	No

Figure 4-21 (Page 2 of 2). Graphics Draw Orders

Graphics Byte (Hex)	Order Name	Described
A5	Fill Polygon	No
A6	Define Shield Area	No

A description of each changed graphics draw order follows. The hexadecimal value of each graphics draw order is in parentheses after the name of the order.

Write Background (A3)

The Write Background order is used to delete the graphics display by using the specified background color.

Format: Figure 4-22 shows the format of the Write Background order.

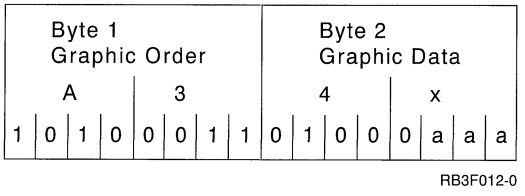


Figure 4-22. Write Background Order Format

Figure 4-23 displays the contents of the bytes in this format.

Figure 4-23. Write Background Order Field Contents and Description

Byte	Field Content	Description
1	X'A3'	Graphics order.
2	Background color	The binary value of the **aaa** bits is the color index this order uses when it writes the background.

Applicable Draw Orders: The Set Color Table draw order is applicable to the Write Background order.

Example: The graphics order and graphics data shown in Figure 4-24 change the entire graphics display to the color indicated by an index of 6 in the color table. The color specified by the color index of the AS/400 color table covers the entire graphics area of the display.

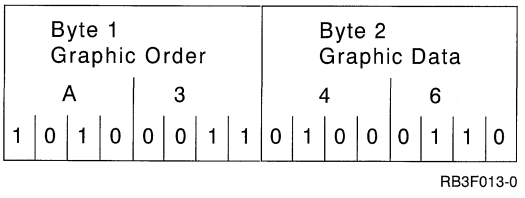


Figure 4-24. Example Write Background Order

Graphics Set Orders

Figure 4-25 lists the graphics set orders. Although all the graphics set orders are listed, only the orders that the work station function uses differently from the 5292-2 are described.

Figure 4-25. Graphics Set Orders

Graphics Byte (Hex)	Order Name	Described
B0	Set Color	Yes
B1	Set Style	Yes
B2	Set Style Offset	No
B3	Set Function	Yes
B4	Set Color Table	Yes
B5	Set Marker	No
B6	Set Line Weight	No
B7	Set Fill Mode	Yes
B8	Set Attributes	Yes

The description of each changed graphics set order follows. The hexadecimal value of each graphics set order is in parentheses after the name of the order.

Set Color (B0)

The Set Color order specifies the color index to use for subsequent draw orders. The default color index is 7.

Format: Figure 4-26 shows the format of the Set Color order.

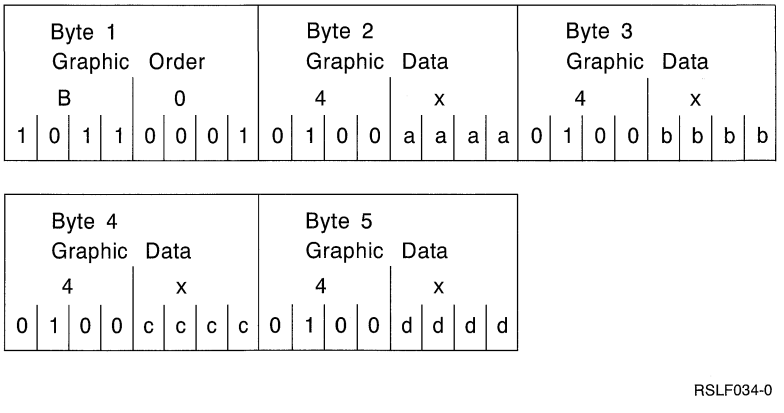


Figure 4-26. Set Color Order Format

Figure 4-27 displays the contents of the bytes in this format.

Figure 4-27. Set Color Order Field Contents and Description

Byte	Field Content	Field Description
1	Hex B0	Order code.
2	Graphics data	The binary value of the **aaa** bits specifies the color index (0 through 7) to be used for subsequent draw orders.

Applicable Draw Orders: Following are applicable draw orders:

- Draw Polyline
- Draw Scanline
- Fill Polygon
- Write Polymarker

Example: The graphics order and data byte shown in Figure 4-28 set the display color to color index 3 for subsequent draw orders.

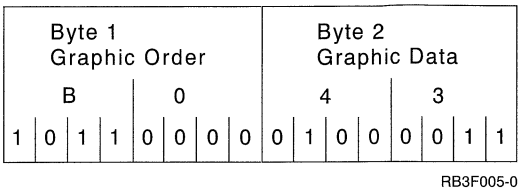


Figure 4-28. Example Set Color Order

Note: The color value is used as an index in a table of selected colors which work station function supplies. This color applies to all subsequent graphics objects except text, which gets its color from its field attributes.

Set Style (B1)

The Set Style order specifies the line style to be used for subsequent draw orders. The default is a solid line.

Format: Figure 4-29 shows the format for the Set Style order.

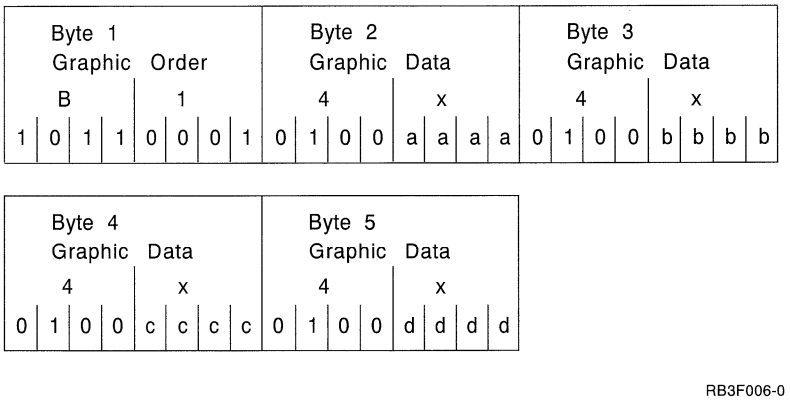


Figure 4-29. Set Style Order Format

Figure 4-30 displays the contents of the bytes in this format.

Figure 4-30 (Page 1 of 2). Set Style Order Field Contents and Description

Byte	Field Content	Field Description
1	B1	Order code.
2	Segment A	Length (binary value of the **aaaa** bits; number of picture elements [pels]) of the first visible segment of the line style.
3	Segment B	Length of the first segment that has not changed (binary value of the **bbbb** bits; number of pels) that follows the first visible segment.

Figure 4-30 (Page 2 of 2). Set Style Order Field Contents and Description

Byte	Field Content	Field Description
4	Segment C	Length of the second visible segment (binary value of the **cccc** bits; number of pels) of the line.
5	Segment D	Length of the second segment that has not changed (binary value of the **dddd** bits; number of pels) that follows the second visible segment.

Applicable Draw Orders: Following are applicable draw orders:

- Draw Polyline
- Fill Polygon

Personal Computer Mapping: The VDI line style is set according to the segment specifications described above and the selections made during operation of the graphics configuration program. The VDI supported line styles are shown in Figure 4-31.

Figure 4-31. VDI Supported Line Styles for the Set Style Order

Style	Description
0	Invisible
1	Solid
2	Long dash
3	Dotted
4	Dash dotted
5	Medium dash
6	Dash with two dots
7	Short dash

Example: The graphics order and graphics data shown in Figure 4-32 set the line style on a blank background.

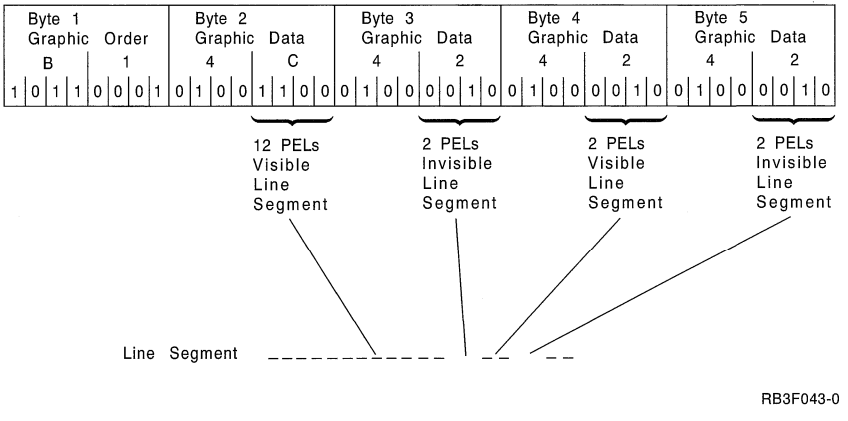


Figure 4-32. Example Set Style Order

Set Function (B3)

The Set Function order determines how subsequent draw orders combine with the existing picture. This order allows you to specify how you replace the existing picture color. You can either draw over the existing picture with a new color, or combine the color you are drawing with the existing picture color.

The default replaces the existing color with the color you are drawing.

Format: Figure 4-33 shows the format of the Set Function order.

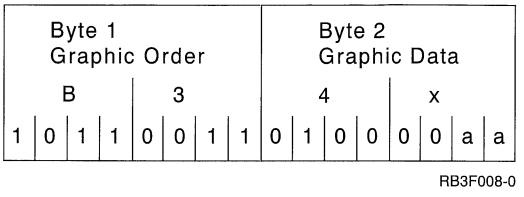


Figure 4-33. Set Function Order Format

Figure 4-34 displays the contents of the bytes in this format.

Figure 4-34. Set Function Order Field Contents and Description

Byte	Field Content	Description
1	Hex B3	Order code.
2	Function (low 2 bits)	Drawing logical operator: 00 Ignored by the personal computer. 01 OR: The color you are drawing is combined with the existing color using OR logic. 10 XOR: The color you are drawing combines with the existing color using exclusive OR logic. 11 REPLACE: The color you are drawing replaces the existing color.

Applicable Draw Orders: Following are applicable draw orders:

- Draw Polyline
- Draw Scanline
- Fill Polygon

Personal Computer Mapping: VDI supports the same writing modes as the 5292-2 does, but VDI line styles may be different from the 5292-2.

Example: If you are drawing a red line and encounter existing white pels, the resulting color, assuming default colors, is:

```
OR      0 1 0     (Color index 2)
        1 1 0     (Color index 6)
        ---------
        1 1 0     Resulting color index
```

The format of this example is shown in Figure 4-35.

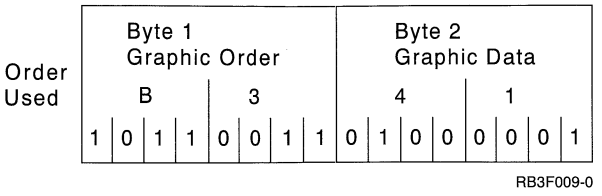


Figure 4-35. Set Function Order Example Format

Set Color Table (B4)

The Set Color Table order loads the display's color table with new intensity values. Each 9-bit entry consists of 3 intensity bits for each primary color (red, green, and blue). This allows for intensities of 0 through 7 for each color. When supported by the graphics adapter, it supplies up to 512 possible colors or shades. A value of 7 indicates the maximum intensity for a primary color.

This command is accepted and ignored when a personal computer display does not support a color table (for example, a CGA or an EGA with a monochrome display).

When display hardware supports a color table, the command is fully done, but the actual color displayed depends on the display's hardware capabilities.

When display hardware supports less than eight colors, but has color table support (for example, the base EGA), the mapping of the eight colors from the AS/400 system to the color of the personal computer hardware is specified by the graphics configuration.

Format: Figure 4-36 shows the format of the Set Color Table order.

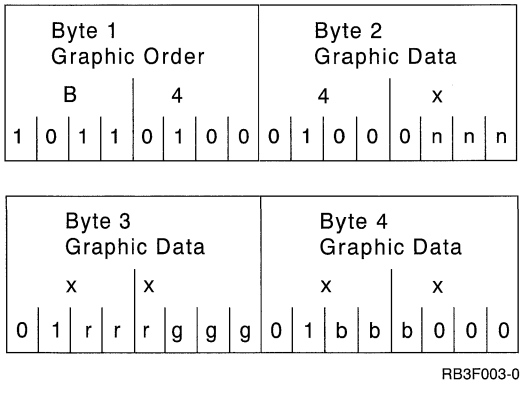


Figure 4-36. Set Color Table Order Format

Figure 4-37 displays the contents of the bytes in this format.

Figure 4-37. Set Color Table Order Field Contents and Description

Byte	Field Contents	Field Description
1	Hex B4	Order code.
2	Table index (low 3 bits)	Index into display's color table.
3, 4	New table value	New value for display's color table entry specified by table index. Data byte format for entry change: **01RRRGGG** **R** Red intensity **G** Green intensity **01BBB000** **B** Blue intensity
5 through n	...	Additional index/value bytes.
n + 1	Hex 92	End of Data order.

Default Colors: Figure 4-38 shows the default values for the display's color table.

Figure 4-38. Color Table Default Values for the Display

3-Bit Color Index	Default Value (RRR GGG BBB)	Color Displayed
0	000 000 000	Black
1	111 000 000	Red
2	000 111 000	Green
3	000 000 111	Blue
4	111 000 111	Pink
5	111 111 000	Yellow
6	000 111 111	Turquoise
7	111 111 111	White

Applicable Draw Orders: Following are applicable draw orders:

- Draw Polyline
- Draw Scanline
- Fill Polygon
- Write Background
- Write Polymarker

Personal Computer Mapping: The VDI color table index value maps to the computer-system-requested value. The red, blue, and green values scale to fit the VDI requirements or 0 to 100 percent.

Set Fill Mode (B7)

The Set Fill Mode order establishes the fill mode control information used with the Fill Polygon order.

Figure 4-39 shows the default values for the Set Fill Mode order.

Figure 4-39. Set Fill Mode Order Default Values

Bits	Value	Description
aa	00	Ignored.
bb	00	Draw a solid boundary line and fill interior with style (from Set Style order).
cccccc	000000	Ignored.

Format: Figure 4-40 shows the format of the Set Fill Mode order.

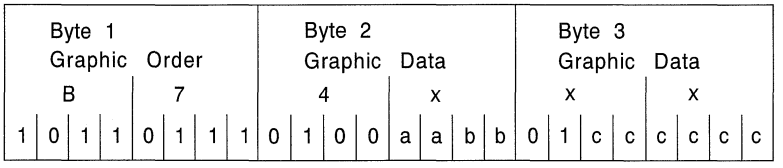


Figure 4-40. Set Fill Mode Order Format

Figure 4-41 displays the contents of the bytes in this format.

Figure 4-41. Set Fill Mode Field Contents and Description

Byte	Field Content	Description	
1	Hex B7	Graphics order.	
2, 3	Fill control	Control information for area pattern:	
		aa bits	Ignore.
		bb bits 00	Draw solid boundary line and fill with style specified in Set Style order.
		bb bits 01	Draw solid boundary line.
		bb bits 10	Fill interior with style specified in Set Style order.
			Note: Fill interior does not include the boundary line.
		bb bits 11	Draw boundary line with style specified in Set Style order.

Applicable Draw Orders: The Fill Polygon draw order is applicable to the Set Fill Mode order.

Personal Computer Mapping: Bits **aa** and bits **cccccc** are ignored. A fill pattern, if requested, is based on the current line style.

Figure 4-42 shows the VDI line style and fill pattern for use with the Set Fill Mode order personal computer mapping.

Figure 4-42. VDI Line Style and Fill Pattern

Line Style	Fill Pattern
0	Hollow
1	Solid
2	Wide-spaced 45° lines crossed with –45° lines
3	Wide-spaced 45° lines
4	Medium-spaced 45° lines
5	Medium-spaced 45° lines crossed with –45° lines
6	Narrow-spaced 45° lines
7	Narrow-spaced 45° lines crossed with –45° lines

Example: The graphics order and graphics data shown in Figure 4-43 result in filling the interior with the style defined by the Set Style order in effect when the Fill Polygon order is sent.

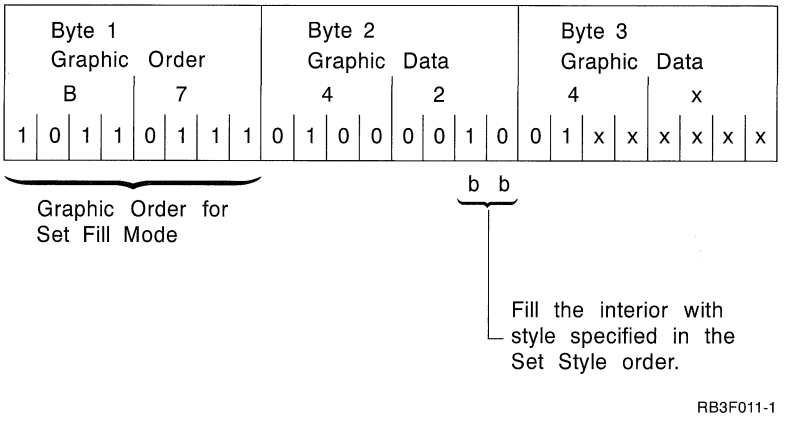


Figure 4-43. Example Set Fill Mode Order

Set Attributes (B8)

The Set Attributes order allows you to determine the attributes of the graphics data stream.

Format: Figure 4-44 shows the format for the Set Attributes order byte.

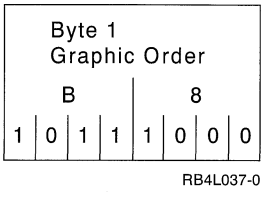


Figure 4-44. Set Attributes Order Byte Format

The Set Attributes order byte is followed by 40 bytes of data, which contain the fields displayed in Figure 4-45.

Figure 4-45. Set Attributes Order Field Contents and Description

Bytes	Field Content	Description
1 to 4	0 through 32767	Number of horizontal units used.
5 to 8	0 through 32767	Number of vertical units used.
9 to 12	256 through 4096	Bytes of graphic buffer size.
13 to 16	2 through device maximum	Number of screen colors. See note.
17 to 20	0 through device maximum	Color palette size. See note.
21 to 40		Reserved.

Note: The values for screen color and palette size are not changed by the host system. Instead, values used by VDI are used.

Each field is coded to a value greater than hex 40. For example, if you place hex 37 in a 2-byte coded field, the result would be as shown in Figure 4-46.

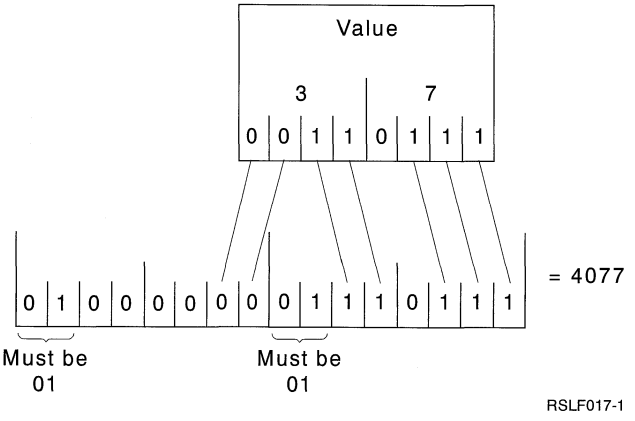


Figure 4-46. Set Attributes Order Example Format

Example: If the horizontal resolution of the graphics display is 480, then the maximum vertical resolution is 479 (479 = 480 − 1), which is hex 1DF. Bytes 5 through 8 of the attribute information would appear as in Figure 4-47.

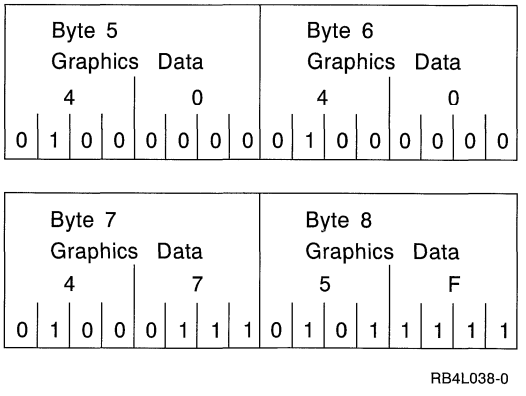


Figure 4-47. Example Set Attributes Order

Printer Orders

Figure 4-48 lists the printer orders. No printer orders are described since the work station function does not support the printer in graphics mode.

Figure 4-48. Graphics Printer Orders

Graphics Byte (Hex)	Order Name	Described
C0	Printer Data Follows	Causes P1 error
C1	Screen Copy	Causes P1 error
C2	Load Printer A/N Color Mix Table	No
C3	Load Printer Graphics Color Mix Table	No
C4	Set Printer Time-Out	No

IEEE-488 Interface Orders

Figure 4-49 lists the IEEE-488 interface orders. Although all the IEEE-488 orders are listed, only the orders that the work station function uses differently from the 5292-2 are described.

Note: The description of the IEEE-488 orders assumes you understand the IEEE-488 standards and the 9914 Adapter.

Figure 4-49. IEEE-488 Local Command Orders

Graphics Byte (Hex)	Order Name	Described
D1	Set Address	No
D2	Set Talker Time-Out	No
D3	Set EOI Mode	No
E0	IEEE-488 Data Follows	Yes
E1	Take Control Asynchronously	Yes
E2	Go to Standby	Yes
E3	Remote Enable	No
E4	Remote Disable	No
E5	Interface Clear	Yes

The AS/400 data stream is in IEEE-488 format and the plotter is attached through the RS-232-C interface. The work station function handles the conversion.

The description of each changed IEEE-488 interface order follows. The hexadecimal value of each order is in parentheses after the name of the order.

IEEE-488 Data Follows (E0)

The IEEE-488 Data Follows order sends pass-through data directly to an IEEE-488 device. Data following this order passes through until an End of Data is encountered. If the data spans blocks, use the More Data to Come order. Refer to the *IBM 5250 Information Display System Functions Reference* for more information.

IEEE-488 Data-Follows-order pass-through data, following a Take Control Asynchronously order and preceding a Go to Standby order, processes as interface messages. All other IEEE-488 Data-Follows-order pass-through data processes as device-dependent messages and is sent to the IEEE-488 device.

Format: Figure 4-50 shows the format of the IEEE-488 Data Follows order.

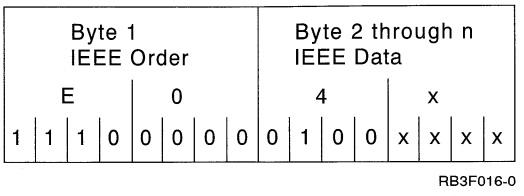


Figure 4-50. IEEE-488 Data Follows Order Format

Figure 4-51 displays the contents of the bytes in this format.

Figure 4-51. IEEE-488 Data Follows Order Field Contents and Description

Byte	Field Content	Description
1	Hex E0	Data Follows order
2, 3	IEEE-488 data	Pass-through data pair 1
4, 5	IEEE-488 data	Pass-through data pair 2
6, n	IEEE-488 data	Additional pass-through data pairs
n+1	Hex 92	End of Data order

Notes:

1. The interface messages (IM) data is the data that configures the listeners and talkers on the IEEE-488 bus.
2. The device-dependent messages (DDM) data is the data associated with a particular plotter.
3. This description of IEEE-488 orders assumes an understanding of the IEEE-488 standards and the 9914 Adapter.

Personal Computer Mapping: This order processes IEEE-488 data transmissions. All interface data is captured and verified for correct IEEE-488 syntax and device configurations. All other data is considered plotter data and passes directly to the plotter.

Note: If the plotter address contained in the interface message data does not match the address specified in the configuration file, an E1 error occurs when trying to send data to the plotter.

Example: This example shows a valid IEEE-488 order sequence. IEEE-488 set orders must have been previously received and processed.

```
E5 E3 E1 E0 (IM) 92 E2 E0 (DDM) 92
```

Take Control Asynchronously (E1)

The Take Control Asynchronously order sends the Take Control Asynchronously Auxiliary command to the IEEE-488 adapter.

Format: Figure 4-52 shows the format of the Take Control Asynchronously order.

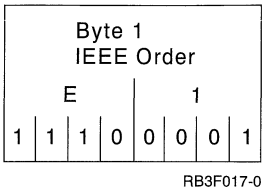


Figure 4-52. Take Control Asynchronously Order Format

Personal Computer Mapping: This order defines the start of the interface message data stream.

Go to Standby (E2)

The Go to Standby order sends the Go to Standby Auxiliary command to the IEEE-488 adapter.

Format: Figure 4-53 shows the format of the Go to Standby order.

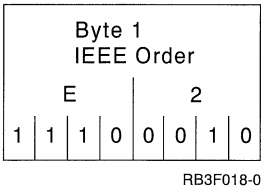


Figure 4-53. Go to Standby Order Format

Personal Computer Mapping: This order marks the end of the interface message data stream.

Interface Clear (E5)

The Interface Clear order sends an Interface Clear Auxiliary command to the IEEE-488 adapter.

Format: Figure 4-54 shows the format of the Interface Clear order.

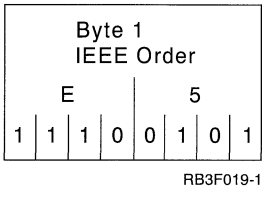


Figure 4-54. Interface Clear Order Format

Personal Computer Mapping: This order removes the talker from the interface and performs the Take Control Asynchronously order function.

Batch Error Level Return Codes

The following work station function programs return batch error level codes:

- WSF
- STARTWSF
- STOPWSF

If you are using a batch file, these programs set a return code that can be verified by the DOS batch file command IF ERRORLEVEL. Batch error level codes do the following functions:

Error Level (Decimal)	Description
0	The process was successful.
10	An error occurred, but the program was able to run successfully.
13	An error occurred because the specified work station function session is already started.
20	An error occurred and the program was not able to continue.
25	An error occurred when the work station function session could not be started.

Work Station Function Using Extended DOS

This section discusses using the work station function with extended DOS.

Note: Extended DOS support allows certain PC Support functions to use extended memory above 1MB (MB equals 1 048 576 bytes), thus making more of the conventional memory available for running other DOS programs. Extended DOS support can be used on personal computers that have an 80286 processor or above and at least 384KB (KB equals 1024 bytes) of extended memory. Most 80286 and above systems with 1MB of memory have 384KB of extended memory available.

Using the Extended DOS version of work station function puts as much code and data into extended memory as possible. It still maintains regular DOS interfaces and data so that other functions are not affected.

Note: The EIMPCS.SYS program is required by the extended DOS version of work station function.

The following functions cannot use extended memory, but are not affected when you use the Extended DOS version of work station function:

- WSF printer emulation
- WSF graphics
- Session manager
- WSF API
- WSF 132 column support
- WSF Thailand support
- STARTWSF command
- STOPWSF command

Extended WSF supports a configurable area to receive host data streams. The default size of the area is 6KB. This area is allocated in extended memory unless

a session is configured for graphics, then conventional memory is used. See "Data Stream Size Restriction" on page 4-11 for the discussion about host data streams.

Chapter 5. Printer Emulation for Work Station Function

This chapter contains printer emulation information, including default values for forms handling on AS/400 printers, AS/400 printer commands that are uniquely supported by work station function printers, and a description of the ASCII Transparent Data (ATRN) printer command.

The work station function printer emulates only the 3812 SNA character string (SCS) data stream.

For extended DOS users on a PS/55, the 5250WS program is available. This program is the DBCS extended version of work station function running with the local area network router and synchronous data link control (SDLC) router.

For more information about configuring these options, refer to the *PC Support/400 DOS Installation and Administration Guide*. For more information about using printer emulation and how to show your printer profile, refer to the *PC Support/400 User's Guide for DOS*.

Forms Handling on AS/400 Printers

This section provides information about the ways that the IBM 3812 printer handles forms. For information on your personal printer and its specifications, refer to your personal printer manual.

The following table shows the default values for forms handling on the 3812 printer.

Printer Specification	**IBM 3812**
Print line length (inches)	14
Paper width: maximum single sheet (inches)	8.5
Paper length: maximum (inches)	14
Lines per inch	See note 1
Characters per inch	See note 2
Characters per line	See note 3
Continuous forms	No
Single sheet forms	Yes

Notes:

1. The 3812 prints 4, 5-1/3, 6, 8, 9.6, 12, 24, or 48 lines per inch with 1/2, 1, 1-1/2, 2, 2-1/2, or 3 spaces between lines. The AS/400 system controls this option.
2. The 3812 prints 5, 10, 12, 15, 17.1, or proportional-spaced characters per inch. The AS/400 system controls this option.
3. The 3812 prints 70, 140, 168, 210, or 239 characters per line (varies if you select proportional spacing). The AS/400 system controls this option.

AS/400 Printer Commands

The commands the AS/400 system uses to communicate with the SCS 3812 printer are explained in the *Guide to Programming for Printing*. This section lists and explains only the printer commands that are supported differently by the work station function SCS 3812 printer support.

Page Presentation Media

The work station function SCS 3812 printer supports the following page presentation media (PPM) parameters:

- Forms-control (FC)
- Source-drawer (SD)
- Destination drawer offset (DDO)
- Quality (Q)
- Duplex (DX)

Other parameters are not inspected.

The work station function depends on your printer session profile to correctly define the type of paper handling equipment that can be attached to your personal printer.

If the AS/400 print job chooses a source drawer that the personal printer is not capable of supporting, the paper-feed-type (PFT) parameter of the Set Print Setup (SPSU) control is overridden. This forces manual insertion. The work station function stops printing and turns on the FORMS indicator with one of the following values in the *Output Data* field; the message corresponding to the value is displayed on the work station function printer operator panel:

01 Feed from drawer 1
02 Feed from drawer 2
0E Feed from the envelope holder

Each of these *Output Data* fields has a message associated with it as follows:

01 6272 Host system is requesting paper drawer 01.
02 6272 Host system is requesting paper drawer 02.
0E 6273 Host system is requesting envelopes.

If the PFT parameter of the SPSU control is overridden, a Change Setup indicator (on the printer operator panel) indicates when the application returns to the paper feed technique in effect prior to the manual insertion override.

If the paper feed technique is continuous forms and a change to the source drawer is requested, the work station function stops printing and turns on the CHANGE SETUP indicator, with the new source drawer value in the *Output Data* field as follows:

D1 Feed from drawer 1
D2 Feed from drawer 2

The decision to override the PFT parameter of the SPSU control depends on what you specified during printer session configuration.

The default settings for the personal printer options are determined by the printer model you select. For more information about configuring the work station function

printer, refer to the manual *PC Support/400 DOS Installation and Administration Guide*.

Set Text Orientation

The work station function SCS 3812 printer supports the set text orientation (STO) page rotation parameter. The character rotation parameter is ignored.

The page rotation parameter can have the following values:

- Normal portrait (0 degrees clockwise rotation of text)
- Landscape left (270 degrees clockwise rotation of text)
- Portrait upside down (180 degrees clockwise rotation of text)
- Landscape right (90 degrees clockwise rotation of text)
- Select Computer Output Reduction
- Selection orientation based on paper size

Begin and End Emphasis

Work station function supports the Begin and End Emphasis commands.

The way the work station function accomplishes emphasis on your attached personal printer depends on the functions defined in your Printer Function Table (PFT). Refer to "Highlighting" on page A-10.

Load Alternate Character

The work station function SCS 3812 printer does not support the Load Alternate Character (LAC) command. Therefore, graphics utilities, such as IBM AS/400 Business Graphics Utility (BGU), are not supported through work station function printing.

Set Character Distance

All values of the parameter for the Set Character Distance (SCD) command are supported. If no change to the current font width results from the SCD parameter value, the control is ignored.

If you have configured your printer session to prompt you for font changes, when the first printable graphic following a valid SCD is encountered, the work station function does the following:

1. Selects the correct font at the personal printer.
2. Stops and returns a *Printer Not Ready* status to the AS/400 system.
3. Turns on the Change Font indicator on the printer operator panel.
4. Displays on the printer operator panel the following message:

   ```
   6270 Host system is requesting type style &&&&&.
   ```

   where `&&&&&` is the decimal number of the requested type style.

You must do the following:

1. Change the print wheel/font cartridge on the attached personal printer.
2. Ensure that the personal printer is ready.
3. Select the Start option on the work station function printer operator panel.

Figure 5-1 on page 5-4 shows the GFID values displayed in the *Output Data* field for valid CD parameter values. Values are listed in hexadecimal.

Figure 5-1. GFID Values

CD Parameter	Action Selected	GFID Value
000A	10 characters per inch	0B
000B	Proportional Spacing	9E
000C	12 characters per inch	57
000F	15 characters per inch	DF
00FF	10 characters per inch	0B

During configuration, you can specify that the attached personal printer does not require you to change fonts. In this case, the work station function sends a command to the personal printer to select the desired font and continues printing.

Set CGCS through Local ID

This control is not fully supported. The work station function supports two code pages, one for use by data processing (DP) applications and one for use by word processing (WP) applications. If a valid Set CGCS through Local ID (SCGL) command is received, the DP code page becomes the active code page.

Set FID through GFID

The Set FID through GFID (SFG) command selects fonts (type styles) on an SCS 3812 printer.

WSF printer emulation uses information in the SFG command to select a font on the personal printer that is being used to emulate an SCS 3812 printer.

For personal printers that support the IBM Personal Printer Data Stream (PPDS) Set Font Global command, the font information in the Set FID through GFID command can be passed directly to the personal printer. The personal printer provides font substitution if the requested font is not supported.

For personal printers that do not support the PPDS Set Font Global command, the printer function table is used to map the font requested in the Set FID through GFID command into personal printer commands that select the font on the personal printer.

The SFG command parameters, font global ID (FGID), font width (FWD), and font attribute (FA) are supported. Any requested FGID is accepted, including FGIDs not supported by a real SCS 3812 printer.

The SFG command's FGID parameter is set by entering a type-style number in a document or font identifier in a print file.

General Font Mapping Considerations

The font mapping is controlled by the following printer function table information:

- Individual Type Style Definitions
- Group Type Style Definitions
- User-Defined Control 988

If the requested FGID has an individual or group type-style definition, that definition is used to select the font on the personal printer.

PFTSETUP, the user interface to create or change printer function tables, supports definition of the following type styles:

- 1-65 (10 Pitch)
- 66-153 (12 Pitch)
- 154-200 (Proportional)
- 211-239 (15 Pitch)
- 240-249 (5 Pitch)
- 250-259 (17.1 Pitch)
- 260-279 (8.55 Pitch)

For more information about PFTSETUP, refer to Appendix A, "Using the Printer Function Table Setup Program" on page A-1.

If the requested FGID is not defined in the printer function table, then User-Defined Control 988 is inspected to determine the data stream supported by the personal printer. For a description of User-Defined Control 988, refer to "User-Defined Control 988" on page A-14.

For personal printers supporting the IBM Personal Printer Data Stream (PPDS) Level 2 and above, the PPDS Set Font Global command is used to select fonts on the personal printer. Fonts on font cards are supported without printer function table modifications.

For personal printers that do not support the IBM Personal Printer Data Stream Level 2 and above, if the requested FGID is not supported, the FWD parameter is used to determine if there is support for the requested font width. If an individual or group type-style definition supports the requested font width, that definition is used to select the font on the personal printer. If the requested FGID is not supported and the requested font width is not supported, the user is prompted at the printer operator panel for the font change. This prompt occurs even if your printer session was configured not to prompt for font changes.

Special Font Mapping Considerations

The following items are special font mapping considerations:

1. Substitution can be provided for the following requested FGIDs:

Requested Host FGID	Substituted PFT Type Style
281 (20 Pitch)	128
285 (25 Pitch)	139
290 (27 Pitch)	129
392	130
751	131
1051	132
1053	133
1056	134

1351	135
1653	136
2103	138

These requested FGIDs are supported by the real SCS 3812 printer. Substitution is provided because PFTSETUP does not allow individual or group type style definitions to be defined for these requested FGIDs.

The substitution is performed if the substituted type style has an individual or group type-style definition present in the printer function table.

2. Substitution may be provided for requested fonts that have a font width of 15 characters per inch (CPI).

   If a 15 CPI font is requested, but no 15 CPI fonts are supported, a 17.1 CPI font is substituted if support is available for a 17.1 CPI font.

3. If no support is available for the requested font, the User-Defined Control 988 is inspected before prompting for a font change. If User-Defined Control 988 is empty or X'00' (null), the font prompt is suppressed.

Set GCGID through GCID

This control is not fully supported. If a valid Set GCGID through GCID (SCG) control with a code-page global ID (CPGID) parameter of 259 decimal is received and the user has selected symbol support using CFGWSF, then the symbols code page becomes the active code page. If the user did not select symbol support using CFGWSF and the SCG control has a CPGID parameter of 259 decimal, the current code page remains active. If a valid SCG control with a CPGID parameter of 256 or 500 decimal is received, the WP code page becomes the active code page. If the CPGID parameter is not 256, 259, or 500 decimal, the DP code page is used.

Set Line Density

The Set Line Density (SLD) command is supported with all values of the line density (LD) parameter. The work station function rounds down to the nearest supported lines-per-inch value.

For work station function printers, when computer output reduction (COR) mode is activated, 70% of the line density set by SLD is used. The common values of 6 lines per inch and 8 lines per inch are changed to 9 lines per inch and 12 lines per inch, respectively.

Set Print Setup

When the work station function receives the Set Print Setup (SPSU) control and the specified paper feed technique parameter is different from the current one, work station function does the following:

1. Stops printing.
2. Turns off the Ready indicator on the work station function printer operator panel.
3. Sends a *Printer Not Ready* status to the AS/400 system.
4. Turns on the Change Setup indicator on the printer operator panel.
5. Displays a code for the paper-feed-type (PFT) parameter received with this control.

6. Displays message 6269, 6268, or 6267 on the work station function printer operator panel.

The codes displayed are shown in Figure 5-2.

Figure 5-2. Set Print Setup (SPSU) Control Codes

PFT Value	Output Data Code	Message
01	01	6269 Install manual forms
02	02	6268 Install continuous forms
03	03	6267 Install automatic cut sheet forms

The operator must manually change the paper handling equipment and then select the Start option on the work station function printer operator panel. Specify the initial paper handling equipment when you configure your printer session.

When the Change Setup indicator is on, work station function assumes that your personal printer may be powered off to change the paper handling equipment. After installing the requested paper handling equipment, turn on the power to your personal printer before selecting the start option on the work station function printer operator panel.

Set Single Line Distance

The Set Single Line Distance (SSLD) control is supported.

For work station function SCS 3812 printer, when computer output reduction (COR) mode is activated, 70% of the line distance set by SSLD is used. The common values of 6 lines per inch and 8 lines per inch are changed to 9 lines per inch and 12 lines per inch, respectively.

Set Vertical Margins

The Set Vertical Margins (SVM) control is supported.

When the SVM control processes, there is no comparison made between the TM value and either the LPI density or page depth. However, a TM value of less than a single line distance or greater than the page depth results in an exception condition (EC=1) during the processing of a form-feeding condition (due to a form-feed command or an automatic page-ending condition).

ASCII Transparent Data Printer Command

The ASCII Transparent Data (ATRN) command is supported for all work station function printers. The work station function supports ATRN within a printer data stream, allowing a block of data to transmit from the AS/400 system directly to the attached personal printer without conversion or inspection.

The hexadecimal code for the command is 03.

ATRN is a grouping control character used with a 1-byte parameter to denote the beginning of a transparent data stream. The format of the command is:

```
ATRN nn data
```

where nn is a hexadecimal count of the number of bytes of transparent data, not including the count byte.

The ATRN command does not affect the internally-maintained presentation position (PP).

There are two forms of ASCII transparency:

- Full ASCII transparent jobs
- ASCII data embedded in an SCS job

Full ASCII transparent jobs contain only ASCII data for a personal printer. The AS/400 system produces full ASCII transparent jobs when the printer is configured to use the host print transform function. These jobs are also produced by a virtual printer using data type 4. Refer to the *PC Support/400 DOS Installation and Administration Guide* or to the *PC Support/400 OS/2 Installation and Administration Guide* for more information on the virtual printer function and data type 4. For more information on the host print transform function, refer to the *PC Support/400 User's Guide for DOS* and the *PC Support/400 DOS Installation and Administration Guide*. Only the data present in the full ASCII job is sent to the printer. No initialization before the job or reset after the job is performed, unless the WFPRINT.ATRN environment variable is used. For more information about optional initialization, refer to "Optional Initialization and Resetting of Personal Printers."

The other form of ASCII transparency is ASCII data that is embedded in an SCS data stream. When the ATRN command (03) is encountered, the next *nn* bytes of data are sent to the personal printer with no EBCDIC to ASCII conversion. The printer is not reset when exiting an ASCII embed, unless the ATRN reset command is received.

If the ATRN reset command (03 00) is received during an SCS job, reset controls are sent to the printer. This is done to return the printer to a known state after printing a block of ASCII data.

Optional Initialization and Resetting of Personal Printers

Work station function printing initializes a personal printer before host system print (SCS) jobs and resets the printer to a power-on state after an SCS job. Work station function printing does not do any initialization before or resetting after ASCII transparent (ATRN) jobs. Only the ASCII codes are sent to the printer. These changes are designed to provide the best integrity of all print jobs coming from the AS/400 system.

The following optional features may help you in deciding when to initialize or reset the printer.

- To initialize before and reset after an ATRN job is determined by the presence of a DOS environment variable (WFPRINT.ATRN). If this variable is present and set to anything but zero (0), then work station function printing sends out the control sequences. If this variable is not present or is set to zero (0), then work station function printing does not send out the control sequences.
- To initialize before and reset after an SCS job is determined by the presence of another DOS environment variable (WFPRINT.SCS). If this variable is present and set to anything but zero (0), the work station function printing does not

send out the control sequences. If this variable is not present or is set to zero (0), then work station function printing sends out the control sequences.

Note: You do not have the authority to turn off initialization before SCS jobs. Work station function printing does some initialization before SCS jobs to ensure the integrity of SCS print jobs.

The initialization controls consist of the start-of-document initialization string from the printer function table (PFT), some internal controls determined by work station function printing, and the printer initialization string from the printer session profile.

The reset controls consist of some internal controls determined by work station function printing and the end-of-document reset string from the PFT.

These DOS environment variables can be typed at the DOS prompt or from within batch files such as AUTOEXEC.BAT or STARTPCS.BAT. Some examples of this are:

```
SET WFPRINT.ATRN=0
SET WFPRINT.SCS=1
```

For initialization and reset processing similar to that available before Version 2 Release 1.0, both DOS environment variables should be set like this:

```
SET WFPRINT.ATRN=1
SET WFPRINT.SCS=1
```

Note: You should be aware that this type of initialization and reset may affect other print jobs, especially PostScript ATRN jobs. The recommended setting for the DOS environment variables is:

```
SET WFPRINT.ATRN=0
SET WFPRINT.SCS=0
```

Chapter 6. User Exit Programs

This chapter supplies information on how to use user exit programs with the AS/400 Change Network Attributes (CHGNETA) command. It also supplies common parameter fields used with the command.

User exit programs work with both the DOS and OS/2 operating systems.

Using User Exit Programs

PC Support/400 supports user-written exit programs. These programs help ensure that your data is both secure and correct. Your exit program can determine if users have authority to perform a particular operation and what data they can access on the AS/400 system.

Note: The creation of memory to pass data to user exit programs, and the authorization checking that the user exit performs, may affect the performance of your system.

Change Network Attributes Command

To select the user exit program, enter the Change Network Attributes (CHGNETA) command from the AS/400 system Command Entry Menu display and specify the PC Support Request Access (PCSACC) parameter. CHGNETA is an Operating System/400* (OS/400*) command.

Although CHGNETA has several parameters, this section only describes the PCSACC parameter. For more information on the CHGNETA command, refer to the *CL Reference* manual.

PCSACC Parameter

Following are valid values for the PCSACC parameter:

***REJECT**
The PC Support system programs reject every request from the personal computer. They send an error message to the PC application.

***OBJAUT**
The PC Support system programs verify normal object authorities. For example, they verify authority to retrieve data from a database file for a transfer request.

library name/program name
The PC Support system program calls the program name you supply, and, based on a returned value from the program, determines if the PC request should be rejected. If the return code indicates the request should be rejected or if an error occurs, an error message is sent to the personal computer.

When you specify an exit program, the PC Support system programs pass the following two parameters to the exit program before running your request:

- A 1-byte return code area
- A character structure containing your user profile name and request

These two parameters allow the exit program to determine if your request is allowed. If the exit program sets the return code to X'F1', then the PC Support system program allows the request. If the return code is set to anything else, the PC Support system program rejects the request and displays the `Requested Function Rejected` message.

Parameters for the Virtual Printer User Exit Program

When you specify a user exit program name and use virtual printer, the following occurs:

1. The virtual printer function verifies your authority to a printer file and its output queue.
2. The virtual printer function calls the specified program.
3. The user exit program determines if the request is allowed to run.

Figure 6-1 shows common parameter fields and their descriptions for the virtual printer user exit program.

Figure 6-1 (Page 1 of 2). Parameter Fields for the Virtual Printer Function User Exit Program

Field	Description	Format	Size
User profile name	This field contains your user profile name.	Character	10
Application name	This field is a constant *VPRT indicating that virtual printer calls the program.	Character	10
Requested function	This field contains one of the following: • EXTRACT indicates performance of a list function. • CHECK indicates verification of both your authority to a printer file, and the output queue of that file. • OPEN indicates opening of the printer file.	Character	10
Object name	• For an EXTRACT request, this field contains one of the following: – Spaces, indicates the making of a list of all printer devices on the system. The library name must also be spaces in this case. – Spaces, indicates the making of a list of the specified libraries. The library name must *not* be spaces in this case. – A generic name (a partial name ending in an asterisk), indicates the making of a list of a partial list of printer files from the specified library. – A single asterisk, indicates making a list of all the printer files from the specified library. • For the CHECK request, this field contains the name of a printer file to verify. • For the OPEN request, this field contains the name of a printer file to open.	Character	10

Figure 6-1 (Page 2 of 2). Parameter Fields for the Virtual Printer Function User Exit Program

Field	Description	Format	Size
Library name	• For an EXTRACT request, this field contains one of the following: – Spaces, indicates the making of a list of all the printer devices on the system. The object name must also be spaces in this case. – *LIBL, indicates the making of a list of the libraries on the library. The object name must be spaces in this case. – *USRLIBL, indicates the making of a list of all the libraries on the user portion of the library list. The object name must be spaces in this case. – A generic name (a partial name ending with an asterisk), indicates list of a partial list of libraries. The object name must be spaces in this case. – *LIBL, indicates the making of a list of printer files from the libraries on the library list. The object name must *not* be spaces in this case. – *USRLIBL, indicates the making of a list of printer files from the libraries on the user portion of the library list. The object name must *not* be spaces in this case. **Note:** The library name and object name must *not* be spaces when a list of printer files from the library is made. • For the CHECK request, this field contains the name of the library containing the printer file to verify. (You can use the special value *LIBL.) • For the OPEN request, this field contains the name of the library containing the printer file to open. (You can use the special value *LIBL.)	Character	10
Member name	This field contains the member name for some applications, but is always spaces for the virtual printer application.	Character	10
Format name	This field contains the format name for some applications, but is always spaces for the virtual printer application.	Character	10
Length of the next field	This field contains the length of the next field. A value of 0 indicates that the next field does not exist.	Zoned decimal	5, 0
Output queue name and library name	This field exists only when the requested function is verification of the user's authority to the printer file and output queue. When the field exists, it contains a 10-character output queue name followed by a 10-character library name. The special value '*JOB' indicates use of the output queue of the job. The special value '*DEV' indicates use of the default output queue for the printer device.	Character	0 or 20

Parameters for the Transfer Function User Exit Program

When you specify user exit program and use the transfer function, the following occurs:

1. The transfer function verifies your authority to the file or library.
2. The transfer function system program calls the specified user exit program.
3. The user exit program determines if the request should run.
4. The transfer function program performs the transfer request.

When you are using the transfer function and the transfer request is a system-to-personal-computer request that contains a FROM prompt with more than one file, the requested function field contains the value of JOIN rather than SELECT. The user exit program is called for each file specified in the FROM prompt with the object name, library name, member name, and record format name containing the appropriate values.

Figure 6-2 displays common parameter fields and their descriptions for the transfer function user exit program.

Figure 6-2 (Page 1 of 2). Parameter Fields for the Transfer Function User Exit Program

Field	Description	Format	Size
User profile name	This field contains your user profile name.	Character	10
Application name	This field is a constant *TFRFCL indicating that the transfer function calls the program.	Character	10
Requested function	This field contains one of the following: • SELECT indicates a AS/400 system-to-personal computer transfer request. • JOIN indicates a AS/400 system-to-personal computer transfer request with more than one file specified. • REPLACE indicates a personal-computer-to-AS/400 system transfer request. • EXTRACT indicates performance of a list function.	Character	10
Object name	• For an EXTRACT request, this field contains one of the following: – A file name, partial file name (name ending in an asterisk), or a single asterisk. – Spaces, indicates the making of a list of all the libraries in the user portion of the library list. • For a SELECT or REPLACE request, this field contains the name of a database file.	Character	10
Library name	• For an EXTRACT request, this field contains one of the following: – A library name, *LIBL, or *USRLIBL – Spaces, indicating the making of a list of all the libraries in the user portion of the library list. • For a SELECT or REPLACE request, this field contains one of the following: – A library name – Spaces, if you do not specify a library name	Character	10

Figure 6-2 (Page 2 of 2). Parameter Fields for the Transfer Function User Exit Program

Field	Description	Format	Size
Member name	• For an EXTRACT request, this field contains one of the following: – Spaces, if you do not specify a member name. – A member name, partial member name (name ending in an asterisk), or single asterisk. – *FIRST, for the first member in the file. • For a REPLACE or SELECT request, this field contains one of the following: – A member name	Character	10
Record format name	• For an EXTRACT request, this field contains one of the following: – Spaces, if you do not specify a format name. – A format name, a partial format name (name ending in an asterisk), or a single asterisk. – *ONLY, for the only format in the file. • For a SELECT or REPLACE request, this field contains one of the following: – A format name	Character	10
Length of next field	The next field is up to 4096 characters long.	Zoned decimal	5, 0
Entire request	The entire EXTRACT, SELECT, or REPLACE request.	Character	Up to 4096

Parameters for the Shared Folders Type 2 User Exit Program

When you specify user exit program and use shared folders type 2, the following occurs:

1. The shared folders type 2 function verifies your authority to the file or directory.
2. The shared folders type 2 function system program calls the specified user exit program.
3. The user exit program determines if the request should run.
4. The shared folders type 2 function program performs the request.

Figure 6-3 shows common parameter fields and their descriptions for the shared folders type 2 user exit program.

Figure 6-3 (Page 1 of 2). Parameter Fields for the Shared Folders Function Type 2 User Exit Program

Field	Description	Format	Size
User profile name	This field contains the user profile name.	Character	10
Application name	This field is always *FLRSRV indicating that the shared folders server calls the user exit program. **Note:** If a user exit program was specified for a previous AS/400 release of shared folders, this value was *DDM.	Character	10

Figure 6-3 (Page 2 of 2). Parameter Fields for the Shared Folders Function Type 2 User Exit Program

Field	Description	Format	Size
Requested function	This field contains one of the following: • CHANGE indicates a change file attributes request. • CREATE indicates a create stream file or create directory request. • DELETE indicates a delete file or delete directory request. • EXTRACT indicates a list file attributes request. • MOVE indicates a move request. • OPEN indicates an open stream file request. • RENAME indicates a rename request.	Character	10
Object name	This field is always *SPC for the shared folders function.	Character	10
Library name	This field is always *SPC for the shared folders function.	Character	10
Member name	This field is always *N for the shared folders function.	Character	10
Format name	This field is always spaces for the shared folders function.	Character	10
Length of the next field	This field contains the length of the next field.	Zoned decimal	5, 0
Other	This field contains the following structure:	Character	99
	• Not Used	Character	10
	• Not Used	Character	10
	If the requested function has the value OPEN, the first four fields apply:		
	• Read access, 1 = yes 0 = no	Character	1
	• Write access, 1 = yes 0 = no	Character	1
	• Read/Write access, 1 = yes 0 = no	Character	1
	• Delete allowed, 1 = yes 0 = no	Character	1
	• File (document) name	Character	12
	• Directory (folder) name	Character	63

Parameters for the Message Function User Exit Program

Figure 6-4 shows common parameter fields and their descriptions for the message function user exit program.

Figure 6-4. Parameter Fields for the Message Function User Exit Program

Field	Description	Format	Size
User profile name	This field contains your user profile name.	Character	10
Application name	This field is a constant *MSGFCL, indicating that the message function calls the program.	Character	10
Requested function	This field has a value of one of the following: • SEND if you want to send messages to others. • RECEIVE if you want to receive messages from others.	Character	10

Message function does not use any other section of the common portion of the second parameter. No application-specific information is used in the variable area.

Parameters for the Data Queues Function User Exit Program

When you specify the user exit program and use the data queues function, the following occurs:

1. The data queues target program calls the specified user exit program.
2. The user exit program determines if the request should run.
3. The data queues target program performs the request.

The user exit program is called for each CREATE, SEND, CLEAR, QUERY, RECEIVE, and PEEK request.

Figure 6-5 shows common parameter fields and their descriptions for the data queues function user exit program.

Figure 6-5 (Page 1 of 2). Parameter Fields for the Data Queues Function User Exit Program

Field	Value	Format	Size
User profile	User profile name	Character	10
Application	Constant *DQSRV	Character	10
Function	This field contains one of the following: • CREATE indicates a request to create a data queue. • RECEIVE indicates a request to receive a message from a data queue. • PEEK indicates a request to receive a message from a data queue without deleting it. • SEND indicates a request to send a message. • DELETE indicates a request to delete a data queue and all of its messages. • CLEAR indicates a request to clear all the messages from a data queue. • QUERY indicates a request to query a data queue.	Character	10

Figure 6-5 (Page 2 of 2). Parameter Fields for the Data Queues Function User Exit Program

Field	Value	Format	Size
Object name	Data queue name	Character	10
Library name	Library name	Character	10
Member name	Blanks	Character	10
Format name	Blanks	Character	10
Length next field	Total request length up to 260	Zoned decimal	5,0
Relational Op	Relational operator for Receive-by-Key request. Set to blanks for Send-by-Key request. EQ = is equal to NE = is not equal to GT = is greater than LT = less than GE = greater than or equal to LE = less than or equal to	Character	2
Key length	Key length	Zoned decimal	5,0
Key value	Key value (of length specified in key length).	Character	256

Parameters for the Remote SQL Function User Exit Program

When you specify the user exit program and use the remote SQL function, the following occurs:

1. Control is transferred to the user exit program.
2. The user exit program either accepts or rejects the remote SQL request.
3. If the request was accepted, the SQL operation is performed.
4. If the request was rejected:
   - An error return code is returned to the application.
   - The remote SQL server sends an SQL communications area (SQLCA) with the SQL code (SQLCODE) set to -901.
   - The SQL state is set to '56000', and the SQL error program (SQLERRP) is set to 'QRQSQL'.
   - The remote SQL server sends a diagnostic message to the host system job indicating the error.

   This happens regardless of which release (Version 2 Release 1 or later) you are using on your personal computer; the user exit processing is the same.

   Two remote SQL APIs can be used to determine why a request was rejected:
   EHNRQSQLCA Returns the remote SQL communications area
   EHNRQRTVMSG Retrieves the error message text
   See the "Remote SQL Function High-Level Application Programming Interface" or the "Remote SQL Windows Application Programming Interface" sections in the *PC Support/400 API Reference* for more details.

The user exit program is called for the Remote SQL APIs listed below. Each of these RQ APIs uses the following requests with a subset of the parameters, as shown in Figure 6-6 on page 6-9.

Note: The user profile, application, request function, and commit level parameters are **always** used by the user exit program. The Reserved parameter is **never** used by the user exit program.

Figure 6-6. Parameters Provided for Each Request Type

RQ API entry point	Request Function	Object Name	Library Name	Update Mode	Cursor ID	State-ment Name	SQL State-ment
EHNRQCONNECT	CONNECT						X
EHNRQDELETE	DELETE	1	1		X	1	X
EHNRQUPDATE	UPDATE	1	1		X	1	X
EHNRQEXEC	EXECUTE						X
EHNRQEXECPM	EXECUTEPM				X		X
EHNRQSELECT	SELECT			X	X		X
EHNRQSELECTPM	SELECTPM				X		X
EHNRQSELECTVAL	SELECTVAL			X	X		
EHNRQINVOKE	RMTCALL	X	X				
EHNRQPREPST	PREPTOPKG	X	X			X	X
EHNRQEXECST	EXECUTEPKG	X	X			X	
EHNRQEXECST	SELECTPKG	X	X	X	X	X	
EHNRQPREPST	CREATEPKG	X	X				

Note:

1 These parameters are also passed for a DELETE or an UPDATE request function if the request is issued after a SELECTPKG request function with the same cursor ID parameter.

User exit program support is not available for the remote SQL server APIs.

Figure 6-7 shows common parameter fields and their descriptions for the remote SQL function user exit program.

Figure 6-7 (Page 1 of 3). Parameter Fields for the Remote SQL Function User Exit Program

Field	Value	Format	Size
User profile	User profile name	Character	10
Application	Constant value *RQSRV	Character	10

Figure 6-7 (Page 2 of 3). Parameter Fields for the Remote SQL Function User Exit Program

Field	Value	Format	Size
Function	This field contains one of the following: • CONNECT indicates a request to connect to a remote database. • DELETE indicates a request to delete the current row of a specified cursor. • UPDATE indicates a request to update the current row of a specified cursor. • EXECUTE indicates a request for the immediate execution of a non-SELECT SQL statement. • EXECUTEPM indicates a request to prepare a non-SELECT SQL statement with a parameter marker. • SELECT indicates a request to select data records from the database for operations. • SELECTPM indicates a request to prepare a SELECT SQL statement with a parameter marker. • SELECTVAL indicates a request to run a prepared SELECT SQL statement with a parameter marker. • RMTCALL indicates a request to invoke an AS/400 application program. • PREPTOPKG indicates a request to prepare an SQL statement into a package. • SELECTPKG indicates a request to open a stored SELECT SQL statement. • EXECPKG indicates a request to execute a stored non-SELECT SQL statement. • CREATEPKG indicates a request to create a new package.	Character	10
Object name	This field contains one of the following: • If the request is a RMTCALL request, the object name is the program the personal computer application calls. • If the request is a PREPTOPKG request, the object name is the SQL package on which the request is acting.	Character	10
Library name	The library name on which the object (program or package) is acting.	Character	10
Commit level	This field contains one of the following: 0 = *NONE 1 = *ALL 2 = *CHG 3 = *CS See the *PC Support/400 API Reference* for a description of these commit levels.	Character	1
Update mode	The update mode specifies whether we are opening a file for read-only mode or for update mode: 0 = Read-only 1 = Update	Character	1
Cursor ID	The cursor ID identifies the active statement.	Character	1

Figure 6-7 (Page 3 of 3). Parameter Fields for the Remote SQL Function User Exit Program

Field	Value	Format	Size
Reserved	This field is reserved for future use.		4
Statement name	The statement name on which the SQL package request is acting.	Character	18
SQL statement	The first 4096 bytes of the SQL request.	Character	Up to 4096

Parameters for the License Management Function User Exit Program

When you specify the user exit program and use the license management function, the following occurs:

1. Control is transferred to the user exit program.
2. The user exit program either accepts or rejects the license management request.
3. If the request is accepted, the requested operation is performed.
4. If the request is rejected, the user exit program has an error return code, or the network attribute is *REJECT.

Figure 6-8 shows common parameter fields and their descriptions for the license management function user exit program.

Figure 6-8. Parameter Fields for the License Management Function User Exit Program

Field	Value	Format	Size
User profile name	This field contains your user profile name.	Character	10
Application name	This field is a constant value, *LMSRV, indicating that the license management function calls the program.	Character	10
Requested function	This field contains one of the following values: • REQUEST indicates you are requesting a license. • RELEASE indicates you are releasing a license.	Character	10
PC ID	This field contains the PC identifier of the PC for which you are requesting or releasing a license.	Character	10
Product ID	This field contains the product identifier of the product for which you are requesting or releasing a license.	Character	10
Feature ID	This field contains the feature identifier of the product for which you are requesting or releasing a license.	Character	10
Release	This field contains the release (in the form VxRyMz) of the product for which you are requesting or releasing a license.	Character	10

Sample User Exit Programs

These examples do not show all programming considerations or techniques. Review the examples before you begin application design and coding.

Creating User Exit Programs Using RPG/400

Figure 6-9 illustrates how to set up a user exit program using RPG/400.* Use Figure 6-10 on page 6-15 as an algorithm to be used in a PC Support application.

```
**
** PC SUPPORT/400 - SAMPLE USER EXIT PROGRAM
**
** THE FOLLOWING RPG/400 PROGRAM UNCONDITIONALLY
** ACCEPTS ALL REQUESTS.  IT CAN BE USED AS A SHELL
** FOR SPECIFIC APPLICATIONS.  NOTE: REMOVE THE
** SUBROUTINES AND CASE STATEMENT ENTRIES FOR PC
** SUPPORT/400 APPLICATIONS THAT DO NOT REQUIRE
** SPECIFIC EXIT PROGRAM HANDLING FOR BETTER PERFORMANCE.
**
E*
E* NECESSARY ARRAY DEFINITIONS FOR TRANSFER FUNCTION
E* AND REMOTE SQL
E*
E                    TFREQ    4096  1
E                    RSREQ    4107  1
I*
I* PC SUPPORT PARAMETER DATA STRUCTURE
I*
IPCSDTA      DS                          4171
I*
I* THE FOLLOWING HEADER INFORMATION IS GIVEN FOR ALL
I* PC SUPPORT FUNCTIONS.  IT IDENTIFIES THE USER
I* PROFILE, PC SUPPORT APPLICATION IDENTIFIER, AND
I* OPERATION BEING PERFORMED
I*
I                                       1  10 USERID
I                                      11  20 APPLID
I                                      21  30 FUNCID
```

Figure 6-9 (Part 1 of 4). Example of an RPG/400 User Exit Program

```
I*
I* SPECIFIC PARAMETERS FOR VIRTUAL PRINTER
I*
I                                       31  40 VPOBJ
I                                       41  50 VPLIB
I                                       71  750VPLEN
I                                       76  85 VPOUTQ
I                                       86  95 VPQLIB
I*
I* SPECIFIC PARAMETERS FOR TRANSFER FUNCTION
I*
I                                       31  40 TFOBJ
I                                       41  50 TFLIB
I                                       51  60 TFMBR
I                                       61  70 TFFMT
I                                       71  750TFLEN
I                                       764171 TFREQ
I*
I* SPECIFIC PARAMETERS FOR SHARED FOLDERS
I*
I                                       96  96 SFREAD
I                                       97  97 SFWRIT
I                                       98  98 SFRW
I                                       99  99 SFDLT
I                                      100 111 SFDOC
I                                      112 174 SFFLR
I*
I* SPECIFIC PARAMETERS FOR DATA QUEUES
I*
I                                       31  40 DQQ
I                                       41  50 DQLIB
I                                       70  75 DQLEN
I                                       76  77 DQROP
I                                       78  82 DQKLEN
I                                       83 338 DQKEY
I*
I* SPECIFIC PARAMETERS FOR REMOTE SQL
I*
I                                       31  40 RSOBJ
I                                       41  50 RSLIB
I                                       51  51 RSCMT
I                                       52  52 RSMODE
I                                       53  53 RSCID
I                                       54  71 RSSTN
I                                       72  75 RSRSV
I                                       764172 RSREQ
```

Figure 6-9 (Part 2 of 4). Example of an RPG/400 User Exit Program

```
I*
I* THERE ARE NO SPECIFIC PARAMETERS FOR MESSAGING FUNCTION
I*
C*
C* EXIT PROGRAM CALL PARAMETERS
C*
C           *ENTRY    PLIST
C                     PARM           RTNCD   1
C                     PARM           PCSDTA
C*
C* INITIALIZE RETURN VALUE TO ACCEPT REQUEST
C*
C                     MOVE '1'       RTNCD
C*
C* COMMON PROCESSING
C*
C*                    COMMON LOGIC GOES HERE
C*
C* PROCESS BASED ON PC SUPPORT/400 APPLICATION
C*
C           APPLID    CASEQ'*VPRT'   VPRT
C           APPLID    CASEQ'*TFRFCL' TFR
C           APPLID    CASEQ'*FLRSRV' FLR
C           APPLID    CASEQ'*MSGFCL' MSG
C           APPLID    CASEQ'*DQSRV'  DATAQ
C           APPLID    CASEQ'*RQSRV'  RSQL
C                     END
C                     SETON                      LR
C                     RETRN
C*
C* SUBROUTINES
C*
C*
C* VIRTUAL PRINT
C*
C           VPRT      BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
C*
C* TRANSFER FUNCTION
C*
C           TFR       BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
C*
C* SHARED FOLDERS
C*
C           FLR       BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
```

Figure 6-9 (Part 3 of 4). Example of an RPG/400 User Exit Program

```
C*
C* MESSAGING FUNCTION
C*
C           MSG       BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
C* DATA QUEUES
C*
C           DATAQ     BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
C*
C* REMOTE SQL
C*
C           RSQL      BEGSR
C*                    SPECIFIC LOGIC GOES HERE
C                     ENDSR
```

Figure 6-9 (Part 4 of 4). Example of an RPG/400 User Exit Program

Using a RPG/400 User Exit Program

Figure 6-10 shows you how you can use the example shown in Figure 6-9 on page 6-12 to create an exit program algorithm for transfer function.

```
C* THE FOLLOWING IS AN EXAMPLE OF SPECIFIC PROCESSING
C* THAT THE EXIT PROGRAM COULD DO FOR TRANSFER FUNCTION.
C*
C* IN THIS CASE, USERS ARE NOT ALLOWED TO SELECT
C* DATA FROM ANY FILES THAT ARE IN LIBRARY QIWS.
C*
C           TFR       BEGSR
C           FUNCTN    IFEQ 'SELECT'
C           TFLIB     ANDEQ'QIWS'
C                     MOVE '0'       RTNCD
C                     END
C                     ENDSR
```

Figure 6-10. An Algorithm for the Transfer Function Written in RPG/400

Creating User Exit Programs Using Control Language

Figure 6-11 on page 6-16 illustrates how to set up a user exit program using control language (CL). Use Figure 6-12 on page 6-21 as an algorithm to be used in a PC Support application.

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                        */
/* PC SUPPORT/400 - SAMPLE USER EXIT PROGRAM                              */
/*                                                                        */
/* THE FOLLOWING CONTROL LANGUAGE PROGRAM UNCONDITIONALLY                 */
/* ACCEPTS ALL REQUESTS.  IT CAN BE USED AS A SHELL FOR DEVELOPING        */
/* EXIT PROGRAMS TAILORED FOR YOUR OPERATING ENVIRONMENT.                 */
/*                                                                        */
/*                                                                        */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
PGM PARM(&STATUS &REQUEST)

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                        */
/* PROGRAM CALL PARAMETER DECLARATIONS                                    */
/*                                                                        */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/

  DCL VAR(&STATUS) TYPE(*CHAR) LEN(1) /* Accept/Reject indicator    */
  /*                                                                      */
  /*  Note: Request is declared as *CHAR LEN(2000) because that is  */
  /*    the limit in CL. The actual length of REQUEST is 4171.      */
  /*                                                                      */
  DCL VAR(&REQUEST) TYPE(*CHAR) LEN(2000) /* Parameter structure    */

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                        */
/* PC SUPPORT PARAMETER DECLARES                                          */
/*                                                                        */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/

  /* COMMON DECLARES */
  DCL VAR(&USER)   TYPE(*CHAR) LEN(10) /* User ID using PC Support  */
  DCL VAR(&APPLIC) TYPE(*CHAR) LEN(10) /* PC Support/400 application*/
  DCL VAR(&FUNCTN) TYPE(*CHAR) LEN(10) /* Function being performed  */

  /* VIRTUAL PRINT DECLARES */
  DCL VAR(&VPOBJ)  TYPE(*CHAR) LEN(10) /* Object name               */
  DCL VAR(&VPLIB)  TYPE(*CHAR) LEN(10) /* Object library name       */
  DCL VAR(&VPLEN)  TYPE(*DEC)  LEN(5 0)/* Length of following fields*/
  DCL VAR(&VPOUTQ) TYPE(*CHAR) LEN(10) /* Output queue name         */
  DCL VAR(&VPQLIB) TYPE(*CHAR) LEN(10) /* Output queue library name */
```

Figure 6-11 (Part 1 of 5). Example of a CL User Exit Program

```
/* TRANSFER FUNCTION DECLARES */
DCL VAR(&TFOBJ)  TYPE(*CHAR) LEN(10) /* Object name              */
DCL VAR(&TFLIB)  TYPE(*CHAR) LEN(10) /* Object library name      */
DCL VAR(&TFMBR)  TYPE(*CHAR) LEN(10) /* Member name              */
DCL VAR(&TFFMT)  TYPE(*CHAR) LEN(10) /* Record format name       */
DCL VAR(&TFLEN)  TYPE(*DEC)  LEN(5 0)/* Length of request        */
DCL VAR(&TFREQ)  TYPE(*CHAR) LEN(1925)/*Transfer request statement*/

/* SHARED FOLDERS DECLARES */
DCL VAR(&SFREAD) TYPE(*CHAR) LEN(1)  /* Read access indicator    */
DCL VAR(&SFWRITE) TYPE(*CHAR) LEN(1) /* Write access indicator   */
DCL VAR(&SFRDWRT) TYPE(*CHAR) LEN(1) /* Read/Write access indicator*/
DCL VAR(&SFDLT)  TYPE(*CHAR) LEN(1)  /* Delete allowed indicator */
DCL VAR(&SFDOC)  TYPE(*CHAR) LEN(12) /* Document object name     */
DCL VAR(&SFFLR)  TYPE(*CHAR) LEN(63) /* Folder name              */

/* DATA QUEUE DECLARES */
DCL VAR(&DQQ)    TYPE(*CHAR) LEN(10) /* Data queue name          */
DCL VAR(&DQLIB)  TYPE(*CHAR) LEN(10) /* Data queue library name  */
DCL VAR(&DQLEN)  TYPE(*DEC) LEN(5 0) /* Total request length     */
DCL VAR(&DQROP)  TYPE(*CHAR) LEN(2)  /* Relational operator      */
DCL VAR(&DQKLEN) TYPE(*DEC) LEN(5 0) /* Key length               */
DCL VAR(&DQKEY)  TYPE(*CHAR) LEN(256)/* Key value                */

/* REMOTE SQL DECLARES */
DCL VAR(&RSOBJ)  TYPE(*CHAR) LEN(10) /* Object name              */
DCL VAR(&RSLIB)  TYPE(*CHAR) LEN(10) /* Object library name      */
DCL VAR(&RSCMT)  TYPE(*CHAR) LEN(1)  /* Commitment control level */
DCL VAR(&RSMODE) TYPE(*CHAR) LEN(1)  /* Block/Update mode indicator*/
DCL VAR(&RSCID)  TYPE(*CHAR) LEN(1)  /* Cursor ID                */
DCL VAR(&RSSTN)  TYPE(*CHAR) LEN(18) /* Statement name           */
DCL VAR(&RSRSV)  TYPE(*CHAR) LEN(4)  /* Reserved                 */
DCL VAR(&RSREQ)  TYPE(*CHAR) LEN(1925)/* SQL statement           */

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                               */
/* OTHER DECLARES                                                */
/*                                                               */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/

  DCL VAR(&WRKLEN) TYPE(*CHAR) LEN(5)
```

Figure 6-11 (Part 2 of 5). Example of a CL User Exit Program

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                       */
/* EXTRACT THE VARIOUS PARAMETERS FROM THE STRUCTURE                     */
/*                                                                       */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/

  /* HEADER */
  CHGVAR VAR(&USER)    VALUE(%SST(&REQUEST  1 10))
  CHGVAR VAR(&APPLIC)  VALUE(%SST(&REQUEST 11 10))
  CHGVAR VAR(&FUNCTN)  VALUE(%SST(&REQUEST 21 10))

  /* VIRTUAL PRINTER */
  CHGVAR VAR(&VPOBJ)   VALUE(%SST(&REQUEST 31 10))
  CHGVAR VAR(&VPLIB)   VALUE(%SST(&REQUEST 41 10))
  CHGVAR VAR(&WRKLEN)  VALUE(%SST(&REQUEST 71  5))
  CHGVAR VAR(&VPLEN)   VALUE(&WRKLEN)
  CHGVAR VAR(&VPOUTQ)  VALUE(%SST(&REQUEST 76 10))
  CHGVAR VAR(&VPQLIB)  VALUE(%SST(&REQUEST 86 10))

  /* TRANSFER FUNCTION */
  CHGVAR VAR(&TFOBJ)   VALUE(%SST(&REQUEST 31 10))
  CHGVAR VAR(&TFLIB)   VALUE(%SST(&REQUEST 41 10))
  CHGVAR VAR(&TFMBR)   VALUE(%SST(&REQUEST 51 10))
  CHGVAR VAR(&TFFMT)   VALUE(%SST(&REQUEST 61 10))
  CHGVAR VAR(&WRKLEN)  VALUE(%SST(&REQUEST 71  5))
  CHGVAR VAR(&TFLEN)   VALUE(&WRKLEN)
  CHGVAR VAR(&TFREQ)   VALUE(%SST(&REQUEST 76 1925))

  /* SHARED FOLDERS */
  CHGVAR VAR(&SFREAD)  VALUE(%SST(&REQUEST  96  1))
  CHGVAR VAR(&SFWRITE) VALUE(%SST(&REQUEST  97  1))
  CHGVAR VAR(&SFRDWRT) VALUE(%SST(&REQUEST  98  1))
  CHGVAR VAR(&SFDLT)   VALUE(%SST(&REQUEST  99  1))
  CHGVAR VAR(&SFDOC)   VALUE(%SST(&REQUEST 100 12))
  CHGVAR VAR(&SFFLR)   VALUE(%SST(&REQUEST 112 63))

  /* DATA QUEUES */
  CHGVAR VAR(&DQQ)     VALUE(%SST(&REQUEST 31 10))
  CHGVAR VAR(&DQLIB)   VALUE(%SST(&REQUEST 41 10))
  CHGVAR VAR(&WRKLEN)  VALUE(%SST(&REQUEST 71  5))
  CHGVAR VAR(&DQLEN)   VALUE(&WRKLEN)
  CHGVAR VAR(&DQROP)   VALUE(%SST(&REQUEST 76  2))
  CHGVAR VAR(&WRKLEN)  VALUE(%SST(&REQUEST 78  5))
  CHGVAR VAR(&DQKLEN)  VALUE(&WRKLEN)
  CHGVAR VAR(&DQKEY)   VALUE(%SST(&REQUEST 83 &DQKLEN))
```

Figure 6-11 (Part 3 of 5). Example of a CL User Exit Program

```
  /* REMOTE SQL */
  CHGVAR VAR(&RSOBJ)   VALUE(%SST(&REQUEST 31 10))
  CHGVAR VAR(&RSLIB)   VALUE(%SST(&REQUEST 41 10))
  CHGVAR VAR(&RSCMT)   VALUE(%SST(&REQUEST 51 1))
  CHGVAR VAR(&RSMODE)  VALUE(%SST(&REQUEST 52 1))
  CHGVAR VAR(&RSCID)   VALUE(%SST(&REQUEST 53 1))
  CHGVAR VAR(&RSSTN)   VALUE(%SST(&REQUEST 54 18))
  CHGVAR VAR(&RSRSV)   VALUE(%SST(&REQUEST 72 4))
  CHGVAR VAR(&RSREQ)   VALUE(%SST(&REQUEST 76 1925))

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                         */
/* BEGIN MAIN PROGRAM                                                      */
/*                                                                         */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/


  CHGVAR     VAR(&STATUS) VALUE('1') /* INITIALIZE RETURN +
                                        VALUE TO ACCEPT THE REQUEST */

  /* ADD LOGIC COMMON TO ALL PC SUPPORT/400 APPLICATIONS HERE */

  /* PROCESS BASED ON PC SUPPORT/400 APPLICATION */
  IF COND(&APPLIC *EQ '*VPRT') THEN(+
     GOTO CMDLBL(VPRT))                  /* IF VIRTUAL PRINTER */
  IF COND(&APPLIC *EQ '*TFRFCL') THEN(+
     GOTO CMDLBL(TFR))                   /* IF TRANSFER FUNCTION */
  IF COND(&APPLIC *EQ '*FLRSRV') THEN(+
     GOTO CMDLBL(FLR))                   /* IF SHARED FOLDERS */
  IF COND(&APPLIC *EQ '*MSGFCL') THEN(+
     GOTO CMDLBL(MSG))                   /* IF MESSAGING FUNCTION */
  IF COND(&APPLIC *EQ '*DQSRV') THEN(+
     GOTO CMDLBL(DATAQ))                 /* IF DATA QUEUES */
  IF COND(&APPLIC *EQ '*RQSRV') THEN(+
     GOTO CMDLBL(RSQL))                  /* IF REMOTE SQL */

  GOTO EXIT
```

Figure 6-11 (Part 4 of 5). Example of a CL User Exit Program

```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/
/*                                                                       */
/* SUBROUTINES                                                           */
/*                                                                       */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **/

/* VITUAL PRINTER */
  VPRT:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
/* TRANSFER FUNCTION */
  TFR:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
/* SHARED FOLDERS */
  FLR:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
/* MESSAGING FUNCTION */
  MSG:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
/* DATA QUEUES */
  DATAQ:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
/* REMOTE SQL */
  RSQL:

    /* SPECIFIC LOGIC GOES HERE */

    GOTO EXIT
 EXIT:
ENDPGM
```

Figure 6-11 (Part 5 of 5). Example of a CL User Exit Program

Using a CL User Exit Program

Figure 6-12 on page 6-21 shows you how you can use the example shown in Figure 6-11 on page 6-16 to create an exit program algorithm for the transfer function.

```
/*      THE FOLLOWING IS AN EXAMPLE OF SPECIFIC PROCESSING          */
/*      THAT THE EXIT PROGRAM COULD DO FOR TRANSFER FUNCTION.       */
/*                                                                  */
/*      IN THIS CASE, USERS ARE NOT ALLOWED TO SELECT               */
/*      DATA FROM ANY FILES THAT ARE IN LIBRARY QIWS.               */
/*                                                                  */

/* TRANSFER FUNCTION */

  TFR:

    IF COND((&FUNCTN *EQ 'SELECT') *AND (&TFLIB *EQ 'QIWS')) +
      THEN(CHGVAR VAR(&STATUS) VALUE('0'))

    GOTO EXIT
```

Figure 6-12. An Algorithm for the Transfer Function Written in AS/400 Control Language

Chapter 7. Submit Remote Command Function

This chapter discusses the Submit Remote Command (RMTCMD) function and its error levels.

The RMTCMD function consists of RMTCMD.EXE which can be started at either the DOS or OS/2 prompt. The function submits control language (CL) commands from a personal computer to an AS/400 system through distributed data management (DDM). The commands submitted with the RMTCMD function cannot be used to open files or update individual records.

The RMTCMD function supports two types of CL commands, bimodal (those that are supported in both batch and interactive environments) and batch. You must have proper authority both for the CL command being submitted and for the remote system on which the command is to operate. Refer to the *CL Reference* for information on which commands are supported in the batch environment and for authority requirements of those commands.

Note: For better AS/400 system performance when submitting bimodal commands, run jobs that do not take much processing time interactively and submit jobs that take a lot of processing time to batch processing. The AS/400 Submit Job Command (SBMJOB) function can be used to submit jobs to batch.

RMTCMD Program

Submit commands to the remote system from the DOS or OS/2 prompt. There are two command line interfaces. Use one interface for submitting a single batch command to the remote system and the other interface for submitting multiple batch commands to the remote system. Multiple batch commands are stored in an ASCII text file. If you are using DBCS, RMTCMD.EXE exists only for the OS/2 program.

Sending a Single Batch Command

The format of a single batch command is:

```
RMTCMD remote command [//RemoteSysName] [/Z]
```

remote command
The CL command, typed as it would be typed on the remote system.

This command should be surrounded by double quotation marks if it contains anything that looks like another command line option, such as /Z.

//RemoteSysName
The remote system name.

This parameter is optional. If you do not specify the remote system name, the default remote system name will be used.

/Z The message suppression option.

This parameter is optional. If selected, informational messages will be suppressed.

The system does not prompt for any input after you enter the single batch command.

If the command is sent, a remote system generated message is displayed. (The remote system sends the message to the personal computer where it is displayed.) If the command fails because of a communications problem, for example, the local RMTCMD program displays a message indicating the problem. Refer to the online help for the cause of, and recovery from, any messages that the RMTCMD program displays.

Sending Multiple Batch Commands

The `/I file name` option is required. `/Q`, `//RemoteSysName`, and `/Z` are optional.

```
RMTCMD /I file name [/Q] [//RemoteSysName] [/Z]
```

/I file name
The name of the file containing the CL remote commands. This parameter is required. The file name must be a valid file name. There must be at least one space between /I and the file name. The file name may be preceded by a valid drive letter, a colon (:) and a valid path.

/Q An instruction to the RMTCMD program not to prompt you if an error level with an optional value greater than or equal to 20 is returned by one of the commands in the file. All commands in the file will process unless a severe error has occurred, for example, a communications error, that prevents further processing.

//RemoteSysName
The remote system name. This parameter is optional.

/Z The message suppression option.

This parameter is optional. If selected, informational messages will be suppressed.

The file that contains the individual commands has the following characteristics:

- Each command string is on a separate line.
- The maximum length of a command is 2000 characters.
- The Carriage-Return+Linefeed character sequence is used as a delimiter.
- Because the commands are submitted sequentially, the commands are in the order in which you want them processed.
- **For OS/2 only:** If a fully-qualified file name is not given, then the file of commands should be in a directory specified in DPATH, or in the current directory.

Any messages for a specific command are displayed after the command is submitted. If an error level with a value greater than or equal to 20 is returned by any of the submitted commands, the following message appears unless /Q is specified.

```
Press Enter to continue or Press Esc to cancel
```

If you press the Enter key (the Carriage-Return), the next command in the file is submitted. If you press the Esc (Escape) key, the RMTCMD program ends without submitting any more commands.

The following error levels may be returned by the RMTCMD program:

0 No errors or warnings.

10 Warning(s) returned from the remote system.

20 Error(s) returned from the remote system, or the key sequence Control+Break was typed, or a bad MRI file was typed.

21 Severe error(s) returned from the remote system.

22 DDM on the remote system has had its ability to access a file damaged.

23 Permanent damage has occurred to permanent objects on the remote system.

24 DDM on the remote system has been damaged. No further communication will be possible during this session.

30 DDM on the remote system has reached the limit of its resources.

31 Command string syntax problem. This error occurs if the command string is longer than 2000 bytes.

41 Remote system name syntax problem. This error occurs if the system name is longer than 8 bytes.

42 Not able to find or open the file containing the remote commands.

44 There is a problem in the command line syntax.

45 An error has occurred while reading the input file.

49 The remote system does not support the DDM command to submit remote commands.

50 Contact with remote system ended.

51 Conversation with remote system unexpectedly ended.

52 DDM on the remote system unexpectedly ended.

53 Conversation allocation unsuccessful because a session was not available.

54 Remote system named not found.

55 Insufficient remote system resource to start DDM on remote system.

56 Contact with remote system temporarily interrupted. Try the operation again.

57 DDM not found on the remote system.

58 User ID or password is not correct.

68 Too many programs are using the OS/2 Communications Manager.

69 The OS/2 Communications Manager is not available.

70 The router is not loaded.

71 The OS/2 Communications Manager's configuration of the local system LU does not match the router's LCLN entry.

72 The API could not dynamically allocate sufficient memory. (Applies to the OS/2 operating system only.)

95 DBCS translation not available.

99 This is an internal error that should not normally occur.

100 Not able to load MRI.

Running the RMTCMD Program Using DOS Function Call 4BH

When you run the RMTCMD program using a DOS function call, make sure the command line in the program segment prefix (PSP) ends with a hexadecimal X'0D' (a Carriage-Return).

Chapter 8. Using PC Support/400 and the Microsoft Windows Program

You should consider the following when you use PC Support/400 with Microsoft Windows:

- You should be using the extended DOS version of PC Support/400.
- Running PC Support/400 in Windows real mode is not supported. You must be using either standard mode or 386 enhanced mode.
- Expanded Memory Specifications (EMS) is not supported by PC Support/400 while the Windows program is active.
- Running the PC Support asynchronous or SDLC router with Windows in enhanced mode is not supported.

  Note: Microsoft Windows does not support the higher baud rates for the PC Support asynchronous router. When running the asynchronous router with Windows in standard mode, you may have to lower the baud rate used by the asynchronous router. This will improve your response time. For information on changing your baud rate, see the description of the ASBR identifier in the *PC Support/400 DOS Installation and Administration Guide*.

Windows Compatibility Program (PCSWIN)

The Microsoft Windows program creates a multiple-task environment running on top of a single-task operating system, DOS. PCSWIN resolves this multiple-task environment between PC Support and Windows and enables the windows to PC Support programs.

Program Considerations

The PC Support programs, like most DOS applications, are non-reentrant, meaning that while they are running, they do not expect to be interrupted and called again. Under Windows, however, a PC Support task that is running can be switched, or stopped, to allow another task to run. When the PC Support task is allowed to run again, the computer may not be in the same state it originally was.

The PCSWIN program prevents Windows from task switching while PC Support functions are in a non-reentrant state, sometimes referred to as a critical section. Critical section handling is performed automatically by PCSWIN for user applications that make use of the PC Support application program interfaces (APIs).

Instancing Data

Each DOS session running in Windows 386 enhanced mode has a unique environment. Information about this environment, such as the current drive and path, is kept for each DOS window. This technique is called instancing. Instancing allows changes to one DOS session without affecting other sessions.

The PCSWIN program provides the instancing interface for PC Support automatically for applications using PC Support functions.

Router Buffer Handling

Applications written to the PC Support router API need to provide a buffer when using the allocate verb. This buffer is usually allocated by the application and used by the router. An application running in a Microsoft Windows environment may not be active when the router needs to use the buffer. As a result, data may be damaged.

The PCSWIN program solves this problem by allocating a buffer pool that is always available to the router. As applications perform allocate verbs, PCSWIN provides buffers out of its pool and gives them to the router. Buffers are automatically allocated by PCSWIN when Windows is running. However, applications need to provide for the possibility of PCSWIN running out of space in its buffer pool. An error return code is displayed. Refer to the *PC Support/400 API Reference* for more information.

The buffer pool for PCSWIN is configurable, and you can start PCSWIN with a buffer as large as 64K. The default size is 16K. The recommended allocation to the buffer pool is 1K larger than you require because PCSWIN uses part of the buffer for its own use. PCSWIN automatically allocates the buffer in the upper memory block, if enough memory is available. No configuration parameter is available because the configuration is done automatically. Refer to the *PC Support/400 DOS Installation and Administration Guide* for the buffer sizes for the various PC Support functions.

Network Driver

The IBM AS/400 network driver provides an integrated Windows user interface for several network tasks.

- Connecting network drives and printers.
- Managing network print jobs.
- Disconnecting network drives and printers.

The PC Support network driver uses the DOS network redirection interface to connect and disconnect drives. It also uses the DOS network redirection interface to connect and disconnect network printers, if the remote system is not an AS/400 system. Therefore, the network driver can be used for connecting and disconnecting network drives and printers other than those attached by the network driver, provided these networks also use the DOS network redirection interface.

Connecting to a Non-PC Support Network Drive

To connect a drive that is not connected through PC Support network, you need to perform the following steps:

1. Select the Windows File Manager from within Windows.
2. Select the Connect Net Drive... option from the Disk menu.
3. Type the network path for the drive to be assigned and the password for the network, if required (for example, \\NETSYS\NETDRIVE and NETPASSWD).

The PC Support network driver issues a DOS redirect device call (INT 21 Function 5F03). Any network using this interface can access the redirection information and respond as appropriate.

Connecting to a Non-PC Support Network Printer

Similarly, to connect a non-PC Support network printer, you need to perform the following steps:

1. Select the Printers icon from the Windows Control Panel.
2. Select the Network... push button.
3. Type the network path for the printer to be assigned and the password for the network, if required (for example, \\NETSYS\NETPRINT and NETPASSWD).

The PC Support network driver determines that the destination system is not an AS/400 system and issues the DOS redirect device call (INT 21 Function 5F03). From here, any network using this interface can access the redirection information and respond as appropriate.

Some of the functions provided by the network driver are specific to PC Support/400 and, therefore, are not compatible with other networks. For example, the browse and print manager functions only work with PC Support/400.

For more information on the DOS network redirection interface, see the *DOS Technical Reference*.

PC Support/400 Dynamic Link Libraries (DLLs)

Several of the existing APIs for PC Support have been extended to the Microsoft Windows environment. They provide the developer of cooperative applications with the tools necessary to create Windows applications that interact with AS/400 programs and resources. The following APIs are available:

- Router
- Shared Folders
- Data Queues
- Transfer Function
- Remote Command
- Remote SQL
- Virtual Print

For more detailed information about the DLLs, see the individual chapters in the *PC Support/400 API Reference*.

Router API for Microsoft Windows

This API provides the same level of function for writing cooperative APPC applications as the API for DOS. The buffer used by the router will be provided from a pool of buffers that have been pre-allocated by the PCSWIN program. See "Router Buffer Handling" on page 8-2 for more information.

Data Queues API for Microsoft Windows

In addition to supporting the same level of function as the API in DOS, the Data Queues API for Windows has additional APIs that take advantage of Windows' ability to do multiple tasks.

The API to receive a message from a data queue is designed to wait until the message arrives from the AS/400 data queue. This stops all activity on the personal computer until the message arrives. Two APIs allow Windows to continue

doing multiple tasks. The first API requests the message from an AS/400 data queue, and the second checks if the message is available.

For more detailed information on the Windows DLLs, refer to the *PC Support/400 API Reference*.

Appendix A. Using the Printer Function Table Setup Program

This appendix contains information on the printer function table (PFT) setup program. It describes how to create a PFT, how to enter control sequences, and how to define printer functions. It also discusses function selection tests.

The PFT setup program is intended for use by programmers, technicians, or printer function experts. If the technical information for your personal printer is not available, contact the personal printer dealer or manufacturer for assistance.

You can use the PFT setup program with DOS, extended DOS, or the OS/2 operating system. In the OS/2 operating system, it must run in DOS compatibility mode. If you use the PFT setup program in the DOS compatibility mode, you create printer function tables that can be used with the DOS work station function and not the OS/2 Communications Manager work station feature.

Printer Function Table Setup Program Overview

The printer function table (PFT) describes the functional characteristics of the attached personal printer to work station function. You need a PFT to use a personal printer with work station function. Create and edit this table with the PFT setup program (PFTSETUP). Refer to the *PC Support/400 DOS Installation and Administration Guide* and to the *PC Support/400 OS/2 Installation and Administration Guide*, or to the *OS/2 System Administrator's Guide for Communications* for a list of supported printers.

You need a minimum of 330KB of usable storage to run PFTSETUP.

While you create or change a PFT, you can run tests to determine the operation and characteristics of your personal printer.

When you run the first test, PFTSETUP automatically runs a test to determine if a form feed control must appear at the end of each print test for you to see the results of each test.

Work station function may not require or use all the functions provided by your personal printer. Also, work station function does not use all the functions provided by the setup program itself. Work station function uses the following PFT functions:

- Start of document initialization control sequence
- End of document reset control sequence
- Lines per inch
- Variable line spacing
- Begin/end underline
- Begin/end superscript
- Begin/end subscript
- 1/2 index up
- 1/2 index down
- 5, 10, 12, 15, 17.1 pitch
- Horizontal motion index
- Forward horizontal relative movement
- Backward horizontal relative movement
- Begin/end emphasis

- Begin/end quality print
- Bottom tray select and feed
- Top tray select and feed
- Envelope tray select and feed
- Eject sheet without paper feed
- Manual feed select
- Paper positioning
- Reverse index
- Set page length
- Landscape printing orientation bottom tray select and feed
- Landscape printing orientation top tray select and feed
- Landscape printing orientation envelope tray select and feed
- Landscape printing orientation manual feed select
- Default type style
- Individual type style
- Group type style
- Character set redefinition
- User-defined control, number 984
- User-defined control, number 985
- User-defined control, number 986
- User-defined control, number 987
- User-defined control, number 988
- User-defined control, number 989
- User-defined control, number 990
- User-defined control, number 991
- User-defined control, number 992
- User-defined control, number 993
- User-defined control, number 994
- User-defined control, number 995
- User-defined control, number 996
- User-defined control, number 997
- User-defined control, number 998
- User-defined control, number 999

Work station function does *not* use the following functions:

- Control sequence file name
- 8.55 pitch
- Continuous feed select
- Ignore/enable paper end sensor
- Collate
- Set left margin
- Carrier return/line feed
- Multiple copies
- Slot selection definition
- User-defined controls, except numbers 984 through 999

PFTs Provided by Work Station Function

Work station function provides the following tables:

DEFAULT.PFT This table allows printing of simple output (no changes in pitch, type style, and so on). You may be able to use this table initially to define your personal printer. If you need to

create a PFT for more complex functions, you can copy this PFT to use as a guideline.

IBM3812.PFT This PFT supports the IBM 3812 Pageprinter. If you have a 3812, you must use the PFT. You cannot recreate this table with the setup program, or create a new table to support the 3812. However, you can view this table to understand the information you specify for your page printer.

If you have an IBM 3812 Pageprinter, do not change the type styles/character sets portion of this table.

IBM2380.PFT This PFT supports the IBM 2380 printer and IBM 2381 printers.

IBM2390.PFT This PFT supports the IBM 2390 printer and IBM 2391 printers.

IBM3816.PFT This PFT supports the IBM 3816 Pageprinter.

IBM4019.PFT This PFT supports the IBM 4019 LaserPrinter in IBM ASCII mode.

IBM4029.PFT This PFT supports the IBM 4029 LaserPrinter in IBM ASCII mode.

IBM4072.PFT This PFT supports the IBM 4072 ExecJet* printer.

IBM4202.PFT This PFT supports the IBM 4202 Proprinter XL, 4202 Proprinter II XL, 4202 Proprinter III XL, 4201 Proprinter III, 4201-1 Proprinter, 4201-2 Proprinter III, and the IBM 4226 Model 302 printer.

IBM4208.PFT This PFT supports the IBM 4207 Proprinter X24, IBM 4207 Proprinter X24E, IBM 4208 Proprinter XL24, and the IBM 4208 Proprinter XL24E.

IBM4212.PFT This PFT supports the IBM 4212 Proprinter (24 pin).

IBM4216.PFT This PFT supports the IBM 4216-010/D10 Personal Pageprinter in HP** mode.

IBM5152.PFT This PFT supports the IBM 5212 Graphics printer.

IBM5201.PFT This PFT supports the IBM 5201-002 Quietwriter* Printer.

IBM5202.PFT This PFT supports the IBM 5202 Quietwriter III Printer and the IBM 5201 Quietwriter Printer.

IBM5204.PFT This PFT supports the IBM 5204 Quickwriter* Printer.

IBM5216.PFT This PFT supports the IBM 5216 Wheelprinter Printer.

HPLASER2.PFT This PFT supports the HP LaserJet** Series II, the HP LaserJet Series IIP, and the IBM 4019 LaserPrinter in HP mode.

HPLASER3.PFT This PFT supports the HP LaserJet Series III, the HP LaserJet Series IIIsi, and the HP LaserJet Series IIIP.

HPLASERD.PFT This PFT supports the HP LaserJet Series IID.

HPLASR3D.PFT This PFT supports the HP LaserJet Series IIID.

HPDSKJT.PFT This PFT supports the HP DeskJet 500.

EFX850.PFT	This PFT supports the Epson** FX-850.
ELX810.PFT	This PFT supports the Epson LX-810.
ELQ510.PFT	This PFT supports the Epson LQ-510.
ELQ2550.PFT	This PFT supports the Epson LQ-2550.
NEC2200.PFT	This PFT supports the NEC** P2200.
NEC5300.PFT	This PFT supports the NEC P5200 and P5300 personal printers.
NEC6300.PFT	This PFT supports the NEC P6200 and P6300 personal printers.
NECP2.PFT	This PFT supports the NEC P2200XE.

Special Considerations When Using IBM-Supplied PFTs

You should read this section if you are using any of the following IBM-supplied PFTs:

- IBM4216.PFT
- HPLASER2.PFT
- HPLASER3.PFT
- HPLASERD.PFT
- HPLASR3D.PFT
- HPDSKJT.PFT

The following restrictions apply:

- Proportional space fonts do not print correctly when text is overstruck.
- Justification is not supported when using proportional space fonts.
- Tabs are not supported when using proportional space fonts.
- The symbol set used is the Roman 8 character set.

  Note: The symbol set can be changed by changing the printer function table.

Creating a Printer Function Table

Use the following procedure to create or revise a PFT:

1. If you have a serial printer, use the DOS MODE command to direct the printer output to the serial port before you run PFTSETUP. At the *DOS* prompt, type the following commands. Press the Enter key at the end of each line.

   ```
   MODE COM 1:1200,n,8,1,p
   MODE LPT1:=COM1
   ```

   The baud rate and other parameters on the COM1 line vary depending on the type of personal printer. See your personal printer manual for exact information.

2. Change directories to the PC Support MRI directory (for example, `\QIWSFLR\MRI2924`).
3. To start the program, type `PFTSETUP` and press the Enter key.
4. Select the option to either create or revise a table.
5. A display of information appears. Read the information and press the Enter key to continue.

6. Enter a PFT file name. You can use one from the list in the previous section, "PFTs Provided by Work Station Function" on page A-2, or create a new PFT. Use the revise option to make changes to a PFT you have created.

   If you are revising an IBM-supplied PFT, make a copy of the original PFT for backup purposes.

7. When the Revise Printer Function Table menu appears, select the option you want to define.

8. The program prompts you for all necessary information about printer functions. Refer to "Defining Printer Functions" on page A-7 for a description of each printer function. Refer to "Entering Control Sequences" for information on entering control sequences in the PFT.

   Refer to your personal printer manual for the correct terminology for each function. Study the functions and the concepts described to determine the appropriate use or need for each PFT item.

9. After you finish defining functions, run the function selection tests (refer to "Function Selection Tests" on page A-19).

10. Change the printer session profile to use the PFT. Refer to the *PC Support/400 DOS Installation and Administration Guide* and to the *PC Support/400 OS/2 Installation and Administration Guide*, or to the *OS/2 System Administrator's Guide for Communications* for information on changing printer session profiles.

Entering Control Sequences

A control sequence is a sequence of hexadecimal values that instruct the personal printer to perform a specific function or print a particular output. Use the following rules when entering control sequences:

- Use two hexadecimal digits for each byte to be sent to the personal printer.
- Leave one space between each pair of hexadecimal digits.
- Use either uppercase or lowercase letters.
- Enter a maximum of 41 characters, including spaces, in the first field of a control sequence (14 bytes of hexadecimal values).
- Enter a maximum of 59 characters, including spaces, in the continuation field (20 bytes of hexadecimal values).
- Use a variable place holder (*n* or N) separated by a space, if a printing function can be changed at print time by specifying a variable to the personal printer.
- You can use more than one control in a control sequence. However, you can use only one variable *n* per sequence.
- Do not enter a variable *n* as the first byte of a control sequence.

For example, if your personal printer manual specifies Esc E for a particular function, type 1B 45 (1B is the ASCII hexadecimal value for Esc, and 45 is the value for E).

Note: If a control sequence sets a printer mode that affects the operation of other control sequences, be sure to reset this mode in the affected sequences.

Variable N

Each personal printer interprets the value for variable *n* differently. The interpretation is based on the algorithm:

(value sent to printer +/- offset) * x/y = n

where:

Maximum value determines the type of coding and number of bytes to send to the personal printer (the *value,* or variable *n*, in the algorithm). Refer to your personal printer manual for the maximum decimal value allowed for each variable *n*. For example, if your personal printer manual defines a control in which a value of zero is equal to a parameter value of X'40', and a maximum number of units supported by the control is X'0F', then the maximum value of *n* is 79. Or if your personal printer manual defines a control sequence, such as Esc US n, where n-1 is the value used by the personal printer and the value of *n* cannot exceed X'7E', then the maximum value of *n* is 126.

Offset added or subtracted by printer provides a way to enter a decimal number that the personal printer adds to or subtracts from the variable *n* when it receives the control sequence. For some personal printers, the value of *n* must be adjusted before being sent to the personal printer, because the personal printer performs an adjustment on the value. For example, if your personal printer manual specifies a chart where a value of zero equates to a parameter of X'40', then the offset is 64, which is subtracted from the variable.

Control increases (x/y inches) represent a scaling factor used to convert the value of variable *n* to inches. For some personal printers, the value of *n* must be defined in fractions of an inch. For example, if x/y = 1/72, the variable represents units of 1/72 inch.

When x/y = 1/1, the variable represents units of 1 inch, and the algorithm is simplified to this:

(value sent to the printer +/- offset) = n

If your personal printer manual defines *n* (or *n* times a fraction) as the value used by the personal printer (for example, n/72), the offset is zero. The algorithm simplifies to this:

(value sent to the printer * x/y) = n

Determining the Format of Variable N

Each variable *n* must be in a specific format for the personal printer. The PFT setup program helps you specify the format. When you press the Enter key in a menu that has just been changed and that defines a variable *n*, the PFT setup program tests variable *n* on the personal printer. You then examine printed output and respond to questions on the display. These responses determine the value of variable *n*. If necessary, you can redefine the control sequence and press the Enter key again to determine the type of variable. If none of the tests provide the correct result, work station function cannot use the function on your personal printer.

Defining Printer Functions

The Revise Printer Function Table menu lists the control sequences you can define for printer functions:

- Initialization and reset
- Vertical/horizontal spacing
- Highlighting
- Paper/page options
- Type styles/character sets
- User-defined controls

Initialization and Reset

PFTSETUP allows you to define start of document initialization and end of document reset.

Start of Document Initialization

Note: The PFTSETUP program sends the control sequence for start of document initialization to your personal printer before every function selection or variable *n* test. When creating a PFT, you should define the start of document initialization control sequence first.

Work station function uses the start of document initialization control sequence at the start of every SCS job and when the command `03 00` is received during a SCS job. Refer to "ASCII Transparent Data Printer Command" on page 5-7 for more information on ASCII transparency. You can also refer to "Optional Initialization and Resetting of Personal Printers" on page 5-8 for information about controlling the initialization and reset of printers.

You may define a control in the start of document control sequence or set a printer switch to set each of the following conditions:

Vertical Units	Units defined for the variable line spacing control.
Line Spacing	6 lines per inch.
Pitch	10 pitch.
Carriage Return	With no line feed. If a carriage return cannot be achieved without a line feed, zero index carriage return (ZICR) is not supported. In addition, line spacing may always be incorrect if the sequence carriage return line feed causes an extra line feed.
Line Feed	Perform a line feed with no carriage return.
Turn Off	Superscript, subscript, underline, overstrike, bold, proportional spaced mode (PSM), justification.
Character Set	Printer default. Set to personal computer character set 2 if the personal printer supports it.
Printing Mode	Two-directional or one-directional mode, depending on the personal printer.
Color	Black.
Line Length	Maximum supported by the personal printer.
Page Length	11 inches.

End of Document Reset

Work station function uses the control sequence for end of document reset after each SCS job. You should define a control, where applicable, in this control sequence to reset the following values:

- Vertical units
- Line spacing (in default vertical units)
- Carriage return status
- Line feed status
- Page length
- Left margin
- Printer font (pitch, type style)
- Printer character set
- Portrait/landscape status
- Color
- Printing mode
- Paper handling

You can also refer to "Optional Initialization and Resetting of Personal Printers" on page 5-8 for information about controlling the initialization and reset of printers.

Vertical/Horizontal Spacing

The vertical/horizontal spacing option allows you to define these functions:

- Vertical line spacing
- Indexing functions
- Horizontal character spacing
- Horizontal relative movement

These functions are described as follows:

Vertical Line Spacing

Vertical line spacing is a measure of the distance between printed lines.

Work station function uses the control for variable line spacing. Otherwise, the controls for 6 or 8 lines per inch are used. Variable line spacing must be defined to support SCS 3812 functions.

- Lines per Inch

  The most common vertical line spacing is 6 or 8 lines per inch. You can define one control sequence to set 6 lines per inch and one to set 8 lines per inch, or if the variable line spacing control is defined, you may leave the control sequences to set 6 lines per inch and 8 lines per inch undefined.

- Variable Line Spacing

  A VLS control sets the vertical line spacing to a variable value. Some personal printer manuals refer to VLS as line feed pitch. Check your personal printer manual, especially if a variable increase is allowed, for the function that controls vertical motion.

  If your personal printer does not support the suggested value, use the smallest increment of vertical spacing that your personal printer supports. Common values are 1/48, 1/72, 1/96, and 1/300.

  Work station function uses VLS to achieve 0.5, 1.5, and 2.5 line spacing in AS/400 documents.

Indexing Functions

Work station function supports the superscript/subscript controls. On some personal printers, these controls move the print position up or down 1/2 line spaces. Other personal printers create superscripts and subscripts by compressing the height of the characters. If your personal printer can print superscripts and subscripts in only one pitch, do not define these controls.

Work station function also supports 1/2 Index Up (reverse), 1/2 Index Down (forward), and Reverse Index to process AS/400 superscript/subscript controls.

The controls are examined in the following order when superscripting or subscripting is requested. The search stops when a defined control is found.

- 1/2 Index Up and 1/2 Index Down
- Reverse Index
- Superscript and Subscript

Horizontal Character Spacing

Use the Group and Individual Type Style section of the PFT and leave the Horizontal Character Spacing section of the PFT undefined to configure a work station function printer.

If there are no Group or Individual Type Styles defined, then work station function will use the Horizontal Character Spacing section of the PFT to support type style (font) changes.

Horizontal character spacing refers to the number of characters in a horizontal inch of printing. You can set up horizontal character spacing as either a pitch value or with horizontal motion index. The value of a pitch is determined by the width of the characters and how far apart they are spaced on the printed page. For example, 10 pitch refers to 10 characters in a horizontal inch. Work station function supports 5, 10, 12, 15, and 17.1 pitches. Work station function does not support 8.55 pitch. Proportional spaced mode (PSM) is supported with the horizontal motion index control.

If you define a horizontal motion index, try multiples of 1/60 inch. If you cannot use these multiples, use the smallest space allowed by your personal printer. Common values are 1/60 and 1/120 inch.

For each type style in a print job, work station function first checks for a defined pitch control sequence. If it is defined, the active pitch is set accordingly. Otherwise, work station function sets the defined pitch control nearest to the one received. If no pitch control sequence is defined, work station function prompts you on the printer operator panel to physically change a font at your attached personal printer.

Work station function also uses the horizontal motion index control to accomplish the word processing function of line justification. Line justification may be specified in AS/400 documents.

Horizontal Relative Movement

The horizontal relative movement (HRM) controls move the print position backward or forward relative to the current print position. Some personal printer manuals refer to HRM as horizontal cursor positioning. Find the function in your personal printer manual that moves the horizontal position of the personal printer relative to the current position.

If possible, define control sequences for forward and backward relative movement. Try multiples of 1/60 inch first. Otherwise, use the smallest increment of horizontal spacing supported by your personal printer. Common values are 1/120 and 1/240 and 1/300 inch.

Work station function uses the forward horizontal relative movement control to accomplish the word processing function of line justification. Line justification may be specified in AS/400 documents.

Work station function uses the backward horizontal relative movement control to accomplish back spacing and tab alignment in proportional spaced mode (PSM) type styles. AS/400 documents may specify PSM type styles.

Work station function may also use Backward and Forward Horizontal Relative Movement controls to accomplish boldface type. If the Begin/End Emphasis controls in the PFT have not been defined, work station function will use HRM controls to overstrike characters with a 1/120 inch horizontal offset.

Highlighting

Work station function supports the begin-quality-print/end-quality print controls in the PFT.

Work station function printers support begin-emphasis/end-emphasis controls. The AS/400 system uses Begin and End emphasis commands to print boldface type in print jobs.

If begin-emphasis/end-emphasis controls are not defined in your PFT, work station function will use Forward or Backward Horizontal Relative Movement controls to overstrike graphic characters with 1/120 inch horizontal offset.

If Forward or Backward Horizontal Relative Movement controls are not defined in your PFT, work station function will backspace and overstrike graphic characters.

Work station function also supports begin-underline/end-underline controls in the PFT. If these controls are not defined in the PFT, work station function emulates these controls using the underscore graphic and a second pass of the print line.

Paper/Page Options

The paper handling equipment configured in the printer session profile can be supported by defining paper handling and page length controls in the PFT. Work station function does not support the set left margin, carriage-return/line-feed, or multiple copies controls in the PFT. If your personal printer does not have advanced paper handling capabilities, work station function provides default values for the controls that are not specifically defined.

Paper Handling

Work station function uses the control sequences for the following:

- Bottom tray select and feed
- Top tray select and feed
- Envelope tray select and feed
- Manual feed select
- Eject sheet without paper feed, automatic cut sheet

Work station function does not support the controls for the following:

- Continuous feed select
- Ignore paper end sensor
- Enable paper end sensor
- Eject sheet without paper feed, manual cut sheet
- Collate

If Landscape Paper Handling controls are defined, the Paper Handling select control sequences should include controls to set the page orientation to portrait.

Some personal printers require two controls for the paper (or envelope) feed function: one for paper tray selection and one for loading paper from that tray. In this case, define both controls in the control sequence.

If your personal printer has an automatic sheet-feed device, define all five paper feed items if possible. If the personal printer supports a paper feed such as single-drawer automatic sheet feed, but does not define a control sequence to select or feed paper from that paper feed, define a null (X'00') for that paper feed.

The select and feed controls in the Paper Handling menu can be defined for the bottom, top, and envelope tray. This control causes a sheet of cut paper (or envelope) to be fed into the personal printer from the specified tray. When an AS/400 print job requests the bottom, top, or envelope tray, work station function checks the number of cut sheet drawers and envelope support configured in the printer session profile. If the requested paper source was configured in the printer session profile, work station function will use the corresponding control sequence in the PFT. If the requested paper source was not configured in the printer session profile, work station function will use the manual feed select control and prompt on the work station function printer operator panel to feed manually.

The manual feed select control allows the automatic sheet-feed device to accept a manually inserted sheet of paper through the manual insertion gate. The personal printer must stop printing to accept the manually fed sheet.

Work station function assumes that continuous forms feed is available.

If the eject sheet without paper feed (automatic cut sheet) control is defined, work station function assumes the paper feed controls eject the current page and feed the next page.

Work station function assumes that the manual feed select control ejects the current page and stops the personal printer for the next manually inserted page.

Run the print test for continuous feed select on a continuous forms feed. Run the print tests for bottom, top, and envelope trays on an automatic feed. Run the print test for manual paper feed insertion on a manual feed or automatic feed with a manual insertion gate.

Landscape Paper Handling

Work station function printers support the following landscape paper handling controls for printing orientation:

- Bottom tray select and feed
- Top tray select and feed
- Envelope tray select and feed
- Manual feed select

In each landscape paper handling select and feed control sequence enter the controls to set the page orientation to landscape.

Some personal printers require two controls for the paper (or envelope) feed function: one for paper tray selection and one for loading paper from that tray. If this is the case, define both controls in the control sequence.

If your personal printer has an automatic sheet-feed device, define all four landscape paper feed items if possible. For more information about landscape controls, see "User-Defined Controls 993, 994, 995, and 996" on page A-16.

Paper Positioning

Work station function printers support the following paper positioning values for automatic feed:

- Distance from top paper edge
- Distance from left paper edge

If values are defined, work station function will assume that the values specify the nonprintable area on the page of your personal printer. The top and left margins used for an AS/400 print job will include these values.

For example, if a value of 1/2 is specified for the distance from the top paper edge, and the AS/400 document sets a top margin of 1 inch, work station function will only advance your personal printer position 1/2 inch. The effect will be that the printed output will have a 1 inch top margin.

Page Length

Work station function uses the page-length controls to set the length of the page. You must specify the page-length controls with one of the options (inches or lines).

If page-length controls are not defined, or the page length exceeds the maximum values allowed for the supported page-length controls, work station function indexes the personal printer as closely as possible to the end of each page. For more information about page size controls, see "User-Defined Control 986 and 987" on page A-14.

Type Styles/Character Sets

Type style defines the appearance (font) and pitch of characters. For example, Gothic characters look different from characters printed in a commonly used type style such as Prestige Elite.** You can define default, individual, and group type styles.

For each individual and group type style definition, you can select a personal-computer character-set.

Character sets are defined groups of graphic characters. For example, a character set can include “a” through “z”, and other print characters.

You can define PC character sets 2 through 9.

If the personal printer has removable print elements, such as a print wheel, you can define a character set to match those on a print element, then associate the character set with a type style number.

You can also define a character set to include characters that can be constructed from those on the print element. For example, if the print element has a tilde, you can define the *n tilde* character as *n backspace tilde*.

Work station function does not support slot selection definition.

Default Type Style

Type style definitions allow you to specify the controls that select the type styles and character sets supported by your personal printer. Type styles are defined with beginning and ending control sequences for the PC character sets 2 through 9. The beginning control sequence selects the print element for a type style and a character set. The ending control sequence is optional.

Individual Type Style

An individual type style definition can be defined to select a type style and associated character set. Use an individual definition to define a type style that can only be selected using unique control sequences for each print element.

PFT Individual Font Type Style Definitions

A printer function table that supports your personal printer is selected when you configure the WSF Printer Emulation session. To determine what printer function table has been selected, you can use the show printer profile option on the printer operator panel.

If your personal printer supports the IBM Personal Printer Data Stream (PPDS) Set Font Global command, no individual or group type styles are defined in the printer function table. User-Defined Control 988 is defined as X'04' (IBM PPDS Level 2 or greater). WSF Printer Emulation then uses the PPDS Set Font Global command to select fonts on your personal printer.

If your personal printer does not support the PPDS Set Font Global command, the printer function table is used to map the IBM type style number into printer commands that select the font on your personal printer.

Group Type Styles

You can define a group type style to select a group of type styles and associated character sets on the personal printer. Each group type style definition is assigned a group type style identifier. There are 26 identifiers (a through z). Use a group type style definition to define a group of type styles that are selected through common control sequences.

User-Defined Control 984 and 985

Work station function printers use user-defined controls 984 and 985 to support envelope size commands. Control 985 contains the set envelope size command and control 984 contains parameters for the command sequence. The first byte of control 985 is the parameter offset into the command sequence. The second byte of control 985 is the parameter length. The third byte is the start of the command sequence. User-defined control 984 is a series of parameters for selecting envelope size. Each parameter can be one or more bytes long. All eight parameters must be the same length. The parameters are listed in order for the following envelope sizes:

Monarch
Number 9
Number 10
JIS number 13
DL
C5
B5
Other

User-Defined Control 986 and 987

Work station function printers use user-defined controls 986 and 987 to support page size commands. User-defined control 987 contains the set page size command and user-defined control 986 contains parameters for the command sequence. The first byte of control 987 is the parameter offset into the command sequence. The second byte is the parameter length. The third byte is the start of the command sequence. User-defined control 986 is a series of parameters for selecting page size. Each parameter can be one or more bytes long (all parameters must be the same length). The parameters are listed in order for the following paper sizes:

A5
half letter
B5
Executive 1
Executive 2
Executive 3
Letter
A4
Legal
Other

User-Defined Control 988

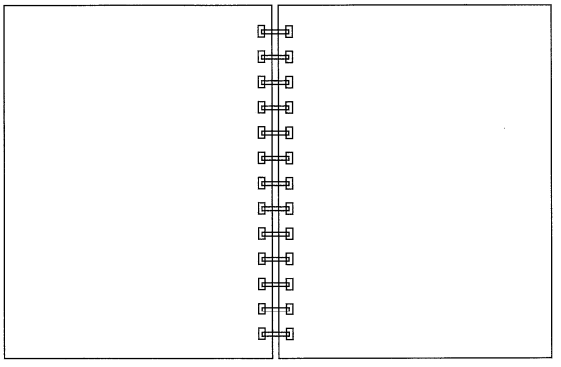
User-defined control 988 defines the printer data stream that is used for the PFT. The values currently defined for this control are:

00 NULL
01 IBM PPDS (personal printer data stream) Level 1
02 IBM 3812
03 HP PCL (printer command language) 4
04 IBM PPDS Level 2 or greater
05 Epson
06 NEC (Nippon Electronic Corporation)
07 IBM other

User-Defined Controls 989, 990, and 991

The HP LaserJet IID controls support simplex and duplex page printing (for example, printing on one or both sides of a sheet of paper, respectively). The edge of the paper to be bound may be either the long edge or the short edge.

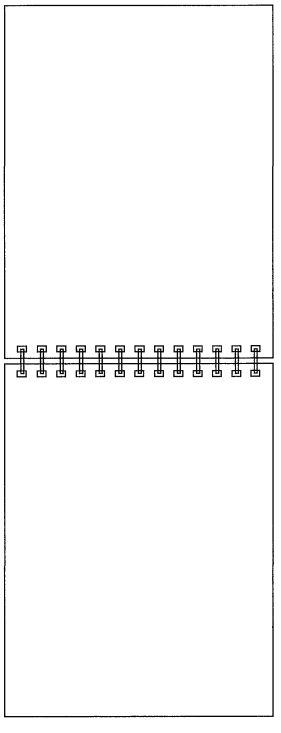
User-defined control 989 selects a duplex-long edge binding. If portrait is specified, the binding appears as shown in Figure A-1. This can also be referred to as normal duplex.



RSLF046-0

Figure A-1. Output Using the Portrait Option with User-Defined Control 989

User-defined control 990 selects a duplex-short edge binding. If portrait is specified, the binding appears as shown in Figure A-2. This can also be referred to as tumble duplex.



RSLF047-0

Figure A-2. Output Using the Portrait Option with User-Defined Control 990

User-defined control 991 selects the simplex mode of operation (one-sided print).

User-Defined Control 992

Work station function personal printers use user-defined control 992 to move the output tray. This allows different printouts or copies of the same printout to be slightly offset from each other. The printouts are then easier to separate after they have been removed from the output tray.

User-Defined Controls 993, 994, 995, and 996

Work station function printers may use user-defined controls 993, 994, 995, and 996.

Depending on your personal printer, there are two cases to consider:

Case I — If your personal printer supports normal portrait and a single landscape orientation (either left or right), then define the controls for Paper Handling and Landscape Paper Handling. See "Paper Handling" on page A-10 and "Landscape Paper Handling" on page A-12 for more information. User-defined controls 993, 994, 995, and 996 should be left undefined.

Case II — If your personal printer supports normal portrait, landscape left, portrait upside down, and landscape right orientations, then define the appropriate user-defined control to specify page orientation.

User-defined control 993 will select normal portrait orientation (0 degree clockwise rotation of text).

User-defined control 994 will select landscape left orientation (270 degree clockwise rotation of text).

User-defined control 995 will select portrait upside down orientation (180 degree clockwise rotation of text).

User-defined control 996 will select landscape right orientation (90 degree clockwise rotation of text).

User-Defined Controls 997, 998, and 999

Work station function printers use user-defined control 997, 998, and 999 to change the type style for computer output reduction (COR).

Computer output reduction consists of rotating the paper to a landscape right orientation, reducing the character size (changing the type style), changing the line density, and changing the top and left margins.

Computer output reduction can be activated when the presentation page size is larger than the paper size configured in your printer session profile. User-defined controls 997, 998, and 999 are used by the work station function to change the type style when computer output reduction is activated in 10, 12, or 15 pitch type styles, respectively.

These user-defined controls should be 5 bytes long. The first 2 bytes should specify a type style number, the next 2 bytes should specify a font width in 1440ths of an inch, and the last byte should specify a font attribute.

For example, if your personal printer supports type style 254 which is a 17.1 pitch (1440/84) type style, you could enter X'00 FE 00 54 01' for the user-defined control 997. The value X'00 FE' for the user-defined control is the hexadecimal representation for type style 254; the value X'00 54' is the hexadecimal representation for 84, which is the width of type style 254 in 1440ths of an inch; and the value X'01' is the hexadecimal value to select a font attribute of fixed-pitch font. Then, if computer output reduction mode is activated while using a 10-pitch type style, work station function uses the information you entered in user-defined control 997 to change to type style 254.

For the above example, type style 254 should be defined in the Group or Individual Type Style section of your PFT if your personal printer does not support the IBM Personal Printer data stream (PPDS).

Printing the Paragraph Symbol

The IBM-supplied EBCDIC-to-ASCII table maps the paragraph symbol EBCDIC X'B6' to the ASCII code point X'14'.

If you are creating a printer function table and your personal printer does not directly support X'14' as the paragraph symbol, you will either have to change the EBCDIC-to-ASCII table (using the CFGWSF program) or set up your PFT to map X'14' to the code point for the paragraph symbol on your personal printer.

No change is required for the IBM-supplied printer function tables. The IBM-supplied printer function tables map X'14' to X'1B 5E 14' in the Character Set section of the PFT.

Changing the PFT for Use with External Font Cards

The PC Support work station function PFTs can be changed so that external fonts, such as font cards, can be used. Use the following steps to change the PFTs.

Note: The following discussion assumes the IBM Personal Printer Data Stream (PPDS). A similar process can be determined using a different data stream, such as HP Printer Control Language (PCL).

1. Make a copy of the PFT to be changed.
2. Start PFTSETUP and when the Revise Printer Function Table menu appears, select option 5 (Typestyles/Character Sets).
3. On the Typestyles/Character Sets menu, select option 3 (Group).
4. Select the group identifier that contains the font you want to isolate. The identifiers are:

   A Lists 10 pitch fonts, including fonts 1 through 65. The initial control sequence hexadecimal string used for this group is X'14 12', which stands for "'Select 10 pitch font'."

   B Lists 12 pitch fonts, including fonts 66 through 153. The initial control sequence hexadecimal string used for this group is X'14 1B 3A', which stands for "'Select 12 pitch font'."

   C Lists 17 pitch fonts, including fonts 250 through 259. The initial control sequence hexadecimal string used for this group is X'14 12 0F', which stands for "'Select 17.1 pitch font'."

D Lists the proportional spaced fonts, including fonts 154 through 200. The initial control sequence hexadecimal string used for this group is X'1B 50 01', which stands for "'Begin Proportional Spacing Mode'."

E Lists five pitch fonts, including 240 through 249. The initial control sequence hexadecimal string for this group is X'14 12 1B 57 01', which stands for "'Begin Continuous Double-wide Printing'."

Group identifiers F through Z are not used.

5. Change the font range. For example, if you want to use the font 86 available on your font card instead of the default font 86, change the range of fonts in Group B from 66 through 86 to 66 through 85.
6. Press Enter to save your changes and then press Esc to return to the Typestyles/Character Sets menu.
7. Select option 2 (Individual). A blank type style selection screen appears.

   Type 86 in the *PC Typestyle Number* field.

   Type 2 in the *PC Character Set* field.

   Leave the *Ending Control Sequence* field blank.

   Use the following steps to determine a value for the *Initial Control Sequence* field:

   a. The "ESC [I Select Font" sequence is used for this control sequence. Refer to the *IBM 5204 Quickwriter Printer Technical Reference* or the *IBM 5202 Quietwriter III Printer Technical Reference* for decimal to hexadecimal conversion values.

   b. The Select Font sequence is as follows:

      ```
      ESC [ I Ln Hn Hfid Lfid Hfwd Lfwd fa
      ```

      where:

      Ln Number of hexadecimal parameters following the Hn parameter (normally X'05').

      Hn X'00'.

      Hfid X'00'.

      Lfid Font ID in hexadecimal. For example, if the font ID is 86, then the Lfid is X'56'.

      Hfwd X'00'.

      Lfwd Value determined by the pitch. See the pitch tables in the *IBM 5204 Quickwriter Printer Guide to Operations* or the *IBM 5202 Quietwriter III Printer Guide to Operations* for more information.

      fa Spacing attribute (fixed or proportional) of the font to be selected. Please refer to your printer's operations guide for more information.

Function Selection Tests

After you create or change a PFT, run the function selection tests to verify the print functions you defined in the table:

- Superscript and subscript
- Underline
- Emphasis (bold)
- Form feed
- Backspace
- Pitches

You do not need to run the tests for the functions not supported by work station function (cursor draw).

Use the following procedure to run the tests:

1. If you have a serial printer, use the DOS MODE command to direct the printer output to the serial port before you run PFTSETUP. At the *DOS* prompt, type the following commands. Press the Enter key at the end of each line.

   ```
   MODE COM 1:1200,n,8,1,p
   MODE LPT1:=COM1
   ```

   The baud rate and other parameters on the COM1 line vary depending on the type of personal printer. See your personal printer manual for exact information.

2. Change directories to the PC Support MRI directory (for example, `\QIWSFLR\MRI2924`).
3. To start the program, type `PFTSETUP` and press the Enter key.
4. Select option 2 (Revise table) and press the Enter key. An information display appears. Press the Enter key. Enter the requested information on the display and press the Enter key. The Revise Printer Function Table display appears.
5. Select option 7 (Function selection tests) and press the Enter key. The Function Selection Tests Options display appears.
6. Move the cursor to a test on the table and press the Enter key. The personal printer will print instructions on running the test.
7. The test will print samples. Answer the questions on the display according to the personal printer output. When more than one sample is printed, choose the first one that provides the appearance you want.
8. If none of the samples appears the way you wanted, choose the sample that is closest. Or, you can redefine the control sequence for the function.
9. If you decide to redefine any controls, run the tests again.

   Note: If a control that is not defined is used in a test, the test is printed without that control.

10. Run the tests to ensure that you have defined the functions correctly. If you do not, the PFT setup program prompts you to run them before you exit the program.

Appendix B. Hypertext File Format

A hypertext file consists of two types of data. The first record of each file is a Table of Offsets, and the rest of the file is the hypertext data. The Table of Offsets is a group of 12-byte entries. The first 6 bytes of each entry are the display name and the second 6 bytes are the offset of the display from the start of the file. The structure of each display in the data section of the file is:

- The first record is an identifier record, *&xxxxhhhh, in which xxxx is a 4-character component identification and hhhhh is the display name.
- The second record is the title line of the display. It must be less than 74 characters long.
- The rest of the display contains the formatted text to be displayed. It can be any number of lines, and contains the hypertext reference phrases.

A hypertext file can be up to 999,999 bytes in size, including the Table of Offsets. The number of hypertext files that can be linked is limited by file space. The number of displays that can be displayed is limited by the amount of available memory. Memory is freed up each time a display is exited.

Figure B-1 shows a sample hypertext file.

```
The first record is the Table of Offsets
|-panel 1--||-panel 2--||last panel||-end file-|
 name offset name offset name offset name offset
|          ||          ||          ||          |
HELPID000050HELPXX000703999999000787zzzzzz000801
*&ABCDHELPID
Title of help panel HELPID
Text of help panel formatted as it will appear on the display.
•    <<LINKIDSame file>>
        This is a hypertext link in the same file, to a panel named
        LINKID.
•    <<(NEWFILE)LINKID Next file>>
        This is a hypertext link to a panel named LINKID in a file named
        NEWFILE.HLP.
•    <<(#FIG.EXE#GRAPH#)BADLNK Figure>>
        This is a hypertext link to a program called FIG.EXE with
        parameters GRAPH.  When the program FIG is ended, control
        returns here.  If there is a non-zero return code from FIG,
        then the hypertext panel BADLNK is displayed.
*&ABCDHELPXX
Title of help panel HELPXX
           |
           |
           v
*&XXXX999999
```

Figure B-1. Hypertext File Example

Appendix C. PC Support/400 Interrupts

PC Support/400 functions can use the interrupts in the personal computer to perform their functions or to communicate with other functions of PC Support/400 or user-defined applications.

The table in Figure C-1 lists the interrupts used by PC Support/400. If you experience problems when using an adapter or software supplied by other equipment manufacturers, you can use this table to see if any of the interrupts used by PC Support/400 conflict with the hardware or software you have installed on your personal computer. Different functions of PC Support/400 use different interrupt vectors, so review the table for all PC Support/400 functions that you use.

Note: All interrupt vectors listed in the table are hexadecimal and not decimal. If specified, interrupt request (IRQ) is the hardware interrupt level used.

Figure C-1 (Page 1 of 3). Interrupts Used by PC Support/400

Function	Interrupt Vectors	Notes
Expanded Memory Interface Manager (EIMPCS.SYS)	69	This interrupt can be configured in the range from 60 to 66, 68, and 69 using the /I= parameter of EIMPCS.SYS in the CONFIG.SYS file. This is the shared interrupt used by the PC Support/400 functions. **Note:** Do not use the value specified for INTL in the CONFIG.PCS file, or 68 (the default value for INTL if not specified in the CONFIG.PCS file).
Router - Async (ASYNRTR.OVL)	8 (IRQ0)	
	B (IRQ3) or C (IRQ4)	The interrupt is based on the serial port in use. If using COM1, interrupt C is used. If using COM2, COM3, or COM4, interrupt B is used.
	68	The router uses this interrupt to allow other PC Support/400 functions to communicate with it. This interrupt can be configured in the range from 60 to 68 using the INTL identifier in the CONFIG.PCS file. **Note:** Do not use interrupt 67 if EMS is in use, because EMS uses interrupt 67.
Router - SDLC (SDLCRTR.OVL)	B (IRQ3) C (IRQ4)	
	68	The router uses this interrupt to allow other PC Support/400 functions to communicate with it. This interrupt can be configured in the range from 60 to 68 using the INTL identifier in the CONFIG.PCS file. **Note:** Do not use interrupt 67 if EMS is in use, because EMS uses interrupt 67.
Router - TDLC (E5250RTR.OVL)	8 (IRQ0)	E5250AH
	9 (IRQ1)	This interrupt is used by E5250AH.
	2	This interrupt is used by WSEAH.

Figure C-1 (Page 2 of 3). Interrupts Used by PC Support/400

Function	Interrupt Vectors	Notes
	D (IRQ5)	The TDLC router can use any interrupt in the range A-F (IRQ2-IRQ7), but D (IRQ5) is the default. This interrupt can be configured with the reference diskette on microchannel architecture (MCA) machines and on the command line of E5250AH for non-MCA machines (/L# where # is the IRQ interrupt level 2-7).
	13	E5250AH, Router
	15	This interrupt is used by WSEAH.
	68	The router uses this interrupt to allow other PC Support/400 functions to communicate with it. This interrupt can be configured in the range from 60 to 68 using the INTL identifier in the CONFIG.PCS file. **Note:** Do not use interrupt 67 if EMS is in use, because EMS uses interrupt 67.
Router - Token-Ring (ITRNRTR.OVL)	A (IRQ2) or B (IRQ3)	The token-ring router does not use the interrupts, but the LAN Support program uses either 2 or 3 (it can be configured).
	A (IRQ2), B (IRQ3), or F (IRQ7)	The token-ring adapter can use several interrupts. On a PS/2 computer you can use your reference diskette to look at all of the interrupt possibilities. On a non-MCA machine you must review the token-ring adapter documentation to determine which interrupt is being used. The interrupt used is based on the switch settings on the adapter.
	5C	This interrupt is used by the IBM LAN Support Program to process application program interface (API) requests.
	68	The router uses this interrupt to allow other PC Support/400 functions to communicate with it. You can configure this interrupt in the range from 60 to 68 using the INTL identifier in the CONFIG.PCS file. **Note:** Do not use interrupt 67 if EMS is in use, because EMS uses interrupt 67.
Message Function (STARTMSG.EXE)	8 (IRQ0), 16, 21,	
	69	This is the PC Support/400 shared interrupt. **Note:** Do not use the value specified for INTL in the CONFIG.PCS file, or 68 (the default value for INTL if not specified in the CONFIG.PCS file).
Organizer (PCO.EXE)	23 and 24	
Network Connector (SFNC.OVL)	20, 21, and 22	
Device Driver Extensions (ECYDDX.SYS)	20, 21, 27, 2F, 30 and 31	
Transfer Function (STF.EXE)	69	This is the PC Support/400 shared interrupt. **Note:** Do not use the value specified for INTL in the CONFIG.PCS file, or 68 (the default value for INTL if not specified in the CONFIG.PCS file).
Virtual Print (VPRT.EXE)	17, 8 (IRQ0)	

Figure C-1 (Page 3 of 3). Interrupts Used by PC Support/400

Function	Interrupt Vectors	Notes
	69	This is the PC Support/400 shared interrupt. **Note:** Do not use the value specified for INTL in the CONFIG.PCS file, or 68 (the default value for INTL if not specified in the CONFIG.PCS file).
Work Station Function (WSF.EXE)	8 (IRQ0), 9 (IRQ1), D (IRQ5),	
	59	Used by WSF Virtual Device Interface (VDI) routines.
	7A	This interrupt is used by the WSF application program interface (API), but can be configured to any value desired by using parameter */I:nn* when running the WSFAPI program. *nn* is any hexadecimal value for an interrupt that is not being used by another program.
	7F	This interrupt is used by HDILOAD on an 8514 display when using WSF in 132-column mode.
Work Station Function Printer Support (WFPRINT.EXE)	5 14 17 21 69	Print Screen interrupt Serial port interrupt Parallel port interrupt DOS function interrupt PC Support/400 shared interrupt
5250 Session Manager (SM5250.EXE)	5, 8 (IRQ0) 10, 16 and 21	
	7F	This interrupt is used by HDILOAD on an 8514 display when using WSF in 132-column mode.

Bibliography

The following AS/400 manuals contain information you may need. The manuals are listed with their full title and order number. When these manuals are referred to in this manual, the short title listed is used.

- *DOS 5.00 Command Reference* contains information about DOS commands that exclusively use file management function calls to perform file-oriented functions.
- *IBM Disk Operating System Technical Reference* contains information about using the DOS operating system.
- *IBM Local Area Network Technical Reference,* SC30-3383, contains information about using IBM local area network adapters and supporting software in IBM Personal Computers, IBM Personal System/55 computers, and IBM Personal System/2 computers.
- *IBM 5204 Quickwriter Printer Technical Reference*, S544-4208, contains information about using the 5204 Quickwriter printer.
- *IBM 5202 Quietwriter III Printer Guide to Operations*, S544-4034, contains information about using the 5202 Quietwriter III printer.
- *IBM 5250 Information Display System Functions Reference* contains information on the orders used by the AS/400 system to communicate with graphics work stations.
- *IBM Operating System/2 Extended Edition Version 1.3 Getting Started* contains an introduction to the OS/2 program and provides the steps and basic information needed to install it, as well as how to use the online overview.
- *IBM Operating System/2 Extended Edition Version 1.3 User's Guide* provides information for users to accomplish the basic tasks provided by the OS/2 program, including the base operating system, Communications Manager, LAN Requester, and Database Manager.
- *IBM OS/2 Extended Edition Version 1.3 System Administrator*, 90X7908, provides information about the hardware and software supported by Communications Manager, along with memory and disk storage requirements.
- *OS/2 Extended Edition 1.3: APPC Programmer's Reference* contains information about application program stack size requirements.
- *OS/2 2.0 Programming Guide* Volumes 1, 2 and 3 (10G6261, 10G6494, and 10G6495) contain information about how to use OS/2 functions, and how the functions relate to one another.
- *OS/2 Standard Edition Version 1.2: Technical Reference* (6280212) describes the OS/2 Standard Edition application program interface (API). It also contains a description of the I/O subsystems and the device driver interfaces.
- *PC Support/400: Application Program Interface Reference*, SC41-8254, provides application programmers with information to write application programs for programmable work stations that use PC Support. This manual is intended for all versions of PC Support. **Short title:** *PC Support/400 API Reference*.
- *PC Support/400: DOS Installation and Administration Guide*, SC41-0006, provides PC Support or AS/400 administrators or supervisors with information for planning and installing PC Support, and configuring and diagnosing the problems for individual PC Support users. This manual is intended for users with personal computers using the DOS operating system. **Short title:** *PC Support/400 DOS Installation and Administration Guide*.
- *PC Support/400: DOS Installation and Administration Guide (PS/55)*, SC41-0008, provides PC Support or AS/400 administrators or supervisors with information for planning and installing PC Support on a PS/55 personal computer, and configuring and diagnosing the problems for individual PC Support users. This manual is intended for users with PS/55 personal computers using the DOS operating system. **Short title:** *PC Support/400 DOS Installation and Administration Guide (PS/55)*.
- *PC Support/400: DOS User's Guide*, SC41-8199, provides all PC Support users with concepts and examples of how to use the PC Support functions with the DOS operating system. It assumes that the PC Support product is already installed and set up. This manual is intended for users with personal computers using the DOS operating system. **Short title:** *PC Support/400 User's Guide for DOS*.
- *PC Support/400: DOS User's Guide (PS/55)*, SC41-2414, provides all PC Support users with concepts and examples of how to use the PC Support functions on a PS/55 personal computer using the DOS operating system. It assumes that the PC Support product is already installed and set up. This manual is intended for users with PS/55 personal computers using the DOS operating system. **Short title:** *PC Support/400 User's Guide for DOS (PS/55)*.
- *PC Support/400: OS/2 Installation and Administration Guide*, SC41-0007, provides the PC Support or AS/400 administrator or supervisor with information for planning and installing PC Support, and

configuring and diagnosing problems for individual PC Support users. This manual is intended for users with personal computers using the OS/2 operating system. **Short title:** *PC Support/400 OS/2 Installation and Administration Guide*.

- *PC Support/400: OS/2 Installation and Administration Guide (PS/55)*, SC41-0009, provides the PC Support or AS/400 administrator or supervisor with information for planning and installing PC Support on a PS/55 personal computer, and configuring and diagnosing problems for individual PC Support users. This manual is intended for users with PS/55 personal computers using the OS/2 operating system. **Short title:** *PC Support/400 OS/2 Installation and Administration Guide (PS/55)*.
- *PC Support/400: OS/2 User's Guide*, SC41-8200, provides users with personal computers attached to an AS/400 system with concepts and examples of how to use the PC Support functions with the OS/2 operating system. It assumes that the PC Support product is already installed and set up. This manual is intended for users with personal computers using the OS/2 operating system. **Short title:** *PC Support/400 User's Guide for OS/2*.
- *PC Support/400: OS/2 User's Guide (PS/55)*, SC41-2415, provides users with PS/55 personal computers attached to an AS/400 system with concepts and examples of how to use the PC Support functions with the OS/2 operating system. It assumes that the PC Support product is already installed and set up. This manual is intended for users with PS/55 personal computers using the OS/2 operating system. **Short title:** *PC Support/400 User's Guide for OS/2 (PS/55)*.
- *Personal System/2 Hardware Interface Technical Reference—Common Interfaces*, S84F-9809, describes the devices and interfaces that are common to PS/2 systems. It includes the technical information describing devices, such as serial port and parallel port controllers, and general information, such as the microprocessor instruction set and the characters associated with each keystroke.
- *Programming: Control Language Reference*, SC41-0030, provides a description of all the AS/400 control language and all of its OS/400 commands. This manual is intended primarily for the application programmer and system programmer. **Short title:** *CL Reference*.
- *Publications Guide*, GC41-9678, identifies and describes all of the information in the AS/400 library. It also provides information needed to verify that the appropriate manuals were shipped with the product and helps to determine what additional manuals to order. This manual also provides guidance on how, when, and where to use the online and printed information.

Index

Numerics

A

B

C

D

E

F

G

H

I

Q

R

S

T

U

V

W

X

Z

Customer Satisfaction Feedback

Application System/400
PC Support/400:
DOS and OS/2 Technical Reference
Version 2

Publication No. SC41-8091-02

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

_____

_____

_____

_____

_____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Customer Satisfaction Feedback
SC41-8091-02

IBM®

Cut c
Alon

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

SC41-8091-02

Cut c
Alon

Program Number: 5738-PC1

Printed in Denmark by
Aalborg Stiftsbogtrykkeri A/S

SC41-8091-02