Personal Communications Version 5.0
for Windows® 95, Windows 98,
Windows NT®, and Windows 2000

Reference Volume I

Personal Communications Version 5.0
for Windows® 95, Windows 98,
Windows NT®, and Windows 2000

Reference Volume I

Note

Before using this information and the product it supports, read the information in"Appendix. Notices" on page 263.

Contents

© Copyright IBM Corp. 1989, 2000   **iii**

Figures

Tables

About This Book

Beginning with Version 4.3, the ® Personal Communications for Windows® 95, Windows 98, Windows NT®, and Windows 2000 Reference book was split into two volumes. These volumes provide information for using IBM® Personal Communications AS/400® for Windows 95, Windows 98, Windows NT, and Windows 2000 (hereafter called PC400) and IBM Personal Communications for Windows 95, Windows 98, Windows NT, and Windows 2000. In this book, *Windows* refers to Windows 95, Windows 98, Windows NT, and Windows 2000. When information is specific to only Windows 95, Windows 98, Windows NT, or Windows 2000, this will be indicated in the text. *PC/3270* refers to the 3270 portion of the combined package. Throughout this book, *workstation* refers to all supported personal computers. When only one model or architecture of the personal computer is referred to, only that type is specified.

Who Should Read This Book

This book is intended for the person who installs and uses Personal Communications on a workstation.

How to Use This Book

This book contains reference information that you might need to refer to when installing or operating Personal Communications.

Personal Communications is designed to use various communication adapters and to work with other workstation and host system software. Refer to the appropriate documentation for the products you use.

Highlighting Conventions

This book uses the following highlighting conventions:

UPPERCASE
: Uppercase type indicates commands and program keywords. You can enter these values in uppercase or lowercase.

Bold
: Bold type indicates the names of window controls, such as lists, check boxes, entry fields, menu choices, and push buttons.

Italics
: Italics indicate:
  - Special emphasis in text or a reference citation.
  - Variables that you supply a value for.
  - Default values.

Example type
: Example type indicates information that the user is instructed to type at a command prompt or in a window.

Command Syntax Symbols

Parentheses, brackets, ellipses, and slashes have the following meanings or uses:

( )
: Parentheses enclose operands that govern the action of certain command options.

[ ]	Brackets indicate an optional command argument. If you do not use the optional item, the program selects a default.
...	Ellipsis after an argument indicates that you can repeat the preceding item any number of times.
/	For 3270, a slash must precede the Time Sharing Option Extensions (TSO/E) password. A slash must also precede parameters of DOS commands entered from the command line. For 5250, a slash must precede parameters of IBM DOS commands entered from the command line.
\	A backslash is included as part of any directory name. An initial backslash indicates the first-level directory, and an additional backslash is inserted in the directory name to indicate another level.

All directives, operands, and other syntax are typed in uppercase or lowercase, unless otherwise indicated.

Part 1. General Information

Chapter 1. Personal Communications Highlights

IBM Personal Communications Version 5.0 provides two different CD-ROM package options. The packages are:

- A CD-ROM (Version 5.0) containing *IBM Personal Communications for Windows 95, Windows 98, Windows NT, and Windows 2000*, which provides 5250, 3270, and VT emulation and connections to System/390® (S/390®), Application System/400 (AS/400), and ASCII host systems.
- A CD-ROM (Version 5.0) containing *IBM Personal Communications AS/400 for Windows 95, Windows 98, Windows NT, and Windows 2000*, which provides 5250 and VT emulation and connections to (AS/400) and ASCII host systems.

The information in this book applies to both packages, unless indicated. Each CD-ROM package supports only a single language with a single level of data encryption.

Personal Communications brings the power of personal networking to your workstation by providing a variety of connectivity options supporting local area network (LAN) and wide area network (WAN) environments. Whether you need host terminal emulation, client/server applications, or connectivity, Personal Communications offers a robust set of communications, networking, and administrative features.

Personal Communications is a full-function emulator package with an easy-to-use graphical interface, which includes many useful features such as file transfer and dynamic configuration, and emulator APIs including the IBM Host Access Class Library.

Personal Communications also provides a variety of SNA-based client application programming interfaces (APIs). You can create applications that use the peer-to-peer client APIs, which are based on LU 6.2 (and provided by Personal Communications). Using these APIs you can simultaneously access and process information on peer workstations.

With Personal Communications, you can participate in Advanced-Peer-to-Peer Networks (APPN) as an end node, and use the advanced network features, high-performance routing (HPR), and dependent LU requester (DLUR).

Personal Communications provides the following functions:

- **System/390 Connections**
  - **LAN**   LAN via IEEE 802.2
    3270 NWSAA or Communications Server for Windows NT
    Telnet3270
    3270 via AS/400
    APPC 3270 via LAN
    Microsoft SNA client over FMI
    Dependent Logical Unit Requester (DLUR)
    VT-over-Telnet (TCP/IP)
    3174 Peer Communication
  - **COAX** SNA Distributed Function Terminal
    Non-SNA Distributed Function Terminal

Highlights

COM port

Telnet 3270
SNA-over-Async
IBM Global Network (not in Japan)
Home3270
IBM Global Network - SNA-over-Async
Dependent Logical Unit Requester (DLUR) via

SNA-over-Async

Hayes AutoSync
APPC 3270 via SNA-over-Async
APPC 3270 via Hayes AutoSync
VT-over-Async
VT-over-Telnet (TCP/IP)
X.25 Hayes AutoSync
X.25 DLUR via Hayes AutoSync
X.25 APPC 3270 via Hayes AutoSync

5250 3270 via AS/400
Dependent Logical Unit Requester (DLUR)

SDLC Synchronous Data Link Control
3270 via AS/400 (passthru)
APPC 3270 via SDLC
Dependent Logical Unit Requester (DLUR)

SNA/IP

LU 0, 1, 2, 3
APPC 3270
LU 0, 1, 2, 3 via DLUR

IBM ISA/MCA WAC

Synchronous Data Link Control
3270 via AS/400 (passthru)
APPC 3270 via SDLC
Dependent Logical Unit Requester (DLUR)
X.25 Qualified Logical Link Control
X.25 3270 via AS/400
APPC 3270 via X.25
X.25 DLUR

API Client

NetWare for SAA Client
Communications Server Client
Communications Server

IBM-EEDLC

LU (0, 1, 2, 3) via DLUR
APPC 3270

OEM LU (0, 1, 2, 3)
3270 via AS/400 (passthru)
APPC 3270
Dependent LU Requester (DLUR)

- **AS/400 Connections**

LAN LAN via IEEE 802.2
Telnet5250 over TCP/IP
VT over Telnet
Telnet5250 over IPX/SPX

5250 Twinaxial Data Link Control (APPC)
Twinaxial Data Link Control (Console)

COM port

SNA-over-Async

        Hayes AutoSync
        SNA-over-Async (Console)
        VT over Async
        VT over Telnet
        Telnet 5250
        X.25 Hayes AutoSync

SDLC Synchronous Data Link Control

SNA/IP
        5250

IBM ISA/MCA WAC
        Synchronous Data Link Control (WAC)
        X.25 Qualified Logical Link Control

IBM-EEDLC
        5250

OEM APPC 5250

- **ASCII Emulator Connections**
  LAN VT over Telnet
  COM port
      VT-over-Async
      VT over Telnet

- **S/3X Emulator Connections**
  5250 Twinaxial Data Link Control (Console)

- **Client/Server Connections**
  LAN LAN via IEEE 802.2
  Twinaxial
      Twinaxial Data Link Control (APPC)
  COM port
      SNA-over-Async
      Hayes AutoSync
      X.25
  SDLC Synchronous Data Link Control
  IBM WAC
      Synchronous Data Link Control
      X.25
  AnyNet SNA over TCP/IP
      APPC
  Enterprise Extender
      HPR over IP

- **Configuration of SNA Node (APPN) Sessions**
  – Emulator
  – Client/server applications

- **SNA Node Operations**
  – Starting and stopping resources
  – Deleting resources
  – Displaying resource information
  – Changing session limits
  – Initiation of path switches

- **Log Viewer**
  – View Message Log, Trace Log, and Merged Log files
  – Summary and Detail views
  – Set default Message Log size and location
  – Filter and search Log files
  – Message Log entries Help

- **Trace Capability**

- 3270/5250 emulator data
- APPN and APPC API data
- Connectivity data, such as LAN or SDLC
- User services data, such as node initialization

- **APPC Applets**
  - Display SNA sense data (GETSENSE)
  - Transfer files (AFTP and AFTPD)
  - Check connection (APING)

- **Sample Programs**
  - Located in \Personal Communications\samples subdirectory

- **Installation and Configuration**
  - Partial installation option
  - Program sharing on a network server
  - Automatic detection of installed communication adapters
  - Dynamic change of communication configurations
  - Automatic Dial Facility (Async (IGN), SDLC, Home3270, SNA-A)
  - Silent Installation
  - ASCII SNA-node configuration
  - Verification of ASCII configuration

- **OEM Adaptor Cards**

  An open API enables vendors and other equipment manufacturers (OEMs) to provide adapter cards that allow for additional connectivity options, for example, additional X.25, ISDN, SDLC or twinax support.

- **Migration**
  - Migration from Personal Communications for Windows V3.0 or later

- **Host Session Function**
  - Up to 26 sessions
  - Variable screen size and automatic font scaling
  - Function settings (of the host code page, for example) for each session

- **Host Graphics Support**
  - Built-in vector graphics support for GDDM and other graphics applications

- **File Transfer Function**
  - Easy operation through graphical user interface (GUI) windows
  - Batch transfer of multiple files
  - Concurrent file transfer through multiple sessions
  - Background file transfer
  - File transfer invocation by macro
  - OfficeVision®/MVS® Import/Export functions
  - VT File Transfer (XModem and YModem)

- **Edit (Cut and Paste) Function**

  You can use the clipboard to cut, copy, and paste a selected area. In addition, you can paste data in other applications, such as spreadsheet programs, that support the PasteLink function.
  - Support of spreadsheet data format (Sylk, Biff3, Wk3 formats)
  - Copy Append
  - Paste Next
  - Paste to Trim Rectangle
  - Paste Stop at Protected Line

- **Graphical User Interface (GUI)**
  - Customizable 3D iconic tool bar
  - 3D-button hotspots
  - Pop-up keypad
  - Macro function, including record and play

– VBScripts, including record and play
– Keyboard-function setup and remapping
– Mouse-button-function setup and remapping
– Display setup (cursor type, graphics, sound, colors, for example)
– Automatic font size adjustment or fixed font size
– Window-appearance setup
– Menu-bar customization
– 3270 Light Pen emulation by using a mouse
– Status bar with history
– Page setup (Text and Graphics)
– Revised Configuration Dialog
– Online help

- **Print Function**
  – Printer session (for PC/3270: SCS, LU 3, or non-SNA)
  – Graphics local print
  – Printing with the Windows NT printer drivers
  – Print function by printer definition table (PDT)
  – Multiple host-print functions in multiple sessions
  – Print-job control by SNA bracket timeout
  – PDF-to-PDT conversion tool
  – PC400 print function by OS/400® Host Print Transform (HPT)
  – PC400 printing supported by the AS/400 Advanced Print Support Utility
  – ZipPrint

- **Programming Interfaces**
  – 16/32-bit Emulator High-Level Language Application Programming Interface (EHLLAPI)
  – 16/32-bit Dynamic Data Exchange (DDE)
  – 32-bit Node Operations Facility (NOF)
  – 16/32-bit Personal Communications API (PCSAPI)
  – 32-bit Advanced Program-to-Program Communication (APPC)
  – 32-bit Common Programming Interface for Communications (CPI-C)
  – 32-bit Automation Object API
  – 32-bit ActiveX/OLE 2.0
  – Host Access Beans for Java™
  – ActiveX Controls

- **PC400 Client Function**
  – Data transfer
  – PC Organizer
  – Text Assist
  – Enhanced Programmable Terminal User Interface (ENPTUI)
  – Network Drives (Shared Folders)

Highlights

Chapter 2. Where to Find More Information

This chapter discusses getting help when you are installing, configuring, or using Personal Communications.

Online Help

This help facility describes how to install, configure, and use Personal Communications. Online help is very extensive and includes information about every aspect of configuring and using Personal Communications.

Use help to obtain the following information:
- Menu choices
- Operation procedures
- Operations in windows
- Meanings of the terms displayed in windows
- Causes of errors and the corresponding actions to take
- Mouse-based operations
- Operation without a mouse
- Detailed explanations of specific terms
- Further technical information about Personal Communications
- Detailed explanations of operator information area (OIA) messages

To display online help, select choices from the Help menu or press **F1**.

You can use Personal Communications online help just as you use the online help for Windows.

Messages and Alerts

Online messages are displayed by Personal Communications, but a message does not always mean an error occurred. For example, a message might tell you that an operation is in progress or has been completed. A message can also prompt you to wait for the completion of an operation.

Messages That Appear in Pop-Up Windows

During your use of Personal Communications, there may be messages that appear in pop-up windows—not as a direct result of your actions; these can occur for a number of reasons, outlined in the following sections.

System-Fault Messages
For Windows NT, if a message does appear in a pop-up window, you can paste its contents into the Windows NT clipboard. To do this:

1. Click **Details** on the pop-up window.
2. Mark the text that you want to copy.
3. Click the marked text with the right mouse button and then click **Copy**.
4. Start an editor, such as Notepad, and click **Paste** from the Edit menu.
5. Save the file in case an IBM Service Representative needs this information to diagnose your problem.

Security-Related Messages

Personal Communications optionally utilizes Secure Sockets Layer (SSL) to establish sessions with servers; this may require input from you (for example, a password). Refer to *IBM Personal Communications Reference Volume II* for details.

System-Policy-Related Messages

Your Personal Communications workstation configuration can be controlled centrally using facilities for managing system policies. Refer to *IBM Personal Communications Reference Volume II* for details.

OIA Messages

Personal Communications displays messages in the operator information area (OIA) or in a pop-up window. Messages from Personal Communications are displayed in the message window; messages from the host system regarding the condition of the session are displayed in the OIA of the session window.

The OIA is the bottom line of the session window. An OIA message indicates the status of Personal Communications as well as information about the workstation, host system, and attachment method.

All of the OIA indicators, reminders, and messages are described in the online help. To view this information:

1. Select **Index** from the Help menu.
2. Select **The operator information area messages**.

   To look up a specific OIA message, select **Search**. When the Search window appears, type the letters that appear in the OIA. For example, MACH or PROG. If a lightning bolt appears, type COMM.
3. Double-click the index entry that matches your search.
4. Scroll through the window until you find the number that appears in your OIA.

Alerts

Alerts may be generated which correspond to specific Personal Communications messages. See "Chapter 8. Alerts" on page 55 for more information.

Personal Communications Library

The Personal Communications library includes the following publications:
- *IBM Personal Communications Version 5.0 Quick Beginnings* , GC31-8679-01
- *IBM Personal Communications Version 5.0 Access Feature*, SC31-8684-01
- *IBM Personal Communications Version 5.0 Reference Volume I*, SC31-8680-01
- *IBM Personal Communications Version 5.0 Reference Volume II*, SC31-8682-01
- *IBM Personal Communications Version 5.0 Emulator Programming*, SC31-8478-03
- *IBM Personal Communications Version 5.0 Client/Server Communications Programming*, SC31-8479-03
- *IBM Personal Communications Version 5.0 System Management Programming*, SC31-8480-03
- *IBM Personal Communications Version 5.0 CM Mouse Support User's Guide and Reference*
- *IBM Personal Communications Version 5.0 Host Access Class Library*, SC31-8685-01
- *IBM Personal Communications Version 5.0 Configuration File Reference*, SC31-8655-02

These books are also available (except in DBCS versions) in .PDF format and are optionally installed or viewed from the CD-ROM.

In addition to the printed books, there are HTML documents provided with Personal Communications:

Host Access Class Library for Java
> This HTML document describes how to write an ActiveX/OLE 2.0–compliant application to use Personal Communications as an embedded object.

Host Access Beans for Java
> This HTML document describes Personal Communications emulator functions delivered as a set of Java™ Beans.

Open Host Interface Objects for Java
> This HTML document describes how to write an OHIO-compliant application to use Personal Communications as an embedded object.

Finding More Information

Chapter 3. Problem Analysis

This chapter describes the information that will help you analyze problems with Personal Communications, and ways to report a problem to IBM. For detailed information about contacting IBM, refer to *Personal Communications Version 5.0 Quick Beginnings*.

For information about Personal Communications and support, refer to the following Web sites:

- The Personal Communications home page provides access to general product information, and download services. To view this page, use an Internet browser and go to the following Internet address or Universal Resource Locater (URL):

  `http://www.ibm.com/software/network/pcomm`

- The Personal Communications support page provides links to code fixes, tips, newsgroups, support options, and services. To view this page or to submit a software defect report, use an Internet browser and go to the following Internet address:

  `http://www.ibm.com/software/network/pcomm/support`

Personal Communications provides several utilities to help you with problem analysis. They can be invoked by selecting their icons from the **Programs>IBM Personal Communications>Administrative and PD Aids** subfolder on the Windows **Start** menu.

The following sections describe these utilities and how to use them.

Log Viewer

The Personal Communications log viewer utility enables you to view, merge, sort, search, and filter information contained in message and trace logs. Use the log viewer during problem analysis to work with message and trace log entries. The default name of the message log output file is *PCSMSG.MLG*; its file extension must be *.MLG*. The file extension for trace logs must be *.TLG*.

To view message or trace logs:

1. From the Administrative and PD Aids subfolder, select **Log Viewer**; or, from an active session, select **Launch > Log Viewer** from the **Actions** menu.
2. From the list of logged messages, double-click a message to display the message text.

For more information about log viewer functions, see "Chapter 7. Log Viewer Functions" on page 53.

Trace Facility

The Personal Communications trace facility enables you to log trace information for certain Personal Communications functions.

To start a trace, perform the following steps:

Problem Analysis

1. From the Administrative and PD Aids folder, click **Trace Facility**; or, from an active session, select **Launch > Trace Facility** from the **Actions** menu. The trace status on the title bar displays the current state:

   Active Trace data is being collected by the trace facility.

   Inactive
   No trace data is being collected.

2. From the main dialog box, click **Set Up** to set the desired trace system parameters.

3. Click **OK** to return to the main trace dialog box.

4. From the main trace dialog box, select the type of data you want to trace from the Function Name, Component Name, and Trace Option list boxes.

   Function Name
   A specific set of Personal Communications features, such as 3270/5250 Emulator or User Services.

   Component Name
   the name of a specific part of a function, such as API data (for the 3270/5250 Emulator function) or Node Initialization (for the User Services function).

   Trace Options
   The options associated with a particular component, such as EHLLAPI (for the API component) or API trace (for the Node Initialization component).

5. Start tracing data by clicking **Start**, or apply changes to the trace options by clicking **Apply**.

6. Run the operation that you want to trace.

7. Optionally, stop the trace by clicking **Stop**.

8. Save the trace data to your hard disk by clicking **Save**.

9. Click **Format** to specify a formatted trace file name and to format the trace data. The Information Bundler utility should be used immediately after the trace is complete to ensure that the correct information is gathered.

   Note: If you have changed the path for the fomatted trace file, the Information Bundler will not find the trace information. Copy the trace files to the Personal Communications private subdirectory.

10. Click **OK**.

11. Click **Clear** to clear the trace buffer where you saved a trace.

12. Use the log viewer to view the formatted trace log.

Information Bundler

The Personal Communications Information Bundler utility gathers system files, trace and log files, and registry information and creates a .RAM file. This .RAM file is sent to support personnel via FTP for problem resolution. This utility should be executed immediately after the trace is complete to ensure that the correct information is gathered. To use the information bundler utility:

1. From the PD Aids folder in the Personal Communications program group, double-click the **Information Bundler** icon; or, from an active session, select **Launch > Information Bundler** from the **Actions** menu.

Problem Analysis

2. A .RAM file containing system and Personal Communications information is created in the Personal Communications directory. By default this file is called 12345.RAM.

Internet Service

The Internet Service utility enables you to send the .RAM file containing diagnostic data collected by the Information Bundler to an FTP server. This utility will not work unless TCP/IP is configured and you have a direct connection to the internet. To use Internet Service:

1. Double-click the **Internet Service** icon located in the Personal Communications program group; or, from an active session, select **Launch > Internet Service** from the **Actions** menu.

   The Internet Service window contains four data fields that must have valid values before you can submit your problem report.

2. Verify that the FTP Address field contains the default address **testcase.software.ibm.com**. This is the service anonymous FTP server.

3. Enter your email address in the field provided.

4. In the Problem Determination Filename field, enter the file name and path of the .RAM file created with the Information Bundler. This file is located in the Personal Communications installation directory.

5. In the PMR Number field, enter the PMR number that you received as a result of contacting IBM support personnel.

6. When all fields in the window have been filled with valid values, click **Transmit** to submit your problem determination information.

Problem Analysis

Chapter 4. Printing

You can use Personal Communications to print from display or printer sessions.

From display sessions, you can print all (**Print Screen**) or part (**Trim Print**) of the screen of your session window on a workstation printer. **Trim Print** is not available for PC400 sessions. For more information, refer to *Quick Beginnings* or the online help.

From printer sessions, you can direct printing from an S/390® or AS/400 system to a workstation printer.

Note: When you use a host application which prints to your workstation's LPT1, you must first select the printer in the Printer Setup dialog of the File menu.

Configure a printer session to designate a workstation printer as a system printer that will use either a Windows printer driver or a printer definition table (PDT) provided with Personal Communications.

- Use Windows printer drivers for Personal Communications to print files based on printer setup parameters, such as scaling, duplex options, and page orientation, that you define in **Printer Setup**.
- Use PDT files for Personal Communications to print files based on page setup information, such as control codes and the printer output format, defined in the PDT. You can customize PDTs to define your own controls, by editing the corresponding printer definition file (PDF) and converting it to a PDT.

From a 3270 display session, you can also use the ZipPrint utility provided in the Emulator Utilities folder to print PROFS or OfficeVision (OV) notes, calendars, documents, CMS files, XEDIT workspaces, and host-session screens. See the online help for more information.

For considerations and restrictions on printing, refer to "Chapter 11. Considerations for the Use of PC/3270 Sessions" on page 95 and "Chapter 16. Considerations for the Use of PC400 Sessions" on page 251.

Setting Up the Printer

Following is a description of how to set up your printer with either a Windows printer driver or a PDT file.

For an explanation of how to customize a PDT, refer to "Chapter 9. Building a Printer Definition Table (PDT) for PC/3270" on page 61 or "Chapter 12. Building a Printer Definition Table (PDT) for PC400" on page 109.

Defining a Printer for a Session

To choose or change the Windows printer driver to be used, follow these steps:

1. Select **Printer Setup** from the **File** menu of the session window.

   The Printer Setup window lists the available printers.

2. Select a printer from the list box. If desired, click the checkbox **Show this dialog before every print**.

> Note: The **Default** selection causes the current Windows default printer to be used.

3. Confirm that **Use PDT** is not selected.

Using PDT Files

To use a PDT file:

1. Click **File - Printer Setup** from the menu bar of the session window.

   The Printer Setup window appears.

2. Click the printer to be used from the list box.

3. Click **Setup**; specify the paper size.

4. Click **OK**.

5. Click the **Use PDT** check box, then **Select PDT**.

   The Select PDT file window appears.

6. To use an existing PDT, select the PDT file to be used; then click **OK**.

   To use a PDF that you have modified, you must first convert it to a PDT. To do so:

   a. Click **Convert PDF**.

   b. Select the PDF file to be converted from the list, then click **Convert**.

      The window displays the result of the conversion. If there are any errors during the conversion, they are listed in the window.

   c. When you select **Save List**, the window list is saved in ***.LST** file in the **PDFPDT** subdirectory.

      To close without saving the list, click **Close**.

      After the file is converted, control returns to the Select PDT file window and the converted PDT file appears in the list.

   d. Select the PDT file; then click **OK**.

7. Click **OK** in the Printer Setup window.

PDT Files

Basic Support

PDTs (printer definition tables) are compiled from PDFs (printer definition files). PDFs contain printer commands that must be understood/supported by your printer.

The following are the basic printer languages:

PCL   Printer Control Language (Hewlett-Packard)

PPDS   ProPrinter Data Stream (IBM)

ESC/P Printer Control Language (Epson)

POSTSCRIPT
   (No PDFs for this language)

Many printers support two or more of these languages. Most print drivers use a PJL (Printer Job Language) to switch between languages and to perform other job control functions, such as setting the number of copies.

You do not need a PDF for each different printer model; with the increasing number of models, PDFs are now named for the printer language, not the printer model (for example, ESC_P, HPPCL5, IBMPPDS).

It is recommended that you use the following PDFs (and associated PDTs):

HPLJ4 for all PCL5 and PCL6 language printers

IBM4212

for all PPDS language printers

LQ870 for all EPSON language printers

The older PDFs and PDTs are also supplied; others are on our website. If you already have modified PDFs, any PDF and PDTs with different names than those below, are retained during an install.

The End_Job statement in a PDF contains the printer commands that are sent to the printer at the end of each print job. If the End_Job contains a character defined as form feed (FFF in the PDT), a form feed (FF) is sent to the printer. It is not needed if the host application ends the job with the FF, as is commonly done. Some print drivers add the FF if needed, and most print drivers ignore extra FFsl. So the FFF usually is protection against the host application not using a FF, and usually causes no problem. However, if you get an extra blank page, remove the FFF.

All PDFs contain the FFF, except those remarked on in the following table. Most PDFs can be used for both 3270 and 5250 printing; those that must be used for 3270 only are also noted. Others are marked as being on the website only.

Table 1. Printer Definition Files

Printer Definition File (PDF) Name	Remarks
actlzr2	
bj300	(3270 only)
cpqpm20	(No FFF)
eap2250	
eap3250	
eap5500	
efx1170	
efx5000	
efx850	
elq1070	
elq2550	
elq510	
elq860	
elx810	
eplpcl4	
eplpcl5	(No FFF)
esc_p	(3270 only) (DBCS only)
esq1170	
hpdj550c	

Printing

Table 1. Printer Definition Files (continued)

Printer Definition File (PDF) Name	Remarks
hpdskjt	
hplj4	(No FFF)
hpljii	
hpljii2	(No FFF)
hpljiii	(No FFF)
hpljiiid	(No FFF)
ibm3812	
ibm3816	
ibm38521	(3270 only)
ibm38522	
ibm4019	
ibm4070	
ibm4070e	
ibm4072	
ibm4201	
ibm4202	
ibm4207	
ibm4208	
ibm4212	
ibm4226	
ibm5152	
ibm5182	
ibm5201	
ibm5202	
ibm5216	
ibm5223	
ibm5577	(No FFF) (3270 only) (DBCS only)
ibm5577a	(No FFF) (3270 only)
ibm5577b	(No FFF) (3270 only)
ibm5585	(No FFF) (3270 only) (DBCS only)
ibm5587	(No FFF) (3270 only) (DBCS only)
ibmd5250	(3270 only)
ibms5250	(3270 only)
lbp4	
lips3a4	(No FFF) (3270 only) (DBCS only)
lips3b4	(No FFF) (3270 only) (DBCS only)
lq870	
nec2200	(No FFF) (Website only)
nec5300	(No FFF) (Website only)
nec6300	(No FFF) (Website only)

Table 1. Printer Definition Files (continued)

Printer Definition File (PDF) Name	Remarks
necp2	(No FFF) (Website only)
nppages	(No FFF) (3270 only)
oki184t	(Website only)
oki320	(Website only)
oki3410	(Website only)
oki390p	(Website only)
oki393p	(Website only)
oki400	(No FFF) (Website only)
oki590	(Website only)
oki800	(No FFF) (Website only)
oki810	(No FFF) (Website only)
pan1123	
pan1124	
pan1180	
pan1695	
pan2123	
pan2124	
pan2180	
pan2624	
panlbp4	
panlbp5	(No FFF)
prn3812	(3270 only)
prn38521	(3270 only)
prn38522	(3270 only)
prn4019	(3270 only)
prn4072	(3270 only)
prn4202	(3270 only)
prn5152	(3270 only)
prn5182	(3270 only)
prn5202	(3270 only)
prn5216	(3270 only)
prn5223	(3270 only)

PFT Migration

You can migrate a PC Support/400 Workstation Feature Printer Function
Table (PFT) to a PDF for PC400. Refer to "PFT Migration Utility" on page 145
for more information.

Double-Byte Character Support

The PDTs provided by Personal Communications and the corresponding printers are listed in the following table.

Note: All PDT files are available in SBCS mode, except for the following:
- IBM5577.PDT
- IBM5577B.PDT
- IBM5585.PDT
- IBM5587.PDT
- ESC_P.PDT
- LIPS3A4.PDT (3270 only)
- LIPS3B4.PDT (3270 only)
- ESC_BIG5.PDT
- ESC_CNS.PDT
- ESC_5550.PDT
- ESC_TCA.PDT
- KS_JO.PDT
- KS_WAN.PDT
- KSSM_JO.PDT
- KSSM_WAN.PDT
- NPPAGES.PDT

The following PDTs are supported for both 3270 and 5250 sessions:

```
File Name        Printer Name/Model
---------        ------------------
IBM5577.PDT      IBM 4208-501, -5C2, -5K2
                 IBM 4216-510
                 IBM 5572-B02
                 IBM 5573-HC2
                 IBM 5575-B02, -F02, -H02, BC2, -FK1E
                 IBM 5577-B02, -G02, -H02, FC2, -HC2, -KC2,-HK2
                 IBM 5579-H02, -KC2
IBM5585.PDT      IBM 5585-H01, -HC1
                 IBM 5587-G01
                 IBM 5587-H01 (with extended functions)
                 IBM 5589-H01
IBM5587.PDT      IBM 5587-G01
                 IBM 5587-H01 (without extended functions)
ESC_P.PDT        ESC/P printer for China or Japan
ESC_BIG5.PDT     ESC/P printer with Big5 DBCS code
ESC_CNS.PDT      ESC/P printer with CNS DBCS code
ESC_5550.PDT     ESC/P printer 5550 DBCS code
ESC_TCA.PDT      ESC/P printer with TCA DBCS code
KS_JO.PDT        KS printer with combination code
KS_WAN.PDT       KS printer with completion code
KSSM_JO.PDT      KSSM printer with combination code
KSSM_WAN.PDT     KSSM printer with completion code
NPPAGES.PDT      IBM NP12, NP17, NP24 printers for Japan
```

The following PDTs are supported for DBCS only:

```
File Name        Printer Name/Model
---------        ------------------
NPPAGES.PDT      IBM NP12, NP17, and NP24 in PAGES mode
(Japan only)
IBM5585.PDT      All types of IBM 5585 printers
IBM5587.PDT      All types of IBM 5587 printers
IBM5577.PDT      All types of IBM 5577 printers
IBM5577B.PDT     IBM 5577-S02/T02 for Postal Bar Code Support (Japan only)
```

If a PDT file does not exist for the printer being used, you can create your own. To do so, first create a printer definition file (PDF) and convert it into a PDT file, as described in "Using PDT Files" on page 18. PDF files are saved in the **PDFPDT** subdirectory under the directory in which Personal Communications has been installed.

Page Setup Parameters

You can set Page Setup parameters, such as the maximum number of lines per page, the maximum number of columns, and fonts. These parameters are initially set to the defaults. Use this function to change specific control items.

Note: When a PDT file is used, this function cannot be used.

You can set the following parameters.

Table 2. Page Setup Parameters — Text

Parameter	Description
CPI	Specifies the number of characters to be printed per inch. If CPI was not set previously, a CPI value suitable for the font selected from the Font list box is assumed as the default.
LPI	Specifies the number of lines to be printed per inch. If LPI was not set previously, an LPI value suitable for the font selected from the Font list box is assumed as the default.
Maximum Number of Lines per Page	Specifies the maximum number of lines per page. A number in the range 1–255 can be specified. The default is **66**. See Note 1.
Maximum Number of Characters per Line	Specifies the maximum number of characters per line. A number in the range 1–255 can be specified. The default is **132**. See Note 2.
Font	Lists, in the list box, the fonts that can be used with the currently selected printer. The fonts in brackets [ ] are device fonts specific to the printer driver. The other fonts are graphics display interface (GDI) fonts printed in bit map formats.

Notes:

1. When you set this to a value other than the default, Personal Communications uses it to scale the LPI and font to the best fit for the page.
2. When you set this to a value other than the default, Personal Communications uses it to scale the CPI and font to the best fit for the page.

From a Personal Communications display session, you can set additional parameters by selecting the **Text Options** tab, as follows:

Printing

Table 3. Page Setup Parameters — Text Options

Group	Options	Setting
Print Options The following options can be seelected as print options. These options are not available for the printer session.	**Suppress null lines**	Determines whether to delete lines containing only null or non-printable characters (null or non-printable field characters, and field attributes) or to print them as null lines. **Selected** Null lines are not printed. **Not selected** Null lines are printed. This option can be used when: • The **Print Screen** command is used • LU type 3 printing, or non-SNA printing is done • The combination of bits 2 and 3 of the WCC is not 00.
	Print nulls as spaces	Determines whether to print NULL control codes as null characters. **Selected** The codes are printed as blanks. **Not selected** The codes are treated as null characters. This option can be used when: • The **Print Screen** command is used • LU type 1 or 3 printing, or non-SNA printing is done.
	Ignore FF when at first print position	Determines whether to ignore the FF code at the beginning of the buffer address. **Selected** The FF code is ignored and not executed. **Not selected** The FF code is executed. This option can be used when: • The **Print Screen** command is used • LU type 1, LU type 2, LU type 3, or non-SNA printing is done. This option cannot be used in 5250 sessions.

Table 3. Page Setup Parameters — Text Options (continued)

Group	Options	Setting
	FF takes a print position if followed by data	Determines whether to print null characters if data follows the FF code. **Selected** — FF is executed, and a blank character is printed at the beginning of the next page. Therefore, the next data is printed in the second column of the line. **Not selected** — FF is executed, and the next data is printed at the beginning of the first line on the next page. This option can be used when: • The **Print Screen** command is used • Host initiated local copy and LU type 3 printing, or non-SNA printing is done. • In both PDT and standard modes. This option cannot be used in 5250 sessions.
Suppress auto new line when:	**CR at maximum print position +1**	Determines whether to perform automatic new line when CR is the maximum number of columns to be printed away from the first column. **Selected** — Automatic new line is not done. The characters after the CR code are printed on the same line. **Not selected** — Automatic new line is done. Determines whether to perform automatic new line/carriage return before new line/carriage return (NL) is done when NL is the maximum number of columns to be printed away from the first column. This option cannot be used for 5250 sessions.

Printing

Table 3. Page Setup Parameters — Text Options (continued)

Group	Options	Setting
	NL at maximum print position +1	**Selected** Automatic new line is not done.
		Not selected After automatic new line is done, new line is done again.
		This option can be used when LU type 3 printing or non-SNA printing is done and the combination of bits 2 and 3 of the WCC is 00.
		This option cannot be used in 5250 sessions.
Form Feed Position **Note:** This cannot be used for 5250 sessions.	**Any position**	Determines that printing can start at any position on the page.
	Column 1 only	Determines that printing starts at column 1 of the page.

From a Personal Communications 3270 session, you can set additional parameters by selecting the **Graphics** tab.

Table 4. Page Setup Parameters - Graphics (PC/3270 Only)

Parameter	Description
Scaling	The screen image is printed with the resolution that is calculated as the actual screen resolution multiplied by the following parameters. **x1** Same resolution as screen **x2** Double the resolution in each dimension **x3** Triple the resolution in each dimension **x4** Four times the resolution in each dimension
Black-on-White	This option allows you to determine how the black pixels on the screen are printed.
	Yes Black pixels are printed as white pixels. Pixels other than black are printed as black pixels when you use a monochrome printer. When you use a color printer, they are printed in the same color as on the screen.
	No Black pixels are printed in black. Pixels other than black are printed as white pixels when you use a monochrome printer. When you use a color printer, they are printed in the same color as on the screen.

When you use a PC400 printer session, you can set additional parameters by clicking the **Orientation** tab, as follows.

Table 5. Parameters — Orientation (PC400 Printer Session Only)

Group	Parameter	Description
Margins These margin settings are used only if Use best fit scaling is selected.	Top Margin	Bottom Margin is assumed to be equal to Top Margin.
	Left Margin	Right Margin is assumed to be equal to Left Margin.

Table 5. Parameters — Orientation (PC400 Printer Session Only) (continued)

Group	Parameter	Description
Page Orientation Allows you to change the default page orientation to specify how to print a document on the workstation printer. If the orientation is explicitly set by the AS/400 page setup code, the explicit orientation is used.	Use automatic page orientation	If selected and the host does not explicitly set the orientation, the best orientation based on the host specified CPI, LPI, and page size will be used. If not selected and the host does not explicitly set the orientation the following drawer orientation will be used.

Table 6. Orientation Parameters (PC400 Printer Session Only)

Group	Setting	Description
Drawer 1 orientation The default page orientation for the paper from drawer 1.	**Computer output reduction**	The document is printed in landscape. The font, pitch, and margins are set to appropriate values to fit on a page.
	Portrait	The document is printed in portrait.
	Landscape	The document is printed in landscape.
Drawer 2 orientation The default page orientation for the paper from drawer 2.	**Computer output reduction**	The document is printed in landscape. The font, pitch, and margins are set to appropriate values to fit on a page
	Portrait	The document is printed in portrait.
	Landscape	The document is printed in landscape.
	Use best fit scaling	5250 print emulation now provides a function to scale your host fomratted page to the Best Fit in the physical Printable Area. This can be done for any 5250 print output. For example, Host formats output to print 66 lines and 85 characters at 6 LPI and 10 CPI. Personal Communications will now adjust the LPI and CPI to compensate for non printable borders enforced by the print drivers. On the Print Page Setup panel, you can turn this option on or off. In addition, you can now specify the margin sizes of the output pane.

When you use a PC400 printer session, you can set additional parameters by clicking the **Form Settings** tab, as follows.

Table 7. Page Setup Parameters — Form Settings (PC400 Printer Session Only)

Parameter	Description
Form Settings	Specifies the form that should be selected when an application program specifies one of the following paper sources: • Drawer-one form • Drawer-two form • Envelope-hopper form Before using this function, you must configure the paper trays and forms in the printer-driver setup.

Printing

Table 8. Page Setup Parameters — Advanced Options (PC400 Printer Session Only)

Option	Item to be set
Printer Font Code Page	Represents the code page being used for printing and displaying on the workstation.
No CR between fields	Represents not sending a CR when printing other fields on the same line.
Print bold as normal	Represents printing bold characters as not bold.
Display print status dialog	Represents showing a dialog window that will display showing printer status.
Use raster fonts	Represents allowing bitmap fonts for display and printing.

To set or change setup items:

1. Select **File > Page Setup** from the menu bar of the session window.

   The Page Setup window appears.
2. Select the tab that contains the parameters you want to change.

   Note::

   To switch from the current settings back to the defaults, select **Default**.
3. When all the items have been set, click **OK** or **Apply**.

Printing to Disk

If you are using a PDT, you can save a host print-job or the contents of the session window (Print Screen) to a workstation file instead of printing it.

Two types of Print-to-Disk function are provided by Personal Communications:

- **Print-to-Disk Append:**

  Appends multiple host print jobs or print screen jobs to a single workstation file.
- **Print-to-Disk Separate:**

  Saves each host-print job or screen to a separate workstation file. You can specify the file name, but the extension is automatically assigned as a decimal number from 000 to 999. If you delete a file, its number will be re-used. When all 999 numbers have been used, the extension is automatically assigned a decimal number from 1000 to 9999.

Notes:

1. Print-to-Disk is not available for the Print-Graphics function.
2. Print-to-Disk can be used only when you use a printer definition table (PDT) file.

To set up Print-to-Disk:

1. Select **File > Printer Setup** from the menu bar in the session window.

   The Printer Setup window lists the supported printers.
2. Select **Print to Disk Append** or **Print to Disk Separate** from the list box.
3. Click **Select PDT**.

   The Select PDT file window appears.
4. Select a PDT file from the list; then click **OK**.

   The Prnter Setup window reappears.
5. Click **Setup**.

   The File Selecting window appears.

6. Specify a file name, drive, and path; then click **OK**.

   Note: If you specify the name of an existing file, subsequent print jobs are appended to the data in the original file in the case of Print to Disk Append.

Printing Japan Postal Bar Code

In support of the new postal bar code implemented in Japan, Personal Communications provides print capabilities for postal bar codes.

Chapter 5. Transferring Files

Personal Communications File Transfer enables one or more files to be transferred between a host system and workstation at the same time. Transfer types and translation tables can be defined in advance.

Host Requirements

For PC/3270 File Transfer in SBCS mode, you need one or more of the following host file-transfer programs (referred to as IND$FILE):
- IBM 3270-PC File Transfer Program, 5665-311 (MVS/TSO)
- IBM 3270-PC File Transfer Program, 5664-281 (VM/SP 2.1)
- IBM CICS®/VS 3270-PC File Transfer Program, 5798-DQH (CICS/VS 1.5)

For PC/3270 File Transfer in DBCS mode, you need one or more of the following host file-transfer programs (referred to as APVUFILE):
- VM/CMS File Transfer Program, 5799-BWK (Japan)
- VM/CMS File Transfer Program, 5799-PGX (Korea, China, Taiwan)
- MVS/TSO File Transfer Program, 5799-BWJ (Japan)
- MVS/TSO File Transfer Program, 5799-PGY (Korea, China, Taiwan)
- CICS (MVS, VSE) File Transfer Program, 5799-BWL (Japan)
- CICS (MVS, VSE) File Transfer Program, 5799-PGZ (Korea, China, Taiwan)

For PC400 File Transfer in SBCS mode, you need one of the following host file-transfer programs (referred to as APVAFILE):
- Personal Communications Tools/400 8mm Tape — 46H8350
- Personal Communications Tools/400 1/2 inch Tape — 85G9973
- Personal Communications Tools/400 1/4 inch Tape — 85G9969

For PC400 File Transfer in DBCS mode, you need one of the following host file-transfer programs (referred to as APVAFILE):
- Personal Communications Tools/400 V1R1, 5799-QBX (Japan)
- Personal Communications Tools for OS/400 V1R1, 5799-FPZ (Korea, China, Taiwan)

Note:
PCT400 was withdrawn from marketing 3/98.

You can perform the following file transfer functions:
- Send files to the host system
- Receive files from the host system
- Use lists of files
- Create templates to define file names and transfer types
- Define transfer types
- Set transfer options
- Modify translation tables
- Import or export files (PC/3270 CICS only)
- Create interactive document profile (IDP) files (PC/3270 CICS only)
- Transfer files via the XMODEM or YMODEM protocols

Sending Files to the Host System

To send a file from your workstation to the host system:

1. Sign on to the host system.

2. Select **Send File to Host** from the **Actions** menu of the session window. (You can also select the **Send** button on the tool bar.)

   The Send File to Host window appears.

3. Specify the name of the workstation file to be sent to the host system by entering the name in the **PC File** text box, or click the **Browse** button to open a dialog box for selecting the file.

4. Enter the name under which the file will be stored on the host; then enter or select the **Transfer Type**. If a template is provided for the file type you are transferring, the host file name and the transfer type appear automatically.

   > **Using List Files**
   >
   > Select **Open List**; then select the list to be used for transfer. See 1 on page 33 for details of how to create list files.

5. Click **Send**.

   The file is sent to the host system. The send status appears in the Send a File Status window.

Receiving Files from the Host System

To transfer a file from the host system to your workstation:

1. Sign on to the host system.

2. Select **Receive File from Host** from the **Actions** menu. (You can also select the **Receive** button from the tool bar.)

   The Receive File from Host window appears.

3. Specify the name of the host file to be received. Enter the name in the **Host File** text box, or specify it as follows:

   > **Using the Clipboard button**
   > If you have copied one or more host file names to the clipboard, you can paste the names into the transfer list; click the **Clipboard** button to open a dialog box for this. Select one or more of the pasted file names to be transferred. Then click **OK**.

4. Enter or modify the suggested name under which the file will be stored on the workstation, and enter or select the **Transfer Type**; or click the **Browse** button to open a dialog box for selecting a location for the file.

   > **Using List Files**
   > Select **Open List**, and select the list to be used for transfer. (See 1 on page 33 for an explanation of how to create list files.)

   If a template is provided for the file type you are transferring, the workstation file name and the generated transfer type appear automatically.

5. Click **Receive**.

The receive status appears in the **Receive a File Status** window.

Using List Files

If the same files are transmitted frequently, you can create a list of the files and save it.

A list file can be used for both Send and Receive. The default list file extension is .SRL.

Creating List Files

To create a list file:

1. Select **Receive File from Host** from the **Actions** menu or **Send File to Host** from the **Actions** menu of the session window; or click the **Send** or **Receive** buttons on the tool bar.

   The corresponding window appears.

2. Select a file to be transferred from the **Host-File Name** or **PC-File Name** list box by pointing to the name of a file to be selected. While holding down the Ctrl key, click the left mouse button.

   The file name, its corresponding workstation or host file name (according to the available templates), and the transfer type appear in the **Transfer List** part of the window.

   Note: You can also click the **Browse** button (for sending files) or the **Clipboard** button (for receiving files) to open the corresponding dialog box, which allows you to select files for transferring; when you click **OK**, the selected files are shown in the **Transfer List**.

3. Click the **Add to List** button to include a selected file in the **Transfer List**.

4. After all desired files have been selected, click **Save List**.

   The Save File-Transfer List File As window appears.

5. Enter or select a list name, and click **OK**.

Editing Lists

To edit the contents of a previously created list:

1. As explained above, display the Send File to Host or Receive File from Host window.

2. Select **Open List**.

   The Open File-Transfer List File window appears.

3. Select the name corresponding to the list file to be edited, then click **OK**.

4. The contents of the selected list appear in the Send File to Host or Receive File from Host window.

5. Edit the contents of the list file.

Changing the contents of a list

Choose the file to be changed from the list, and overwrite the items to be changed in the text box; then click the **Update in List** button.

> **Removing a file from the list**
> Choose the file to be removed, and click **Remove from List**.

> **Adding a file to the list**
> Double-click the file to be added from the list of host or workstation files.

6. Select **Save List**.

   The Save File-Transfer List File As window appears.

7. Enter a name and then click **OK**.

Managing Templates

A *template* is a set of rules to be used by the workstation to automatically generate a workstation or host file name and transfer type when you specify a file to be sent or received.

You can have up to 32 templates. They are automatically numbered from 1 to 32.

When you specify a file to be transferred, the workstation scans the templates, starting from template 1. It uses the first matching template to generate a name for the transferred file and the transfer type.

To manage a template:

1. Select **Receive File from Host** from the **Actions** menu or **Send File to Host** from the **Actions** menu of the session window; or click the **Send** or **Receive** buttons on the tool bar.

   The Send File to Host or Receive File from Host window appears.

2. Select **Template**.

   The Template window appears. The contents of the window depend on the connected host system.

Adding Templates

The list box for the Template window lists the currently stored templates.

To add a template:

1. Select any template from the list box.

   The contents of the selected template appear under the list box.

2. Change the workstation or host file names or extensions by overwriting them; then select the transfer type. (For details of the transfer types, see "Defining Transfer Types" on page 35.)

3. Click **Add**.

   The window for determining where in the list to display the new template appears.

4. Select a template number and specify whether to display the new template before or after the template that has that number. Click **OK**.

   The new template is added to the list in the appropriate position.

Replacing and Deleting Templates

To change the contents of a currently stored template, or to delete a template:

1. Select the template to be changed or deleted.

   The contents of the selected template appear under the list box.

2. To change the contents, overwrite the appropriate part and then click **Replace**.

   To delete a template, click **Delete**.

   The selected template is changed or deleted, and the contents of the template list box are changed.

Testing Templates

To test the contents of an added or changed template:

1. Select the template to be tested from the list box.

   The number of the selected template appears in the Test Templates box in the lower part of the window.

2. Select or enter data for the following items:

   Test Mode
   Determine which mode is to be used for the test: the mode in which a file is transmitted from the workstation to the host system (send), or the mode in which a file is transmitted from the host system to the workstation (receive).

   Templates
   Determine which templates to test: only the template selected in step 1, or all registered templates.

   Source File
   Enter the name of the file to be used for the test.

3. Click **Test**.

   Target File indicates the name that has been generated by the template.

   Note: Testing a template does not transfer a file.

Defining Transfer Types

Transfer types define the option information used for controlling file transfer. Up to 32 transfer types can be defined for each host system. Text, binary, and append (excluding CICS) are the defaults.

To add or change transfer types:

1. Select **Preferences > Transfer** from the **Edit** menu of the session window.

2. Click the tab for your host type or modem protocol.

   The property page for the selected host or modem protocol appears. The items that appear depend on the selected host system.

3. Enter transfer-type names in the **Transfer Type** box, or select them from the drop-down list.

4. Select or enter the required items (see "Explanation of Items to Be Defined" on page 36).

   To add or replace a transfer type, click **Save**. To delete a transfer type, click **Delete**.

5. A dialog box displays, asking for confirmation. Click **OK**.

Explanation of Items to Be Defined

Choosing the appropriate property page enables you to set the items described in the following sections.

File Options

The file options that can be used depend on the type of the connected host system and the host code page selected when the session was configured. Table 10 lists the file options and the host systems and modes in which the options can be used. The mode values are:

Table 9. Mode Values for File Transfer Options

Mode	Host Code Page
DBCS	930 (Japan Katakana) 930 (Japan Katakana - Extended) 939 (Japan Latin - Extended) 1390 (New Japanese Katakana - Extended) 1399 (New Japanese Latin - Extended) 933 (Hangeul) 1364 (Hangeul 1364) 935 (Simplified Chinese) 937 (Traditional Chinese) 1371 (Taiwan 1372)
SBCS	Others

The following chart lists options for PC/3270. Refer to "Chapter 15. File Transfer for PC400" on page 247 for information on PC400 options.

Table 10. Transfer File Options (3270 Only)

File Option	Host System	Mode	Conversion Details
ASCII	VM/CMS MVS/TSO ICS	SBCS DBCS	Converts codes as follows when a file is sent: • Converts 1-byte workstation codes to EBCDIC codes • Converts 2-byte workstation codes to IBM host DBCS codes • Adds SO (hex 0E) and SI (hex 0F) before and after the DBCS field • Converts RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F) Converts codes as follows when a file is received: • Converts EBCDIC codes to 1-byte workstation codes • Converts IBM host DBCS codes to 2-byte workstation codes • Removes SO (hex 0E) and SI (hex 0F) from before and after a DBCS field

Table 10. Transfer File Options (3270 Only) (continued)

File Option	Host System	Mode	Conversion Details
JISCII	VM/CMS MVS/TSO CICS	DBCS	Converts codes as follows when a file is sent: • Converts 1-byte workstation codes to EBCDIC codes • Converts 2-byte workstation codes to IBM kanji codes • Adds SO (hex 0E) and SI (hex 0F) before and after the kanji field • Converts RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F) Converts codes as follows when a file is received: • Converts EBCDIC codes to 1-byte workstation codes • Converts IBM kanji codes to 2-byte workstation codes • Removes SO (hex 0E) and SI (hex 0F) from before and after a kanji field
CRLF	VM CMS MVS/TSO CICS	SBCS DBCS	Converts codes as follows when a file is sent: • Does not remove CRLF (hex 0D0A) from the end of each line. The code is treated as a delimiter for each record. • Removes EOF (hex 1A) from the end of the file. Converts codes as follows when a file is received: • Adds CRLF (hex 0D0A) to the end of each line. • Adds EOF (hex 1A) to the end of the file. Removes EOF from the existing file, and appends EOF to the end of the added file when APPEND is specified.
APPEND	VM/CMS MVS/TSO	SBCS DBCS	Appends the sent file to the existing host file. Appends the received file to the existing workstation file.
SO	VM/CMS MVS/TSO CICS	DBCS	This option is valid only for DBCS when used with the *JISCII* and *ASCII* options when receiving a file. SO (hex 0E) and SI (hex 0F) are converted to RS (hex 1E) and US (hex 1F).
NOSO	VM/CMS MVS/TSO CICS	DBCS	This option is valid only for DBCS when used with the *JISCII* and *ASCII* options when sending a file. • SO (hex 0E) and SI (hex 0F) are not added before and after the DBCS field. • RS (hex 1E) and US (hex 1F) are not converted to SO (hex 1F) and SI (hex 0F).
BLANK	VM/CMS MVS/TSO CICS	DBCS	This option is valid with the *CRLF* option when receiving a file. BLANK (hex 40) is not removed from the end of each line.
USER	VM/CMS MVS/TSO CICS	DBCS	This option is valid only for DBCS when used with the *JISCII*, *ASCII*, and *SO* options when receiving a file. SO (hex 0E) and SI (hex 0F) are not converted to RS (hex 1E) and US (hex 1F); they are written to a file without being converted.

Record Format
Valid only for VM/CMS and MVS/TSO when APPEND is not specified for file transmission. Any of the these can be selected:
• **Default**
• **Fixed** (fixed length)
• **Variable** (variable length)
• **Undefined** (undefined mode for MVS/TSO only)

If the **Default** value is selected, the record format is selected automatically by the host system.

Specifying **Variable** for VM file transfer enables host disk space to be used efficiently.

Logical Record Length (LRECL)

Valid only for VM/CMS and MVS/TSO when APPEND is not specified for file transmission.

Enter the **logical record length** to be used (host record byte count) in the **LRECL** text box. If **Variable** and **Undefined Mode** are specified as the record format, the logical record length is the maximum record length within a file. The maximum value is 32767.

The record length of a file sent from a workstation to the host system might exceed the logical record length specified here. If so, the host file transfer program divides the file by the logical record length.

When sending a text file from a workstation to a host, if the text file contains 2-byte workstation codes (such as kanji codes), the record length of the file is changed because SO and SI have been inserted.

To send a file containing long records to the host system, specify a sufficiently long logical record length.

Because the record length of a workstation file exceeds the logical record length, a message does not appear normally if each record is divided. To display a message, add the following specification to the [Transfer] item of the workstation profile:

```
DisplayTruncateMessage = Y
```

TSO Allocation Parameter (MVS/TSO)

Valid only for MVS/TSO when **APPEND** is not specified for file transmission. The following items can be specified:

[Allocation Amounts]

Primary
　Enter the number of tracks or cylinders allocated to this file transfer.

Secondary
　If the primary allocation is not sufficient for the entire file transfer, enter additional storage capacity allocated to the file transfer.

[Allocation Units]

Tracks
　Specify this parameter to allocate a host file by track. Ask your system manager whether to use tracks or cylinders as the unit.

Cylinders
　Specify this parameter to allocate a host file in units of cylinders.

AVblocks
　Specify this parameter to allocate a host file in units of blocks.

[Block size]

This item is used only to create a new data set. Enter the block size of a new host data set, in bytes, in the text box. If this item is omitted, the workstation assumes

the value that appears in the **Logical Record Length** box. The maximum value is 32767. If **AVblocks** is selected, the block size is the block size of the new data set.

Additional Options
The required host command options can be entered in the **Additional Options** text box.

Setting Advanced Transfer Options

To set advanced options:

1. Select **Preferences > Transfer** from the **Edit** menu of the session window. The setup dialog is displayed.
2. Change the required settings on the property page labeled **General**.
3. Click **OK**.

The following sections contain information about the items which can be defined for file transfer options.

Host Type (3270 Only)

You can specify from the drop-down list box the type of host (MVS/TSO, VM, or CICS) to which your workstation is connected.

Data Transfer (5250 Only)

You can choose whether the Data Transfer function (see "Chapter 14. Data Transfer for PC400" on page 175) is to be used instead of the normal file transfer function.

Protocol Type (ASCII Only)

You can specify from the drop-down list box the type of modem protocol (XMODEM or YMODEM) to be used for transfers.

Host Command

You can specify host command to be called when file transfer starts. If nothing is entered in this text box, IND$FILE or its equivalent for other countries is used for 3270 SBCS sessions, APVUFILE is used for 3270 DBCS sessions, and APVAFILE is used for 5250 sessions.

Default PC Directory

You can specify the default directory that appears in the Send File to Host or Receive File From Host window. To select the directory, click the **Browse** button.

Default Partitioned Data Set (MVS/TSO Only)

You can specify the MVS partitioned data set to be used as the default.

Default VM Disk (VM Only)

You can specify the VM disk to be used as the default.

Default Library Name (AS/400 Only)

You can specify the AS/400 library to be used as the default.

PC Code Page

When a file is transferred, EBCDIC codes are converted to 1-byte workstation codes, and vice versa. A valid value is automatically selected from among the following values for SBCS sessions: 437, 737, 813, 819, 833, 850, 852, 854, 857, 858, 860, 861, 862, 863, 864, 865, 866, 869, 874, 912, 915, 916, 920, 921, 922, 1008, 1089, 1124, 1125, 1127, 1129, 1131, 1133, 1153, 1155, 1156, 1157, 1158, 1160, 1164, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, and 1258; and from the following values for DBCS sessions: 897 and 1041 (Japanese); 1088 and 1126 (Hangeul); 1114 (Simplified Chinese and Traditional Chinese)—according to the host code page specified when the workstation is configured. For an explanation of how to select host code pages, see the online help for the host code page.

Packet Size (3270 Only)

The amount of memory (in bytes) used by the workstation for transmission and reception. If a large value is entered, a file is transferred more quickly, but the memory overhead is larger. The default value is 12288. In the case of Telnet3270, you can specify a packet size larger than 8,000 bytes by adding the following line to the Telnet3270 stanza in your workstation profile:

```
SendBufferSize=nnnn
```

File-Transfer Timeout

You can define the time the workstation waits for a response from the host system (in seconds). If the host system does not respond, the transfer is canceled, and an error message appears. A number in the range 20–65535 (or 0) can be specified. The default is 60 seconds for ASCII sessions; for all others, it is 30 seconds. Specify an appropriate value such that the error message does not appear too early. If you specify 0, a timeout is not set.

If a packet or block size is relatively large for low-speed lines, such as SDLC or COM port lines, it is recommended that 150 seconds or greater be specified.

Extension for List-Files

You can change the default extension (.SRL) of file-transfer list files.

Clear Session Before Transfer (3270 Only)

You can specify whether a Clear command is sent to the host system before a file is transferred. Choose any of these option buttons:

Default
> A Clear command is sent before a file is transferred (VM/CMS or CICS only).

Yes A Clear command is sent for MVS also.

No A Clear command is not sent for any host system.

Show Status Window

You can choose the method of displaying the file-transfer-progress status.

In Session
> When file transfer starts, the status window appears. The name of the file being transferred and the transfer progress appear.

In icon

When file transfer starts, the status icon appears on the screen. If the icon is restored, the status window appears.

Enhanced Protocol (SBCS Global Network Connection Only)

This option enables faster file transfer in most cases. The packet size for Enhanced Protocol is fixed, so the File Transfer Packet Size setting is ignored.

Lam-Alef Expansion (Arabic Only)

If you want the Lam-Alef characters transferred from the host to be expanded into the two-character representation for Windows, click this check box.

Note: This will change the size of the record on the personal computer; this should not be a problem for normal text files, but exercise caution when formatted files and databases are being transferred.

Setting Up the Translation Table

You can create or edit the translation table to be used for sending or receiving files.

When you use a DBCS session as the host session, that is, when 930 or 939 (Japanese), 933 or 1364 (Hangeul), 935 (Simplified Chinese), or 937 (Traditional Chinese) is specified during configuration as the host code page, you can create and change a translation table for the user-font area. A DBCS translation table is then used for displaying a screen, printing, and sending and receiving files.

Changing the Translation Table

To change the translation table:
1. Select **Preferences > Transfer** from the **Edit** menu of the session window.
2. Click the **Translation Table** tab on the resulting window.

   The Translation-Table Setup property page appears.
3. The table currently being used (IBM default or the name of a user-defined table) is shown. Choose either **IBM Default** or **User-Defined**.
4. If you choose **User-Defined**, enter a translation-table name in the **File Name** text box, or select a name by clicking **Browse**.
5. Click **OK**.

Customizing the Translation Table

You can create a user-specific translation table for transmission or reception, or you can edit an existing translation table.

To create or edit a translation table:
1. On the **Translation Tables** property page, click **Customize** in the Upload or Download window.

   The Customize Translation window appears.

   If you chose **IBM Default** or if you chose **New** from the File menu, the default values appear in the table.

   Translation source codes
   PC code-points when an upload translation table is edited. Host code-points when a download translation table is edited.

Translation target codes
Host code-points when an upload translation table is edited. PC code-points when a download translation table is edited.

2. Double-click the code to be changed in the table, and change the value in the entry field that subsequently appears.

3. Click **Save** or **Save As** from the File menu.

4. If asked, enter a name in the Save Translation File As window and click **OK**.

5. Click **Exit** from the File menu of the Customize Translation window.

User-Font Area (DBCS Only)

For DBCS sessions, the following code ranges can be defined by the user with the DBCS translation table:

User-font area
Range

Host kanji code
First byte: from X'69' to X'7F' Second byte: from X'41' to X'FE'

Workstation kanji code
First byte: from X'F0' to X'F9' Second byte: from X'40' to X'7E', from X'80' to X'FC'

Host kanji numbers
The host kanji-numbers area corresponds to the user-font area of host kanji codes. For example, the host kanji number 10561 corresponds to the host kanji code 6941.

JIS KUTEN numbers
From section 95 to section 114 (from 1 to 94)

Traditional Chinese host code
From X'C241' to X'E2FD' (low byte X'41' to X'FD', skipping X'7F')

Traditional Chinese workstation code
The following ranges:
From X'FA40' to X'FEFE' (low byte X'40' to X'7E', X'A1' to X'FE')
From X'8E40' to X'A0FE' (low byte X'40' to X'7E', X'A1' to X'FE')
From X'8140' to X'8DFE' (low byte X'40' to X'7E', X'A1' to X'FE')
From X'8181' to X'8C82' (low byte X'81' to X'A0')
From X'F9D6' to X'F9FE'

Simplified Chinese host code
From X'7641' to X'7FFD' (low byte X'41' to X'FD', skipping X'7F')

Simplified Chinese workstation code
From X'8DA1' to X'A0FE' (low byte X'A1' to X'FE')

Hangeul host code
From X'D441' to X'DDFD' (low byte X'41' to X'FD', skipping X'7F')

Hangeul workstation code
The following ranges:
From X'C9A1' to X'C9FE' (low byte X'A1' to X'FE')
From X'FEA1' to X'FEFE' (low byte X'A1' to X'FE')
From X'8FA1' to X'A0FE' (low byte X'A1' to X'FE')

Import/Export (3270 CICS Only)

Import/Export is an office system communication program and an application program executed under the IBM Customer Information Control System (CICS).

Clicking **Import/Export** loads a module into workstation memory. You can then start Import or Export from a menu on the host screen.

When you export a document from the host, the workstation receives two files: one is the file itself, and the other is the interchange document profile (IDP) file, which contains document header information.

When you **Import** a file to a host system, it must be accompanied by an IDP file of the same name. If the necessary IDP file does not exist, you can create it as described in "Creating IDP Files".

To transmit files using Import/Export:
1. Verify that the window of the host session is active and ready for file transfer.
2. Select **Import/Export** from the **Actions** menu of the session window.

   The minimized Import/Export Status window appears.
3. Select **Import** or **Export** from the host application menu.
4. Specify the host and workstation file names of the file to be transferred. Run Import or Export.

   When Import or Export starts, the Import/Export Status window is maximized.

   After the file is transferred, the window is closed.

Creating IDP Files

To send a document to the host system using Import, you must have an IDP file that includes information on the document. If you have previously exported the file, you should have an IDP. If not, or if you have deleted the IDP file, you must create an IDP file.

To create an IDP file:
1. Select **Preferences > Transfer** from the **Edit** menu of the session window.
2. Click the **IDP Files** tab on the setup window.
3. Enter the name for the IDP file to be created, or click the **Browse** button to select it.

   The IDP file has the same name as the file to be transferred, and the extension **(.IDP)**.
4. Select **OK**.

Transferring Files

Chapter 6. SNA Node Operations

SNA Node Operations enable you to display information and manage the node for certain Personal Communications resources, which can be useful when operating the SNA node. The following tasks are supported:
- Starting and stopping resources
- Deleting resources
- Displaying resource information
- Changing session limits
- Initiating path switches

The configuration information for a SNA Node is stored in an ASCII file, with the extension .ACG. You can edit the file by launching SNA Node Configuration. To ensure that your new configuration is correct, use the Verification tool.

You can display information or take action on resources if you start 3270, 5250, or any client/server sessions with one of the following attachment types:

System/390 Connections

LAN   LAN via IEEE 802.2
> 3270 via AS/400
> APPC 3270 via LAN
> Dependent Logical Unit Requester (DLUR)
> 3174 Peer Communication

COM port
> SNA-over-Async
> IBM Global network — SNA-over-Async
> Dependent Logical Unit Requester (DLUR) via:
>
>> SNA-over-Async
>>
>> Hayes AutoSync
>
> APPC 3270 via:
>
>> SNA-over-Async
>>
>> Hayes AutoSync
>>
>> X.25 Hayes AutoSync
>>
>> X.25 DLUR via Hayes AutoSync
>>
>> X.25 APPC 3270 via Hayes AutoSync

5250   3270 via AS/400
> Dependent Logical Unit Requester (DLUR)

SDLC Synchronous Data Link Control
> 3270 via AS/400 (passthru)
> APPC 3270 via SDLC
> Dependent Logical Unit Requester (DLUR)

SNA/IP
> LU 0, 1, 2, 3
> APPC 3270
> LU 0, 1, 2, 3 via DLUR

IBM ISA/MCA WAC
> Synchronous Data Link Control
> 3270 via AS/400 (passthru)

       45

> APPC 3270 via SDLC
> Dependent Logical Unit Requester
> X.25 Qualified Logical Link Control
> X.25 3270 via AS/400
> APPC 3270 via X.25
> X.25 DLUR

IBM-EEDLC
> LU (0, 1, 2, 3) via DLUR)
> APPC 3270

OEM LU (0, 1, 2, 3)
> 3270 via AS/400 (passthru)
> APPC 3270
> Dependent Logical Unit Requester (DLUR)

AS/400 Connections
LAN LAN via IEEE 802.2
5250 Twinaxial Data Link Control (APPC)
COM port
> SNA-over-Async
> Hayes AutoSync
> X.25 Hayes AutoSync

SDLC Synchronous Data Link Control
SNA/IP
> 5250

IBM ISA/MCA WAC
> Synchronous Data Link Control (WAC)
> X.25 Qualified Logical Link Control

IBM-EEDLC
> 5250

OEM APPC 5250

Client/Server Connections
LAN LAN via IEEE 802.2
Twinaxial
> Twinaxial Data Link Control (APPC)

COM port
> SNA-over-Async
> Hayes AutoSync
> X.25

SDLC Synchronous Data Link Control
IBM WAC
> Synchronous Data Link Control
> X.25

AnyNet SNA over TCP/IP
> APPC

Enterprise Extender
> HPR over IP

SNA resources that can be displayed or managed include the following:

AnyNet® Sockets Devices
> Shows the currently defined AnyNet Sockets over SNA devices on a system.

AnyNet Sockets Routes
> Shows the AnyNet routes through the network.

AnyNet IP to LU Mapping
Shows the attributes used by AnyNet Sockets over SNA to map an IP address to an LU.

AnyNet Sockets Connections
Displays the active AnyNet Sockets over SNA access node and gateway connections.

Connection Networks
Allow APPN nodes in a LAN to have direct links with each other without requiring logical link definitions at each node.

Connections
Link stations to the adjacent nodes

CPI-C Side Information
Associate a set of parameters with a specified symbolic destination name.

Data Link Controls
How data is formatted for transmission on the physical connection.

Devices
Parts of the workstation hardware, such as adapters, that are used to transmit and receive data.

DLUR PUs
A dependent logical unit requester (DLUR) physical unit (PU) is a PU in an APPN end node that owns dependent LUs, but requests that a dependent LU server provide the SSCP services for those dependent LUs.

Focal Points
A system that provides centralized network management services.

Local LU 0 to 3
A local logical unit (LU) type 0, 1, 2, or 3 is a dependent LU on the workstation that provides services for 3270 terminal and printer emulation applications.

Local LU 6.2
A local logical unit (LU) type 6.2 is an independent or dependent LU that provides APPC services.

LU 6.2 Sessions
Transports data between two partner LU 6.2s. Conversations between transaction programs use LU 6.2 sessions.

Mode The name used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service (COS) within the transport network.

Node The control point that manages the node and its associated resources. The local node provides APPN services.

Partner LU 6.2
A remote computer that communicates through an APPC session with local LU 6.2s.

RTP Connections
In high-performance routing (HPR), a rapid transport protocol (RTP) connection is the connection established between the endpoints of the route to transport session traffic.

Transaction Programs
A transaction program (TP) is a program that processes transactions in an SNA network. There are two kinds of TPs: application transaction programs and service transaction programs.

Starting Node Operations

To start Node Operations, click the **SNA Node Operations** icon in the Personal Communications folder; or, from an active session window select **Launch > SNA Node Operations** from the **Actions** menu.

Note: This utility can also be started using a command ("Command-Line Utilities" on page 50).

Using the Menu Bar

From the menu bar of the SNA Node Operations utility, select one of the following:

Operations
To start or stop a SNA node.

To start a node, select the configuration file you want to use.

To change session limits (CNOS).

To apply a new configuration file, select the configuration file you want to use.

Launch
To launch to other Personal Communications programs, such as Log Viewer, Trace Facility, or SNA Node Configuration.

View To modify the layout of your Personal Communications window (Tool bar or Status bar).

To select resource attributes, which allows you to customize the display of SNA resource attributes.

To view frequently used resources, such as:
- Node
- Connections
- Local LU 0 to 3
- Partner LUs
- LU 6.2 sessions

Window
To modify the layout of your Personal Communications window.

Help Displays the Personal Communications online help.

Displaying a Resource

To display a resource, select a resource from the pull-down list on the toolbar.

Starting a Resource

To start a resource:
1. Select a resource from the pull-down list on the toolbar.
2. Select the first column of the item, and right mouse click to display the pop-up.

3. Select **Start** to start the resource. A started (active) resource can be stopped, but not deleted.

Stopping a Resource

There are two ways of stopping a resource:

Normal Stop
Performs clean up and then suspends the resource.

Abnormal Stop
Immediately suspends the resource.

To stop a resource:
1. Select a resource from the pull-down list on the tool bar.
2. Select the first column of the item, and right mouse click to display the pop-up.
3. Select **Normal Stop** or **Abnormal Stop** to suspend the resource. If a resource is stopped (inactive), it can be deleted or redefined.

Deleting a Resource

To delete a resource:
1. Select a resource from the pull-down list on the tool bar.
2. Select the first column of the item, and right mouse click to display the pop-up.
3. Select **Delete** to delete the resource.

A resource to be redefined need not be deleted, but it should be stopped (inactive).

Changing an HPR Path Switch

To perform a high performance routing (HPR) path switch:
1. Select **RTP connections** from the pull-down list on the tool bar.
2. Select the first column of the item, and right mouse click to display the pop-up.
3. Select **HPR path switch** to change the path for the data on this RTP connection.

Updating SNA Resources Dynamically

It is not necessary to delete an SNA resource before redefining it. A resource must be inactive, however, when you redefine it.

If a redefinition is rejected for a reason other than that the resource is active, a STATE_CHECK return code is returned as well as a specific secondary return code, as listed in the NOF specification. Most rejections are due to inconsistencies in the configuration, such as one of the following:

- Attempting to define an LS with the adjacent CP name set to the local CP name.
- Attempting to define two link stations on a non-switched port with the same destination address.

Such rejections occur regardless of whether you are defining a resource for the first time or redefining it.

Keep in mind the following when redefining resources:

- You may not change the ls_role of a port that has link stations defined on it. The link stations must be deleted before the port role can be changed (because an LS may default to the port ls_role).
- You may not change the type of a named DLC. The DLC definition must be deleted before the same name can be reused for a DLC of a different type.

Command-Line Utilities

Another way to perform SNA Node operations is to use the command line. Five command line programs are available to perform basic operations for the SNA Node.

CSSTART [-a | -d | -m | -h] [-p | -q] [cfgfile]

The CSSTART command is used to start the SNA Node with a specified configuration. The **-a** flag enables automatic start up after rebooting the machine. The **-d** flag sets the default SNA node configuration file. The **-m** flag stands for manual start which disables automatic start up after rebooting the machine. If the **-p** flag is used, all messages are shown in a message box pop-up. Messages are written to *stdout* by default. The **-q** flag suppresses all output. The **-h** flag summons the help data for this command.The **cfgfile** parameter is the name of the configuration file you want to use to start the product. If no **cfgfile** parameter is specified, the default configuration file is used. If no default configuration file has been set, an error is reported. If the node is successfully started, CSSTART returns zero, otherwise a non-zero value is returned.

CSSTOP [-p | -q][-h]

The CSSTOP command is used to stop the SNA Node. The flags have the same meaning as for CSSTART. If the node is successfully stopped then CSSTOP returns zero, otherwise a non-zero value is returned.

CSQUERY [-p | -q][-h]

The CSQUERY command is used to query the status of the SNA Node. The flags have the same meaning as for CSSTART. Along with the status, the default and active configuration file names are shown. If the node is running, then CSQUERY returns zero; otherwise a non-zero value is returned.

CSDSPY {resource [object_id] [/D#]} | [-h]
Displays information about SNA node resources, where:

resource
Is one of the following:

AIL	AnyNet IP to LU mapping
ASC	AnyNet Sockets IP connections
ASD	AnyNet Sockets devices
ASR	AnyNet Sockets routes
CNT	Connection networks
CON	Connections
CPS	CPI-C side information
DLC	Data link controls
DEV	Devices

DPU	DLUR PUs
FPT	Focal Points
LU0	Local LU 0 to 3
L62	Local LU 6.2
L6S	LU 6.2 sessions
MOD	Modes
NOD	Nodes
PLU	Partner LU 6.2
RTP	RTP connections
TRP	Transaction programs

object_id
> Specifies a case-sensitive value (for example, LU001 or LINK001) to search for. Only objects whose first attribute matches this value are displayed. The object id must be spelled correctly, including case (upper and lower). An incorrect object id will result in nothing being displayed.

/D# Specifies the level of detail to display. # is a number between 1 and 3, where 1 displays the least level of detail and 3 shows all information. 2 is the default.

CSMODIFY {resource object_id action} | [-h]
> Enables you to control SNA node resources, where:

resource
> Is one of the following:

CON	Connections. You can start, stop, or delete this resource.
CPS	CPI-C side information. You can delete this resource.
DLC	Data link controls. You can start, stop, or delete this resource.
DEV	Devices. You can start, stop, or delete this resource.
DPU	DLUR DPUs. You can start or stop this resource.
LU0	Local LU 0 to 3. You can delete this resource unless the LU is in use.
L62	Local LU 6.2. You can delete this resource.
RTP	RTP connections. You can switch the path of this resource.

object_id
> Specifies a case-sensitive value (for example, LU001 or LINK001) to search for. Only objects whose first attribute matches this value are displayed. The object id must be spelled correctly, including case (upper and lower). An incorrect object id will result in an error being displayed.

action One of the following:

SN	Normal stop
SI	Immediate stop
ST	Start

DL   Delete

PS   Path switch

[-h]   This switch is used to summon help data for this command.

Chapter 7. Log Viewer Functions

The Personal Communications log viewer utility enables you to view, merge, sort, search, and filter information contained in message and trace logs. You can use the viewer during problem determination to work with message and trace log entries. The default name of the message log output file is *PCSMSG.MLG*; its file extension must be *.MLG*. The file extension for trace logs must be *.TLG*. Please note that the Help per Message Log Item functionality is only available for message logs.

Viewing Message and Trace Logs

To view message or trace logs:
1. From the Administrative and PD Aids folder, click **Log Viewer**; or, from an active session window, select **Launch > Log Viewer** from the **Actions** menu.
2. From the list of logged messages, click on one of the details on the log entry in the bottom pane.

Note: If the logger device driver determines that the product kernel driver-generated log is full and cannot log a message, it will create an entry in the Windows log. The information logged may include which log failed, as well as the location and reason for the failure.

Changing Message Log Size and Location

The Personal Communications log viewer utility allows you to modify the size and location of message log files, and change the name of the default message log file. The size of a log file is counted in kilobytes and can range from a minimum of 4Kb to a maximum limited only by available hard disk space.

To modify the location and size of the log:
1. From the **Log Viewer** main menu, choose **Options** and then choose **Configure Message Log Settings...** A Windows common dialog box is displayed.
2. From the dialog box, browse the directory structure and choose the destination directory and filename for the message log.
3. Using the spin control counter field, use the up and down arrows to increase or decrease the log file size (in kilobytes).
4. Select **OK** to save settings and exit the window.

Merging Message and Trace Logs

The Personal Communications log viewer utility allows you to open and merge message and trace log entries in the same log viewer window. You can merge any combination of message and trace log files.

- .MLG into .TLG

- .TLG into .TLG

- .TLG into .MLG

- .MLG into .MLG

To merge message and trace log files:
1. From the Log Viewer window, select the message or trace log file window where the files will be merged.
2. Select the **File** menu, then select **Merge** and choose a file to be merged.

Sorting Message and Trace Logs

The Personal Communications log viewer utility allows you to sort message and log files in ascending and descending order. To sort files in ascending order, click the column header one time, or right mouse click the data. Click the column header, or right mouse click the data a second time to sort in descending order.

Message and logs can also be sorted by selecting the data to be sorted, and right mouse clicking to display the pop-up menu. Click **Sort**.

Searching Personal Communications Logs

To search the log files, click **Edit > Find** on the Main menu.

Type your search string in the provided box. You can refine your search by checking the **Match case** checkbox if your search is to be case-specific. If you want to limit your search to only complete words, check the **Match whole word only** check box.

Clicking **Find Next** takes you to the next instance where your search string appears highlighted in the log.

Clicking **Cancel** will stop the search.

Filtering Personal Communications Logs

Messages can be filtered by component only. Traces can be filtered by component, by process ID, and by thread ID.

To filter the message or trace record list, do the following:

1. Decide what you want to filter by. For example, in the message log, you may want to filter your view so that it contains only messages issued by a particular component. (Filtering by component is the default.)
2. Click the left mouse button to highlight the item that has the value that you want to filter by.
3. Right click in the appropriate column list area.
4. Click either **Filter In**, **Filter Out**.

   Filter In allows only those items selected to be in the resulting view. Filter out removes the selected items from the resulting view. **Sort** allows you to sort entries in ascending or descending order based upon the entry selected. This function works much like sorting by column header. Filters are cumulative, so you can filter the results of your first filter.

To restore your orignal view, click **View** and then **Refresh**, or you can click **F5**.

Chapter 8. Alerts

Alerts are generated by components of Personal Communications; corresponding messages are logged in the message log and can be viewed with the Personal Communications log viewer utility. See "Viewing Message and Trace Logs" on page 53 for more information.

Note that some alerts can be caused by different situations. Each situation may generate a different message. Other alerts are informational and do not generate specific messages in the log, although messages about problems relating to the situation that generated the alert may be logged.

```
  Alert ID number      Messages logged
```

APPN

```
  X'034A6F0B'          PCS4066E
                       PCS4068E
                       PCS4093E
  X'0DF28A14'          PCS4065E
                       PCS4066E
                       PCS4068E
                       PCS4069E
                       PCS4070E
  X'170F7710'          No specific message
  X'21745F28'          No specific message
  X'2313A399'          PCS4364A
                       PCS4365A
                       PCS4379A
  X'32CDF4E2'          PCS4073E
  X'47302521'          No specific message
  X'6D27D125'          PCS4066E
                       PCS4068E
  X'7599A7D8'          No specific message
  X'769022F0'          PCS4504E
  X'9DCD7CCA'          PCS4275E
                       PCS4280E
                       PCS4282E
                       PCS4283A
                       PCS4284E
                       PCS4304E
                       PCS4305E
                       PCS4310A
                       PCS4311E
                       PCS4312E
  X'9E452D9C'          PCS4593A
  X'A89646AA'          PCS4275E
                       PCS4280E
                       PCS4282E
                       PCS4283A
                       PCS4284E
                       PCS4304E
                       PCS4305E
                       PCS4310A
                       PCS4311E
                       PCS4312E
  X'B558D310'          PCS4324E
                       PCS4342E
                       PCS4347E
  X'C781E91E'          No specific message
  X'EBAA3C4F'          PCS4593A
```

Alerts

```
              X'EBEE390E'      PCS4063E
                               PCS4064E
                               PCS4066E
                               PCS4067E
                               PCS4068E
                               PCS4071A
                               PCS4091E
                               PCS4092E
                               PCS4094E
                               PCS4123E
                               PCS4124E
                               PCS4125E
              X'F52A0C01'      PCS4061E
                               PCS4062E
              X'FE1C42EB'      No specific message
```
LLC2 SAP
```
              X'016E5F4E'      PCS1066A
                               PCS1054A
              X'3BA03B6D'      PCS1066A
                               PCS1005E
              X'55BF3E1C'      PCS1066A
                               PCS1054A
              X'A676B230'      PCS1066A
                               PCS1005E
              X'CAF3C58A'      PCS1066A
                               PCS1054A
              X'D2E24978'      PCS1066A
                               PCS1005E
              X'D615A61E'      PCS1066A
                               PCS1054A
              X'EB1D6ABB'      PCS1066A
                               PCS1005E
              X'EB61E14F'      PCS1066A
                               PCS1005E
```

LLC2 Link Station
```
              X'216D1033'      PCS1065A
                               PCS1003E
              X'25AC0D84'      PCS1065A
                               PCS1004E
              X'28EF2B5D'      PCS1065A
                               PCS1001E
                               PCS1004E
                               PCS1006E
              X'5B8F5BA7'      PCS1065A
                               PCS1050A
              X'83D91642'      PCS1065A
                               PCS1000E
              X'87180BF5'      PCS1065A
                               PCS1000E
              X'8A5B2D2C'      PCS1065A
                               PCS1000E
              X'8E9A309B'      PCS1065A
                               PCS1000E
              X'E65B0B7F'      PCS1065A
```

pDLC

Alert ID number	Messages logged	Alert type	Alert description
X'0E499026'	PCS8607	01	3300
X'0F935B3E'	PCS8603	01	3300
X'21C346F0'	PCS8619	01	3300
X'25025B47'	PCS8620	01	3300
X'28417D9E'	PCS8617	01	3300

```
      X'2C806029'        PCS8618              01         3300
      X'4227687B'        PCS8610              01         3300
      X'6C6E2505'        PCS8604              01         8000
                         PCS8612
      X'7EA9C871'        PCS8608              01         3300
      X'8CEC6B74'        PCS8609              01         3300
      X'AB218ADF'        PCS8700              01         3300
      X'BB5C288E'        PCS8600              01         3300
      X'C16E9922'        PCS8615              01         3300
      X'C5AF8495'        PCS8616              01         3300
      X'C8ECA24C'        PCS8613              01         3300
      X'CC2DBFFB'        PCS8614              01         3300
      X'D3F9C6D8'        PCS8611              01         3300
      X'EBB67B65'        PCS8606              01         3300
```

Twinax DLC

```
   Alert ID number    Messages logged    Alert type   Alert description

   X'24077791'        PCS8700              01         3300
   X'C91C002A'        PCS8107              01         3300
   X'D7D5F953'        PCS8101              01         3300
```

Alerts

Part 2. Personal Communications/3270

Chapter 9. Building a Printer Definition Table (PDT) for PC/3270

This chapter, in combination with "Chapter 12. Building a Printer Definition Table (PDT) for PC400" on page 109, explains how to customize a printer definition table (PDT file) for PC/3270. PDTs for PC/3270 and PC400 differ only slightly; most of the information you will need is in the other chapter. If you use a common PDT (used for both PC/3270 and PC400), the additional statements for PC400 are ignored for PC/3270 processing. Similarly, statements unique to PC/3270 are ignored for PC400 processing.

"ASCII_PASSTHRU?" and "EBCDIC_PASSTHRU?"

The ASCII_PASSTHRU? and EBCDIC_PASSTHRU? PDF statements are new options available for PC/3270. See Chapter 12 for details.

Supplemental Explanation of PDF Statements for PC/3270

The PDF statements listed below have the following functions:

MAXIMUM_PAGE_LENGTH
> Printed lines per page. If you change this value, you must change the value in the "SET_PAGE_LENGTH=SFL value" statement to be the same.

MAXIMUM_PRINT_POSITION
> Printed characters per line.

COMPRESS_LINE_SPACING?
> (For LU 3 only) Whether blank or null lines are to be printed if all characters on that line are nulls.

FORM_FEED_ANY_POSITION?
> Whether a form feed is to be valid in any position. If NO, a form feed will be valid only in the following positions:
> * First print-position of the buffer
> * After a valid new line operation
> * First print-position of a line

OVERRIDE_FORMATTED_PRINT?
> Whether nulls are to be printed as blanks.

INTERV_REQ_TIMER
> This statement is ignored.

INTERV_TIMER_ON_PE_ONLY?
> This statement is ignored.

RESELECT_TIME_EXCPT_5204
> This statement is ignored.

ESC/P_LINE_FEED?
> If YES, the line feed (LF) function is emulated when the line feed command is received. This is useful when you do not want a line feed accompanied by a carriage return (CR) on a printer using the ESC/P printer language. If NO, the value defined in the LINE_FEED statement is sent to the printer.

IGNORE_FORM_FEED_AT_FIRST_POS?

If YES, the form feed (FF) function is ignored at the first position (for LU 2, LU 3, and non-SNA sessions) or at the beginning of the print job (for LU 1 sessions). Using this option eliminates extra blank pages at the beginning of each print job.

FORM_FEED_TAKES_POSITION?

If YES, the form feed (FF) function is effective if followed by data (LU 2, LU 3, and non-SNA sessions only).

KANJI_CODE?

If YES, kanji codes (JIS or SHIFT_JIS) have been used. This value cannot be changed.

ZENKAKU_SPACE

The size (adjustment unit) of a user-defined character and a HANKAKU character. This value cannot be changed.

SBCS_FONT_LOAD

Registration of a HANKAKU GAIJI. This value cannot be changed.

DBCS_FONT_LOAD

Registration of a ZENKAKU GAIJI. This value cannot be changed.

SET_LOCAL_FONT

Set a font set of user-defined characters. Remove it when user-defined characters are not loaded to a printer.

RESET_LOCAL_FONT

Reset a font set of user-defined characters. Remove it when user-defined characters are not loaded to a printer.

KANJI_ON

Set kanji mode.

KANJI_OFF

Reset kanji mode.

ATTRIBUTE_GRID_LINE

Set to grid-line print. This value cannot be changed.

START_DOUBLE_WIDTH_CHARACTER

Set a double-width character.

END_DOUBLE_WIDTH_CHARACTER

Reset a double-width character.

Notes:

1. When using IBM5577.PDF, change FORM_FEED=EJC to FORM_FEED in the file when a continuous form job does not feed correctly.

2. When using ESC_P.PDF, DBCS (2-byte) characters are placed to the left of the character box, not in the center. To place DBCS characters in the center of the character box, change the IBM-supplied values in ESC_P.PDF:

```
P10 EQU 1B 50 1B 20 00 1C 53 00 0C 1C 54 00 00
P12 EQU 1B 4D 1B 20 00 1C 53 00 06 1C 54 00 00
P13 EQU 1B 67 1B 20 01 1C 53 00 03 1C 54 00 01
P15 EQU 1B 67 1B 20 00 1C 53 00 00 1C 54 00 00
```

to the following:

```
P10 EQU 1B 50 1B 20 00 1C 53 06 06 1C 54 00 00
P12 EQU 1B 4D 1B 20 00 1C 53 03 03 1C 54 00 00
P13 EQU 1B 67 1B 20 01 1C 53 02 01 1C 54 00 01
P15 EQU 1B 67 1B 20 00 1C 53 00 00 1C 54 00 00
```

SCS TAB Setting

A PC/3270 printer session LU type 1 can accept any number of tab positions, and the host printer session can send any number of tabs to the printer session. However, the workstation printer you are using might support fewer tab positions than the host application sets; for example, the IBM Proprinter supports 27 tab positions.

If the number of tab positions that the host application sets exceeds the maximum number of positions that the printer supports, your printed output will not look as you expect it to. You can avoid this situation by modifying the PDF file and reconfiguring PC/3270 as follows:

1. Modify the SET_HORIZONTAL_TABS statement as follows:

   `SET_HORIZONTAL_TABS=`*number*

2. Save the file under a new name.
3. Convert the PDF file (with the procedure described on Selecting and Changing the Printer Definition Table (PDT File)).
4. Select the new PDT file created by the above step.

Printer Color Mixing

Some printers, such as the IBM 5182, compose certain colors by mixing colors. Colors are mixed by printing the text in one color and then printing over the same text in another color on a second pass.

PC/3270 will compose a color if the color is not defined in the printer definition table of a color printer that is capable of mixing colors. Therefore, if you are using a printer that composes some colors by mixing two colors, leave the definition of the composed colors blank in the printer definition file. Only the composite colors defined in Table 11 are created by double-printing the primary colors.

Table 11. Color Mixes

Composite Color	Primary Colors
Red	yellow, magenta
Green	yellow, cyan
Blue	magenta, cyan

For example, to create red, you must define yellow and magenta. The primary colors must be defined in the printer definition table.

Printer Session Data Stream Support

There are two types of host-directed print data streams supported by PC/3270: the 3270 data stream and the SNA character string (SCS) data stream.

3270 Data Stream

The 3270 data stream is a buffer-oriented data stream. The print data is formatted as if it were going to be displayed on a screen. The host system sends commands to format the presentation space. These commands can change the presentation space in any location at any time. Once the host system completes formatting the presentation space, it issues a START PRINT command and the presentation space

Building PC/3270 PDTs

is printed as accurately as the printer hardware allows. The 3270 data stream can be sent to PC/3270 workstations over an SNA LU 3 session or a non-SNA DFT printer session.

Table 12 lists the commands that can be sent in the 3270 data stream.

Table 12. 3270 Data Stream Commands

Command	Meaning
W	Write
EW	Erase/Write
EWA	Erase/Write Alternate
RB	Read Buffer
RM	Read Modified
RMA	Read Modified All
EAU	Erase All Unprotected
WSF	Write Structured Field

The RB, RM, and RMA commands cannot be used with an SNA attachment.

Table 13 lists the orders that can be sent in the 3270 data stream.

Table 13. 3270 Data Stream Orders

Order	Meaning
SBA	Start Buffer Address
SF	Start Field
IC	Insert Cursor
PT	Program Tab
RA	Repeat to Address
EUA	Erase Unprotected to Address
SFE	Start Field Extended
SA	Set Attribute
MF	Modify Field

The last three orders in this list manage the color, extended highlighting, and programmed symbols attributes for fields and individual characters. The programmed symbols attribute is not supported by PC/3270.

In addition to the commands and orders in the preceding list, there are special printer formatting control codes that can be included in the 3270 data stream.

Table 14 lists the control codes that can be sent in the 3270 data stream.

Table 14. 3270 Data Stream Format Control Orders

Code	Description
NL	New Line control code moves the print position to the left margin and down one line.
CR	Carriage Return control code moves the print position to the left margin.
EM	End of Message control code ends the print operation.

Table 14. 3270 Data Stream Format Control Orders (continued)

Code	Description
FF	Form Feed control code moves the print position to the left margin at the top of the next page.

Note: NL, CR, and EM are valid only if a line-length format specified by the WCC is not used. The FF code is valid in any buffer position.

PC/3270 printer support interprets each 3270 attribute and printer control code and translates them into a sequence of one or more workstation printer control codes. For more information about the 3270 data stream, refer to *IBM 3270 Information Display Data Stream Programmer's Reference*

SCS Data Stream

The SCS data stream is a sequential data stream that is oriented toward line printers. The characters are translated according to the host system code page and printed according to the attributes and formatting specified by the control commands. The way in which SCS characters are printed depends only on the control codes that precede them. Therefore, it is not necessary to buffer the characters before they are printed. The SCS data stream flows across the SNA LU 1 session only.

The valid SCS control codes supported by PC/3270 are listed in Table 15.

Table 15. Supported SCS Control Codes

Code	Value	Command
NL	X'15'	New Line
LF	X'25'	Linefeed (Index)
CR	X'0D'	Carriage Return
FF	X'0C'	Form Feed
BS	X'16'	Backspace
BEL	X'2F'	Bell Function
IRS	X'1E'	Interchange-Record Separator
SA	X'28'	Set Attribute
TRN	X'35'	Transparent
NUL	X'00'	Null
HT	X'05'	Horizontal Tab
VT	X'0B'	Vertical Tab
SHF	X'2B C1'	Set Horizontal Format
SVF	X'2B C2'	Set Vertical Format
SLD	X'2BC 6'	Set Line Density
SPD	X'2BD2 xx29'	Set Print Density
VCS	X'04 xx'	Vertical Channel Select
ENP	X'14'	Enable Presentation
INP	X'24'	Inhibit Presentation
GE	X'08'	Graphic Escape
PPM	X'2BD2 xx48'	Page Presentation Media

Table 15. Supported SCS Control Codes (continued)

Code	Value	Command
SO	X'0E'	Shift Out
SI	X'0F'	Shift In
RNL	X'06'	Required New Line
RCR	X'06'	Required Carriage Return
SPS	X'09'	Superscript
WUS	X'23'	Word Underscore
IR	X'33'	Index Return
PP	X'34'	Presentation Position
EBS	X'36'	Expanded Backspace
NBS	X'36'	Numeric Backspace
SBS	X'38'	Subscript
IT	X'39'	Indent Tab
RFF	X'3A'	Required Form Feed
RPE	X'3A'	Required Page End
RSP	X'41'	Required Space
NSP	X'E1'	Numeric Space
ESP	X'E1'	Expandable Space
SH	X'CA'	Syllable Hyphen

Note: The ENP and INP commands are accepted, but are ignored, in an LU 1 session.

For more information about the SCS printer data stream, refer to *IBM Systems Network Architecture - Sessions Between Logical Units*

Delimiting Print Jobs

Many print jobs can be sent over a single PC/3270 printer session. PC/3270 allows multiple sessions and applications to share a single workstation printer on a between-jobs basis. PC/3270 needs to know when each print job starts and ends so that printers can be shared properly and begin and end job strings can be sent at the appropriate times. The emulator recognizes a number of different methods of delimiting print jobs:

By Session
> PC/3270 printer support assumes, by default, that all print jobs are delimited by sessions. That is, in the absence of all other indicators, PC/3270 assumes that a print job begins when a printer session is started and ends when it is reset.

Time-Out Interval
> On non-SNA DFT sessions, print jobs can be delimited by a user-specified timeout interval. A print job on a non-SNA DFT printer session begins when the first host-outbound data for that job is received, and ends when no host-outbound data is received for a period of time exceeding the user-specified non-SNA DFT timeout interval. For PC/3270, this interval is specified during configuration.

By Brackets

On an SNA session with an LU 1 or LU 3 host application, PC/3270 can use bracket indicators to delimit print jobs. The emulator opens the printer when it receives the begin bracket, then it processes and prints data. The emulator closes the printer after it receives an end bracket.

When the emulator receives another begin bracket, it again opens a printer and begins processing a print job.

Structured Fields

The host can use structured fields to indicate to the device that a new file is beginning or that the current file is completed. PC/3270 delimits print jobs with Begin of File and End of File structured fields (SF) to perform host-directed printing. Structured fields are described in the next section.

Structured Fields

The host uses Begin of File and End of File structured fields to indicate to a device that a file is beginning or ending.

Begin or End of File Query Reply

The Begin or End of File query reply indicates that a device supports Begin of File and End of File to delineate print jobs. The PC/3270 sends a query reply, as shown in Table 16, to the host in response to a Read Partition General query.

Table 16. Begin/End of File Query Reply Format

Byte	Contents	Description
0–1	X'0005'	The length of this structure
2	X'81'	Query reply
3	QCODE X'9F'	Begin/End of File
4	FLAGS	Reserved; must be set to 0's

Begin/End of File Structured Fields

Begin/End of File structured fields are accepted on either LU 1 or LU 3 sessions. Table 17 shows the format of the Begin/End of File structured fields.

Table 17. Begin/End of File Structured Field Format

Byte	Bit	Contents	Description
0–1		X'0007'	The length of this structure
2–3		X'0F85'	Begin/End of File
4		PID	Partition ID
5		FLAG1[1]	
	0–1	B'00'	Reserved
		B'01'	End of File is being sent
		B'10'	Begin of File is being sent
		B'11'	Reserved
	2–7		Reserved
6		FLAG2	Reserved; must be set to 0's.

:

[1]   This byte indicates whether Begin of File or End of File is being sent.

Processing Begin or End of File Structured Fields

When the Begin or End of File structured fields are used with brackets or non-SNA timeout intervals, the Begin or End of File SFs take precedence over the brackets or timeout intervals in determining when a print job begins or ends. Refer to the following examples:

- Begin or End of File structured fields overriding brackets:

```
Begin Bracket, Begin of File Structured Field, ...Data...,
End Bracket
```

  The device will wait indefinitely until the End of File structured field is received before ending the print job.

- Begin or End of File structured fields overriding non-SNA timeout intervals:

```
Begin of File structured fields, ...Data..., pause > timeout value
```

  The device will wait indefinitely until the End of File structured fields is received before ending the print job.

- Inconsistent use of Begin or End of File structured fields and brackets:

```
Begin Bracket, ...Data1...,
Begin of File Structured Fields,...Data2..,
End of File Structured Fields, ...Data3...,
End Bracket
```

  When you use the Begin of File and End of File structured fields inconsistently with brackets, the results are unpredictable. In the preceding example, the device might process Data1, Data2, and Data3 as separate jobs or combine two or more of them into one file.

For predictable results, each data block must be enclosed by a Begin of File structured field and an End of File structured field. The following example shows three print jobs all delimited by Begin or End of File structured fields:

```
Begin Bracket, Begin of File Structured Field, ...Data1...,
End of File Structured Field,(job1)
Begin of File Structured Field, ...Data2..., End of File Structured Field, (job2)
Begin of File Structured Field, ...Data3..., End of File Structured Field,
End Bracket(job3)
```

PC/3270 always keeps track of brackets and timeout intervals. After the emulator receives a Begin of File structured field, it takes no action on Begin Brackets, End Brackets, or timeout until it receives an End of File structured field. Once a valid End of File SF is processed, the emulator defaults to delimiting jobs by brackets or timeout intervals until it receives the next Begin of File structured field.

Begin or End of File Structured Field Error Conditions

PC/3270 does not accept transmission of data belonging to two separate print jobs in the same chain. To be accepted by the emulator, Begin of File structured fields must be the first structured field of a chain and End of File structured fields must be the last structured field of a chain.

PC/3270 rejects transmission in the following instances:

- The emulator receives an End of File structured field without first receiving a Begin of File structured field.
- The emulator receives a second Begin of File structured field without receiving an intervening End of File structured field.

- The emulator receives a Begin of File structured field that is not the first structured field following a **write structured field** command (LU 2, LU 3, and non-SNA sessions) or a Function Management Header 1 (LU 1 sessions).
- The emulator receives an End of File structured field that is not the last structured field following a **write structured field** command (LU 2, LU 3, and non-SNA sessions) or a Function Management Header 1 (LU 1 sessions).

Processing SCS Data Streams

When processing an SCS data stream, PC/3270 treats Begin or End of File structured fields as follows:

- A Begin of File structured field indicates that all SCS data in the same transmission until an End of File structured field is received is part of a new print job.
- An End of File structured field indicates that any SCS data received in the same chain as the End of File structured field is the last data of the current print job.

Processing 3270 Data Streams

When processing a 3270 data stream, PC/3270 treats Begin or End of File structured fields as follows:

- A Begin of File structured field indicates that the next presentation space print (initiated by a **write type** command with the start print bit turned on in the write control character) is the first in a print job.
- An End of File structured field indicates that the last presentation space print was the last of the current print job. The emulator immediately sends a terminate string to the printer to close the printer session.

If PC/3270 receives a Begin of File structured field and an End of File structured field without at least one presentation space separating them, it ignores the structured fields.

Chapter 10. File Transfer Commands for PC/3270

You can send data files to and receive them from IBM host systems that are running:

CICS/MVS
> Customer Information Control System running under MVS

CICS/VSE
> CICS running under Virtual Storage Extended

MVS/TSO
> Multiple Virtual Storage/Time Sharing Option

OV/MVS
> OfficeVision/MVS

VM/CMS
> Virtual Machine/Conversational Monitor System

For more information on using these commands, select **Send File to Host** from the **Actions** menu and **File Transfer from Command prompt** in the help panel.

File Transfer Methods

You can transfer files in the following ways with PC/3270:
- By selecting **Receive File from Host** from the **Actions** menu or **Send File to Host** from the **Actions** menu of the workstation window
- By using the **SEND** and **RECEIVE** commands at the DOS command prompt
- By using an EHLLAPI application that invokes file transfer
- By using a macro that has send or receive commands as macro statements
- By clicking the **Send** or **Recv** icon on the tool bar

Requirements and Restrictions

Install the file transfer program, IND$FILE (or PCT/400 for 5250 sessions), on your host system. Ask your system administrator for additional file transfer procedures and precautions. An alternate host command name can be used by defining a DOS environment variable IND$FILE in **AUTOEXEC.BAT** or in a particular DOS box. For example:

```
SET IND$FILE = MYXFER
```

Similarly, for DBCS sessions, you can define a DOS environment variable APVUFILE in **AUTOEXEC.BAT** or in a particular DOS box. For example:

```
SET APVUFILE = MYXFER
```

You should not use the following words as a VM file name or file type, as an MVS data set name, or as a CICS file name, because they are reserved for use as option commands.

> ASCII, APPEND, TIME, CLEAR, NOCLEAR, SILENT, QUIET, PROGRESS, JISCII, SO, NOSO, BLANK, USER, CRLF, BINARY, NOCRLF

If you want to send to or receive from a subdirectory other than **\Personal Communications**, you must specify the full path name.

Sending and Receiving Files from the DOS Command Prompt

The workstation is the point of reference for the SEND and RECEIVE commands: You send from the workstation to the host and receive from the host to the workstation.

To send or receive a file:

1. Make sure you are logged on to your host.
2. Make sure the **Ready** message of the host system is displayed, except if you are transferring files through the command option of the ISPF application.

   Note: In the latter case, you must specify the NOCLEAR option for the file transfer command.

   If your screen is blank, make sure that no applications are running and that your host session is not in a *holding* state.

   Note: If you receive any messages from host application programs while you are transferring files, the transfer might not succeed. To prevent messages from interfering, enter the appropriate host command to set messages off temporarily. When file transfer is finished, set messages on again.

3. Switch to your DOS window session or DOS full-screen session.
4. If you use a hard disk, make sure the **SEND.EXE** and **RECEIVE.EXE** files are in your current directory or in your path. If you want to send to or receive from a subdirectory other than **\Personal Communications**, you must specify the full path name.
5. Type the appropriate SEND or RECEIVE command at the DOS command prompt.

   Details on the SEND and RECEIVE commands and their options are explained in the following sections.

Using the VM/CMS SEND Command

Use the following information when sending a file to VM/CMS:

Figure 1 shows the command and information that you must provide. Enter it as shown (including parentheses). You can use either uppercase or lowercase letters.

Notes:

1. `Blank` means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **(options**.
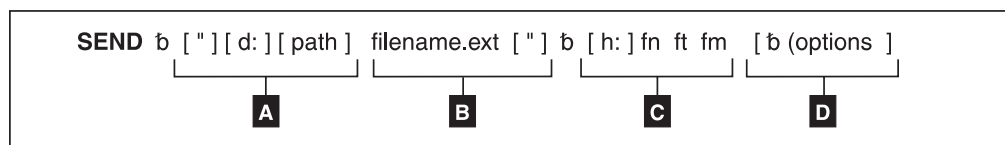
Figure 1. VM/CMS SEND Command Syntax

A   The workstation drive and path of the file to send.

B   The name of the workstation file to send.

C   Host session specifications for the file to be sent to the host.

h: The short name of the session (which can be omitted if it is **a**)

fn File name

ft File type

fm File mode

D Optional changes made to the file during transfer. More than one option can be selected. Valid options are:
- APPEND
- ASCII
- CLEAR
- CRLF
- JISCII (for Japanese DBCS sessions)
- LRECL n
- NOCLEAR
- NOSO
- PROGRESS
- QUIET
- RECFM x
- TIME(n)

The parts of the VM/CMS SEND command are:

SEND The command.

d: The name of the diskette or hard disk drive on which the file is located.

path The path to the subdirectory that the file is in.

filename.ext
The name of the file to be sent, including the extension.

h: The short name of the host session to which you want to send the file. The default is `a:`.

fn ft fm
The name the file is to have on your VM/CMS disk. You must specify the file name (`fn`) and file type (`ft`). You can omit file mode (`fm`) if you want the file placed on your A-disk. You can create a new name or use a name that is already on your disk. If you use a new name, the file that you send is added to your disk. If you use the name of an existing file, the file that you send either replaces or is added to the old file. (Refer to the description of the APPEND option.)

(options or **[(options (for DBCS)**
These options can be specified:

APPEND
Specifies that the file being sent is to be added to the end of an existing VM/CMS file. Omit this option if you want the file to replace an existing file. You cannot specify the LRECL n or RECFM x option if you use the APPEND option.

ASCII Performs the following:
- Converts 1-byte workstation codes to EBCDIC codes.
- Converts 2-byte codes to IBM host DBCS codes.
- Inserts SO/SI characters into the DBCS field and, if control characters (→ or ←) are found, converts the control characters to SO/SI characters.

CLEAR

Clears the workstation window at the beginning of the file transfer. **CLEAR** is the default.

CRLF Specifies preserving of the carriage return and line feed codes. You need the ASCII and CRLF options for text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files, such as programs.

JISCII This option is valid for Japanese DBCS sessions only and performs the following:

- Converts 1-byte workstation codes to EBCDIC codes.
- Converts 2-byte codes to IBM kanji codes.
- Inserts SO/SI characters into the DBCS field and, if control characters (→ or ←) are found, converts the control characters to SO/SI characters.

LRECL n

Specifies the file's record length. Include a record length only if you want the file to have a record length on your VM/CMS disk other than 80. Replace n with the record length you want. If you omit this option, the record length is set to 80 for fixed-length records or to a maximum of 80 for variable-length records.

NOCLEAR

Suppresses the sending of a Clear command at the beginning of the file transfer.

NOSO

This option is valid only for DBCS when used with the JISCII or ASCII option and performs the following:

- Does not convert SO (hex 0E) and SI (hex 0F) before and after the DBCS field.
- Does not convert RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F).

PROGRESS

Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.

QUIET

Does not show any messages.

RECFM x

Specifies the file record format. Use this parameter to specify variable-length or fixed-length records in the file. Replace x with V for variable or F for fixed. By default, the file has fixed-length records unless you specify the CRLF option; then the file has variable-length records unless you specify otherwise.

TIME(n)

Specifies the length of time n, in units of 30 seconds, that the program waits for a response from the host before it sends an error message. Replace n with an integer value in the range from 0 through 2184. If you specify 0, timeout will not be set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, of large block sizes, or for

slow communication lines (such as SDLC and COM port), *5* (150 seconds) is recommended. There should be no blank spaces is between TIME and (n).

Examples

The following examples show the command syntax you can use to send files. The parameters of the SEND command can be combined into a single set of parentheses.

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To send a workstation file from your default drive and add it as a new file on your VM/CMS A-disk:

  `SEND pc.txt a:cmsfile script a (ASCII CRLF LRECL 72 RECFM V`

  `SEND pc.txt a:cmsfile script a [(JISCII CRLF LRECL 72 RECFM V` (for DBCS)

  Note: If you use a command that exceeds one line, do not press Enter when you fill that line; continue typing your command.

  This command sends a workstation file named **PC.TXT** from your default drive to your host in your host session named **a**. You do not need to specify the workstation drive if the file you are sending is on the current drive. The command creates a new file, named **CMSFILE SCRIPT**, on your A-disk. The records in the file can vary in length up to 72 characters.

- To send a workstation file from your default drive to replace a file on your VM/CMS A-disk:

  `SEND pc.txt a:cmsfile script a (ASCII CRLF`

  `SEND pc.txt a:cmsfile script a [(JISCII CRLF` (for DBCS)

  This command sends a workstation file named **PC.TXT** from your default drive to your VM/CMS A-disk in your host session named **a**. You do not need to name the workstation drive if the file you are sending is on the default drive. The file replaces a **SCRIPT** file named **CMSFILE**. The new **CMSFILE** has the same record length and format as the old **CMSFILE**.

  If you do not have a file called **CMSFILE SCRIPT** on your A-disk, **PC.TXT** is added to your A-disk as a new file called **CMSFILE SCRIPT**. The records in the file are 80 characters long and have fixed length.

- To send a binary workstation file from a drive other than your default drive:

  `SEND a:pc.exe c:cmsfile exebin b (recfm v`

  This command sends a workstation file named **PC.EXE** from a diskette in drive A to your VM/CMS B-disk in your host session named **c**. It is a new file, or it replaces a file named **CMSFILE**.

  When transferring a binary file, you must specify a variable record format (**recfm v**), otherwise, blank characters are added to the file.

- To send a file from your hard disk and add it to the end of a file on your VM/CMS A-disk:

  `SEND c:pc.txt cmsfile script a (ASCII CRLF APPEND`

  `SEND c:pc.txt cmsfile script a [(JISCII CRLF APPEND` (for DBCS)

File Transfer Commands for PC/3270

This command sends a workstation file named **PC.TXT** from your hard disk to your host session. You do not need to name the host session if you are sending to the **a** session. The file is added to the end of a script file named **CMSFILE** on your VM/CMS A-disk.

- To send a file from a subdirectory on your hard disk to your VM/CMS A-disk:

```
SEND c:\sd1\pc.txt cmsfile script a (ASCII CRLF
SEND c:\sd1\pc.txt cmsfile script a [(JISCII CRLF (for DBCS)
```

This command sends a file named **PC.TXT** from subdirectory **SD1** on your hard disk to your host session. It replaces a **SCRIPT** file named **CMSFILE** on your VM/CMS A-disk.

Using the VM/CMS RECEIVE Command

Use the following information when receiving a file from VM/CMS:

Figure 2 shows the command and information you must provide. Enter it as shown (including parentheses), except that you can use either uppercase or lowercase letters.

Notes:

1. ƀ means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **(options**.

```
RECEIVE  ƀ  [ " ] [ d: ] [ path ]   filename.ext  [ " ]  ƀ  [ h: ] fn  ft  fm    [ ƀ (options  ]
                    └──────────A──────────┘   └────────B────────┘   └────────C────────┘   └──────D──────┘
```
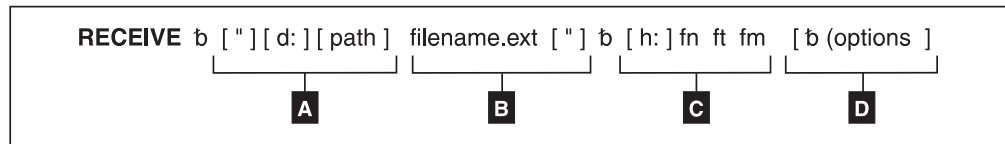
Figure 2. VM/CMS RECEIVE Command Syntax

A   The workstation drive and path of the file to be received.

B   The name of the workstation file to be received.

C   Host session specifications for the file to be received from the host.
- **h:**   The short name of the session (which can be omitted if it is **a**)
- **fn**   File name
- **ft**   File type
- **fm**   File mode

D   Optional changes made to the file during transfer. More than one option can be selected. Valid options are:
- APPEND
- ASCII
- BLANK
- CLEAR
- CRLF
- JISCII (for Japanese DBCS sessions)
- NOCLEAR
- PROGRESS
- QUIET
- SO
- TIME(n)
- USER

The parts of the VM/CMS RECEIVE command are:

RECEIVE
>   The command.

d: The name of the diskette or hard disk drive on which the file is to be received.

path   The path indicating the directory to which the file is to be stored.

filename.ext
>   The name of the workstation file, including the extension. Use a new name or one that already exists. If you use a new name, the file that you receive is added to your diskette or hard disk. If you use the name of an existing file, the file that you receive either replaces or supplements the existing file. (Refer to the APPEND option.)

h: The short name of the host session from which you want to get the file. The default is `a:`.

fn ft fm
>   The name of the file you want to receive from your VM/CMS disk. The file name `fn` is required.

(options or [(options (for DBCS)
>   These options can be specified:

>   **APPEND**
>   >   Specifies that the file being received is to be added to the end of an existing file. Omit this part of the VM/CMS file that is received to replace an existing file.

>   **ASCII**   Performs the following:
>   - Converts EBCDIC codes to 1-byte workstation codes.
>   - Converts IBM host DBCS codes to 2-byte codes.
>   - Removes SO and SI characters from a DBCS field.

>   **BLANK**
>   >   This option is valid with the CRLF option. Use it to retain BLANK (x'40') at the end of each line.

>   **CRLF**   Specifies the carriage return and line feed codes. You need ASCII and CRLF for text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files, such as programs.

>   **CLEAR**
>   >   Clears the workstation window at the beginning of the file transfer.

>   **JISCII** This option is valid for Japanese DBCS sessions only and performs the following:
>   - Converts EBCDIC codes to 1-byte workstation codes.
>   - Converts IBM kanji codes to 2-byte codes.
>   - Removes SO and SI characters from a DBCS field.

>   **NOCLEAR**
>   >   Suppresses the sending of a Clear command at the beginning of the file transfer.

PROGRESS

Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.

QUIET

Does not show any messages.

SO This option is valid only for DBCS when used with the JISCII or ASCII option; it converts SO (hex 0E) and SI (hex 0F) to RS (hex 1E) and US (hex 1F).

TIME(n)

Specifies the length of time, in units of 30 seconds, that the program waits for a response from the host before it sends an error message. The value n is an integer value in the range from 0 through 2184. If you specify 0, timeout is not set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, of large block sizes, or for slow communication lines (such as SDLC and COM port), *5* (150 seconds) is recommended. There should be no blank spaces between TIME and *(n)*.

USER This option is valid only for DBCS with the options JISCII, ASCII, and SO. SO (hex 0E) and SI (hex 0F) are not converted to RS (hex 1E) and US (hex 1F); they are written to a file without being converted.

Examples

The following examples show the command syntax you can use to receive files from a VM/CMS host. The parameters of the RECEIVE command can be combined into a single set of parentheses.

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To receive a file from your VM/CMS A-disk to your default drive for a workstationsession:

  ```
  RECEIVE pc.txt a:cmsfile script a (ASCII CRLF
  RECEIVE pc.txt a:cmsfile script a [(JISCII CRLF (for DBCS)
  ```

  This command sends a **SCRIPT** file **CMSFILE** from your VM/CMS A-disk in a host session named **A** to your workstation session. It adds the file to your default drive (diskette or hard disk) with the name **PC.TXT**.

- To receive a file from your VM/CMS B-disk and replace a file on a drive other than your default:

  ```
  RECEIVE a:pc.txt a:cmsfile script b (ASCII CRLF
  RECEIVE a:pc.txt a:cmsfile script b [(JISCII CRLF (for DBCS)
  ```

  This command sends a **SCRIPT** file named **CMSFILE SCRIPT** from your VM/CMS B-disk in a host session named **A** to a drive other than the default for your PC session. It replaces a file named **PC.TXT** on a diskette in drive A.

- To receive a file from your VM/CMS A-disk and add it to the end of a file on your hard disk:

  ```
  RECEIVE c:pc.txt a:cmsfile script a (ASCII CRLF APPEND
  RECEIVE c:pc.txt a:cmsfile script a [(JISCII CRLF APPEND (for DBCS)
  ```

This command sends a **SCRIPT** file named **CMSFILE SCRIPT** from your VM/CMS A-disk in a host session named **A** to your workstation session. It adds the contents of **CMSFILE** to the end of a file named **PC.TXT** on your hard disk.

- To receive a file from your VM/CMS A-disk and place it in a subdirectory on your default drive:

  `RECEIVE \sd1\pc.txt a:cmsfile script a (ASCII CRLF`

  `RECEIVE \sd1\pc.txt a:cmsfile script a [(JISCII CRLF` (for DBCS)

  This command sends a **SCRIPT** file named **CMSFILE SCRIPT** from your VM/CMS A-disk to your default drive. It creates or replaces a file named **PC.TXT** in a subdirectory named **\SD1**.

Using the MVS/TSO SEND Command

Use the following information when entering the SEND command to the MVS/TSO host:

Figure 3 shows the command and information you must provide. Enter text as shown (including parentheses), except that you can use either uppercase or lowercase letters.

Notes:

1. Þ means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **options**.

```
SEND Þ [ " ] [ d: ] [ path ]  filename.ext  [ " ]  Þ  [ h: ] data-set-name
           |_____|  |_____|       |_____|
                   A                   B                       C

         [ Þ (member-name) ]    [ /password ]    [ Þ  options ]
         |_____|    |_____|    |_____|
                  D                   E                F
```
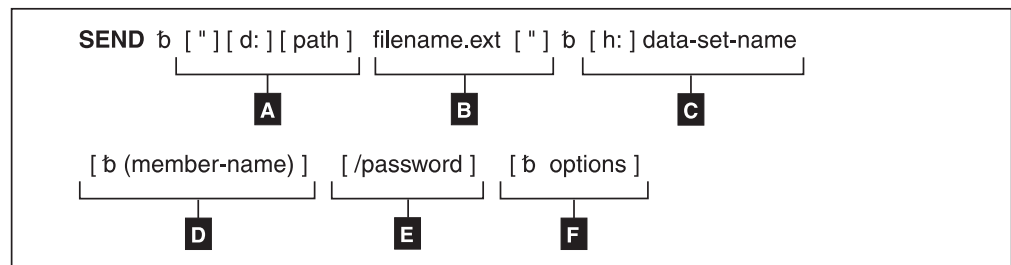
Figure 3. MVS/TSO SEND Command Syntax

A   The workstation drive and path of the file to send.

B   The name of the workstation file to send.

C   The short name of the host session and the data set name of the file to send.

D   The member name if the file is put in a partitioned data set.

E   The password of the data set if it has one.

F   Optional changes made to the file during transfer. More than one option can be specified. Valid options are:
- APPEND
- ASCII
- BLKSIZE(n)
- CLEAR
- CRLF
- JISCII (for Japanese DBCS sessions)
- LRECL(n)

File Transfer Commands for PC/3270

- NOCLEAR [1]
- NOSO
- PROGRESS
- QUIET
- RECFM(x)
- SPACE(n[,n1]) unit
- TIME(n)

The parts of the MVS/TSO SEND command are:

SEND The command.

d: The name of the diskette or hard disk drive where the file is located.

path The path indicating the directory where the file is located.

filename.ext
The name of the file to be sent. Include the extension if the file has one.

h: The name of the MVS/TSO host session to which you want to send the file. You can omit this name if you have only one host. If you have more than one host, this is the short name of the MVS/TSO host session. The default short name is **A**.

data-set-name
The data set name that the file you send is to have on your MVS/TSO volume; this name is required. Enclose the data set name with the member name in single quotation marks if you are using a fully qualified data set name.

This option creates a new name or uses a data set name already on your TSO volume. If you use a new name, the file that you send is added to your MVS/TSO volume. If you use the name of an existing data set, the file you send either replaces or supplements the existing data set. Refer to the APPEND option.

(member-name)
The member name if the file is to be put into a partitioned data set. If you use member-name, you cannot use LRECL(n), BLKSIZE(n), RECFM(x), and SPACE(n,[n1]) unit.

Note: If someone else is using the partitioned data set, you cannot send a file to your MVS/TSO host.

/password
The password of the data set, if the data set has a password.

options or [(options (for DBCS)
These options can be specified:

APPEND
Specifies that the file being sent is added to the end of an existing MVS/TSO data set. Omit this option if you want the file to replace an existing MVS/TSO data set. You cannot use LRECL(n), RECFM(x), SPACE(n[,n1]) unit, or BLKSIZE(n) options if you use the APPEND option.

ASCII Performs the following:

- Converts 1-byte workstation codes to EBCDIC codes.

---

1. You must use the NOCLEAR option when you are transferring files while in ISPF command mode on the host.

- Converts 2-byte codes to IBM host DBCS codes.
- Inserts SO and SI characters in the DBCS field.

BLKSIZE(n)

Specifies the size of the blocks of data in a new data set on your MVS/TSO volume. This part is optional. To set the block size for a new data set, replace `n` with the new size. If you omit this option, the block size is determined in the following manner:

- If the record format is variable, the block size is 6233.
- If the record format is fixed, the block size is the largest multiple of the record length that is less than 6233:

```
BLKSIZE = LRECL * (6233/LRECL)
```

If you use the `(member-name)` or APPEND option, do not use this option.

CLEAR

Clears the workstation window at the beginning of the file transfer.

CRLF   Specifies the global use of carriage return and line feed codes. You need to specify ASCII and CRLF options for sending text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files.

JISCII   This option is valid for Japanese DBCS sessions only and performs the following:

- Converts 1-byte workstation codes to EBCDIC codes.
- Converts 2-byte codes to IBM kanji codes.
- Inserts SO and SI characters in the DBCS field.

LRECL(n)

Specifies the record length for a new data set on your MVS/TSO volume, where `n` is a whole number from 1 through 32760 representing the number of characters per record. If you want to set the record length for a new data set, replace `n` with the new length. If you omit this option, the record length is set to 80 for fixed-length records and to 255 for variable-length records. If you use the `(member-name)` or APPEND options, do not use this option.

NOCLEAR

Suppresses the sending of a Clear command at the beginning of the file transfer. This option is required for ISPF command mode.

NOSO

This option is valid only for DBCS when used with the JISCII or ASCII option and performs the following:

- Does not insert SO (hex 0E) and SI (hex 0F) before and after a DBCS field.
- Does not convert RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F).

PROGRESS

Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.

QUIET

Does not show any messages.

RECFM(x)

Specifies the record format for a new data set on your MVS/TSO volume, where x = V, F, or U. For variable-, fixed- or undefined-length records in the data set, replace the x with V, F, or U, respectively.

If you omit this option, the record format of the host data set is determined by the setting of the CRLF parameter: if you specify CRLF, the data set has variable-length records; if you do not specify CRLF, it has fixed-length records. If you use the (member-name) or APPEND options, do not use this option.

SPACE(n[,n1]) unit

Specifies an amount of space to be set aside for a new data set on your MVS/TSO volume. To set aside a certain number of blocks, tracks, or cylinders for the new data set:

- Provide unit as the type of space you want (AVBLOCK, TRACKS, or CYLINDERS).
- Give n as the amount of space that you want the data set to occupy (in the unit of measure you select).
- If the data set needs more space than you ask for with n, give n,n1 where n1 is the size of additional space to be used only when necessary.

These values are similar to the values on the ALLOCATE command of MVS/TSO.

If you omit this option, you get space for one block. The length of the block is set by the BLKSIZE(n) or LRECL(n) options. If you use the (member-name) or APPEND options, do not use this option.

TIME(n)

Specifies the length of time, in units of 30 seconds, that the program waits for a response from the host before it sends an error message. The value n is an integer value in the range from 0 through 2184. If you specify 0, timeout is not set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, of large block sizes, or for slow communication lines (such as SDLC and COM port), 5 (150 seconds) is recommended. There should be no blank spaces between TIME and *(n)*.

Examples

The following examples show the command syntax you can use to send files from your workstation to an MVS/TSO host:

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To send a file from your default drive to replace a file on the MVS/TSO host:

  SEND pc.txt g:ds.script ASCII CRLF

  SEND pc.txt g:ds.script [JISCII CRLF (for DBCS)

  This command sends a workstation file named **PC.TXT** from your default drive to your MVS/TSO host in a host session named **G**. It creates or replaces a data set named **DS.SCRIPT** on your MVS/TSO volume.

- To send a file from a drive other than the default to your MVS/TSO host:

```
SEND a:pc.txt g:ds.script ASCII CRLF
SEND a:pc.txt g:ds.script [JISCII CRLF (for DBCS)
```

This command sends a workstation file named **PC.TXT** from a diskette in drive A to your MVS/TSO host in a host session named **G**. It replaces a data set named **DS.SCRIPT** on your MVS/TSO volume.

- To send a file from your default drive to your MVS/TSO host and add it to the end of an MVS/TSO data set:

```
SEND a:pc.txt g:ds.script ASCII CRLF APPEND
SEND a:pc.txt g:ds.script [JISCII CRLF APPEND (for DBCS)
```

This command sends a workstation file named **PC.TXT** from a diskette in drive A to your MVS/TSO host in a host session named **G**. It adds the file to the end of a data set named **DS.SCRIPT** on your MVS/TSO volume.

- To send a file to your MVS/TSO host and add it to the end of a data set that has a password:

```
SEND a:pc.txt g:ds.script/odyssey8 ASCII CRLF APPEND
SEND a:pc.txt g:ds.script/odyssey8 [JISCII CRLF APPEND (for DBCS)
```

This command sends a workstation file named **PC.TXT** from a diskette in drive A to your MVS/TSO host in a host session named **G**. It adds the file to the end of a data set named **DS.SCRIPT** on your MVS/TSO volume. This data set has a password of odyssey8.

- To send a file from a subdirectory on your hard disk to a partitioned data set on your MVS/TSO host:

```
SEND c:\sd1\pc.txt g:ds.script (m1) ASCII CRLF
SEND c:\sd1\pc.txt g:ds.script (m1) [JISCII CRLF (for DBCS)
```

This command sends a workstation file named **PC.TXT** from a subdirectory named **\SD1** on your hard disk to your MVS/TSO host in a host session named **G**. It creates or replaces a member named **M1** in a partitioned data set named **DS.SCRIPT** on your MVS/TSO volume.

- To send a file to a partitioned data set that has a password:

```
SEND a:pc.txt g:ds.script (m2)/ili11 ASCII CRLF APPEND
SEND a:pc.txt g:ds.script (m2)/ili11 [JISCII CRLF APPEND (for DBCS)
```

This command sends a workstation file named **PC.TXT** from a diskette in drive A to your MVS/TSO host in a host session named **G**. It adds the file as a member named **M2** to a partitioned data set named **DS.SCRIPT** on your MVS/TSO volume. In this example, the data set has a password of ili11.

- To send a file from your default drive and add it as a new data set on your MVS/TSO volume:

```
SEND pc.txt g:ds.script/aeneid20 ASCII CRLF LRECL(132)
     BLKSIZE(132) RECFM(V) SPACE(20,10) TRACKS
SEND pc.txt g:ds.script/aeneid20 [JISCII CRLF LRECL(132)
     BLKSIZE(132) RECFM(V) SPACE(20,10) TRACKS (for DBCS)
```

This command sends a workstation file named **PC.TXT** from your default drive to your MVS/TSO host. It adds the file as a new data set named **DS.SCRIPT** on your MVS/TSO volume. A password of aeneid20 is assigned. The records in the data set can vary in length up to 132 characters. Data blocks are the same length as the records. Twenty tracks are set aside for this data set. If more tracks are needed, they are added in groups of 10.

Using the MVS/TSO RECEIVE Command

Use the following information when receiving a file from MVS/TSO:

Figure 4 shows the command and information you must provide. Enter it as shown (including parentheses), except that you can use either uppercase or lowercase letters.

Notes:

1. ƀ means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **options**.

```
RECEIVE  ƀ  [ " ] [ d: ] [ path ]   filename.ext  [ " ]  ƀ  [ h: ] data-set-name
                     └─────┬─────┘   └──────┬──────┘        └────────┬────────┘
                           A                 B                       C

         [ ƀ (member-name) ]    [ /password ]    [ ƀ  options ]
         └─────────┬─────────┘  └──────┬──────┘  └──────┬──────┘
                   D                    E                F
```
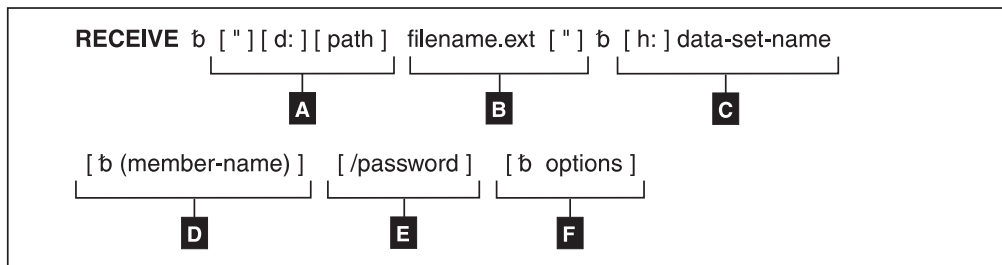
Figure 4. MVS/TSO RECEIVE Command Syntax

A   The workstation drive and path to the directory where the file is to be stored.

B   The name of the workstation file to receive.

C   The short name of the host session, and the data set name of the file you are receiving.

D   The member name if the file is put in a partitioned data set.

E   The password of the data set, if any.

F   Optional changes made to the file during transfer. More than one option can be specified. Valid options are:
   - APPEND
   - ASCII
   - BLANK
   - CRLF
   - JISCII (for Japanese DBCS sessions)
   - PROGRESS
   - QUIET
   - SO
   - TIME(n)
   - USER

The parts of the MVS/TSO RECEIVE commands are:

RECEIVE
        The command.

d: The name of the diskette or hard disk drive where the file is to be located. Use **A:**, **B:**, **C:**, **D:** through **Z:**. This part is optional if the file is received on the current drive.

path   The subdirectory where you want the data set located. This part is optional.

filename.ext
> The name the file is to have on your diskette or hard disk. Creates a new name or uses a name that is already on your diskette or hard disk.
>
> If you use a new name, the data set that you receive is added to your diskette or hard disk. If you use the name of an existing file, the data set that you receive either replaces or supplements the existing file. (Refer to the APPEND option on page 85.)

h:
> The short name of the MVS/TSO session where the data set is located. If you have only one host, this part is optional. Use this option if you have more than one host. The default short name is **A**.

data-set-name
> The name of the data set or the partitioned data set that contains the member you want to send to your workstation session. You must use the qualified name. Enclose the data set name with the member name in single quotation marks if you are using a fully qualified data set name.

(member-name)
> The member name of a partitioned data set to send to your workstation session. This part is optional. Use it only if the data set is a member of a partitioned data set.

/password
> The password of the data set. Use it only if the data set has a password.

(options or [(options (for DBCS)
> These options can be specified:
>
> **APPEND**
> > Adds the data set to the end of an existing file. Omit this part if you want the MVS/TSO data set to replace an existing workstation file.
>
> **ASCII** Performs the following:
> > - Converts EBCDIC codes to 1-byte workstation codes.
> > - Converts IBM host DBCS codes to 2-byte codes.
> > - Removes SO and SI characters from a DBCS field.
>
> **BLANK**
> > This option is valid with the option CRLF; it retains BLANK (hex 40) at the end of each line.
>
> **CRLF** Specifies the use of carriage return and line feed codes. You need ASCII and CRLF for text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files.
>
> **JISCII** This option is valid for Japanese DBCS sessions only and performs the following:
> > - Converts EBCDIC codes to 1-byte workstation codes.
> > - Converts IBM kanji codes to 2-byte codes.
> > - Removes SO and SI characters from a DBCS field.
>
> **PROGRESS**
> > Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.
>
> **QUIET**
> > Does not show any messages.

> **SO** This option is valid only for DBCS when used with the JISCII or ASCII option; it converts SO (hex 0E) and SI (hex 0F) to RS (hex 1E) and US (hex 1F).
>
> **TIME(n)**
> Specifies the length of time, in units of 30 seconds, the program waits for a response from the host before it sends an error message. Replace n with an integer value in the range from 0 through 2184. If you specify 0, timeout is not set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, of large block sizes, or for slow communication lines (such as SDLC and COM port), *5* (150 seconds) is recommended. There should be no blank spaces between TIME and (n).
>
> **USER** This option is valid only for DBCS when used with the JISCII, ASCII and SO options. SO (hex 0E) and SI (hex 0F) are not converted to RS (hex 1E) and US (hex 1F); they are written to a file without being converted.

Examples

The following examples show the command syntax you can use to receive files from your MVS/TSO host to your workstation:

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To receive a data set from an MVS/TSO host to the default drive for your workstation session:

  `RECEIVE pc.txt g:ds.script ASCII CRLF`

  `RECEIVE pc.txt g:ds.script [JISCII CRLF` (for DBCS)

  This command sends a data set named **DS.SCRIPT** from your MVS/TSO volume in a host session named **G** to your OS/2 session. It creates or replaces the file on the default drive with the name **PC.TXT**.

- To receive a data set from an MVS/TSO host to a drive other than your default drive:

  `RECEIVE A:pc.txt g:ds.script ASCII CRLF`

  `RECEIVE A:pc.txt g:ds.script [JISCII CRLF` (for DBCS)

  This command sends a data set named **DS.SCRIPT** from your MVS/TSO volume in a host session named **G**. It replaces a file named **PC.TXT** on a diskette in drive A.

- To receive a data set from an MVS/TSO host and add it to a workstation file:

  `RECEIVE a:pc.txt g:ds.script ASCII CRLF APPEND`

  `RECEIVE a:pc.txt g:ds.script [JISCII CRLF APPEND` (for DBCS)

  This command sends a data set named **DS.SCRIPT** from your MVS/TSO volume in a host session named **G**. It adds the data set to the end of a file named **PC.TXT** on the diskette in drive A.

- To receive a data set from an MVS/TSO host and place it in a subdirectory on your hard disk:

  `RECEIVE c:\sd1\pc.txt ds.script ASCII CRLF`

  `RECEIVE c:\sd1\pc.txt ds.script [JISCII CRLF` (for DBCS)

This command sends a data set named **DS.SCRIPT** from your MVS/TSO volume in a host session named **G**. It creates or replaces a file named **PC.TXT** in a subdirectory named **\SD1** on your hard disk.

- To receive a data set that has a password from an MVS/TSO host to your default drive:

  `RECEIVE A:pc.txt g:ds.script/odyssey8 ASCII CRLF APPEND`

  `RECEIVE A:pc.txt g:ds.script/odyssey8 [JISCII CRLF APPEND (for DBCS)`

  This command sends a data set named **DS.SCRIPT** from your MVS/TSO volume in a host session named **G**. The data set has the password `odyssey8`. The data set is added to the end of a file named **PC.TXT** on the diskette in drive A.

- To receive a member of a partitioned data set from an MVS/TSO host to your DOS session:

  `RECEIVE c:\sd1\pc.txt g:ds.script (m1) ASCII CRLF`

  `RECEIVE c:\sd1\pc.txt g:ds.script (m1) [JISCII CRLF (for DBCS)`

  This command sends a member named **M1** from a partitioned data set named **DS.SCRIPT** in a host session named **G**. The member is placed on your hard disk in a subdirectory named **\SD1**. It replaces or creates a file named **PC.TXT**.

- To receive a member of a partitioned data set that has a password to your Windows session:

  `RECEIVE a:pc.txt g:ds.script (m2)/ili1 ASCII CRLF APPEND`

  `RECEIVE a:pc.txt g:ds.script (m2)/ili1 [JISCII CRLF APPEND (for DBCS)`

  This command sends a member named **M2** from a partitioned data set named **DS.SCRIPT** in a host session named **G**. The data set has a password of `ili1`. The member is added to a file named **PC.TXT** on the diskette in drive A.

Using the CICS SEND Command

Use the following information when sending a file to CICS:

Figure 5 shows the command and information you must provide. Enter it as shown (including parentheses), except that you can use either uppercase or lowercase letters.

Notes:

1. ƀ means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **(options**.
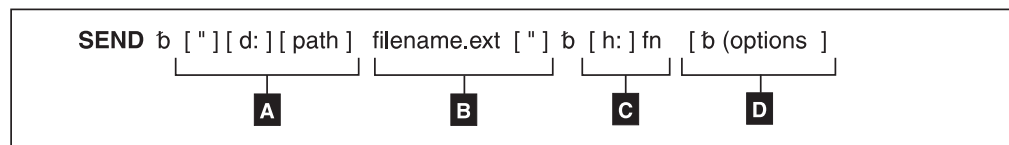
Figure 5. CICS SEND Command Syntax

A   The workstation drive and path of the file to send.

B   The name of the workstation file to send.

C   The short name of the host session, and the host file name of the file to send.

File Transfer Commands for PC/3270

D   Optional changes made to the file during transfer. More than one option can be specified. Valid options are:
- ASCII
- BINARY (for SBCS sessions)
- CLEAR
- CRLF
- JISCII (for Japanese DBCS sessions)
- NOCLEAR
- NOCRLF (for SBCS sessions)
- NOSO
- PROGRESS
- QUIET
- TIME(n)

> **Note:** For SBCS sessions, the default options are ASCII and CRLF; for DBCS sessions, the default option is BINARY.

The parts of the CICS SEND command are:

SEND   The command.

d: The name of the diskette or hard disk drive where the file is located.

path   The path to the subdirectory that the file is in.

filename.ext
   The name of the file to be sent, including the extension.

h: The short name of the host session where you want to send the file. If you have only one host, this part is optional. The default is session **A**.

fn   The name the file is to have on your CICS disk. You must specify the file name. You can create a new name or use a name that is already on the disk.

(options or [(options (for DBCS)
   These options can be specified:

ASCII   Performs the following:
- Converts 1-byte workstation codes to EBCDIC codes.
- Converts 2-byte codes to IBM host DBCS codes.
- Inserts SO and SI characters in a DBCS field and converts the control characters ← and →, if they are present, to SO and SI characters.

The default is ASCII CRLF. You need these control terms for text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files.

Notes:
1. CRLF and NOCRLF are mutually exclusive options.
2. BINARY and ASCII are mutually exclusive options.
3. The assumed defaults, if the optional parameters are omitted, are CRLF ASCII.

BINARY
   Specifies that the data in the file is binary data. The data can be

encrypted, compiled programs, or other data. It is not translated by the host file transfer program but copied unaltered into a temporary storage queue.

This option is valid for SBCS sessions only.

CLEAR
Clears the workstation window at the beginning of the file transfer.

CRLF Specifies carriage return and line feed codes in the text file.

JISCII This option is valid for Japanese DBCS sessions only and performs the following:
- Converts 1-byte workstation codes to EBCDIC codes.
- Converts 2-byte codes to IBM kanji codes.
- Inserts SO and SI characters in a DBCS field and converts the control characters ⊣ and →, if they are present, to SO and SI characters.

NOCLEAR
Suppresses the sending of a Clear command at the beginning of file transfer. This option is required for ISPF command mode.

NOCRLF
Specifies that the PC file does not consist of logical records delimited by carriage return and line feed characters. No concatenation or splitting of records is performed by the CICS file transfer program.

The file is written into a temporary storage using one item on the queue to represent each inbound data buffer. The items on the CICS temporary storage queue can be of different lengths, but none can be more than 32767 characters.

This option is valid for SBCS sessions only.

NOSO
This option is valid only for DBCS when used with the JISCII or ASCII option and performs the following:
- Does not insert SO (hex 0E) and SI (hex 0F) before and after a DBCS field.
- Does not convert RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F).

PROGRESS
Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.

QUIET
Does not show any messages.

TIME(n)
Specifies the length of time, in units of 30 seconds, the program waits for a response from the host before it sends an error message. Replace n with an integer value in the range from 0 through 2184. If you specify 0, timeout is not set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, large block sizes, or for slow

communication lines (such as SDLC and COM port), *5* (150 seconds) is recommended. There should be no blank spaces between TIME and (n).

Examples

The following examples show the command syntax you can use to send files from your workstation to your CICS host.

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To send a workstation file from your default drive and add it as a new file on your CICS host:

  SEND pc.txt a:cicsfile (ASCII CRLF)

  SEND pc.txt a:cicsfile [(JISCII CRLF) (for DBCS)

  Note: Enter the complete CICS SEND command on one line.

  This command sends a workstation file named **PC.TXT** from your default drive on your workstation to your host session **A**. You do not need to provide the workstation drive name if the file you are sending is on the current drive. The command creates a new file named **CICSFILE**.

- To send a basic workstation file from a drive other than your default to replace a file on your CICS host:

  SEND a:myprog.exe a:basprog

  This command sends a workstation file named **MYPROG.EXE** from a diskette in drive A to your CICS host in your host session named **A**. It is written to a file named **BASPROG**, replacing any existing file by that name in host session **A**.

Using the CICS RECEIVE Command

Use the following information when receiving files from CICS:

Figure 6 shows the command and information you must provide. Enter it as shown (including parentheses), except that you can use either uppercase or lowercase.

Notes:

1. ƀ means to insert a space. There must *not* be a space between **h:** and **fn**.
2. For DBCS sessions, insert a left bracket ([) before **(options**.

```
RECEIVE  ƀ  [ " ] [ d: ] [ path ]   filename.ext  [ " ]  ƀ  [ h: ] fn   [ ƀ (options  ]
              └──────────────┘       └─────────┘       └────┘   └────────┘
                     A                    B                C         D
```
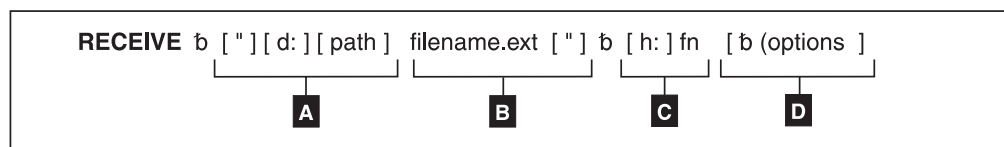
Figure 6. CICS RECEIVE Command Syntax

A   The workstation drive and path where the file is to be received.

B   The name of the workstation file.

C   The short name of the host session (h:) from which you are receiving the file, and the host file name (fn).

D   Optional changes made to the file during transfer. More than one option can be specified. Valid options are:

- ASCII
- BINARY (for SBCS sessions)
- BLANK
- CLEAR
- CRLF
- JISCII (for Japanese DBCS sessions)
- NOCLEAR
- NOCRLF (for SBCS sessions)
- PROGRESS
- QUIET
- SO
- TIME(n)
- USER

> **Note:** The default options for SBCS sessions are ASCII and CRLF; the default option for DBCS sessions is BINARY.

The parts of the CICS RECEIVE command are:

RECEIVE
> The command.

d: The name of the diskette or hard disk drive where the file is to be received.

path   The path to the subdirectory where the file is to be located.

filename.ext
> The name of the workstation file, including the extension. You can create a new name or use a name that is already on your workstation diskette or hard disk. If you use a new name, the file that you receive is added to your diskette or hard disk. If you use the name of an existing file, the file that you receive either replaces or supplements the existing file. Refer to the APPEND option.

h: The short name of the CICS session where the data set is located. If you have only one host, this part is optional. The default session is **A**.

fn   The name of the file you want to receive from your CICS host.

(options or [(options (for DBCS)
> These options can be specified:

> **ASCII** Performs the following:
> - Converts EBCDIC codes to 1-byte workstation codes.
> - Converts IBM host DBCS codes to 2-byte codes.
> - Inserts SO and SI characters in a DBCS field and converts the control characters ← and →, if they are present, to SO and SI characters.
>
> The default is ASCII CRLF. You need ASCII and CRLF control terms for text or source files that you want to view or edit, such as SCRIPT files. You do not need them for binary files.
>
> **Notes:**
> 1. CRLF and NOCRLF are mutually exclusive options.
> 2. BINARY and ASCII are mutually exclusive options.

3. The assumed defaults, if the optional parameters are omitted, are CRLF ASCII.

BINARY

The data in the file is binary data. The data can be encrypted data, compiled programs, or other data. It is not translated by the host file transfer program but is copied without changes into the workstation file.

This option is valid for SBCS sessions only.

BLANK

This option is valid only when used with the CRLF option; it retains BLANK (hex 40) at the end of each line.

CLEAR

Clears the workstation window at the beginning of the file transfer.

CRLF Specifies the use of the carriage return and line feed codes.

JISCII This option is valid for Japanese DBCS sessions only and performs the following:

- Converts EBCDIC codes to 1-byte workstation codes.
- Converts IBM kanji codes to 2-byte codes.
- Inserts SO and SI characters in a DBCS field and converts the control characters ← and →, if they are present, to SO and SI characters.

NOCLEAR

Suppresses the sending of a Clear command at the beginning of file transfer. This option is required for ISPF command mode.

NOCRLF

Specifies that the host computer file does not consist of logical records. The items in the temporary storage queue are sent in order and concatenated in your workstation into a single string of data.

This option is valid for SBCS sessions only.

PROGRESS

Shows a message indicating that the file transfer is in progress or has ended. Such messages do not show the current transferred bytes.

QUIET

Does not show any messages.

SO This option is valid only for DBCS when used with the JISCII or ASCII option; it converts SO (hex 0E) and SI (hex 0F) to RS (hex 1E) and US (hex 1F).

TIME(n)

Specifies the length of time, in units of 30 seconds, the program waits for a response from the host before it sends an error message. Replace n with an integer value in the range from 0 through 2184. If you specify 0, timeout is not set. The default is 1. To avoid a premature error message, specify an adequate value. In cases of large packet sizes, of large block sizes, or for slow communication lines (such as SDLC and COM port), *5* (150 seconds) is recommended. There should be no blank spaces between TIME and (n).

USER   This option is valid only for DBCS when used with the options JISCII, ASCII, and SO. SO (hex 0E) and SI (hex 0F) are not converted to RS (hex 1E) and US (hex 1F); they are written to a file without being converted.

Examples

The following examples show the command syntax you can use to receive files from your CICS host to your workstation.

Note: For Japanese DBCS sessions, use the JISCII rather than the ASCII option. For Hangeul, Simplified Chinese, and Traditional Chinese DBCS sessions, use the ASCII rather than the JISCII option.

- To receive a file from your CICS host to your default drive for a workstation session:

  RECEIVE pc.txt A:cicsfile (ASCII CRLF)

  RECEIVE pc.txt A:cicsfile [(JISCII CRLF) (for DBCS)

  This command sends a file named **CICSFILE** from your CICS host in session **A** to your workstation session. It adds the file to your default drive (diskette or hard disk) with the name **PC.TXT**.

- To receive a basic file from your CICS host and replace a file on a drive other than your default:

  RECEIVE a:myprog.exe a:myprog

  This command sends a file named **MYPROG** from your CICS host in session **A** to a drive other than the default for your workstation session. It replaces a file named **MYPROG.EXE** on a diskette in drive A.

Configuring File-Transfer Code Translation

When you transfer a file between the host and the workstation using the ASCII option (JISCII, for Japanese DBCS sessions), the host-system file-transfer program performs translation from EBCDIC to ASCII (JISCII, for Japanese DBCS sessions) or vice versa, according to the host and PC code pages specified during PC/3270 configuration. However, you might want to use different translation from that supplied by IBM. For the details, refer to "Chapter 5. Transferring Files" on page 31.

Chapter 11. Considerations for the Use of PC/3270 Sessions

This chapter contains additional hints and tips for using PC/3270 sessions.

Considerations and restrictions on the use of PC/3270 and supplementary information other than the items described in this chapter may be included in README.TXT in the Personal Communications directory. Double-click the **Readme - Please!** icon to open this file.

Host-Session Window Operations

Cursor Color

PC/3270 assigns white to the cursor color as the product default, and mixes the background color (if it exists) with an XORed (exclusive or) operation. If you want to change this default cursor color assignment, modify the PCSWIN.INI file and insert the following line into the [Session] section:

```
[Session]
CursorColor=<red-value> <green-value> <blue-value>
```

Where <red-value>, <green-value>, and <blue-value> are integers from 0 to 255 that specify the color intensity for each color primitive respectively; 0 0 0 for black, and 255 255 255 for white.

Releasing Insert Mode with Attention Keys

As on a non-programmable terminal, you can release insert mode when you press an Attention key. If you want this to happen, add this parameter to the [Keyboard] section of the workstation profile (*.WS):

```
[Keyboard]
ResetInsertByAttn=Y                <- Added.
```

Scroll Bar

If you choose **Font** from the Appearance menu in the host session window and choose **Fixed Size** from the Select Display Font window, the entire operator information area might not appear on the screen. If you specify **With Scroll Bar**, the OIA will not scroll. The session window size is restricted to be smaller than the screen size.

Scroll-Lock Key

When the Scroll Lock keyboard indicator is turned on, the cursor movement keys and the Page Up and Page Down keys are used to scroll windows only when you specify **With Scroll-Bar** in the Window Setup window. If you specify **Without Scroll-Bar**, you cannot use the Scroll Lock key, because the entire screen is displayed. For example, cursor-movement keys do nothing in Scroll Lock mode.

3270-Session Screen-Size Control

Although you can specify the screen size of the 3270 session in the PC/3270 configuration, the final screen size is determined by the host according to the SNA BIND negotiation.

Considerations for PC/3270 Sessions

Sometimes, you might want to use a particular screen size when it is difficult to have the host VTAM table changed. The following method will satisfy your requirement; however, because it violates SNA rules, some applications that refer to the VTAM table only might not work.

Modify your workstation profile (.WS) to insert the following line into the [LU] section:

```
[LU]
InfScreenSize=Y
```

Personal Communications processes byte 24 of the host BIND image correctly and according to the SNA architecture. Byte 24 is the last 2 bytes of
```
PSERVIC=X'028000000000000000000300' (03 in this case)
```

in the log mode.

If byte 24 is X'03', the default screen size is 24 x 80 and the alternate is what you configured with PC/3270.

If byte 24 is X'00' or X'02', the default and alternate screen sizes are 24 x 80.

If byte 24 is X'7E' or X'7F', bytes 20–23 in the BIND control the default and alternate screen sizes.

If you have X'03' in the BIND image, you can customize whatever screen size you want. The host will query Personal Communications and use the alternate screen size.

The following table shows the details of the BIND image processing:

```
   Byte in BIND by Host ==>    Result (Screen size)
   -------------------         --------------------
   24  20  21  22  23           Default  Alternate
   ==  ==  ==  ==  ==           =======  =========
a) 00  --  --  --  --   ==>     24 x 80  24 x 80

b) 02  --  --  --  --   ==>     24 x 80  24 x 80

c) 03  --  --  --  --   ==>     24 x 80   R x C

d) 7E  Rd  Cd  --  --   ==>     Rd x Cd  Rd x Cd

e) 7F  Rd  Cd  Ra  Ca   ==>     Rd x Cd  Ra x Ca

Legend:  RxC: Screen Size defined by PC/3270 configuration
             (R,C)  = (24,80) , (32,80) , (43,80) or (27,132)

         Rd: Rows in the default screen size mode
         Cd: Columns in the default screen size mode
             (Rd,Cd) = (24,80) , (32,80) , (43,80) or (27,132)
             Rd*Cd <= R*C

         Ra: Rows in the alternate screen size mode
         Ca: Columns in the alternate screen size mode
             (Ra,Ca) = (24,80) , (32,80) , (43,80) or (27,132)
             Ra*Ca <= R*C
```

PC/3270 supports Query Replies. Make sure that the Query Reply of Implicit Partition returns the final screen size upon receipt of the BIND command.

Customizing a Display Translation Table

PC/3270 displays the host EBCDIC character using the workstation (ANSI) graphic symbol so that the character defined by the S/390 EBCDIC host code page is displayed correctly using the same graphic symbol defined by ANSI. However, you might need your original translation, because your host or workstation application is not designed to use the standard translation.

You can use your original translation table if you refer to the following procedure as an example. Note that the data integrity caused by the user-defined table is your responsibility.

The following procedure is an example of how to remap left and right brackets.
1. Terminate all running 3270 sessions
2. Modify the PC/3270 workstation profile (***.WS**).
   ```
   [Translation]
   IBMDefaultView=N
   DefaultView=C:\Personal Communications\PRIVATE\BRACKET.XLT
   ```
3. Create the display translation table file (**.XLT**). In this example, the following BRACKET.XLT file is created in the Personal Communications private subdirectory.
   ```
   [Profile]
   id=XLT
   Description=User-defined Display Translation Table

   [Option]
   Replace=Y

   [SB Xlate]
   ; EBCDIC=ANSI
   ; The next line displays EBCDIC X'AD' as
   ; an ANSI X'5B' (left bracket)
   AD=5B
   ; The next line displays EBCDIC X'BD' as
   ; an ANSI X'5D' (right bracket)
   BD=5D
   ```
4. Create your own keyboard layout (.KMP) if you need to enter your new left and right brackets graphic symbols:
   ```
   [Keyboard]
   KEY27=ansi dd
   KEY28=ansi a8
   ```
   The information on the right should be lowercase characters. PC/3270 translates ANSI X'dd' into EBCDIC X'ad'. It is displayed as [by the table created in step 3.
5. Click on the PC/3270 icon corresponding to the modified workstation profile.

Support for Long File Names

Like Windows, Personal Communications supports long file names. You can give any name (up to 255 characters) to a file; you are not limited to eight characters with a three-character extension. You can use spaces in the file name, but not the symbols /, \, :, *, ?, ", <, >, or |. In addition, the tilde () character should not be used in CMS or MVS host file names.

File Transfer Function

Host File Name and Reserved Words

You should not use the following words as a VM file name or file type, as a MVS data set name, or as a CICS file name, because they are reserved for use as option commands:
- ASCII
- APPEND
- TIME
- CLEAR
- NOCLEAR
- SILENT
- QUIET
- PROGRESS
- JISCII
- SO
- NOSO
- BLANK
- USER
- CRLF
- BINARY
- NOCRLF

Changing the Packet Size When Import/Export Is Idle

When import/export is idle, select **Preferences > Transfer** from the **Edit** menu. When you change the packet size on the Setup window, end import/export, and then rerun it.

Wait Option for Multiple File Transfer

If multiple file transfers do not succeed, insert the following statement into the [Transfer] section of your **.**WS file:

```
[Transfer]
wait=1000
```

This parameter causes a 1000 msec (1 sec) delay between file transfers. If this does not help, you might need to increase the value again.

NOTRUNC and BLANK Options (SBCS Only)

If you want to add trailing blanks (spaces) to fill the logical record length for each record when downloading a text file, use the following options in the **Additional Options** edit field of the Transfer-Type Definition window.
NOTRUNC : for VM/CMS (PTF# UR35492)
NOTRUNC : for MVS/TSO (PTF# UR34797)
BLANK : for CICS

Setting the VTAM PSERVIC Statement

File transfer problems can occur if extended attribute support has not been set on in the VTAM PSERVIC statement. For extended attribute support, set on the high order bit in PSERVIC byte1 (zero byte origin) as follows: x'xx80xxxxxx...'.

Graphic Functions

This section provides information, restrictions, and considerations for graphic functions.

Graphics Protocols

Personal Communications allows you to use host graphics applications, such as GDDM and others. Two types of graphics are supported:
- Vector
- Programmed symbols

Two protocols are supported for vector graphics:
- Advanced
- Native

See Configuring Graphics in the online helps for a description of these protocols and to learn how to configure your sessions for graphics.

The following functions ar supported:
- Multiple mixed alphanumeric and graphics host sessions
- Use of standard OS/2 printing and plotting facilities
- Creation of PIF (Picture Interchange Format) files
- Clipping graphics data into the clipboard

Vector Graphics

Vector graphics are computer graphics in which display images are generated from display commands and coordinate data. Personal Communications provides vector graphics support for the OS/2-Link (advanced) or the 3179G or GOCA (native) protocols. Choose the protocol that is appropriate for your host applications.

Advanced Protocol: Use the advanced protocol when you have GDDM Version 2 Release 3 or later and are using any of the following operating systems:
- MVS
- VSE
- VM/SP
- VM/XA SP

Note: The advanced protocol is not supported by the CICS pseudo-conversational mode with versions of GDDM earlier than Version 3, and not by IMS/VS at all. It is, however, supported by the CICS pseudo-conversational mode with GDDM Version 3 Release 1 or later.

The advanced protocol is equivalent to that used by OS/2-Link, so it supports the same subsystems. However, no download of code from the host system is required for Personal Communications because all the OS/2-Link graphics modules are integrated into the program.

Native Protocol: Choose the native protocol when you intend to use older GDDM versions or non-GDDM host-graphics applications, such as those originally intended for use on 3270 nonprogrammable terminals as the 3179G, 3192G or 3472G. The native protocol also allows IMS/VS users to display GDDM graphics.

Note: A 3174 controller requires a terminal to respond within a certain time; if it does not, a machine check 207 message appears. If you are using a coaxial attachment for graphics in native mode, this can happen for various reasons,

such as the complexity of the graphics drawing-orders or the speed of your workstation. In this case, you should change to advanced mode.

Programmed Symbols

Raster graphics are displayed with programmed symbols, which are downloaded to your workstation. Personal Communications support up to six sets (PSA through PSF) of triple-plane and multiple-color programmed symbols.

Use programmed symbols as the graphics type when you intend to use host graphics applications originally written for the 3279G terminal.

Graphics applications use one or both of these methods to display graphical screens. Personal Communications allows you to enable or disable support for vector graphics and programmed symbols. Choose the type of support that our host applications require.

Note: If you use the OS2-Link (advanced) protocol under the GDDM program, do not choose programmed symbols. Also, do not choose programmed symbols when you use the OS2-Link protocol with other applications.

Enabling Programmed Symbol Sets: PC/3270 provides up to six sets of triple-plane programmed symbols, depending on the type of graphics support that you choose. By default:

- Two sets (PSA and PSB) of single-plane programmed symbols are usable if you choose both programmed symbols and vector graphics.
- Three sets (PSA, PSB, and PSE) of single-plane programmed symbols and three sets (PSC, PSD, and PSF) of triple-plane programmed symbols are usable if you choose programmed symbols, but not vector graphics.

You can change the number of programmed-symbol sets and triple or single planes available for each programmed-symbol set by editing the [3270] section of the workstation profile:

```
PSSPlanes=xxxxxx
```

Each x represents a number (0, 1, or 3) that indicates how many planes are to be available for each set; the first column indicates the number of planes for PSA, the second column for PSB, and so on. For example, to enable six triple-plane programmed symbol sets, enter the following:

```
PSSPlanes=333333
```

To enable two single-plane and two triple-plane sets, enter the following:

```
PSSPlanes=113300
```

How to Handle Errors Caused by Insufficient Memory

Graphic execution module PCSGRP.DLL uses a large amount of global memory for graphic drawing or printing. When the workstation has insufficient installed memory, results might not be correct. For example, an area might not be clearly shaded.

In this case, increase the amount of installed workstation memory by at least 1 MB. For host graphic printing, add 1 more megabyte.

Memory might have to be further extended depending on the host graphic application and printer driver used.

Drawing-Buffer Size

The drawing-buffer size varies depending on the contents set for Redraw of a graphic function.

To set **Redraw**, select **Preferences > Appearance > Display Setup** from the **Edit** menu in the session window. Select **Graphics** from **Category**.

Selecting **Host** from the optional items of **Redraw** requires no buffer.

If you select **Retained**, the graphic execution module stores all redrawing data into a buffer. Such a buffer is called a *retained buffer*. The buffer size varies depending on the complexity of the graphic data from an application program.

Example:

Simple table   ... 10 KB to 20 KB
Complicated graphic image... 200 KB to 300 KB

When you select **Bitmap** to set **Redraw**, the buffer size will be the same as the sum of the retained buffer size and compatible bit map size. For example:

(Height) x (Width) x (Number of planes) x (Bits/Pixel) / 8 bytes

Example:
When you select a 7x12 font for a VGA 16-Color Display Model 2 (24x80), the bitmap size is:
Bitmap size = (7x80) x (12x24) x 1 x 4 / 8 = 80 KB

When you select a 12x20 font for an IBM PS/55 High-Resolution 256-Color Display Model 2, the bitmap size is as follows:
Bitmap size = (12x80) x (20x24) x 1 x 8 / 8 = 460 KB

Using Bitmaps for Drawing

The graphic execution module uses a bit map compatible with the display unit to draw an area instruction in overpaint mode. An image instruction requires one plane bit map.

(Buffer for area) = (Area width) x (Area height) x (Number of planes) x
(Bits/Pixel) / 8

(Image buffer) = (Image width) x (Image height) / 8

Print Buffer Size

The retained buffer must be used for printing. The retained buffer is the same size as that used for redrawing. This is also applied when you specified **Bit Map** for **Redraw** on the Display Setup window.

If graphic printing is called in Bitmap mode, the graphic printing module generates a bitmap compatible with the connected printer, draws an image on the bitmap, and transfers the bit image to the printer.

This operation is generally performed quickly. When memory is frequently swapped, the process slows down in proportion to the number of swap operations.

If a large bit map is not allocated, the graphic printing module prints a graphic image normally using only the retained buffer. Example: Proprinter (240x144 DPI) character size: Bitmap size = (240x8) x (144x11) x 1 x 1 / 8 = 380 KB Example: EPSON (ESC/P) (360x180 DPI color) character size: Bitmap size = (360x8.5) x (180x11) x 3 x 1 / 8 = 2.3 MB

Note: With some printers, different printing results might be obtained in bitmap mode and non-bitmap mode. If the desired results are not obtained, change the current bitmap mode. For example, specify non-bitmap mode to print in bitmap mode.

Edit-Copy Buffer

An editing operation causes the graphic execution module to copy a bit map and DIBitmap to the clipboard. The bit map is compatible with the display; DIBitmap is a 4-bit/pixel bit map.

Printer Fonts

The printer driver can handle two font sets, the device font and GDI font. The device font is a hardware font built into the printer. The GDI fonts are System (without brackets) or other software fonts for Windows.

When you select a font set for graphic printing from the Printer Control window, use the GDI fonts for the following reasons:

- In bitmap mode, a GDI font can be used for printing. However, the device font cannot be used, because an image cannot be drawn on a memory bit map when using the device font.
- In bitmap mode, the device font cannot be used for printing when OR and exclusive OR attributes are mixed.

Plotter

Because a plotter does not support a raster, the following restrictions are imposed on drawing. Use a plotter for figures and tables that have mainly lines.

- No shading is supported.
- Some shading patterns cannot be distinguished.
- Image order drawing requires much time, and the final printout is of poor quality.
- The OR and exclusive OR are not correctly reflected.

Hole in Screen Caused by Clearing a Graphic Character

When a character overlaps a graphic image, the graphic image is cleared at the position where the character is to be displayed. When you enter a null character or space having the transparent attribute at the position where a graphic image is displayed, the graphic image in that character cell is not cleared.

If you select **Host** or **Retained** to set **Redraw** on the Display Setup window, when characters in a graphic image are cleared, a hole appears in the graphic area. This is because these two modes do not have a bitmap image, and partial redrawing cannot be performed on the screen.

If you select **Bitmap** mode as **Redraw Graphics**, you can find no hole on the graphic region by the application that overrides any alphanumeric characters (as well as NULL and SPACE) on the graphic image.

To restore the screen, perform either of the following actions:

- Press the PA3 key to have the application program redraw the screen.
- Minimize and restore the graphic image retained in Retained mode, then redraw it or select Bitmap mode.

Note: When you change the setting of **Redraw** in the Display Setup window, the set contents are valid from the next drawing.

Miscellaneous Restrictions for Graphic Functions

If advanced protocol is selected, graphic functions cannot be used in the IMS/VS and CICS pseudo-conversational mode with versions of GDDM earlier than Version 3.

Considerations for Graphics Functions

Native-Graphics Datastream

If the host sends an Object Structured Field (Object Picture, Object Data, Object Control) with a zero value in the length field, Personal Communications rejects it and displays PROG754.

Printout to LPT1

When you use a host application that prints to your PC's LPT1, you must first select the printer in the Printer Setup dialog of the File menu.

Print Processing

Using a Printer with the Non-SNA DFT Attachment

For the non-SNA DFT attachment, PC/3270 regards any print job within 30 seconds after the previous one as a continuous job with the previous one. It determines the job end artificially by a timeout value.

You can change the standard timeout value by modifying the [CNDFT] section of the workstation profile (*.WS):

```
[CNDFT]
PrtTimeOut=nn
```

Note: The variable nn is the decimal value of the timeout in seconds.

Print Job Control Using SNA Timeout

PC/3270 treats the data packet, which begins with an SNA Begin Bracket (BB) and ends with an SNA End Bracket (EB), as one job, and spools it to the Windows print manager. However, some applications might not use SNA BB and EB appropriately, and an unexpected page eject can occur. This section describes two timeout controls that you can customize.

Concatenation of Multiple Print Jobs

When a new print job comes from the host before the timer specified by the workstation profile expires, the job is concatenated to the previous one, and they are regarded as one job. You can change the timer value by specifying the following parameter in the [printers] section of the workstation profile (*.WS):

```
[printers]
ConcatenateTime=nn
```

Note: The variable nn is as follows; its unit is one second:

> **0**   No concatenation of print jobs
> **1 – 32767:**
> > Concatenation of print jobs by this timeout value

When a non-SNA attachment is used, the timer should not be specified in the workstation profile.

Termination of a Print Job: If an end of a print job (SNA End of Bracket) does not come from the host before the timer specified by the workstation profile has expired, the job is automatically terminated. You can change the timer value by specifying the following parameter in the [printers] section of the workstation profile (*.WS):

```
[printers]
TerminateTime=nn
```

Note: The variable nn is as follows; its unit is one second:
> **0**   No automatic termination of a print job
> **1 – 32767**
> > Automatic termination of a print job by this timeout value

When a non-SNA attachment is used, the timer should not be specified in the workstation profile.

SCS TRN Command (LU 1)

You can change the operation of the SCS TRN command by adding a parameter to the [printers] section of the workstation profile (*.WS):

ATRN=Y
> Transfers data that is controlled by the SCS TRN command to the printer driver without conversion. In the cases below, however, this parameter is assumed to be ATRN=N.
> - The printer driver does not support PASSTHRU.
> - The printer driver supports the BANDING function.
> - BANDING=Y is specified in the [printers] section of the workstation profile (*.WS).
>
> When you use a printer driver for which the PASSTHRU function is incomplete, sense code X'1003' is returned to the host system for the SCS TRN command.

ATRN=N
> Converts data that is controlled by the SCS TRN command and transfers the result to the printer driver.

The default value is:

```
ATRN=Y
```

Note: The SCS TRN command (LU 1) is available in PDT mode only.

SCS SPD/SLD Command (LU 1)

If the selected font cannot be used with the values specified for CPI or LPI, this command temporarily switches to a valid font.

Ignoring Blank Pages

If you cannot print a blank page correctly, check for the following things:

- Some printers ignore a blank page if so specified with the **Printer Setup** option (or setting the printer's DIP switches). Check whether the printer is set to enable this option.
- Some printer drivers ignore a blank page if so set with the **Printer Driver Setup** option. Check whether this option is supported by the printer driver.

Printing Reverse-Display Characters with GDI Fonts

Some printer drivers that do not support reverse-display character printing print reverse-display characters as spaces. To avoid this, add the `REVERSE=N` line to the [printers] section of the workstation profile (`*.WS`).

CPI/LPI of Device Fonts

If the printer driver cannot use the device font for the specified CPI or LPI, data can be printed with other CPI or LPI values.

Printing Reverse-Display Characters with Device Fonts

Most printer drivers do not support reverse-display-character printing with device fonts.

PCSERR999 Error Messages

Message `PCSERR999 - Personal Communications internal error:` *module-name* *- xxx* might appear when there is insufficient memory. Because the print job remains in the print manager, you should delete it.

Printable Area

Characters might not be printed over a large enough area, depending on the printer driver used. When using a printer driver that allows you to set the margins, specify the minimum margins to maximize the printable area.

SVF Control Code

If the maximum print line (MPL) is less than the paper size in the SVF control code, a page feed is performed when the printing position reaches MPL.

PDT Mode

Printing in PDT mode is restricted as follows:

- A graphic image is printed through the Windows printer driver specified in "Printer Setup", even if the printer is set up for PDT mode.
- APL characters cannot be printed.
- PostScript printers are not supported.

SDDLU Considerations

PC/3270 supports the self-defining dependent logical units (SDDLU) function of ACF/VTAM Version 3 Release 4.1 (MVS) or Version 3 Release 4 (VM). SDDLU provides the ability to dynamically create dependent LUs on predefined PUs and to reconfigure dependent LUs without interruption to any other user on the network and without intervention by system programming or operations staff.

PC/3270 SDDLU is supported for all connections except DFT (coaxial connection).

PC/3270 network stations support SDDLU by:

- Accepting a format-1 Activate PU (ACTPU), which signals that VTAM requires a Network Management Vector Transport (NMVT) containing the Product Set ID (PSID) and LOCADDR of the LU.
- A PC/3270 network station will use LOCADDRs 02, 03, 04, and so on, in that sequence, by default. However, if you specify the LU addresses explicitly, it will use the addresses that you specify.

Considerations for PC/3270 Sessions

- Sending an unsolicited NMVT to VTAM when an LU switches on.

PSID Definitions

PC/3270 defines the default PSID for each LU type. Generally, the PSID contains such things as the machine type and the model number. The following table describes the machine type and model number definitions for each LU type supported by PC/3270.

Machine Type	Model Number	LU Type	Screen/Buffer Size
3270	002	Display Model 2 screen	24 rows by 80 columns
3270	003	Display Model 3 screen	32 rows by 80 columns
3270	004	Display Model 4 screen	43 rows by 80 columns
3270	005	Display Model 5 screen	27 rows by 132 columns
3270	DS2*	Printer 3270 data stream	24 rows by 80 columns
3270	DS3*	Printer 3270 data stream	32 rows by 80 columns
3270	DS4*	Printer 3270 data stream	43 rows by 80 columns
3270	DS5*	Printer 3270 data stream	27 rows by 132 columns
3270	00A	Display	48 rows by 80 columns
3270	00B	Display	62 rows by 160 columns
3270	00C	Display	24 rows by 132 columns

:
*   PC/3270 assumes that 3270 data stream is the default printer-session type. If you use the SNA Character String (SCS) data stream, you must use a different designation for the PSID, such as 3270SCS. You must define this yourself, as shown in the next example.

Assigning User-Defined PSIDs

You can replace the default PSID values with your own definitions or define new PSIDs, by preparing a PSID definition file or by changing the Windows workstation profile. The PSIDs you use must be defined to VTAM.

Add a statement such as the following one to the [LU] definition in the workstation profile. This example supersedes the default PSID with LUA0001.

```
PSID = LUA0001
```

Part 3. Personal Communications AS/400

Chapter 12. Building a Printer Definition Table (PDT) for PC400

This chapter explains how to create and change the printer definition table (PDT file) used for PC400. The PDT file is created by converting the printer definition file (PDF file). The PDF and PDT define the transmission of characters and control codes to the printer and the format of printer output. To create a PDF (or change an existing one—the recommended method), use a text editor that can produce or update an ASCII file.

Selecting and Changing the Printer Definition Table (PDT File)

To use the PDT file:
1. Select **File** from the menu bar of the workstation window.
2. Select **Printer Setup** from the File menu.

   The Printer Setup window appears.
3. Select the **Use PDT file** check box and **Select PDT**.

   The Select PDT file window appears.

To build the PDT file (required only if the PDF has been changed or created):
1. Select **Convert PDF**. Select the PDF file to be converted from the list in the Convert PDF to PDT window; then select **Convert**. The PDF File Converter window appears. After the file has been converted, click on **Close**, then click **Close** in the Convert PDF to PDT window.
2. Click **OK** in the Select PDT file window.
3. Select **OK** on the Printer Setup window.

   After printer setup is complete, the Printer Setup window is closed.

Printer Definition File (PDF File) Format

A PDF contains 3 main sections:
- Macro definitions
- Formatting controls
- Character definitions

Macro Definitions
This section of a PDF contains user-defined macros. A macro is a single mnemonic that stands for a control code or a sequence of control codes. A mnemonic simplifies defining control sequences for PC printers and makes it easier to read the information in the PDF.

The following table shows the structure of a macro definition statement. A macro definition is composed of four parts:

1	2	3	4
name	EQU	PC Printer Control Codes	Comments

For example:

```
   FFF    EQU   0C            /* Form Feed */
```

Building PC400 PDTs

The first part is the user-defined mnemonic or macro name. This name must be exactly three characters long and must not begin with a number. It is helpful to define a meaningful mnemonic, such as P17 for 17.1 pitch.

The second part, EQU, stands for equate and must be coded as EQU.

The third part is the hexadecimal control code, which is specific to a PC printer. Each PC printer manufacturer can define different control codes for the same printer function. For example, the IBM 4019 LaserPrinter uses control codes defined by the IBM Personal Printer Data Stream (PPDS). Hewlett-Packard printers use control codes defined by the Hewlett-Packard Printer Control Language (PCL). These control codes are usually defined in the manual that comes with the printer.

Some PC printer manuals describe control sequences as a string of ASCII symbols, such as ESC J 1 K: others use hexadecimal numbers, such as 1B 57 01; while others use decimal values, such as 27 28 1. The printer definition table compiler accepts any of these formats.

The control codes in the macro definitions section can be any of the following:
- Single characters that are interpreted as their ASCII value
- Two digit numbers that are interpreted as hexadecimal values
- Three digit numbers that are interpreted as decimal values

If you leave the control code section blank or if you delete it, the character or control code is interpreted as a null string. If a character or control code is defined more than once in the file, the last definition is used.

The fourth part is the comment section. The symbols /* indicate the beginning of a comment and the symbols */ indicate the end of a comment. Comments can be coded at any point in the printer definition file and are ignored by the printer definition table compiler.

The following are example macro definition statements that are specific to the IBM 4019 LaserPrinter.

Sample IBM 4019 LaserPrinter Macro Definition Statement

```
LFF EQU 0A                            /* Line Feed                 */
VTB EQU 0B                            /* Vertical Tab              */
FFF EQU 0C                            /* Form Feed                 */
CRR EQU 0D                            /* Carriage Return           */
P05 EQU 1B 57 01                      /* 5 Pitch-Characters/inch   */
P10 EQU 12                            /* 10 Pitch-Characters/inch  */
CDW EQU 1B 57 00                      /* Cancel Double Wide contin.*/
P17 EQU 12 0F                         /* 17.1 Pitch-Character/inch */
LL8 EQU 1B 41 09 1B 32                /* Set line length 8 lines/inch*/
P12 EQU 1B 3A                         /* 12 Pitch-characters/inch  */
RES EQU 1B 5B 4B 07 00 05 31 01 A4 00 00 90
                                      /* The above macro resets the  */
                                      /* printer using the factory   */
                                      /* defaults.  See the IBM Laser*/
                                      /* Printer Technical Reference */
                                      /* manual.                     */
```

To illustrate how macros are coded, an example follows that is specific to the IBM 4019 LaserPrinter. To have the host print job printed in double-wide characters, you must know what control code turns on double-wide printing on your PC printer. On the IBM4019 LaserPrinter, the control code to turn on double-wide printing is X'1B5701'. This would be coded as:

```
BDW EQU 1B 57 01
```

where BDW stands for Begin Double Wide.

This alone would not cause 3270 host print to send this control to the printer. The mnemonic would have to be included in a control code statement, such as START_JOB which is described in the control codes section of the file.

Note: The IBM 4019 LaserPrinter printer definition file that comes with IBM Personal Communications already has this control code defined as the P05 macro. See line 9 of the sample IBM 4019 LaserPrinter Macro Definition Statements ("IBM4019.PDF File Contents" on page 115).

Macro Name Examples: The following are example mnemonics or macros. The control codes are for the IBM 4019 LaserPrinter:

```
LND     EQU 1B 26 6C 31 4F              /* LANDSCAPE    */
POR     EQU 1B 26 6C 30 4F              /* PORTRAIT     */
P12     EQU 1B 28 73 31 32 2E 30 30 48  /* PITCH_12.00  */
T10     EQU 1B 28 73 31 30 2E 30 56     /* POINT_10.0   */
LTR     EQU 1B 26 6C 32 41              /* LETT_PAPER   */
G66     EQU 1B 26 6C 36 36 50           /* PG_LENGTH_66 */
```

Formatting Controls

The controls section of a PDF contains the following:

- Session parameters
- Control codes
- Color specifications
- Highlight specifications

Session parameters: MAXIMUM_PAGE_LENGTH, MAXIMUM_PRINT_POSITION, and SET_PAGE_LENGTH

The parameter to the left of the equal sign is a keyword and MUST be coded exactly as shown. The keyword is used to define a statement. The values to the right of the equal sign are macros or control codes. Because the values to the right of the equal sign can be both macros and control codes, they will sometimes be referred to as parameters.

The MAXIMUM_PAGE_LENGTH, MAXIMUM_PRINT POSITION, and SET_PAGE_LENGTH parameters specify the dimensions of the output job. The number specified in the MAXIMUM_PAGE_LENGTH parameter is used in the SET_PAGE_LENGTH parameter and is substituted for the value keyword. In other words, if SET_PAGE_LENGTH and MAXIMUM_PAGE_LENGTH are coded as follows:

```
MAXIMUM_PAGE_LENGTH=066 /* Printed lines per page */
SET_PAGE_LENGTH=SFL 066
```

The results would be:

SET_PAGE_LENGTH=SFL 066

Because SFL is coded as X'1B 43' in the macro section, the actual control code that would be sent to the printer to set the maximum page length is:

X'1B 43 42'

where X'42' is decimal 66.

Note: Setting MPL=255 causes suppression of form feeds (FFs).

Control Codes: The control codes section of a PDF is used by the PDT function to determine what specific PC printer control code is to be sent to the PC printer when an SCS control code is received. Some of the statements used by Communications Manager are in this section.

```
START_JOB=SEL CDW CDL CUL CDS CP8 CS2
END_JOB=CAT CDW CDL CUL CDS CP4 FFF
SET_PAGE_LENGTH=SFL value
SET_VARIABLE_LINE_DENSITY=ESC A value ESC 2
SET_10_CHARACTERS_PER_INCH=P10
```

Note: The above example lines come from the IBM 4019 LaserPrinter PDF.

START_JOB and END_JOB

The control codes associated with START_JOB are sent at the start of each host print job. It is best to set all printer options to a known or desired state at the beginning of each host print job. The PC printer changes its state or changes the options selected only when instructed to do so by control codes. Therefore, the previous PC application could have left the printer in portrait mode with a Courier font selected, and unless your job changed them, these would be the printer options used.

Even though the START_JOB and END_JOB control codes in Control Codes are the ones shipped with the IBM4019.PDF, many users change them to be more like the ones below:

```
    START_JOB=RES P12 LL8
    END_JOB=RES
```

In this example, the RES, P12 and LL8 macros are used on the START_JOB statement (these macros are defined in Sample IBM 4019 LaserPrinter Macro Definition Statements). This translates into the following control codes being sent to the printer at the beginning of your print job:

X'1B 5B 4B 07 00 05 31 01 A4 00 00 90 1B 3A 1B 41 09 1B 32'

This sequence of control codes:
- Resets the printer to the IBM PPDS factory default settings (RES)
- Begins printing in 12 pitch (P12)
- Begins printing at 8 lines per inch (LL8)

The END-JOB statement sends the following control code to the printer at the end of your job:

X'1B 5B 4B 07 00 05 31 01 A4 00 00 90'

This control code resets the printer to IBM PPDS factory default settings (RES).

SET_VARIABLE_LINE_DENSITY

The SET_VARIABLE_LINE_DENSITY statement is used in combination with the panel where you can select the number of lines per inch (LPI), which can be either 6 or 8.

The PDT process uses whatever is selected in the lines per inch field, and substitutes this number for the value keyword in the SET_VARIABLE_LINE_DENSITY parameter. In other words, if Lines per inch is set to 8, the SET_VARIABLE_LINE_DENSITY parameter is

```
SET_VARIABLE_LINE_DENSITY=ESC A 09 ESC 2.
```

Where the 09 comes from 72/8. 42 is the number of typographic points in 1 inch. Thus a value of 12 points would indicate six lines to an inch.

The control code that is sent to the printer to set the line density is:

```
X'1B 41 09 1B 32'
SET_10_CHARACTERS_PER_INCH
```

On most printers, the default pitch is 10 characters per inch. In most PDTs, Personal Communications uses this same convention and sends the control code found on the SET_10_CHARACTERS_PER_INCH statement. This is usually coded in the PDF as:

```
SET_10_CHARACTERS_PER_INCH=P10
```

where P10 is coded in the macro definition section as:

```
P10 EQU 12
```

Transparent Print Capability

"ASCII_PASSTHRU?" Keyword Support

If you add the following line to your PDF, Personal Communications will send host data without any translation:

```
    ASCII_PASSTHRU? = YES
```

This option is for special host applications that generate PC printer control codes directly.

Even if ASCII_PASSTHRU? is set, control codes defined START_JOB and END_JOB are sent to the printer at the start and the end of a print job respectively. To remove those commands, you need to rebuild the PDT file after removing the PDF keyword definitions for START_JOB and END_JOB.

If ASCII_PASSTHRU? is set, all character definition lines, for example, EBCDIC_xx, are ignored.

If both ASCII_PASSTHRU? and EBCDIC_PASSTHRU? are set, ASCII_PASSTHRU? has priority.

"EBCDIC_PASSTHRU?" Keyword Support

If you add the following line to your PDF, Personal Communications will ignore all SCS commands and send data to the printer after EBCDIC-to-ASCII translation:

```
    EBCDIC_PASSTHRU? = YES
```

For example, the default EBCDIC-ASCII translation table used for U.S. English host code page 037 is as follows:

```
  Hex |  0 1  2 3  4 5  6 7  8 9  A B  C D  E F
 -----+---------------------------------------
    0 | 2020 2020 2020 2020 2020 2020 2020 2020
   10 | 2020 2020 2020 2020 2020 2020 2020 2020
   20 | 2020 2020 2020 2020 2020 2020 2020 2020
   30 | 2020 2020 2020 2020 2020 2020 2020 2020
```

```
40 │ 20FF 8384 85A0 C686 87A4 BD2E 3C28 2B7C
50 │ 2682 8889 8AA1 8C8B 8DE1 2124 2A29 3BAA
60 │ 2D2F B68E B7B5 C78F 80A5 DD2C 255F 3E3F
70 │ 9B90 D2D3 D4D6 D7D8 DE60 3A23 4027 3D22
80 │ 9D61 6263 6465 6667 6869 AEAF D0EC E7F1
90 │ F86A 6B6C 6D6E 6F70 7172 A6A7 91F7 92CF
A0 │ E67E 7374 7576 7778 797A ADA8 D1ED E8A9
B0 │ 5E9C BEFA B8F5 F4AC ABF3 5B5D EEF9 EF9E
C0 │ 7B41 4243 4445 4647 4849 F093 9495 A2E4
D0 │ 7D4A 4B4C 4D4E 4F50 5152 FB96 8197 A398
E0 │ 5CF6 5354 5556 5758 595A FDE2 99E3 E0E5
F0 │ 3031 3233 3435 3637 3839 FCEA 9AEB E9FF
```

You can modify this code page using EBCDIC_xx keywords. Note that the EBCDIC_PASSTHRU? line precedes any EBCDIC_xx lines in your PDF file because Personal Communications reinitiallizes the EBCDIC-to-ASCII translation table when it finds that EBCDIC_PASSTHRU? is set.

Even if EBCDIC_PASSTHRU? is set, control codes defined START_JOB and END_JOB are sent to the printer at the start and the end of a print job respectively. To remove those commands, you need to rebuild the PDT file after removing PDF keyword definitions for START_JOB and END_JOB.

"EBCDIC_xx" Keyword Support

EBCDIC_xx keywords are used to remap entries of EBCDIC-to-ASCII translation table for printer sessions that receive SNA character string (SCS) data streams. For example, if you want to print host space characters using ASCII X'1B', add the following line to your PDF file and recompile to update the PDT file:

```
EBCDIC_40 = 1B
```

You can specify two or more characters for one EBCDIC character. The following definition:

```
EBCDIC_F1 = 1 2 3
```

translates EBCDIC X'F1' into "123" in ASCII.

You can also remap EBCDIC code points (X'00'-X'3F') which are normally assigned to SCS control characters. The following line:

```
EBCDIC_15 = 0C
```

overrides the SCS new line function with X'0C', which normally works as a form feed function. If you remap a code point that is not used for a SCS command (for example, X'27') Personal Communications sends the defined code to the printer and does *not* send SNA sense data to the host. The following example

```
EBCDIC_27 = 1B
```

defines EBCDIC code point X'27' as ASCII X'1B', which is used as an escape character on most personal computer printers.

If you do not need to use any SCS command processing but still need EBCDIC-to-ASCII translation, the EBCDIC_PASSTHRU? is a better solution. If you do not need any SCS command processing or EBCDIC-to-ASCII translation, ASCII_PASSTHRU? is an easier solution.

Printer Definition Tables

Standard printer definition table file names are of the form **IBMnnnnn.PDT**, and PDT's ASCII to ASCII character defintions are of the form **PRNnnnnn.PDT**, where nnnnn is a machine type. See character definition descriptions for more details.

See the help panel or "PDT Files" on page 18 for a list of the PDT files provided by Personal Communications.

The PDT files contained in the PC400 installation diskette can be used as is. However, you might want to do special formatting by changing the definitions of some fields. To do so, copy an existing PDF file, modify it, and then convert it to a new PDT file.

An example PDF file is shown below. Do not attempt to change the statements in a field for which modification is specifically prohibited. If a PDT file created according to a changed PDF file is used, the results of printing cannot be guaranteed.

Example Printer Definition Files

The following examples are annotated versions of printer definition files for the IBM LaserPrinter 4019 (for SBCS sessions) and for the IBM 5577 Printer (for DBCS sessions). These are examples only; the actual files may differ.

IBM4019.PDF File Contents

```
/**********************************************************************/
/*                                                                    */
/*     PRINTER SESSION DEFINITION FILE FOR: LaserPrinter 4019/4019-E  */
/*                                                                    */
/**********************************************************************/
/**********************************************************************/
/*                     Macro Definitions                             */
/* Define values here that will be used commonly throughout your      */
/* definitions.  Then use the left hand side of the equate as you     */
/* define your characters and control strings.  The printer compiler  */
/* will substitute the right hand side of the equate for each         */
/* occurrence of the left hand side throughout the file.              */
/*                                                                    */
/* Macro names must be at least three characters long and may not     */
/* begin with a number.                                               */
/*                                                                    */
/* Format                                                             */
/* A Macro Name is associated with a value or string of values by the */
/* EQU statement.  The right hand side of an EQU statement must be a   */
/* string of zero or more two digit hexadecimal numbers.  If a macro  */
/* definition is more than one line long, you may extend it to the    */
/* next line by ending the first line with a comma.  In this manner   */
/* you may define a macro which is many lines long by terminating each*/
/* line except the last with a comma.  No macro names are allowed on  */
/* right hand side.                                                   */
/**********************************************************************/
BEGIN_MACROS
/* The following values are standard for most printers.  Check your   */
/* printer manual to verify that these are correct for your printer   */
NUL EQU 00           /* Nul character                                 */
BEL EQU 07           /* Beeper                                        */
BAK EQU 08           /* Back Space                                    */
TAB EQU 09           /* Tab                                           */
LFF EQU 0A           /* Line Feed                                     */
VTB EQU 0B           /* Vertical Tab                                  */
FFF EQU 0C           /* Form Feed                                     */
CRR EQU 0D           /* Carriage Return                               */
```

Building PC400 PDTs

```
            P05 EQU 1B 57 01      /* 5 Pitch-Characters/inch                    */
                                  /* Same as Double Wide                        */
            SEL EQU 11            /* Select Printer                             */
            P10 EQU 12            /* 10 Pitch-Characters/inch                   */
            CDW EQU 1B 57 00      /* Cancel Double Wide contin.                 */
            CDL EQU 14            /* Cancel Double Wide line                    */
            ESC EQU 1B            /* Escape                                     */
            CAN EQU 18            /* Cancel Data                                */
            SPA EQU 20            /* Space                                      */
            P17 EQU 12 0F         /* 17.1 Pitch-Characters/inch                 */
            CS2 EQU 1B 36         /* Select Character Set 2                     */
            CS1 EQU 1B 37         /* Select Character Set 1                     */
            P12 EQU 1B 3A         /* 12 Pitch-characters/inch                   */
            SVT EQU 1B 42         /* Set Vertical Tabs                          */
            SFL EQU 1B 43 00      /* Set Form Length                            */
            SHT EQU 1B 44         /* Set Horizontal Tabs                        */
            SDS EQU 1B 47         /* Start Double Strike                        */
            CDS EQU 1B 48         /* Cancel Double Strike                       */
            SSP EQU 1B 4E         /* Set skip perforation                       */
            CSP EQU 1B 4F         /* Cancel skip perforation                    */
            CAT EQU 1B 52         /* Cancel all tabs Clears VT                  */
                                  /* and sets HT every 8 position               */
            CSS EQU 1B 54         /* Cancel Subscript or Superscript            */
            SSO EQU 1B 53 00      /* Set Superscript over                       */
            SSU EQU 1B 53 01      /* Set Subscript under                        */
            SUL EQU 1B 2D 01      /* Start Underline                            */
            CUL EQU 1B 2D 00      /* Cancel Underline                           */
            SCP EQU 1B 5B 54 04 00 00 00 /* ESC T - select code page           */
            CP8 EQU 1B 5B 54 04 00 00 00 03 52    /* select code page 850       */
            CP4 EQU 1B 5B 54 04 00 00 00 01 B5    /* select code page 437       */
            LL2 EQU 1B 41 24 1B 32 /* Setline length 2 lines/inch              */
            LL3 EQU 1B 41 18 1B 32 /* Setline length 3 lines/inch              */
            LL4 EQU 1B 41 12 1B 32 /* Setline length 4 lines/inch              */
            LL6 EQU 1B 41 0C 1B 32 /* Set line length 6 lines/inch             */
            LL8 EQU 1B 41 09 1B 32 /* Set line length 8 lines/inch             */
            LL0 EQU 1B 41 07 1B 32 /* Set line length 10 lines/inch            */
                                  /* actually 7/72 inch                         */
            SD1 EQU 1B 5B 46 05 00 00 01 01 00 00 /* Select Drawer 1            */
            SD2 EQU 1B 5B 46 05 00 00 01 02 00 00 /* Select Drawer 2            */
            ENV EQU 1B 5B 46 05 00 00 02 00 00 00 /* Select Envelope           */
            FRM EQU 1B 64         /* Forward Relative Movement                  */
            VLF EQU 1B 4A         /* Vertical Line Feed 1/216 inch units        */
            SPO EQU 1B 6B         /* Set Portrait Orientation                   */
            SLO EQU 1B 6C         /* Set Landscape Orientation                  */
            SFG EQU 1B 5B 49 08 00 /* Set Font Global                          */
            END_MACROS
            /*              Session Parameters                                  */
            /* These parameters determine the way in which output will be       */
            /* formatted for your printer.                                      */
            /* Numeric Parameters                                               */
            /* These parameters should be defined with a two digit hex number   */
            /* or a three digit decimal number.  The range of the number is zero */
            /* to 255 (decimal).                                                */
            MAXIMUM_PAGE_LENGTH=066               /* Printed lines per page     */
            MAXIMUM_PRINT_POSITION=080            /* Printed characters per line */
            INTERV_REQ_TIMER=001
            RESELECT_TIME_EXCPT_5204=001
            INTERV_TIMER_ON_PE_ONLY?=NO
            HORIZONTAL_PEL=120
            VERTICAL_PEL=216
            LINE_SPACING_RATIO=072
            PAGE_LENGTH_TYPE?=INCH                /* SET_PAGE_LENGTH "value" is */
                                                  /* values                     */
            /* YES/NO Parameters                                                */
            /* These parameters should be defined with either "YES" or "NO" on the*/
            /* right hand side of the '='                                       */
            COMPRESS_LINE_SPACING?=NO             /* Should blank or null lines */
```

```
                                        /* be printed?              */
FORM_FEED_ANY_POSITION?=YES          /* Should the form feed be    */
                                        /* valid in any position?   */
OVERRIDE_FORMATTED_PRINT?=YES        /* Should nulls be printed as */
                                        /* blanks?                  */
AUTO_NEWLINE_AT_MAX_POS?=NO
/*                       Control Codes                             */
/* These definitions tell the emulator what control strings to send to*/
/* your printer to issue control commands.                         */
/*                                                                 */
/* Format                                                          */
/* The name of the control command should always be at the beginning */
/* of a line followed by a '=' and then a definition string.       */
/* A Definition String is any combination of macro names, hexadecimal */
/* numbers, and characters separated by blanks.  A macro must have */
/* previously defined in the macro definitions section above. A    */
/* hexadecimal number must be two digits (0,..,F) long. and a      */
/* character must be preceded and followed by a blank.  If a       */
/* definition string will not fit on a line, it may be continued   */
/* as many lines as you wish by ending each line except the last with */
/* a comma;  ','.   You made add any comments you wish to by including*/
/* them between a slash*  and a *slash where slash is the symbol /.  */


/* START_JOB is the control string which will be sent to your printer */
/* at the beginning of each print job.                             */
START_JOB=SEL CDW CDL CUL CDS CP8 CS2
/* END_JOB is the string which will be sent to your printer at the end*/
/* of each print job.                                              */
END_JOB=CAT CDW CDL CUL CDS CP4 FFF
BACKSPACE=BAK
BEL=BEL
CARRIAGE_RETURN=CRR
NEW_LINE=CRR LFF
LINE_FEED=LFF
FORM_FEED=FFF
HORIZONTAL_TAB=TAB
VERTICAL_TAB=VTB
START_SUBSCRIPT=SSU
END_SUBSCRIPT=CSS
START_SUPERSCRIPT=SSO
END_SUPERSCRIPT=CSS
DUP=*
FIELD_MARK=;
/* The following commands specify control codes for which most PC  */
/* printers require command strings which contain a variable value */
/* or values somewhere in the middle of the string.                */
/* Place the word "value(s)" in the position of your definition    */
/* string where the Personal Communications 5250 should fill in    */
/* the hexadecimal value(s) indicated.                             */
/* For example, on the IBM Proprinter, the SET_HORIZONTAL_TABS     */
/* definition is:                                                  */
/* SET_HORIZONTAL_TABS=ESC D values NUL                            */
SET_HORIZONTAL_TABS=SHT values NUL   /* "values" are the tab stops */
                                        /* in column numbers        */
SET_VERTICAL_TABS=SVT values NUL     /* "values" are the tab stops */
                                        /* in line numbers          */
SET_HORIZONTAL_MARGINS=
SET_PAGE_LENGTH=SFL value             /* "value"=inch of the page   */
SET_AUTO_PERFORATION_SKIP=SSP value
                                        /* "value"=number of lines to */
                                        /* skip over the perforation */
                                        /* between pages.  Used to set */
                                        /* top and bottom margins.   */
SET_VARIABLE_LINE_DENSITY=ESC A value ESC 2
                                        /* "value"=number of points. */
                                        /* A point is                */
                                        /* 1/(LINE_SPACING_RATIO) inch.*/
```

Building PC400 PDTs

```
                   SET_CHARACTER_SET=
                   /*SET_CHARACTER_SET=ESC I NULL  selects the normal font          */
                   /*SET_CHARACTER_SET=ESC I 02  selects the NLQ (near letter quality)  */
                   /*SET_CHARACTER_SET=ESC I 04  selects the normal downloaded font   */
                   /*SET_CHARACTER_SET=ESC I 06  selects the NLQ downloaded font      */
                   /*SET_CHARACTER_SET=CS1       selects the Character set 1          */
                   /*SET_CHARACTER_SET=CS2       selects the Character set 2          */

                   /* These control codes set the printer lines per inch and characters  */
                   /* per inch to fixed amounts.                                      */
                   /* If your printer does not support setting the line density in points*/
                   /* then you can enter control strings for the following commands.   */
                   /* When Personal Communications 5250 gets a command from the host to  */
                   /* set the lines per  inch, it will round it to the closest line per  */
                   /* inch setting that you provide.  Note that if you provide a command */
                   /* for the SET_VARIABLE_LINE_DENSITY command above that it will be    */
                   /* used and any control strings you provide for the set lines per inch*/
                   /* commands below will not be used.                                */

                   SET_2_LINES_PER_INCH=LL2
                   SET_3_LINES_PER_INCH=LL3
                   SET_4_LINES_PER_INCH=LL4
                   SET_6_LINES_PER_INCH=LL6
                   SET_8_LINES_PER_INCH=LL8
                   SET_10_LINES_PER_INCH=LL0          /* 7/72 inch or 9/96 inch        */
                   SET_10_CHARACTERS_PER_INCH=P10
                   SET_12_CHARACTERS_PER_INCH=P12
                   SET_13_CHARACTERS_PER_INCH=        /*                             */
                   SET_15_CHARACTERS_PER_INCH=        /* The proprinter does not     */
                                                       /* support 15 pitch except in  */
                                                       /* graphic mode                */
                   SET_17_CHARACTERS_PER_INCH=P17     /* Condensed mode              */
                   SET_20_CHARACTERS_PER_INCH=
                   START_DOUBLE_WIDTH_CHARACTERS=P05
                   END_DOUBLE_WIDTH_CHARACTERS=CDW

                   /* These control codes are used to select the source drawer number  */
                   /* when your printer has the dual drawer sheetfeed option.          */
                   SELECT_DRAWER1=SD1
                   SELECT_DRAWER2=SD2
                   SELECT_DRAWER3=
                   SELECT_ENVELOPE=ENV                /* Envelope                    */

                   /* These control codes select the print mode (quality of print).   */
                   SELECT_DRAFT_QUALITY=
                   SELECT_LETTER_QUALITY=
                   SELECT_ENHANCED_QUALITY=
                   SELECT_SETUP_QUALITY=

                   /* These control codes                                          */
                   SET_DUPLEX=
                   SET_DUPLEX_TUMBLE=
                   RESET_DUPLEX=

                   /* These control codes set page orientation                      */
                   SET_PORTRAIT_ORIENT=SPO
                   SET_LANDSCAPELEFT_ORIENT=SLO
                   SET_PORTRAITUPDWN_ORIENT=SPO
                   SET_LANDSCAPERGHT_ORIENT=SLO

                   /* These control codes move the print position (Horizontal/Vertical)  */
                   FORWARD_HORIZONTAL_SKIP=FRM word-value(LH)
                   FORWARD_VERTICAL_STEP_FEED=VLF byte-value

                   /* These control codes select the printer font via global font ID   */
                   SET_FONT_GLOBAL=
                   SET_GFID_0003=SFG 00 03 00 90 01 01 03 52 CDW    /* OCR-B.10        */
```

```
SET_GFID_0005=SFG 00 05 00 90 01 01 03 52 CDW   /* Orator.10           */
SET_GFID_0011=SFG 00 0B 00 90 01 01 03 52 CDW   /* Courier.10          */
SET_GFID_0012=SFG 00 0C 00 90 01 01 03 52 CDW   /* Prestige.10         */
SET_GFID_0013=SFG 00 0B 00 90 01 01 03 52 CDW   /* Artisan.10          */
SET_GFID_0018=SFG 00 12 00 90 01 01 03 52 CDW   /* Courier.Italic.10  */
SET_GFID_0019=SFG 00 13 00 90 01 01 03 52 CDW   /* OCR-A.10            */
SET_GFID_0020=SFG 00 14 00 90 01 01 03 52 CDW   /* Pica.10             */
SET_GFID_0030=SFG 00 1E 00 90 01 01 03 52 CDW   /* Math-Symbol.10      */
SET_GFID_0038=SFG 00 26 00 90 01 01 03 52 CDW   /* Orator.Bold.10      */
SET_GFID_0039=SFG 00 27 00 90 01 01 03 52 CDW   /* Gothic.Bold.10      */
SET_GFID_0040=SFG 00 28 00 90 01 01 03 52 CDW   /* Gothic-Text.10      */
SET_GFID_0041=SFG 00 29 00 90 01 01 03 52 CDW   /* Roman-text.10       */
SET_GFID_0042=SFG 00 2A 00 90 01 01 03 52 CDW   /* Serif-text.10       */
SET_GFID_0043=SFG 00 2B 00 90 01 01 03 52 CDW   /* Serif-text.Italic.10*/
SET_GFID_0044=SFG 00 2C 00 90 01 01 03 52 CDW   /* Katakana-gothic.10 */
SET_GFID_0045=SFG 00 2D 00 90 01 01 03 52 CDW   /* APL.10              */
SET_GFID_0046=SFG 00 2E 00 90 01 01 03 52 CDW   /* Courier.Bold.10     */
SET_GFID_0050=SFG 00 32 00 90 01 01 03 52 CDW   /* Shalom.10           */
SET_GFID_0066=SFG 00 42 00 78 01 01 03 52 CDW   /* Gothic-text.12      */
SET_GFID_0068=SFG 00 44 00 78 01 01 03 52 CDW   /* Gothic-text.Italic.12*/
SET_GFID_0069=SFG 00 45 00 78 01 01 03 52 CDW   /* Gothic.Bold.12      */
SET_GFID_0070=SFG 00 46 00 78 01 01 03 52 CDW   /* Serif-text.12       */
SET_GFID_0071=SFG 00 47 00 78 01 01 03 52 CDW   /* Serif-text.Italic.12*/
SET_GFID_0072=SFG 00 48 00 78 01 01 03 52 CDW   /* Serif.Bold.12       */
SET_GFID_0080=SFG 00 73 00 78 01 01 03 52 CDW   /* Math-Symbol.12      */
SET_GFID_0084=SFG 00 54 00 78 01 01 03 52 CDW   /* Script.12           */
SET_GFID_0085=SFG 00 55 00 78 01 01 03 52 CDW   /* Courier.12          */
SET_GFID_0086=SFG 00 56 00 78 01 01 03 52 CDW   /* Prestige.12         */
SET_GFID_0087=SFG 00 57 00 78 01 01 03 52 CDW   /* Letter-gothic.12   */
SET_GFID_0091=SFG 00 70 00 78 01 01 03 52 CDW   /* Light.Italic.12    */
SET_GFID_0107=SFG 00 55 00 78 01 01 03 52 CDW   /* Courier.12          */
SET_GFID_0110=SFG 00 6E 00 78 01 01 03 52 CDW   /* Letter-Gothic.Bold.12*/
SET_GFID_0111=SFG 00 6F 00 78 01 01 03 52 CDW   /* Prestige-Elite.Bold.12*/
SET_GFID_0112=SFG 00 70 00 78 01 01 03 52 CDW   /* Prestige.Italic.12 */
SET_GFID_0115=SFG 00 73 00 78 01 01 03 52 CDW   /* Math-Symbol.12      */
SET_GFID_0155=SFG 00 9B 00 78 02 01 03 52 CDW   /* Boldface.Italic.PSM*/
SET_GFID_0158=SFG 00 9E 00 78 02 01 03 52 CDW   /* Modern.PSM          */
SET_GFID_0159=SFG 00 9F 00 78 02 01 03 52 CDW   /* Document.PSM        */
SET_GFID_0160=SFG 00 A0 00 78 02 01 03 52 CDW   /* Essay.PSM           */
SET_GFID_0162=SFG 00 A2 00 78 02 01 03 52 CDW   /* Essay.Italic.PSM   */
SET_GFID_0163=SFG 00 A3 00 78 02 01 03 52 CDW   /* Essay.Bold.PSM      */
SET_GFID_0168=SFG 00 A8 00 78 02 01 03 52 CDW   /* Barak.PSM           */
SET_GFID_0173=SFG 00 AD 00 78 02 01 03 52 CDW   /* Essay.Light.PSM    */
SET_GFID_0175=SFG 00 AF 00 78 02 01 03 52 CDW   /* Document.PSM        */
SET_GFID_0176=SFG 00 B0 00 78 02 01 03 52 CDW   /* Boldface.PSM        */
SET_GFID_0177=SFG 00 9B 00 78 02 01 03 52 CDW   /* Boldface.Italic.PSM*/
SET_GFID_0193=SFG 00 73 00 78 01 01 03 52 CDW   /* Math-Symbol.12      */
SET_GFID_0198=SFG 00 1E 00 90 01 01 03 52 CDW   /* Math-Symbol.10      */
SET_GFID_0204=SFG 00 CC 00 6C 01 01 03 52 CDW   /* Gothic-text.13      */
SET_GFID_0221=SFG 00 DD 00 60 01 01 03 52 CDW   /* Prestige.15         */
SET_GFID_0222=SFG 00 E6 00 60 01 01 03 52 CDW   /* Gothic-text.15      */
SET_GFID_0223=SFG 00 DF 00 60 01 01 03 52 CDW   /* Courier.15          */
SET_GFID_0225=SFG 00 E1 00 60 01 01 03 52 CDW   /* Math-symbol.15      */
SET_GFID_0229=SFG 00 E5 00 60 01 01 03 52 CDW   /* Serif-text.15       */
SET_GFID_0230=SFG 00 E6 00 60 01 01 03 52 CDW   /* Gothic-text.15      */
SET_GFID_0245=SFG 00 2E 00 90 01 01 03 52 P05   /* Courier.Bold.5      */
SET_GFID_0252=SFG 00 FC 00 54 01 01 03 52 CDW   /* Courier.15          */
SET_GFID_0253=SFG 00 FD 00 54 01 01 03 52 CDW   /* Courier.Bold.17     */
SET_GFID_0254=SFG 00 FE 00 55 01 01 03 52 CDW   /* Courier.17          */
SET_GFID_0280=SFG 01 18 00 48 01 01 03 52 CDW   /* APL.20              */
SET_GFID_0281=SFG 01 19 00 48 01 01 03 52 CDW   /* Gothic-text.20      */
SET_GFID_0290=SFG 01 22 00 36 01 01 03 52 CDW   /* Gothic-text.27      */
SET_GFID_0751=SFG 11 37 00 A0 01 03 03 52 CDW   /* Sonoran-serif.8pt  */
SET_GFID_1051=SFG 11 37 00 C8 01 03 03 52 CDW   /* Sonoran-serif.10pt */
SET_GFID_1053=SFG 11 4B 00 C8 01 03 03 52 CDW   /* Sonoran-serif.bold.10pt*/
SET_GFID_1056=SFG 11 B7 00 C8 01 03 03 52 CDW   /* Sonoran-serif.italic.10pt*/
SET_GFID_1351=SFG 11 37 00 F0 01 03 03 52 CDW   /* Sonoran-serif.12pt  */
```

Building PC400 PDTs

```
SET_GFID_1653=SFG 11 4B 01 40 01 03 03 52 CDW    /* Sonoran-serif.Bold.16pt*/
SET_GFID_2103=SFG 11 4B 01 E0 01 03 03 52 CDW    /* Sonoran-serif.Bold.24pt*/

/*                         Color Specifications                         */
START_COLOR_BLUE=
END_COLOR_BLUE=
START_COLOR_GREEN=
END_COLOR_GREEN=
START_COLOR_CYAN=
END_COLOR_CYAN=
START_COLOR_RED=
END_COLOR_RED=
START_COLOR_MAGENTA=
END_COLOR_MAGENTA=
START_COLOR_YELLOW=
END_COLOR_YELLOW=
START_COLOR_BLACK=
END_COLOR_BLACK=
START_COLOR_WHITE=
END_COLOR_WHITE=

/*                       Highlight Specifications                       */
/* These definitions will determine how things which are sent by the   */
/* host to be displayed or printed as underlined, reverse video, or     */
/* blinking will be highlighted on your printer.                        */

START_HIGHLIGHT_INTENSE=SDS              /* This is double strike        */
END_HIGHLIGHT_INTENSE=CDS
START_HIGHLIGHT_UNDERLINE=SUL
END_HIGHLIGHT_UNDERLINE=CUL
START_HIGHLIGHT_REVERSE_VIDEO=
END_HIGHLIGHT_REVERSE_VIDEO=
START_HIGHLIGHT_BLINK=
END_HIGHLIGHT_BLINK=

/*                        Character Definitions                        */
SPACE=SPA
EXCLAMATION_POINT=21
QUOTATION_MARKS=22
NUMBER_SIGN=23
DOLLAR_SIGN=24
PERCENT_SIGN=25
AMPERSAND=26
APOSTROPHE=27
LEFT_PARENTHESIS=28
RIGHT_PARENTHESIS=29
ASTERISK=2A
PLUS_SIGN=2B
COMMA=2C
HYPHEN=2D
PERIOD=2E
SLASH=2F
ZERO=0
ONE=1
TWO=2
THREE=3
FOUR=4
FIVE=5
SIX=6
SEVEN=7
EIGHT=8
NINE=9
COLON=3A
SEMICOLON=3B
LESS_THAN_SIGN=3C
EQUAL_SIGN=3D
GREATER_THAN_SIGN=3E
```

```
QUESTION_MARK=3F
AT_SIGN=40
A_CAPITAL=A
B_CAPITAL=B
C_CAPITAL=C
D_CAPITAL=D
E_CAPITAL=E
F_CAPITAL=F
G_CAPITAL=G
H_CAPITAL=H
I_CAPITAL=I
J_CAPITAL=J
K_CAPITAL=K
L_CAPITAL=L
M_CAPITAL=M
N_CAPITAL=N
O_CAPITAL=O
P_CAPITAL=P
Q_CAPITAL=Q
R_CAPITAL=R
S_CAPITAL=S
T_CAPITAL=T
U_CAPITAL=U
V_CAPITAL=V
W_CAPITAL=W
X_CAPITAL=X
Y_CAPITAL=Y
Z_CAPITAL=Z
LEFT_BRACKET=5B
BACKSLASH=5C
RIGHT_BRACKET=5D
CIRCUMFLEX_ACCENT=5E
UNDERLINE=5F
GRAVE_ACCENT=60
A_SMALL=a
B_SMALL=b
C_SMALL=c
D_SMALL=d
E_SMALL=e
F_SMALL=f
G_SMALL=g
H_SMALL=h
I_SMALL=i
J_SMALL=j
K_SMALL=k
L_SMALL=l
M_SMALL=m
N_SMALL=n
O_SMALL=o
P_SMALL=p
Q_SMALL=q
R_SMALL=r
S_SMALL=s
T_SMALL=t
U_SMALL=u
V_SMALL=v
W_SMALL=w
X_SMALL=x
Y_SMALL=y
Z_SMALL=z
LEFT_BRACE=7B
VERTICAL_BAR=7C
RIGHT_BRACE=7D
TILDE_ACCENT=7E
C_CEDILLA_CAPITAL=80
U_DIAERESIS_SMALL=81
E_ACUTE_SMALL=82
```

Building PC400 PDTs

```
A_CIRCUMFLEX_SMALL=83
A_DIAERESIS_SMALL=84
A_GRAVE_SMALL=85
A_OVERCIRCLE_SMALL=86
C_CEDILLA_SMALL=87
E_CIRCUMFLEX_SMALL=88
E_DIAERESIS_SMALL=89
E_GRAVE_SMALL=8A
I_DIAERESIS_SMALL=8B
I_CIRCUMFLEX_SMALL=8C
I_GRAVE_SMALL=8D
A_DIAERESIS_CAPITAL=8E
A_OVERCIRCLE_CAPITAL=8F
E_ACUTE_CAPITAL=90
AE_DIPTHONG_SMALL=91
AE_DIPTHONG_CAPITAL=92
O_CIRCUMFLEX_SMALL=93
O_DIAERESIS_SMALL=94
O_GRAVE_SMALL=95
U_CIRCUMFLEX_SMALL=96
U_GRAVE_SMALL=97
Y_DIAERESIS_SMALL=98
O_DIAERESIS_CAPITAL=99
U_DIAERESIS_CAPITAL=9A
O_SLASH_SMALL=9B
POUND_SIGN=9C
O_SLASH_CAPITAL=9D
MULTIPLY_SIGN=9E
A_ACUTE_SMALL=A0
I_ACUTE_SMALL=A1
O_ACUTE_SMALL=A2
U_ACUTE_SMALL=A3
N_TILDE_SMALL=A4
N_TILDE_CAPITAL=A5
ORDINAL_INDICATOR_FEMININE=A6
ORDINAL_INDICATOR_MASCULINE=A7
QUESTION_MARK_INVERTED=A8
REGISTERED_TRADEMARK_SYMBOL=A9
LOGICAL_NOT=AA
ONE_HALF=AB
ONE_QUARTER=AC
EXCLAMATION_POINT_INVERTED=AD
LEFT_ANGLE_QUOTES=AE
RIGHT_ANGLE_QUOTES=AF
A_ACUTE_CAPITAL=B5
A_CIRCUMFLEX_CAPITAL=B6
A_GRAVE_CAPITAL=B7
COPYRIGHT_SYMBOL=B8
CENT_SIGN=BD
YEN_SIGN=BE
A_TILDE_SMALL=C6
A_TILDE_CAPITAL=C7
INTERNATIONAL_CURRENCY_SYMBOL=CF
ETH_ICELANDIC_SMALL=D0
ETH_ICELANDIC_CAPITAL=D1
E_CIRCUMFLEX_CAPITAL=D2
E_DIAERESIS_CAPITAL=D3
E_GRAVE_CAPITAL=D4
I_DOTLESS_SMALL=D5
I_ACUTE_CAPITAL=D6
I_CIRCUMFLEX_CAPITAL=D7
I_DIAERESIS_CAPITAL=D8
VERTICAL_LINE_BROKEN=DD
I_GRAVE_CAPITAL=DE
O_ACUTE_CAPITAL=E0
SHARP_S_SMALL=E1
O_CIRCUMFLEX_CAPITAL=E2
```

```
O_GRAVE_CAPITAL=E3
O_TILDE_SMALL=E4
O_TILDE_CAPITAL=E5
MICRO_SYMBOL=E6
THORN_ICELANDIC_SMALL=E7
THORN_ICELANDIC_CAPITAL=E8
U_ACUTE_CAPITAL=E9
U_CIRCUMFLEX_CAPITAL=EA
U_GRAVE_CAPITAL=EB
Y_ACUTE_SMALL=EC
Y_ACUTE_CAPITAL=ED
OVERLINE=EE
ACUTE_ACCENT=EF
SYLLABLE_HYPHEN=F0
PLUS_OR_MINUS_SIGN=F1
THREE_QUARTERS=F3
PARAGRAPH_SYMBOL=F4
SECTION_SYMBOL=F5
DIVIDE_SIGN=F6
CEDILLA=F7
DEGREE_SYMBOL=F8
DIAERESIS=F9
MIDDLE_DOT_ACCENT=FA
ONE_SUPERSCRIPT=FB
THREE_SUPERSCRIPT=FC
TWO_SUPERSCRIPT=FD
REQUIRED_SPACE=SPA
/*******************************************************************/
/*                 PC5250 Internal Data Area.                      */
/* Do not change these statements.                                 */
/*******************************************************************/
TOP_MARGIN=
LEFT_MARGIN=
DYNAMIC_START_JOB=00 00 00 00 00 00 00 00 00 00 00
DYNAMIC_END_JOB=00 00
DYNAMIC_SET_PAGE_LENGTH=00 00 00
PRINTER_ID=40 19
/*                    End of Definition File                       */
```

IBM5577.PDF File Contents

```
/*******************************************************************/
/*     PDF FILE (PRINTER DEFINITION FILE) FOR: PS/55 Printer       */
/*******************************************************************/
BEGIN_MACROS
NUL EQU 00
BEL EQU 07
BAK EQU 08
TAB EQU 09
LFF EQU 0A
VTB EQU 0B
FFF EQU 0C
CRR EQU 0D
SEL EQU 11
DC3 EQU 13
ESC EQU 1B
CAN EQU 18
SPA EQU 20
P10 EQU 1B 7E 02 00 01 32
P12 EQU 1B 7E 02 00 01 3C
P13 EQU 1B 7E 02 00 01 43
P15 EQU 1B 7E 02 00 01 4B
SDW EQU 1B 7E 0E 00 01 09
EDW EQU 1B 7E 0E 00 01 0A
SVT EQU 1B 7E 19
SHT EQU 1B 7E 18
CSS EQU 1B 7E 0E 00 01 0F
SSO EQU 1B 7E 0E 00 01 0D
```

Building PC400 PDTs

```
                    SSU EQU 1B 7E 0E 00 01 0E
                    SUL EQU 1B 7E 11 00 01 01
                    CUL EQU 1B 7E 11 00 01 00
                    LL2 EQU 1B 7E 03 00 01 14
                    LL3 EQU 1B 7E 03 00 01 1E
                    LL4 EQU 1B 7E 03 00 01 28
                    LL6 EQU 1B 7E 03 00 01 3C
                    LL7 EQU 1B 7E 03 00 01 4B
                    LL8 EQU 1B 7E 03 00 01 50
                    SPL EQU 1B 7E 04 00 03 00
                    INZ EQU 1B 7E 01 00 00
                    EJC EQU 1B 7E 0E 00 01 06
                    END_MACROS
/********************************************************************/
/*                      Session Parameters                          */
/********************************************************************/
MAXIMUM_PAGE_LENGTH=066
MAXIMUM_PRINT_POSITION=132
DEFAULT_CPI?=010
DEFAULT_LPI?=006
COMPRESS_LINE_SPACING?=NO
FORM_FEED_ANY_POSITION?=YES
OVERRIDE_FORMATTED_PRINT?=YES
HORIZONTAL_PEL=180
VERTICAL_PEL=120
UNITS_OF_DRAW_LINE=
KANJI_CODE?=SHIFT_JIS
ZENKAKU_SPACE=
PAGE_LENGTH_TYPE?=6INCH
/********************************************************************/
/*                        Control Codes                             */
/********************************************************************/
START_JOB=INZ SEL LL6 P10
END_JOB=INZ
BACKSPACE=BAK
BEL=BEL
CARRIAGE_RETURN=CRR
NEW_LINE=CRR LFF
LINE_FEED=LFF
FORM_FEED=EJC
HORIZONTAL_TAB=TAB
VERTICAL_TAB=VTB
DESELECT=DC3
START_SUBSCRIPT=SSU
END_SUBSCRIPT=CSS
START_SUPERSCRIPT=SSO
END_SUPERSCRIPT=CSS
DUP=*
FIELD_MARK=;
SET_HORIZONTAL_TABS=SHT length(HL)-values
SET_VERTICAL_TABS=SVT length(HL)-values
SET_HORIZONTAL_MARGINS=
SET_PAGE_LENGTH=SPL word-value(HL)
SET_2_LINES_PER_INCH=LL2
SET_3_LINES_PER_INCH=LL3
SET_4_LINES_PER_INCH=LL4
SET_6_LINES_PER_INCH=LL6
SET_7.5_LINES_PER_INCH=LL7
SET_8_LINES_PER_INCH=LL8
SET_10_CHARACTERS_PER_INCH=P10
SET_12_CHARACTERS_PER_INCH=P12
SET_13.4_CHARACTERS_PER_INCH=P13
SET_15_CHARACTERS_PER_INCH=P15
START_DOUBLE_WIDTH_CHARACTERS=SDW
END_DOUBLE_WIDTH_CHARACTERS=EDW
IMAGE_TRANSMISSION=1B 25 31 length(HL)-images
FORWARD_HORIZONTAL_SKIP=1B 25 33 word-value(HL)
```

```
          FORWARD_VERTICAL_STEP_FEED=1B 25 35 word-value(HL)
          SET_FONT_SIZE=1B 7E 20 00 03 word-value(HL) 02
          SET_TATEGAKI_MODE=1B 7E 0E 00 01 0B
          RESET_TATEGAKI_MODE=1B 7E 0E 00 01 0C
          SBCS_FONT_LOAD=1B 7E 81 00 28 F0 40 00 18 byte-values F0 40
          DBCS_FONT_LOAD=1B 7E 81 00 4C F0 40 02 18 byte-values F0 40
          SELECT_DRAWER=
          SET_LOCAL_FONT=
          RESET_LOCAL_FONT=
          ABS_HORIZONTAL_COLUMN_SKIP=1B 7E 1C 00 02 00 byte-value
          REL_HOR_COLUMN_SKIP_TO_RIGHT=1B 7E 1C 00 02 01 byte-value
          SET_SOLID_LINE_TYPE=
          SET_DOTTED_LINE_TYPE=
          SET_LINE_WIDTH_THIN=
          SET_LINE_WIDTH_BOLD=
          DRAW_LINE=
          KANJI_ON=
          KANJI_OFF=
          ATTRIBUTE_GRID_LINE=1B 7E 16 length(HL)-values
          /*******************************************************************/
          /*                  Highlight Specifications             */
          /*******************************************************************/
          START_HIGHLIGHT_INTENSE=ESC 7E 0E 00 01 17
          END_HIGHLIGHT_INTENSE=ESC 7E 0E 00 01 18
          START_HIGHLIGHT_UNDERLINE=1B 7E 11 00 01 01
          END_HIGHLIGHT_UNDERLINE=1B 7E 11 00 01 00
          /*******************************************************************/
          /*                  Internal Data Area.                  */
          /* Do not change these statement.                        */
          /*******************************************************************/
          TOP_MARGIN=
          LEFT_MARGIN=
          DYNAMIC_START_JOB=00 00 00 00 00 00 00 00 00 00 00
          DYNAMIC_END_JOB=00 00
          DYNAMIC_SET_PAGE_LENGTH=00 00 00
          PRINTER_ID=55 77
          /*                  End of Definition File               */
```

Field Names of Printer Definition Files

The following table lists the field names of the printer definition files (PDF files) and their meanings:

Table 18. Field Names of PDF Files

Field Name	Meaning	Remarks
MAXIMUM_PAGE_LENGTH	Default MPL	Default is 66
MAXIMUM_PRINT_POSITION	Default MPP	Default is 132
DEFAULT_CPI?	Default CPI	Default is 10
DEFAULT_LPI?	Default LPI	Default is 6
COMPRESS_LINE_SPACING?	Specifies whether to print a line containing only space characters or nonprint characters.	
FORM_FEED_ANY_POSITION?	Specifies whether to validate the FF control code on the first line.	Do not change this field.
OVERRIDE_FORMATTED_PRINT?	Specifies whether to print NULL characters as blanks.	Do not change this field.
HORIZONTAL_PEL	FORWARD_HORIZONTAL_SKIP length unit	
VERTICAL_PEL	FORWARD_VERTICAL_SKIP length unit	

Building PC400 PDTs

Table 18. Field Names of PDF Files (continued)

Field Name	Meaning	Remarks
IMAGE_HORIZONTAL_PEL	IMAGE_TRANSMISSION horizontal length unit	
IMAGE_VERTICAL_PEL	IMAGE_TRANSMISSION vertical length unit	
LINE_SPACING_RATIO	SET_VARIABLE_LINE_DENSITY length unit	
PAGE_LENGTH_TYPE?	SET_PAGE _LENGTH page length parameter type	
FIRST_LEFT_POSITION	Distance from left paper edge	
FIRST_TOP_POSITION	Distance from top paper edge	
DRAWER1_ORIENTATION	Default page orientation for drawer 1	Default is `COR`
DRAWER2_ORIENTATION	Default page orientation for drawer 2	Default is `COR`
AUTOMATIC_ORIENTATION	Specifies whether to calculate the page orientation.	
START_JOB	Printer control code sent to a printer to start printing	If the control code specifying LPI/CPI is defined, also change DEFAULT_CPI, DEFAULT_LPI.
END_JOB	Printer control code sent to a printer when printing ends	
BACKSPACE	Backspace control code	
BEL	Bell control code	
CARRIAGE_RETURN	Carriage return control code	
NEW_LINE	New line (CR/LF) control code	
LINE_FEED	New line control code	
FORM_FEED	Form feed (FF) control code	
HORIZONTAL_TAB	Horizontal tab control code	
VERTICAL_TAB	Vertical tab control code	
DESELECT	Device control 3 control code	
START_SUBSCRIPT	Subscript character specification	
END_SUBSCRIPT	Subscript character specification release	
START_SUPERSCRIPT	Superscript character specification	
END_SUPERSCRIPT	Superscript character specification release	
DUP	Character used for printing DUP codes	
FIELD_MARK	Character used for printing FIELD MARK characters	
SET_HORIZONTAL_TABS	Horizontal tab setup	Do not change this field.
SET_VERTICAL_TABS	Vertical tab setup	Do not change this field.
SET_HORIZONTAL_MARGINS	Right and left margin setup	Do not change this field.
SET_PAGE_LENGTH	Page length setup	Assign the unit used for the defined control code page length to the PAGE_LENGTH_TYPE field. When single sheets are to be used, delete this field.

Table 18. Field Names of PDF Files (continued)

Field Name	Meaning	Remarks
SET_VARIABLE_LINE_DENSITY	Line density setup	Assign the unit used for the control code length defined to LINE_SPACING_RATIO field.
SET_2_LINES_PER_INCH	New line pitch (2LPI) setup	
SET_3_LINES_PER_INCH	New line pitch (3LPI) setup	
SET_4_LINES_PER_INCH	New line pitch (4LPI) setup	
SET_6_LINES_PER_INCH	New line pitch (6LPI) setup	
SET_8_LINES_PER_INCH	New line pitch (8LPI) setup	
SET_10_LINES_PER_INCH	New line pitch (10LPI) setup	
SET_10_CHARACTERS_PER_INCH	Character pitch (10CPI) setup	
SET_12_CHARACTERS_PER_INCH	Character pitch (12CPI) setup	
SET_13_CHARACTERS_PER_INCH	Character pitch (13CPI) setup	
SET_15_CHARACTERS_PER_INCH	Character pitch (15CPI) setup	
SET_17_CHARACTERS_PER_INCH	Character pitch (17CPI) setup	
SET_20_CHARACTERS_PER_INCH	Character pitch (20CPI) setup	
IMAGE_TRANSMISSION	Image data setup (vertical 24-dot image)	
SELECT_DRAWER1	Page tray (Primary) setup	
SELECT_DRAWER2	Page tray (Alternate) setup	
SELECT_ENVELOPE	Envelope tray setup	
SELECT_DRAFT_QUALITY	Draft print quality setup	
SELECT_LETTER_QUALITY	Letter print quality setup	
SELECT_ENHANCED_QUALITY	Enhanced print quality setup	
SET_DUPLEX	Duplex printing setup	
SET_DUPLEX_TUMBLE	Duplex (tumble) printing setup	
RESET_DUPLEX	Duplex printing release	
SET_PORTRAIT_ORIENT	Page orientation (Normal portrait (upright)) setup	
SET_LANDSCAPELEFT_ORIENT	Page orientation (Landscape left (270 degree clockwise rotation of text)) setup	
SET_PORTRAITUPDWN_ORIENT	Page orientation (Portrait upside down (180 degree clockwise rotation of text)) setup	
SET_LANDSCAPERGHT_ORIENT	Page orientation (Landscape right (90 degree clockwise rotation of text)) setup	
FORWARD_HORIZONTAL_SKIP	Variable skip (relative position/dot unit)	Assign the unit of the defined control code length to the HORIZONTAL_PEL field.
FORWARD_VERTICAL_STEP_FEED	Variable line feed (relative position/dot unit)	Assign the unit of the defined control code length to the VERTICAL_PEL field.
SET_FONT_GLOBAL	Global font ID setup	Do not change this field.
SET_GFID_0003	GFID 3 (OCR-B) setup	

Building PC400 PDTs

Table 18. Field Names of PDF Files (continued)

Field Name	Meaning	Remarks
SET_GFID_0005	GFID 5 (Orator) setup	
SET_GFID_0011	GFID 11 (Courier 10) setup	
SET_GFID_0012	GFID 12 (Prestige Pica) setup	
SET_GFID_0013	GFID 13 (Artisan 10) setup	
SET_GFID_0018	GFID 18 (Courier Italic 10) setup	
SET_GFID_0019	GFID 19 (OCR-A) setup	
SET_GFID_0020	GFID 20 (Pica) setup	
SET_GFID_0030	GFID 30 (Math Symbol 10) setup	
SET_GFID_0038	GFID 38 (Orator Bold) setup	
SET_GFID_0039	GFID 39 (Gothic Bold 10) setup	
SET_GFID_0040	GFID 40 (Gothic Text 10) setup	
SET_GFID_0041	GFID 41 (Roman Text 10) setup	
SET_GFID_0042	GFID 42 (Serif Text 10) setup	
SET_GFID_0043	GFID 43 (Serif Italic 10) setup	
SET_GFID_0044	GFID 44 (Katakana 10) setup	
SET_GFID_0045	GFID 45 (APL 10) setup	
SET_GFID_0046	GFID 46 (Courier Bold 10) setup	
SET_GFID_0050	GFID 50 (Shalom 10) setup	
SET_GFID_0066	GFID 66 (Gothic Text 12) setup	
SET_GFID_0068	GFID 68 (Gothic Italic 12) setup	
SET_GFID_0069	GFID 69 (Gothic Bold 12) setup	
SET_GFID_0070	GFID 70 (Serif Text 12) setup	
SET_GFID_0071	GFID 71 (Serif Italic 12) setup	
SET_GFID_0072	GFID 72 (Serif Bold 12) setup	
SET_GFID_0080	GFID 80 (Math Symbol 12) setup	
SET_GFID_0084	GFID 84 (Script 12) setup	
SET_GFID_0085	GFID 85 (Courier 12) setup	
SET_GFID_0086	GFID 86 (Prestige Elite) setup	
SET_GFID_0087	GFID 87 (Letter Gothic 12) setup	
SET_GFID_0091	GFID 91 (Light Italic 12) setup	
SET_GFID_0110	GFID 110 (Letter Gothic Bold 12) setup	
SET_GFID_0111	GFID 111 (Prestige Elite Bold) setup	
SET_GFID_0112	GFID 112 (Prestige Elite Italic) setup	
SET_GFID_0115	GFID 115 (Math Symbol 12) setup	
SET_GFID_0155	GFID 155 (Boldface Italic) setup	
SET_GFID_0158	GFID 158 (Modern) setup	
SET_GFID_0159	GFID 159 (Boldface) setup	
SET_GFID_0160	GFID 160 (Essay) setup	
SET_GFID_0162	GFID 162 (Essay Italic) setup	
SET_GFID_0163	GFID 163 (Essay Bold) setup	

Table 18. Field Names of PDF Files (continued)

Field Name	Meaning	Remarks
SET_GFID_0168	GFID 168 (Barak PSM) setup	
SET_GFID_0173	GFID 173 (Essay Light) setup	
SET_GFID_0175	GFID 175 (Document) setup	
SET_GFID_0176	GFID 176 (Boldface) setup	
SET_GFID_0177	GFID 177 (Boldface Italic) setup	
SET_GFID_0193	GFID 193 (Math Symbol 12) setup	
SET_GFID_0198	GFID 198 (Math Symbol 10) setup	
SET_GFID_0204	GFID 204 (Gothic Text 13) setup	
SET_GFID_0221	GFID 221 (Prestige 15) setup	
SET_GFID_0222	GFID 222 (Gothic Text 15) setup	
SET_GFID_0223	GFID 223 (Courier 15) setup	
SET_GFID_0225	GFID 225 (Math Symbol 15) setup	
SET_GFID_0229	GFID 229 (Serif Text 15) setup	
SET_GFID_0230	GFID 230 (Gothic Text 15) setup	
SET_GFID_0245	GFID 245 (Courier Bold 5) setup	
SET_GFID_0252	GFID 252 (Courier 17) setup	
SET_GFID_0253	GFID 253 (Courier Bold 17) setup	
SET_GFID_0254	GFID 254 (Courier 17 (sub/super)) setup	
SET_GFID_0280	GFID 280 (APL 20) setup	
SET_GFID_0281	GFID 281 (Gothic Text 20) setup	
SET_GFID_0290	GFID 290 (Gothic Text 27) setup	
SET_GFID_0751	GFID 751 (Sonoran-Serif 8-pt Roman Medium) setup	
SET_GFID_1051	GFID 1051 (Sonoran-Serif 10-pt Roman Medium) setup	
SET_GFID_1053	GFID 1053 (Sonoran-Serif 10-pt Roman Bold) setup	
SET_GFID_1056	GFID 1056 (Sonoran-Serif 10-pt Italic Medium) setup	
SET_GFID_1351	GFID 1351 (Sonoran-Serif 12-pt Roman Medium) setup	
SET_GFID_1653	GFID 1653 (Sonoran-Serif 16-pt Roman Bold) setup	
SET_GFID_2103	GFID 2103 (Sonoran-Serif 24-pt Roman Bold) setup	
START_HIGHLIGHT_INTENSE	Highlight printing setup	
END_HIGHLIGHT_INTENSE	Highlight printing release	
START_HIGHLIGHT_UNDERLINE	Underline setup	
END_HIGHLIGHT_UNDERLINE	Underline release	
TOP_MARGIN	Default top margin	Do not change this field.
LEFT_MARGIN	Default left margin	Do not change this field.

Building PC400 PDTs

Table 18. Field Names of PDF Files (continued)

Field Name	Meaning	Remarks
DYNAMIC_START_JOB	Printer control code sent to a printer to start printing (internal use)	Do not change this field.
DYNAMIC_END_JOB	Printer control code sent to a printer to stop printing (internal use)	Do not change this field.
DYNAMIC_SET_PAGE_LENGTH	Page length setup control code sent to a printer at the start of printing (internal use)	Do not change this field.
PRINTER_ID	Printer ID	Do not change this field.
KANJI_CODE?	Used kanji codes (JIS or SHIFT_JIS)	Do not change this field.
ZENKAKU_SPACE	The size (adjustment unit) of a user-defined character and a HANKAKU character	Do not change this field.
SBCS_FONT_LOAD	Registration of a HANKAKU GAIJI	Do not change this field.
DBCS_FONT_LOAD	Registration of a ZENKAKU GAIJI	Do not change this field.
SET_LOCAL_FONT	Set a font set of user-defined characters	Remove this field when user-defined characters are not loaded to a printer.
RESET_LOCAL_FONT	Reset a font set of user-defined characters	Remove this field when user-defined characters are not loaded to a printer.
KANJI_ON	Set kanji mode	
KANJI_OFF	Reset kanji mode	
ATTRIBUTE_GRID_LINE	Grid-line print	Do not change this field.
START_DOUBLE_WIDTH_CHARACTER	Set a double-width character	
END_DOUBLE_WIDTH_CHARACTER	Reset a double-width character	

Notes:

1. When using IBM5577.PDF, change FORM_FEED=EJC to FORM_FEED in the file when a continuous form job does not feed correctly.

2. When using ESC_P.PDF, DBCS (2-byte) characters are placed to the left of the character box, not in the center. To place DBCS characters in the center of the character box, change the IBM-supplied values in ESC_P.PDF:

```
P10 EQU 1B 50 1B 20 00 1C 53 00 0C 1C 54 00 00
P12 EQU 1B 4D 1B 20 00 1C 53 00 06 1C 54 00 00
P13 EQU 1B 67 1B 20 01 1C 53 00 03 1C 54 00 01
P15 EQU 1B 67 1B 20 00 1C 53 00 00 1C 54 00 00
```

to the following:

```
P10 EQU 1B 50 1B 20 00 1C 53 06 06 1C 54 00 00
P12 EQU 1B 4D 1B 20 00 1C 53 03 03 1C 54 00 00
P13 EQU 1B 67 1B 20 01 1C 53 02 01 1C 54 00 01
P15 EQU 1B 67 1B 20 00 1C 53 00 00 1C 54 00 00
```

The following table lists the session parameter field names and their effective values:

Table 19. Effective Values for PDF File Field Names

Field Name	Effective Value
MAXIMUM_PAGE_LENGTH	001 to 255
MAXIMUM_PRINT_POSITION	001 to 255

Table 19. Effective Values for PDF File Field Names (continued)

Field Name	Effective Value
DEFAULT_CPI?	010/012/015
DEFAULT_LPI?	004/006/008
COMPRESS_LINE_SPACING?	YES/NO
FORM_FEED_ANY_POSITION?	YES/NO
OVERRIDE_FORMATTED_PRINT?	YES/NO
HORIZONTAL_PEL	FORWARD_HORIZONTAL_SKIP length unit
VERTICAL_PEL	FORWARD_VERTICAL_ STEP_FEED length unit
IMAGE_HORIZONTAL_PEL	IMAGE_TRANSMISSION horizontal unit
IMAGE_VERTICAL_PEL	IMAGE_TRANSMISSION vertical unit
LINE_SPACING_RATIO	SET_VARIABLE_LINE_DENSITY length unit
PAGE_LENGTH_TYPE?	LINE/INCH/6INCH*
FIRST_LEFT_POSITION	000 to 1440 in units of 1/1440 inch
FIRST_TOP_POSITION	000 to 1440 in units of 1/1440 inch
DRAWER1_ORIENTATION	LANDSCAPE/PORTRAIT/COR
DRAWER2_ORIENTATION	LANDSCAPE/PORTRAIT/COR
AUTOMATIC_ORIENTATION	YES/NO

* 6/INCH indicates that page length should be specified in units of 1/6 inch.

Notes:

1. If one of the desired CPI/LPI settings is not exactly supported by the printer, set the nearest value. The results of printing might not be as desired.

2. If the units used to specify the control code length defined in FORWARD_HORIZONTAL _SKIP and FORWARD_VERTICAL _STEP_FEED are not the same as the units used to specify the HORIZONTAL_PEL and VERTICAL_PEL, the desired output will not be obtained.

3. If the units used to specify the control code length defined in SET_VARIABLE_LINE_DENSITY are not the same as the units used to specify the LINE_SPACING_RATIO, the desired output will not be obtained.

4. When FIRST_LEFT_POSITION and FIRST_TOP_POSITION are specified, their values are regarded as specifying the unprintable area in the page of the printer. These values are included in the top margin and the left margin specified by the AS/400 printer control code.

Parameter Definition of Printer Control Codes

This section explains how to define the parameters, if any, of the printer control codes. Some printer control codes require that parameters be specified. Pay particular attention when defining a parameter because how this is done depends on the parameter type. If the definition method for another type is used, the desired output will not be obtained.

Building PC400 PDTs

The following table lists printer control code parameters and their meanings:

Table 20. Printer Control Code Parameters

Parameter Type	Meaning
byte-value	One-byte parameter.
byte-values	Multibyte parameter. Used if operands are fixed.
word-value(HL)	One-word parameter (higher and lower bytes).
word-value(LH)	One-word parameter (lower and higher bytes).

length(HL)-values	Multibyte parameter requiring operands.
	    An operand consists of a higher and lower byte.
	    Used if operands are variable.

length(LH)-values	Multibyte parameter requiring operands.
	    An operand consists of a lower and higher byte.
	    Used if operands are variable.

length(HL)-images	Image data requiring operands. An operand consists of a higher and lower byte.
length(LH)-images	Image data requiring operands. An operand consists of a lower and higher byte.
decimal-characters	Decimal characters parameter.

Symbols of Printer Definition Files

The following table lists the symbols that are defined for printer definition files.

Table 21. Printer Symbol Definitions

Field Name	Symbol
SPACE	
EXCLAMATION_POINT	!
QUOTATION_MARKS	"
NUMBER_SIGN	#
DOLLAR_SIGN	$
PERCENT_SIGN	%
AMPERSAND	&
APOSTROPHE	'
LEFT_PARENTHESIS	(
RIGHT_PARENTHESIS	)
ASTERISK	*
PLUS_SIGN	+
COMMA	,
HYPHEN	-
PERIOD	.
SLASH	/
ZERO	0
ONE	1
TWO	2

Table 21. Printer Symbol Definitions (continued)

Field Name	Symbol
THREE	3
FOUR	4
FIVE	5
SIX	6
SEVEN	7
EIGHT	8
NINE	9
COLON	:
SEMICOLON	;
LESS_THAN_SIGN	<
EQUAL_SIGN	=
GREATER_THAN_SIGN	>
QUESTION_MARK	?
AT_SIGN	@
A_CAPITAL	A
B_CAPITAL	B
C_CAPITAL	C
D_CAPITAL	D
E_CAPITAL	E
F_CAPITAL	F
G_CAPITAL	G
H_CAPITAL	H
I_CAPITAL	I
J_CAPITAL	J
K_CAPITAL	K
L_CAPITAL	L
M_CAPITAL	M
N_CAPITAL	N
O_CAPITAL	O
P_CAPITAL	P
Q_CAPITAL	Q
R_CAPITAL	R
S_CAPITAL	S
T_CAPITAL	T
U_CAPITAL	U
V_CAPITAL	V
W_CAPITAL	W
X_CAPITAL	X
Y_CAPITAL	Y
Z_CAPITAL	Z

Building PC400 PDTs

Table 21. Printer Symbol Definitions (continued)

Field Name	Symbol
LEFT_BRACKET	[
BACKSLASH	\
RIGHT_BRACKET	]
CIRCUMFLEX_ACCENT	^
UNDERLINE	_
GRAVE_ACCENT	
A_SMALL	a
B_SMALL	b
C_SMALL	c
D_SMALL	d
E_SMALL	e
F_SMALL	f
G_SMALL	g
H_SMALL	h
I_SMALL	i
J_SMALL	j
K_SMALL	k
L_SMALL	l
M_SMALL	m
N_SMALL	n
O_SMALL	o
P_SMALL	p
Q_SMALL	q
R_SMALL	r
S_SMALL	s
T_SMALL	t
U_SMALL	u
V_SMALL	v
W_SMALL	w
X_SMALL	x
Y_SMALL	y
Z_SMALL	z
LEFT_BRACE	{
VERTICAL_BAR		
RIGHT_BRACE	}
TILDE_ACCENT	
C_CEDILLA_CAPITAL	Ç
U_DIAERESIS_SMALL	ü
E_ACUTE_SMALL	é
A_CIRCUMFLEX_SMALL	â

Table 21. Printer Symbol Definitions (continued)

Field Name	Symbol
A_DIAERESIS_SMALL	ä
A_GRAVE_SMALL	à
A_OVERCIRCLE_SMALL	å
C_CEDILLA_SMALL	ç
E_CIRCUMFLEX_SMALL	ê
E_DIAERESIS_SMALL	ë
E_GRAVE_SMALL	è
I_DIAERESIS_SMALL	ï
I_CIRCUMFLEX_SMALL	î
I_GRAVE_SMALL	ì
A_DIAERESIS_CAPITAL	Ä
A_OVERCIRCLE_CAPITAL	Å
E_ACUTE_CAPITAL	É
AE_DIPTHONG_SMALL	æ
AE_DIPTHONG_CAPITAL	Æ
O_CIRCUMFLEX_SMALL	ô
O_DIAERESIS_SMALL	ö
O_GRAVE_SMALL	ò
U_CIRCUMFLEX_SMALL	û
U_GRAVE_SMALL	ù
Y_DIAERESIS_SMALL	ÿ
O_DIAERESIS_CAPITAL	Ö
U_DIAERESIS_CAPITAL	Ü
O_SLASH_SMALL	ø
POUND_SIGN	£
O_SLASH_CAPITAL	Ø
MULTIPLY_SIGN	×
A_ACUTE_SMALL	á
I_ACUTE_SMALL	í
O_ACUTE_SMALL	ó
U_ACUTE_SMALL	ú
N_TILDE_SMALL	ñ
N_TILDE_CAPITAL	Ñ
ORDINAL_INDICATOR_FEMININE	ª
ORDINAL_INDICATOR_MASCULINE	º
QUESTION_MARK_INVERTED	¿
REGISTERED_TRADEMARK_SYMBOL	®
LOGICAL_NOT	¬
ONE_HALF	½
ONE_QUARTER	¼

Building PC400 PDTs

Table 21. Printer Symbol Definitions (continued)

Field Name	Symbol
EXCLAMATION_POINT_INVERTED	¡
LEFT_ANGLE_QUOTES	«
RIGHT_ANGLE_QUOTES	»
A_ACUTE_CAPITAL	Á
A_CIRCUMFLEX_CAPITAL	Â
A_GRAVE_CAPITAL	À
COPYRIGHT_SYMBOL	©
CENT_SIGN	¢
YEN_SIGN	¥
A_TILDE_SMALL	ã
A_TILDE_CAPITAL	Ã
E_CIRCUMFLEX_CAPITAL	Ê
E_DIAERESIS_CAPITAL	Ë
I_ACUTE_CAPITAL	Í
I_CIRCUMFLEX_CAPITAL	Î
I_DIAERESIS_CAPITAL	Ï
VERTICAL_LINE_BROKEN	¦
I_GRAVE_CAPITAL	Ì
O_ACUTE_CAPITAL	Ó
O_CIRCUMFLEX_CAPITAL	Ô
O_GRAVE_CAPITAL	Ò
O_TILDE_SMALL	õ
O_TILDE_CAPITAL	Õ
MICRO_SYMBOL	µ
U_ACUTE_CAPITAL	Ú
U_CIRCUMFLEX_CAPITAL	Û
U_GRAVE_CAPITAL	Ù
ACUTE_ACCENT	´
SYLLABLE_HYPHEN	-
PLUS_OR_MINUS_SIGN	±
THREE_QUARTERS	¾
PARAGRAPH_SYMBOL	¶
SECTION_SYMBOL	§
DIVIDE_SIGN	÷
DEGREE_SYMBOL	°
ONE_SUPERSCRIPT	¹
THREE_SUPERSCRIPT	³
TWO_SUPERSCRIPT	²
REQUIRED_SPACE	
INTERNATIONAL_CURRENCY_SYMBOL	¤

Table 21. Printer Symbol Definitions (continued)

Field Name	Symbol
ETH_ICELANDIC_SMALL	ð
ETH_ICELANDIC_CAPITAL	Ð
SHARP_S_SMALL	ß
THORN_ICELANDIC_SMALL	þ
THORN_ICELANDIC_CAPITAL	Þ
Y_ACUTE_SMALL	ý
Y_ACUTE_CAPITAL	Ý
OVERLINE	‾
CEDILLA	¸
DIAERESIS	..
MIDDLE_DOT_ACCENT	·

Using Printer Control Codes

This section explains the SNA Character String (SCS) control codes, or Final Form Text: Document Content Architecture (FFT DCA).

For details of AS/400 printer control codes, see *AS/400 Guide to Programming for Printing*.

Printer Control Code Format

Some printer control codes perform single, specific functions by themselves, while others perform multiple functions according to the parameters specified after the control code.

A printer control code with parameters has the following format:

Control Code	Count	Parameter 1	Parameter 2...
1 or 2 bytes	1 byte	1 or 2 bytes	1 or 2 bytes

A count consists of 1 byte, and indicates the length of the parameters (including the count) after the control code, in bytes. For example, a count and two 1-byte parameters is shown as X'03', because the count itself is included. Some control codes, such as the Printing Position (PP) control code, do not have counts.

A parameter can be 1 or 2 bytes in length. The number of bytes depends on the control codes. Not all control codes have parameters.

Note: In this manual, counts and parameters that are actually processed in binary are all expressed in hexadecimal (0–F) to improve readability.

Supported Control Codes

Personal Communications supports all control codes for the 3812 printer.

Supported SCS Commands

Table 22. Supported SCS Commands

SCS Command	Abbreviation	Hexadecimal Code	Function
Character Attribute			
Set Font Size Scaling	SFSS	2BFDnn02xxyy	Specify the font-size scaling for expansion
Character Set Control			
Load Alternate Character	LAC	2BFEnnmmee	Load the alternate character font
Set CGCS[1] Through Local ID	SCL	2BD1nn810B	Specifies the character set for each language
Set Character Distance	SCD	2BD2nn29xxxx	Selects the pitch and font.
Set Font ID Through GFID[2]	SFG	2BD1nn05xxxxyyyyzz	Selects the font.
Set GCSGID[3] Through GCID[4]	SCG	2BD1nn01xxxxyyyy	Specifies the code page.
Set Presentation of Control Character	SPCC	2BFDnn03xxxx	Specify the treatment of SO and SI
Shift Out	SO	0E	Start of DBCS field
Shift In	SI	0F	End of DBCS field
Device Control			
Bell	BEL	2F	Ignored.
Null	NUL	00	Ignored.
Repeat	RPT	0A	Ignored.
Set Exception Action	SEA	2BD2nn85p1–pn	Specifies the explicit action.
Set Initial Conditions	SIC	2BD2nn45xx	Sets the environment of the EBCDIC transform.
Set Print Setup	SPSU	2BD2nn4Cp1–pn	Ignored.
Switch	SW	2A	Ignored.
Generation Control			
Begin Emphasis	BES	2BD1nn8Axx	Begins emphasis.
Begin Overstrike	BOS	2BD4nn72wwxxyyyyzzzz	Begins overstriking.
Begin Underscore	BUS	2BD4nn0Axxyy	Begins underscoring.
Enable Presentation	ENP	14	Ignored.
End Emphasis	EES	2BD1nn8Exx	Ends emphasis.
End Overstrike	EOS	2BD4nn76	Ends overstriking.
End Underscore	EUS	2BD4nn0E	Ends underscoring.
Inhibit Presentation	INP	24	Ignored.
Justify Text Field	JTF	2BD2nn03xxxxyy	Justifies the following text.
Set Graphic Error Action	SGEA	2BC8nnxxyy	Specifies how to process a received unprintable character font.
Set Justify Mode	SJM	2BD2nn0Dxxyy	Justifies each line of text between the left and right margins.
Subscript	SBS	38	Subscript.
Substitute	SUB	3F	Synonym for the graphic exception character.

Table 22. Supported SCS Commands (continued)

SCS Command	Abbreviation	Hexadecimal Code	Function
Superscript	SPS	09	Superscript.
Word Underscore	WUS	23	Underscore the entire word immediately preceding the control.
Grid Control			
Define Grid Line	DGL	2BFDnn00 xxyyG1~Gn	Specify and print the grid line
Print Format Control			
Form Feed	FF	0C	Moves the printing position to the next page.
Page Presentation Media	PPM	2BD2nn48ssssuuvv wwxxyyzz	Specifies the forms setting, the source drawer, the quality, and the duplex function.
Required Form Feed	RFF	3A	Moves the printing position to the next page and restores the indent level to the left margin.
Set Horizontal Format	SHF	2BC1nnxxyyzzt1–tn	Specifies the maximum print position, the left and right margins, and the horizontal tab stops in units of 1 character.
Set Horizontal Margins	SHM	2BD2nn11xxxxyyyy	Sets the left and right margins in units of 1/1440 inch.
Set IGC Type	SIT	2BFDnn010000	Specify the pitch of DBCS characters
Set Presentation Page Size	SPPS	2BD2nn40xxxxyyyy	Specifies the presentation surface width and depth.
Set Text Orientation	STO	2BD3nnF6xxxxyyyy	Specifies the page rotation.
Set Vertical Format	SVF	2BC2nnxxyyzzt1–tn	Specifies the maximum print line, the top and bottom margins, and the vertical tab stops in units of 1 character.
Set Vertical Margins	SVM	2BD2nn49xxxxyyyy	Sets the top and bottom margins in units of 1/1440 inch.
Printing Position Control			
Backspace	BS	16	Moves the printing position to the left one font width.
Carriage Return	CR	0D	Moves the printing position to the left margin.
Horizontal Tab	HT	05	Moves the printing position right to the next tab stop.
Indent Tab	IT	39	Moves the printing position right to the next tab stops and sets the indent level one tab stop further to the right.
Index Return	IRT	33	Synonym for "Required New Line."
Interchange Record Separator	IRS	1E	Synonym for "New Line."
Line Feed	LF	25	Moves the printing position one line vertically.
New Line	NL	15	Moves the printing position to the left margin of the next line.

Table 22. Supported SCS Commands (continued)

SCS Command	Abbreviation	Hexadecimal Code	Function
Numeric Backspace	NBS	36	Synonym for "Backspace."
Presentation Position	PP	34fpxx	Moves the printing position.
Release Left Margin	RLM	2BD2nn0B	Ignored.
Required New Line	RNL	06	Moves the printing position to the left margin of the next line and resets any pending indent level.
Set Horizontal Tab Stops	STAB	2BD2nn01xxt1–tn	Specifies the tab stops.
Set Indent Level	SIL	2BD2nn07xx	Sets the indent level to the specific tab stop.
Set Line Density	SLD	2BC6nnxx	Specifies the line pitch in units of 1/72 inch.
Set Line Spacing	SLS	2BD2nn09xx	Specifies the number of lines spaced by a new line control.
Set Single Line Distance	SSLD	2BD2nn15xxxx	Specifies the line pitch in units of 1/1440 inch.
Unit Backspace	UBS	1A	Moves the printing position 1/60 inch to the left.
Vertical Tab	VT	0B	Moves the printing position down to the next tab stop.
Transparent Control			
ASCII Transparent	ATRN	03	Allows printer commands and ASCII data to be passed to the printer.
Transparent	TRN	35nn	Allows printer commands and ASCII data to be passed to the printer.

Note:
1   Coded Graphic Character Set
2   Global Font ID
3   Graphic Character Set Global Identifier
4   Graphic Character set ID

Legend:
nn: Count
fp: Function parameter
ee, gg, hh, il–ix, mm, nn, p1–pn, ss, t1–tn, uu, vv, ww, xx, yy, zz: Numeric parameter

Programming Notes

This section briefly explains how a printer reacts if a partial control code is received, or if an incomplete control code is sent.

If the transmission of a control code is interrupted, the printer waits for the remaining part of the code. If the data stream (the series of data units and control codes) sent after the interruption is consistent with the data stream sent before the interruption, (that is, if one complete printer control code is restored by chaining), the control code is processed as is.

If the two parts of the data stream are inconsistent, an error occurs. A negative response to an "Invalid Printer Parameter" is sent to the host system, or treated as a no-op (no operation: ignored because of a meaningless code). The main point to

note here is that detailed information is not sent to the host system if an error occurs in a control code. After programming, the data stream must be checked thoroughly by repeating the printing test.

Restrictions and Notes for AS/400 Printer Commands and Printer Setup

This section provides supplementary notes and explains restrictions for printing.

Printer Control Codes

Printer Control Code	If *Use PDT file* is Selected:	If a Windows Printer Driver is Used:
Set the Character Density (SCD)	Select the GFID for the valid character distance (CD) parameter as shown in Table 23 on page 142.	
	If the specified font is not supported, a substitution is provided. For example, when 15 CPI font is specified but the font is not supported, the supported 17 CPI font is substituted.	See "How to Determine Font" on page 142.
Set Font ID through GFID (SFG)	Recognizable GFIDs are restricted. (See "Field Names of Printer Definition Files" on page 125.) If the specified GFID is not supported and it is out of the range from 154 through 200, the closest font width from the fonts shown in Table 24 on page 142 is substituted. If the specified GFID is not supported and it is between 154 and 200, the following font is substituted: **Font name:** Document; **GFID value:**175.	
		See "How to Determine Font" on page 142.
Begin Underscore (BUS)		The selected font might not support the underscore.
Begin Emphasis (BES)		The selected font might not support the emphasis.
Set Text Orientation (STO)		The page orientation can be changed to portrait or landscape. The direction, which is up, down, left, or right on the paper, depends on the Microsoft Windows printer driver. If the page orientation is changed, the current paper is ejected.
Page Presentation Media (PPM)	The following parameters are supported for PDTs in DBCS mode: • Forms Control (FC) • Source Drawer (SD) • Duplex (DX) Other parameters are ignored.	

Building PC400 PDTs

Table 23. SCD GFID Values

CD Parameter	Character Pitch (Normal)	GFID Value (COR)
000A	10 CPI	13 CPI 204
000B	Proportional	13 CPI 175
000C	12 CPI	15 CPI 86
000F	15 CPI	20 CPI 230
00FF	10 CPI	13 CPI 204

All supported GFIDs are listed in the PDF. The following table lists only the most commonly used GFIDs.

Table 24. SGF GFID Values

Font Name	GFID value
Courier Bold 5	245
Courier 10	11
Prestige Elite 12	86
Gothic-text 13	204
Gothic-text 15	230
Courier 17	252
Gothic-text 20	281
Gothic-text 27	290

When you use the Windows printer driver, the spooler must be on.

How to Determine Font

This section explains how PC400 determines a font.

When a print job is created on AS/400 a certain font, identified by font ID (GFID) is associated with it. Such a font can be specified by the following parameters:

- Font family
- Pitch and family
- Character set
- Width
- Height
- Weight
- Style

System fonts are available with all print drivers, and more flexible device fonts are unique to each printer and printer driver. These fonts are also more fixed as to CPI, weight, code pages, and other criteria that can preclude use of it when matching to the host specified needs.

There are problems associated with mapping an AS/400 font to a PC font. When your PC has all of the fonts that you need, use the following table to determine the best font to use.

Entry ID	Font Family	GFID	Pitch & Family	Character Set	Width	Height	Weight	Style
GFID0003	OCR-B	3	49	0	144	240	400	0
GFID0005	Orator	5	49	0	144	240	400	0
GFID0011	Courier	11	49	0	144	240	400	0

Entry ID	Font Family	GFID	Pitch & Family	Character Set	Width	Height	Weight	Style
GFID0012	Prestige	12	49	0	144	240	400	0
GFID0018	Courier Italic	18	49	0	144	240	400	255
GFID0019	OCR-A	19	49	0	144	240	400	0
GFID0020	Pica	20	49	0	144	240	400	0
GFID0030	Symbol	30	49	2	144	240	400	0
GFID0038	Orator	38	49	0	144	240	400	0
GFID0039	Gothic	39	49	0	144	240	800	0
GFID0040	Gothic	40	49	0	144	240	800	0
GFID0041	Roman	41	49	0	144	240	400	0
GFID0042	Serif	43	49	0	144	240	400	255
GFID0044	Katakana	44	49	0	144	240	400	0
GFID0045	APL	45	49	0	144	240	400	0
GFID0046	Courier Bold	46	49	0	144	240	800	0
GFID0066	Gothic	66	49	0	144	240	400	0
GFID0068	Gothic	68	49	0	120	240	400	0
GFID0069	Gothic	69	49	0	120	240	800	0
GFID0070	Serif	70	49	0	120	240	400	0
GFID0071	Serif	70	49	0	120	240	400	255
GFID0072	Serif	72	49	0	120	240	800	0
GFID0080	Symbol	80	49	2	120	240	400	0
GFID0084	Script	84	49	0	120	240	400	0
GFID0085	Courier	85	49	0	120	240	400	0
GFID0086	Prestige	87	49	0	120	240	400	0
GFID0087	Letter-Gothic	87	49	0	120	240	400	0
GFID0110	Letter-Gothic	110	49	0	120	240	800	0
GFID0111	Prestige	111	49	0	120	240	800	0
GFID0112	Prestige	112	49	0	120	240	400	255
GFID0115	Symbol	115	49	2	120	240	400	0
GFID0155	Boldface	155	18	0	120	240	400	0
GFID0158	Document	158	18	0	120	240	400	0
GFID0159	Boldface	159	18	0	120	240	800	0
GFID0175	Document	175	18	0	120	240	400	0
GFID0176	Boldface	176	18	0	120	240	800	0
GFID0177	Boldface	177	18	0	120	240	800	255
GFID0204	Gothic	204	49	0	108	210	400	0
GFID0221	Prestige	221	49	0	96	210	400	0
GFID0222	Gothic	222	49	0	96	210	400	0

Building PC400 PDTs

Entry ID	Font Family	GFID	Pitch & Family	Character Set	Width	Height	Weight	Style
GFID0223	Courier	223	49	0	96	210	400	0
GFID0225	Symbol	225	49	2	96	240	400	0
GFID0229	Serif	229	49	0	94	210	400	0
GFID0230	Gothic	230	49	0	96	210	400	0
GFID0252	Courier	252	49	0	84	240	400	0

Notes:

1. Default GFID from host is 011, we use Courier 10 CPI.

2. A print driver will change the font to its default if we ask for a font name that it does not recognize. Some drivers recognize Gothic, but the DeskJet's do not.

3. Most print drivers default to Courier New, instead of Courier. So Courier switched to Courier New works, but Gothic switched to Courier New changes the font family.

4. Special fonts like CourHEB and GRCOUR869 (for Greek) has required them to be added to PCSPD.DAT to work. Note if the operating system properly/fully supports a language, that font could be the default font instead of Courier New.

Avoiding AS/400 System Dump

If you are running OS/400 Version 3 Release 1, and you attempt to perform a Telnet 5250 mode host print operation, you may experience an AS/400 system dump under certain conditions. To prevent this from occurring you should apply PTF SF35327 on OS/400.

This table matches the AS/400 table of fonts. PC400 builds a PC spool file with the selected font in it. The printer driver picks up the spool file and the font and sends it to the printer where the expected font is used. Refer to *Printer Device Programming (SC41–5713–01)* for additional information on other useful tables.

The ideal picture can be disturbed by the following factors:

- Not all fonts are available on a PC or printer device.

  If the font that was selected from the Table and incorporated into the spool file can not be found on a PC or on a printer the printer driver determines how to present the data on a printer. For example HP printer drivers have the following order of considerations:

  - HP Font Priority Considerations:

    1. Symbol Set
    2. Spacing
    3. Pitch
    4. Height
    5. Style
    6. Stroke Weight
    7. Typeface Family
    8. Resolution
    9. Orientation

  - Location: printer ROM, SIMM module ROM, cartridge ROM, printer RAM

    Priority of locations:

    1. Soft font (lowest ID first)
    2. Cartridge Font

    3. SIMM Font

    4. Internal Font

– 600 dpi has priority over 300 dpi

To avoid this uncertainty we recommend that you update the table so that only the fonts that are available in the given environment are used.

- NLS

The AS/400 font has NLS characters in it, which are not part of a corresponding PC font. Our table does not know anything about character sets other that ANSI, although it provides a field for it. In this case PC400 builds a PC spool file with a font that doesn't recognize NLS.

To fix the problem, we recommend that you either change the font names in the table to the NLS enabled on a PC/printer, or if the font has the same name as the one in the table, update a Character Set value.

- After we decided to scale a printout which may be a result of CORig or BesFitting, we usually decrease the distance horizontally as well as vertically between characters. This can result in overlapping. PC400 attempst to adjust the given character size to a new one. A problem may occur when a font defined in the table is not a scaleable font. Like GFID011, the most heavily used AS/400 font is mapped to Courier. Courier is a non scaleable font which has only a limited number of character box sizes. To avoid possible problems it is recommended that you use Courier New instead, which is a scaleable TTF font.

PFT Migration Utility

The PFT migration utility converts the printer function table (PFT) for the PC Support/400 workstation feature to a printer definition file (PDF) for PC400.

This section describes the operator interface of the PFT migration utility.

For details of PFT, see *AS/400 PC Support: DOS and OS/2 Technical Reference* SC21-8091.

How to Use the PFT Migration Utility

The file name of the PFT Migration Utility program is `PCSPFC.EXE`. It is a Windows application, and you can execute it by:

- Double-clicking the program name using the Windows Explorer utility.
- Specifying the program name (and parameters) in the Windows Run utility as follows:

```
PCSPFC [ [drive:] [path] PFT-file-name[.extension] ]
```

If no parameter is specified, PCSPFC.EXE displays the Convert PFT to PDF dialog box.
If you omit a drive name and a directory name, PCSPFC.EXE uses the current drive and the current directory. If you omit an extension, PCSPFC.EXE adds ".PFT" to the PFT file name.

When you execute the PFT Migration Utility, the Convert PFT to PDF dialog box appears.

On the Convert PFT to PDF dialog box, select a PFT file from the list box or type a specific PFT file name, and select **OK**. The PFT Migration Utilitystarts the conversion and displays the PFT File Converter dialog box to show the conversion status.

After the conversion, if you select **Save List** on the PFT File Converter dialog box, conversion messages in the dialog box are saved into a list file. The list file is created in the same directory and with the same name as the PFT file, except the extension. The extension of the list file is ".LS2".

If the conversion was completed successfully, you can select **Convert PDF to PDT** from the PFT File Converter dialog box to convert the PDF file to a PDT file. You can also create a PDT file by selecting **Printer Setup** from the File pull-down menu as explained in "Selecting and Changing the Printer Definition Table (PDT File)" on page 109.

Considerations

When the base PDF file already exists, the converted PDF fields are appended to the end of the base PDF file. The name of the base PDF file is decided as follows:

Table 25. PDF File Name

PFT File Name	Base PDF File Name
xxxxxxxx.**PFT**	xxxxxxxx.**PDF**
IBMxxxxx.**MNL**	MNLxxxxx.**PDF**
zzzxxxxx.**MNL***	zzzxxxxx.**PDF***
: * "zzz" is not "IBM".	

Even if the same fields are already defined in the PDF file, the appended fields are effective because the last definition is always effective in a PDF file.

If the base PDF file does not exist in the directory, the PFT Migration Utilitycreates a new PDF file that has only the converted fields from the PFT file. In this case, you should append this file to an appropriate base file manually, because the fields converted from the PFT file do not cover all of the necessary PDF fields.

Therefore, it is recommended that you prepare both the PFT file and its base PDF file in the same directory before the conversion.

Details of Migration

This section describes how the PFT Migration Utility migrates the printer function table (PFT) to the printer definition file (PDF).

Migration from the Printer Function Table

The following table shows the target fields of the PDF for the data in the PFT.

Table 26. Migration from the Printer Function Table to the Printer Definition File

PFT Field Name	PDF Field Name
	Description
INITIALIZATION AND RESET	
Initialization	

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Initialization	START_JOB=
	When the data is defined in the Initialization field of PFT, the data is appended to the string START_JOB=.
Filename	
Filename	None
	This field is ignored.
Reset	
Reset	END_JOB=
	When the data is defined in the Reset field of PFT, the data is appended to the string END_JOB=.
VERTICAL LINE SPACING	
6 lines per inch	
6 lines per inch	SET_6_LINES_PER_INCH=
	When the data is defined in the 6 lines per inch field of PFT, the data is appended to the string SET_6_LINES_PER_INCH=.
8 lines per inch	
8 lines per inch	SET_8_LINES_PER_INCH=
	When the data is defined in the 8 lines per inch field of PFT, the data is appended to the string SET_8_LINES_PER_INCH=.
Variable line spacing	
Control Sequence	SET_VARIABLE_DENSITY=
	When the data is defined in the control sequence field of the PFT for the variable line spacing, the data is appended to the string SET_VARIABLE_DENSITY=. The parameter n in the control sequence is replaced with the PDF parameter type.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	LINE_SPACING_RATIO=
	When the data is defined in the X/Y field of the PFT for the variable line spacing, the value Y/X is appended to the string LINE_SPACING_RATIO= as three-digit or four-digit numbers to indicate the decimal number. For example, when Y/X is 72, the **072** is appended to the string LINE_SPACING_RATIO= and **LINE_SPACING_RATIO=072** is written to the output file. When the value is greater than 255, the four-digit number is migrated. When the value is less than 255, the three-digit number is migrated.
Indexing Functions	

Building PC400 PDTs

PFT Field Name	PDF Field Name
	Description
Begin Superscript	START_SUPERSCRIPT=
	When the data is defined in the Begin Superscript field and End Superscript field of the PFT for the indexing functions, the data is appended to the string START_SUPERSCRIPT=. If the data for the End Superscript is not defined, the data for the Begin Superscript is ignored.
End Superscript	END_SUPERSCRIPT=
	When the data is defined in the End Superscript field and Begin Superscript field of the PFT for the indexing functions, the data is appended to the string END_SUPERSCRIPT=. If the data for the Begin Superscript is not defined, the data for the End Superscript is ignored.
Begin Subscript	START_SUBSCRIPT=
	When the data is defined in the Begin Subscript field and End Subscript field of the PFT for the indexing functions, the data is appended to the string START_SUBSCRIPT=. If the data for the End Subscript is not defined, the data for the Begin Subscript is ignored.
End Subscript	END_SUBSCRIPT=
	When the data is defined in the End Subscript field and Begin Subscript field of the PFT for the indexing functions, the data is appended to the string END_SUBSCRIPT=. If the data for the Begin Subscript is not defined, the data for the End Subscript is ignored.
Reverse 1/2 Index	START_SUPERSCRIPT= END_SUBSCRIPT=
	The data is appended to the START_SUPERSCRIPT= for all of the following conditions: • When the data is not defined in the Begin Superscript field in the PFT or when the data is not defined in the End Superscript field in the PFT. • When the data is defined in the Reverse 1/2 index and Forward 1/2 index field in the PFT. The data is appended to END_SUBSCRIPT= for all of the following conditions: • When the data is not defined in the Begin Subscript field in the PFT or when the data is not defined in the End Subscript field in the PFT. • When the data is defined in the Reverse 1/2 index and Forward 1/2 index field in the PFT.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Forward 1/2 Index	END_SUPERSCRIPT= START_SUBSCRIPT=
	The data is appended to END_SUPERSCRIPT= for all of the following conditions: • When the data is not defined in Begin Superscript field in the PFT or when the data is not defined in End Superscript field in the PFT. • When the data is defined in the Reverse 1/2 index and Forward 1/2 index field in the PFT. The data is appended to START_SUBSCRIPT= for all of the following conditions: • When the data is not defined in the Begin Subscript field in the PFT or when the data is not defined in the End Subscript field in the PFT. • When the data is defined in the Reverse 1/2 index and Forward 1/2 index field in the PFT.
Reverse Index	None
	This field is ignored.
HORIZONTAL LINE SPACING	
5 pitch	
5 pitch	None
	This field is ignored.
8.55 pitch	
8.55 pitch	None
	This field is ignored.
10 pitch	
10 pitch	SET_10_CHARACTERS_PER_INCH=
	When the data is defined in the 10 pitch field of the PFT for the horizontal character spacing, the data is appended to the string SET_10_CHARACTERS_PER_INCH=.
12 pitch	
12 pitch	SET_12_CHARACTERS_PER_INCH=
	When the data is defined in the 12 pitch field of the PFT for the horizontal character spacing, the data is appended to the string SET_12_CHARACTERS_PER_INCH=.
15 pitch	
15 pitch	SET_15_CHARACTERS_PER_INCH=
	When the data is defined in the 15 pitch field of the PFT for the horizontal character spacing, the data is appended to the string SET_15_CHARACTERS_PER_INCH=.
17.1 pitch	
17.1 pitch	SET_17_CHARACTERS_PER_INCH=
	When the data is defined in the 17.1 pitch field of the PFT for the horizontal character spacing, the data is appended to the string SET_17_CHARACTERS_PER_INCH=.
Horizontal Motion Index	

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Control Sequence	Ignored
	This field is ignored.
Maximum	Ignored
	This field is ignored.
Offset	Ignored
	This field is ignored.
X/Y	Ignored
	This field is ignored.
HORIZONTAL RELATIVE MOVEMENT	
Forward Relative Movement	
Control Sequence	FORWARD_HORIZONTAL_SKIP=
	When the data is defined in the Forward Relative Movement field of PFT, the data is appended to the string FORWARD_HORIZONTAL_SKIP=. The parameter n in the control sequence is replaced with the PDF parameter type.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	HORIZONTAL_PEL=
	When the data is defined in the X/Y field of the PFT for the forward relative movement, the value Y/X is appended to the string HORIZONTAL_PEL= as three-digit or four-digit number to indicate the decimal number. For example, when Y/X is 120, **120** is appended to the string HORIZONTAL_PEL= and **HORIZONTAL_PEL=120** is written to the output file. When the value is greater than 255, the four-digit number is migrated. When the value is less than 255, the three-digit number is migrated.
Backward Relative Movement	
Control Sequence	None
	This field is ignored.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	None
	This field is ignored.
HIGHLIGHTING	
Begin Emphasis	
Begin Emphasis	START_HIGHLIGHT_INTENSE=
	When the data is defined in the Begin Emphasis of PFT, the data is appended to the string START_HIGHLIGHT_INTENSE=.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
End Emphasis	
End Emphasis	END_HIGHLIGHT_INTENSE=
	When the data is defined in the End Emphasis of PFT, the data is appended to the string END_HIGHLIGHT_INTENSE=.
Begin Underline	
Begin Underline	START_HIGHLIGHT_UNDERLINE=
	When the data is defined in the Begin Underline of PFT, the data is appended to the string START_HIGHLIGHT_UNDERLINE=.
End Underline	
End Underline	END_HIGHLIGHT_UNDERLINE=
	When the data is defined in the End Underline of PFT, the data is appended to the string END_HIGHLIGHT_UNDERLINE=.
Begin Quality Print	
Begin Quality Print	None
	This field is ignored.
End Quality Print	
End Quality Print	None
	This field is ignored.
PAPER HANDLING	
Bottom Tray Feed	
Bottom Tray Feed	SELECT_DRAWER2=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control of the portrait orientation. For this, the PFT Migration Utility divides this control sequence into two parts and migrates the control sequence for the drawer select and paper feed. The control sequence of the portrait orientation is migrated to SET_PORTRAIT_ORIENT=. If no data is defined in any fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility migrates this control sequence to the SELECT_DRAWER2=. (See "Definition of PAPER HANDLING Migration" on page 168.)
Top Tray Feed	
Top Tray Feed	SELECT_DRAWER1=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control of the portrait orientation. For this, the PFT Migration Utility divides this control sequence into two parts and migrates the control sequence for the drawer select and paper feed. The control sequence of the portrait orientation is migrated to SET_PORTRAIT_ORIENT=. If no data is defined in any fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility migrates this control sequence to SELECT_DRAWER1=. (See "Definition of PAPER HANDLING Migration" on page 168.)
Envelope Feed	

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name	
	Description	
Envelope Feed	SELECT_ENVELOPE=	
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control of the portrait orientation. For this, the PFT Migration Utility divides this control sequence into two parts and migrates the control sequence for the drawer select and paper feed. The control sequence of the portrait orientation is migrated to SET_PORTRAIT_ORIENT=. If no data is defined in any fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility migrates this control sequence to SELECT_ENVELOPE=. (See "Definition of PAPER HANDLING Migration" on page 168.)	
Manual Feed		
Manual Feed	SELECT_DRAWER3=	
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control of the portrait orientation. For this, the PFT Migration Utility divides this control sequence into two parts and migrates the control sequence for the drawer select and paper feed. The control sequence of the portrait orientation is migrated to SET_PORTRAIT_ORIENT=. If no data is defined in any fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility migrates this control sequence to SELECT_DRAWER3=. (See "Definition of PAPER HANDLING Migration" on page 168.)	
Continuous Feed		
Continuous Feed	None	
	This field is ignored.	
Ignore Paper End Sensor		
Ignore Paper End Sensor	None	
	This field is ignored.	
Enable Paper End Sensor		
Enable Paper End Sensor	None	
	This field is ignored.	
Eject Automatic Cut Sheet		
Eject automatic Cut Sheet	None	
	This field is ignored.	
Eject Manual Cut Sheet		
Eject Manual Cut Sheet	None	
	This field is ignored.	
Collate		
Collate	None	
	This field is ignored.	
PAPER POSITIONING		
Continuous Forms		

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Dist. from Top Paper Edge	None
	This field is ignored.
Dist. from Left Paper Edge	None
	This field is ignored.
Location of First Print Column	None
	This field is ignored.
Manual Feed	
Dist. from Top Paper Edge	None
	This field is ignored.
Dist. from Left Paper Edge	None
	This field is ignored.
Location of First Print Column	None
	This field is ignored.
Automatic Feed	
Dist. from Top Paper Edge	None
	This field is ignored.
Dist. from Left Paper Edge	None
	This field is ignored.
Location of First Print Column	None
	This field is ignored.
SET PAGE LENGTH (INCHES)	
Control Sequence	PAGE_LENGTH_TYPE?=INCH SET_PAGE_LENGTH=
	When the data is defined in the control sequence for SET PAGE LENGTH (INCHES) and any data is not defined in the control sequence for SET PAGE LENGTH (LINES), this field is migrated.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	None
	This field is ignored.
Top Margin Size	None
	This field is ignored.
Bottom Margin Size	None
	This field is ignored.
SET PAGE LENGTH (LINES)	
Control Sequence	PAGE_LENGTH_TYPE?=LINE SET_PAGE_LENGTH=
	When the length is defined in the control sequence for SET PAGE LENGTH (LINES), this field is migrated.

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	None
	This field is ignored.
Top Margin Size	None
	This field is ignored.
Bottom Margin Size	None
	This field is ignored.
SET LEFT MARGIN (INCHES)	
Control Sequence	None
	This field is ignored.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
X/Y	None
	This field is ignored.
SET LEFT MARGIN (COLUMNS)	
Control Sequence	SET_HORIZONTAL_MARGIN=
	When the data is defined in the control sequence field for SET LEFT MARGIN (COLUMNS), this field is migrated.
Maximum	None
	This field is ignored.
Offset	None
	This field is ignored.
CARRIER RETURN/LINE FEED	
Continuous Forms	

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Carrier Return (Continuous Forms)	CARRIAGE_RETURN=

	This field is migrated for the following cases:
	
	• Case 1
	  – When the data is defined in the Carrier Return field for the Continuous Forms and
	  – Any data is not defined in the Carrier Return field for the Manual Feed and Automatic Feed.
	
	• Case 2
	  – When the data is defined in the Carrier Return field for the Continuous Forms and
	  – The same data is defined in the Carrier Return field for the Manual Feed and
	  – Any data is not defined in the Carrier Return field for the Automatic Feed.
	
	• Case 3
	  – When the data is defined in the Carrier Return field for the Continuous Forms and
	  – The same data is defined in the Carrier Return field for the Automatic Feed and
	  – Any data is not defined in the Carrier Return field for the Manual Feed.
	
	• Case 4
	  – When the data is defined in the Carrier Return field for the Continuous Forms and
	  – The same data is defined in the Carrier Return field for the Automatic Feed and Manual Feed.

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Line Feed (Continuous Forms)	LINE_FEED=

	This field is migrated for the following cases:
	
	• Case 1
	  – When the data is defined in the Line Feed field for the Continuous Forms and
	  – Any data is not defined in the Line Feed Line field for the Manual Feed and Automatic Feed.
	
	• Case 2
	  – When the data is defined in the Line Feed field for the Continuous Forms and
	  – The same data is defined in the Line Feed field for the Manual Feed and
	  – Any data is not defined in the Line Feed field for the Automatic Feed.
	
	• Case 3
	  – When the data is defined in the Line Feed field for the Continuous Forms and
	  – The same data is defined in the Line Feed field for the Automatic Feed and
	  – Any data is not defined in the Line Feed field for the Manual Feed.
	
	• Case 4
	  – When the data is defined in the Line Feed field for the Continuous Forms and
	  – The same data is defined in the Line Feed field for the Automatic Feed and Manual Feed.

Manual Feed	

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Carrier Return (Manual Feed)	CARRIAGE_RETURN=
	This field is migrated for the following cases:

	• Case 1
	  – When the data is defined in the Line Feed field for the Manual Feed and
	  – Any data is not defined in the Line Feed Line field for the Continuous Forms and Automatic Feed.

	• Case 2
	  – When the data is defined in the Line Feed field for the Manual Feed and
	  – The same data is defined in the Line Feed field for the Continuous Forms and
	  – Any data is not defined in the Line Feed field for the Automatic Feed.

	• Case 3
	  – When the data is defined in the Line Feed field for the Manual Feed and
	  – The same data is defined in the Line Feed field for the Automatic Feed and
	  – Any data is not defined in the Line Feed field for the Continuous Forms.

	• Case 4
	  – When the data is defined in the Line Feed field for the Manual Feed and
	  – The same data is defined in the Line Feed field for the Automatic Feed and Continuous Forms.

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Line Feed (Manual Feed)	LINE_FEED=

	This field is migrated for the following cases:
	
	 • Case 1
	   – When the data is defined in the Line Feed field for the Manual Feed and
	   – Any data is not defined in the Line Feed Line field for the Continuous Forms and Automatic Feed.
	
	 • Case 2
	   – When the data is defined in the Line Feed field for the Manual Feed and
	   – The same data is defined in the Line Feed field for the Continuous Forms and
	   – Any data is not defined in the Line Feed field for the Automatic Feed.
	
	 • Case 3
	   – When the data is defined in the Line Feed field for the Manual Feed and
	   – The same data is defined in the Line Feed field for the Automatic Feed and
	   – Any data is not defined in the Line Feed field for the Continuous Forms.
	
	 • Case 4
	   – When the data is defined in the Line Feed field for the Manual Feed and
	   – The same data is defined in the Line Feed field for the Automatic Feed and Continuous Forms.

Automatic Feed	

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Carrier Return (Automatic Feed)	CARRIAGE_RETURN=
	This field is migrated for the following cases: • Case 1 – When the data is defined in the Carrier Return field for the Automatic Feed and – Any data is not defined in the Carrier Return field for the Continuous Feed and Manual Feed. • Case 2 – When the data is defined in the Carrier Return field for the Automatic Feed and – The same data is defined in the Carrier Return field for the Continuous Forms and – Any data is not defined in the Carrier Return field for the Manual Feed. • Case 3 – When the data is defined in the Carrier Return field for the Automatic Feed and – The same data is defined in the Carrier Return field for the Manual Feed and – Any data is not defined in the Carrier Return field for the Continuous Forms. • Case 4 – When the data is defined in the Carrier Return field for the Automatic Feed and – The same data is defined in the Carrier Return field for the Manual Feed and Continuous Forms.

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Line Feed (Automatic Feed)	LINE_FEED=

	This field is migrated for the following cases:
	
	• Case 1
	  – When the data is defined in the Line Feed field for the Automatic Feed and
	  – Any data is not defined in the Line Feed Line field for the Continuous Forms and Manual Feed.
	
	• Case 2
	  – When the data is defined in the Line Feed field for the Automatic Feed and
	  – The same data is defined in the Line Feed field for the Continuous Forms and
	  – Any data is not defined in the Line Feed field for the Manual Feed.
	
	• Case 3
	  – When the data is defined in the Line Feed field for the Automatic Feed and
	  – The same data is defined in the Line Feed field for the Manual Feed and
	  – Any data is not defined in the Line Feed field for the Continuous Forms.
	
	• Case 4
	  – When the data is defined in the Line Feed field for the Automatic Feed and
	  – The same data is defined in the Line Feed field for the Manual Feed and Continuous Forms.

MULTIPLE COPIES	
Print without Clearing Page from	
Top Tray	None
	This field is ignored.
Bottom Tray	None
	This field is ignored.
Envelope Tray	None
	This field is ignored.
Manual Tray	None
	This field is ignored.
Clear Page Buffer	None
	This field is ignored.
Multiple Copies Variable Control	
Control Sequence	None
	This field is ignored.
Maximum	None
	This field is ignored.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Offset	None
	This field is ignored.
LANDSCAPE PAPER HANDLING

PFT Field Name	PDF Field Name / Description
	Bottom Tray Feed
Bottom Tray Feed	SET_LANDSCAPELEFT_ORIENT=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control sequence for the landscape orientation. For this, the PFT Migration Utility divides the control sequence into two parts. The control sequence for the drawer select and paper feed is ignored because this control sequence is migrated when the data for PAPER HANDLING is processed. The control sequence for landscape orientation is migrated to SET_LANDSCAPE_ORIENT=. (See "Definition of PAPER HANDLING Migration" on page 168.)
	Top Tray Feed
Top Tray Feed	SET_LANDSCAPELEFT_ORIENT=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control sequence for the landscape orientation. For this, the PFT Migration Utility divides the control sequence into two parts. The control sequence for the drawer select and paper feed is ignored because this control sequence is migrated when the data for PAPER HANDLING is processed. The control sequence for landscape orientation is migrated to SET_LANDSCAPE_ORIENT=. (See "Definition of PAPER HANDLING Migration" on page 168.)
	Envelope Feed
Envelope Feed	SET_LANDSCAPELEFT_ORIENT=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control sequence for the landscape orientation. For this, the PFT Migration Utility divides the control sequence into two parts. The control sequence for the drawer select and paper feed is ignored because this control sequence is migrated when the data for PAPER HANDLING is processed. The control sequence for landscape orientation is migrated to SET_LANDSCAPE_ORIENT=. (See "Definition of PAPER HANDLING Migration" on page 168.)
	Manual Feed

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Manual Feed	SET_LANDSCAPELEFT_ORIENT=
	If the data is defined in some fields for LANDSCAPE PAPER HANDLING, the PFT Migration Utility assumes that this control sequence includes the control sequence for the landscape orientation. For this, the PFT Migration Utility divides the control sequence into two parts. The control sequence for the drawer select and paper feed is ignored because this control sequence is migrated when the data for PAPER HANDLING is processed. The control sequence for landscape orientation is migrated to SET_LANDSCAPE_ORIENT=. (See "Definition of PAPER HANDLING Migration" on page 168.)
	TYPESTYLE DEFINITION
	Default TypeStyle Definition
PC Character Set	None
	This field is ignored.
Initial Control Sequence	None
	This field is ignored.
Ending Control Sequence	None
	This field is ignored.
	Individual Typestyle Definition
Typestyle number	SET_GFID_
	This number is appended as a four-digit number after the string SET_GFID_. For example, when the typestyle number **9** is defined, **0009** is appended after the string SET_GFID_ and SET_GFID_0009= is migrated. And the numbers supported by PC400 are migrated. The numbers not supported by PC400 are ignored.
PC Character Set	None
	This field is ignored.
Initial Control Sequence	SET_GFID_xxxx=
	This control sequence is appended after the string SET_GFID_xxxx=, where xxxx is the four-digit number defined in the typestyle number field.
Ending Control Sequence	None
	This field is ignored.
Characters	None
	This field is ignored.
Initial Control Sequence	None
	This field is ignored.
Ending Control Sequence	None
	This field is ignored.
Symbols	None
	This field is ignored.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Initial Control Sequence	None
	This field is ignored.
Ending Control Sequence	None
	This field is ignored.
Group Typestyle Definition	
Group Identifier	None
	This field is ignored.
Group Identifier Comment	None
	This field is ignored.
Typestyle number	SET_GFID_
	This number is appended as a four-digit number after the string SET_GFID_. For example, when the typestyle numbers 1, 2, 3, and 4 are defined, **0001**, **0002**, **0003**, and **0004** are appended after the string SET_GFID_ and SET_GFID_0001=, SET_GFID_0002=, SET_GFID_0003=, and SET_GFID_0004= are migrated because the PC400 does not have the group typestyle definition. The typestyle numbers that are not supported by the PC400 are not migrated.
PC Character Set	None
	This field is ignored.
Initial Control Sequence	SET_GFID_xxxx=
	This control sequence is appended after the string SET_GFID_xxxx=, where xxxx is the four-digit number defined in the typestyle number field.
Ending Control Sequence	None
	This field is ignored.
Characters	None
	This field is ignored.
Initial Control Sequence	None
	This field is ignored.
Ending Control Sequence	None
	This field is ignored.
Symbols	None
	This field is ignored.
Initial Control Sequence	None
	This field is ignored.
Ending Control Sequence	None
	This field is ignored.
Character Set Number	
Character Set Number	None
	This field is ignored.
SLOT SELECTION	

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Slot 1 Sequence	
Slot 1 Sequence	None
	This field is ignored.
Slot 2 Sequence	
Slot 2 Sequence	None
	This field is ignored.
Slot 3 Sequence	
Slot 3 Sequence	None
	This field is ignored.
Stop Sequence	
Stop Sequence	None
	This field is ignored.
USER DEFINED CONTROL	
Parameters of SET ENVELOP SIZE Command	
Control Number: 984	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
ESC Sequence of SET ENVELOP SIZE Command	
Control Number: 985	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
Parameters of SET PAGE SIZE Command	
Control Number: 986	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
ESC Sequence of SET PAGE SIZE Command	
Control Number: 987	None
	This field is ignored.
Control Sequence	None
	This field is ignored.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Control Sequence File Name	None
	This field is ignored.
Printer Data Stream	
Control Number: 988	None
	This field is ignored.
Control Sequence	SET_FONT_GLOBAL=1B 5B 49 word-value(LH) word-value(HL) word-value(HL) byte-value word-value(HL)
	When **04** is defined in this field, SET_FONT_GLOBAL=1B 5B 49 ... word-value(HL) is migrated. When the other value is defined, this field is ignored. **04** means IBM Personal Printer Data Stream Level 2 or higher. When the migration is done for PAPER HANDLING, use this information. (See "Definition of PAPER HANDLING Migration" on page 168.)
Control Sequence File Name	None
	This field is ignored.
Duplex long edge	
Control Number: 989	None
	This field is ignored.
Control Sequence	SET_DUPLEX=
	When the data is defined in the control sequence for the duplex long edge, the data is appended to the string SET_DUPLEX=.
Control Sequence File Name	None
	This field is ignored.
Duplex short edge	
Control Number: 990	None
	This field is ignored.
Control Sequence	SET_DUPLEX_TUMBLE=
	When the data is defined in the control sequence for the duplex short edge, the data is appended to the string SET_DUPLEX_TUMBLE=.
Control Sequence File Name	None
	This field is ignored.
Simplex	
Control Number: 991	None
	This field is ignored.
Control Sequence	RESET_DUPLEX=
	When the data is defined in the control sequence for the simplex, the data is appended to the string RESET_DUPLEX=.
Control Sequence File Name	None
	This field is ignored.
Jog the output tray	

Building PC400 PDTs

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Control Number: 992	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
Normal portrait orientation	
Control Number: 993	None
	This field is ignored.
Control Sequence	SET_PORTRAIT_ORIENT=
	When the data is defined in the control field for the normal portrait orientation, the data is appended to the string SET_PORTRAIT_ORIENT=.
Control Sequence File Name	None
	This field is ignored.
Landscape left	
Control Number: 994	None
	This field is ignored.
Control Sequence	SET_LANDSCAPELEFT_ORIENT=
	When the data is defined in the control field for the landscape left, the data is appended to the string SET_LANDSCAPELEFT_ORIENT=.
Control Sequence File Name	None
	This field is ignored.
Portrait upside down orientation	
Control Number: 995	None
	This field is ignored.
Control Sequence	SET_PORTRAITUPDWN_ORIENT=
	When the data is defined in the control field for the portrait upside down orientation, the data is appended to the string SET_PORTRAITUPDWN_ORIENT=.
Control Sequence File Name	None
	This field is ignored.
Landscape right	
Control Number: 996	None
	This field is ignored.
Control Sequence	SET_LANDSCAPERGHT_ORIENT=
	When the data is defined in the control field for the landscape right, the data is appended to the string SET_LANDSCAPERGHT_ORIENT=.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
Control Sequence File Name	None
	This field is ignored.
COR in 10 pitch	
Control Number: 997	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
COR in 12 pitch	
Control Number: 998	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
COR in 15 pitch	
Control Number: 999	None
	This field is ignored.
Control Sequence	None
	This field is ignored.
Control Sequence File Name	None
	This field is ignored.
FUNCTION SELECTION TEST RESPONSES	
Superscript /Subscript	None
	This data is ignored.
Underline	None
	This data is ignored.
Emphasis (Bold)	None
	This data is ignored.
Form Feed	None
	This data is ignored.
Back Space	BACKSPACE=
	This field is migrated when you type YES or NO in response to the prompt during the Backspace Function Selection Test.
Mid-line pitch change	None
	This data is ignored.
Horizontal Character spacing	None
	This data is ignored.

Table 26. Migration from the Printer Function Table to the Printer Definition File (continued)

PFT Field Name	PDF Field Name
	Description
First character position	None
	This data is ignored.
PSM	None
	This data is ignored.
Cursor Draw	None
	This data is ignored.

Note: When you migrate **IBM3812.PFT** and **IBM3812.MNL**, the following fields are added to **IBM3812.PDF**:
- `FORWARD_VERTICAL_STEP_FEED=1B 5B 43 03 00 E3 word-value(LH)`
- `VERTICAL_PEL=240`

Definition of PAPER HANDLING Migration

The PFT Migration Utility migrates the data for PAPER HANDLING and LANDSCAPE PAPER HANDLING as follows:
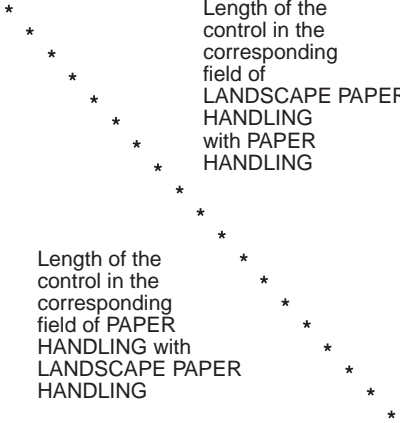
1. The PFT Migration Utility checks whether the LANDSCAPE PAPER HANDLING controls are defined.

2. If the LANDSCAPE PAPER HANDLING controls are defined, the PFT Migration Utility assumes that the PAPER HANDLING control sequences include controls to set the page orientation to portrait. Each LANDSCAPE PAPER HANDLING control sequence has the controls to set the page orientation to landscape in addition to the tray select and feed controls.

3. If the LANDSCAPE PAPER HANDLING controls are not defined, the PAPER HANDLING controls do not have the controls to set the page orientation to portrait. If a personal printer supports normal portrait, landscape left, portrait upside down, and landscape right orientation, the page orientation controls are defined in the appropriate user-defined controls.

If the LANDSCAPE PAPER HANDLING controls are not defined: When the PAPER HANDLING controls are defined, the controls are migrated as follows:

PFT Fields		PDF Fields
Bottom Tray Select & Feed	→	SELECT_DRAWER2=
Top Tray Select & Feed	→	SELECT_DRAWER1=
Envelope Tray Select & Feed	→	SELECT_ENVELOPE=
Manual Select & Feed	→	SELECT_DRAWER3=

If the LANDSCAPE PAPER HANDLING controls are defined: When the PAPER HANDLING controls are defined, these controls have the controls to set the page orientation to portrait. The LANDSCAPE PAPER HANDLING controls have the controls to set the page orientation to landscape. The PFT Migration Utility

migrates the controls as follows:

Length of the control in the corresponding field of LANDSCAPE PAPER HANDLING with PAPER HANDLING // Length of the control in the corresponding field of PAPER HANDLING with LANDSCAPE PAPER HANDLING	(Length1) ==0	(Length1) > 0
(Length2) ==0	CASE 1	CASE 2
(Length2) > 0	CASE 3	Length1 != Length2 CASE 4 ———————— Length1 == Length2 CASE 5

- **CASE 1**

  Since both of the fields are not defined, no data is migrated.

  Example: No data is migrated to SELECT_DRAWER2= under the following conditions. For this, SELECT_DRAWER2= is not written in the output file.
  – No data is defined in the bottom tray select and feed for PAPER HANDLING.
  – No data is defined in the bottom tray select and feed for LANDSCAPE PAPER HANDLING.

- **CASE 2**

  Since no data is defined in the field for PAPER HANDLING, the PFT Migration Utility cannot compare the data in the corresponding field with the data for LANDSCAPE PAPER HANDLING. The data for LANDSCAPE PAPER HANDLING is migrated to the drawer selection field of PDF regardless, including control of the landscape orientation.

  Example: The control of the top tray select and feed for LANDSCAPE PAPER HANDLING is migrated to SELECT_DRAWER1= under the following conditions.
  – No data is defined in the top tray select and feed for PAPER HANDLING.
  – The control is defined in the top tray select and feed for LANDSCAPE PAPER HANDLING.

- **CASE 3**

  Since no data is defined in the field for LANDSCAPE PAPER HANDLING, the PFT Migration Utility cannot compare the data in the corresponding field with

the data for PAPER HANDLING. The data for PAPER HANDLING is migrated to the drawer selection field of PDF regardless, including the control of the portrait orientation.

Example: The control of the manual select and feed for PAPER HANDLING is migrated to SELECT_DRAWER3= under the following conditions.
– The control is defined in the manual select and feed for PAPER HANDLING.
– No data is defined in the top tray select and feed for LANDSCAPE PAPER HANDLING.

- **CASE 4**

  Since the length is different, no data is migrated.

  Example: No data is migrated to SELECT_DRAWER2= under the following conditions. For this, SELECT_DRAWER2= is not written in the output file.
  – The data is defined in the bottom tray select and feed for PAPER HANDLING and the length is 8.
  – The data is defined in the bottom tray select and feed for LANDSCAPE PAPER HANDLING and the length is 10.

- **CASE 5**

  The PFT Migration Utility compares the data for PAPER HANDLING with the data for LANDSCAPE PAPER HANDLING as follows:

  – If a different value is found in the data, search the control backward for the escape character X'1B'.

  – If the escape character is found in the middle of the control, divide the control into two parts. The PFT Migration Utility assumes that the first part is the control for the tray select and feed, and the second part is the control for the page orientation. The PFT Migration Utility migrates the first part of the PAPER HANDLING control into the tray select and feed, the second part of the PAPER HANDLING control into the SET_PORTRAIT_ORIENT=, and the second part of the LANDSCAPE PAPER HANDLING control into SET_LANDSCAPELEFT_ORIENT=.

  – If the escape character is found at the top of the control and 03 (= HP PCL) is defined in the user-defined control 988, the PFT Migration Utility assumes that the controls are combined. For this, the control begins with the escape character and the two shared characters. For this, the PFT Migration Utility assumes that the last character of the first part is a lowercase letter and converts it to an uppercase letter to indicate that it is a terminating character.

The PFT Migration Utility divides the controls into the two parts as follows:

< Control for the PAPER HANDLING >

Migrated into SET_PORTRAIT_ORIENT=

| ESC | shared character | oo oo oo oo | XX | •• •• •• |

Different value

Converted uppercase

Migrated into the tray select and feed
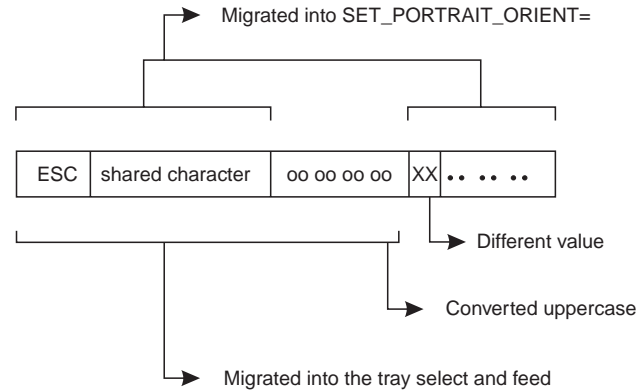
< Control for LANDSCAPE PAPER HANDLING >

Migrated into SET_LANDSCAPELEFT_ORIENT=

| ESC | shared character | oo oo oo oo | XX | :: :: :: |

Different value
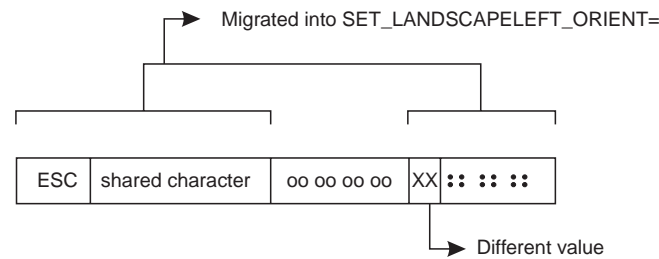
- – If the escape character is found, but the data stream is not HP PCL, the PFT Migration Utility displays the error message to indicate that the PFT Migration Utility cannot migrate the data.
- – If the escape character is not found, the PFT Migration Utility displays the error message to indicate that the PFT Migration Utility cannot migrate the data.
- – If a different value is not found, the same control is defined for PAPER HANDLING and LANDSCAPE PAPER HANDLING.

The PFT Migration Utility migrates the data in the following order:
1. Top Tray Select and Feed
2. Bottom Tray Select and Feed
3. Manual Select and Feed
4. Envelope Tray Select and Feed

The migration stops for the page orientation, SET_PORTRAIT_ORIENT= and SET_LANDSCAPELEFT_ORIENT=, when the controls for the page orientation are found. For example, the controls for the page orientation are found when the data is migrated for the Bottom Tray Select and Feed. The PFT Migration Utility does not care about the page orientation when migrating the data Manual Select and Feed, and Envelop Tray Select and Feed.

Chapter 13. Field Attribute Codes

The following table explains the field attribute codes used in PC400.

Table 27. Meanings of Field Attribute Codes

Hex Code	Meaning
20	Green
21	Green, reversed image
22	White
23	White, reversed image
24	Green, underlined
25	Green, underlined, reversed image
26	White, underlined
27	Nondisplay
28	Red
29	Red, reversed image
2A	Red
2B	Red, reversed image
2C	Red, underlined
2D	Red, underlined, reversed image
2E	Red, underlined
2F	Nondisplay
30	Sky blue with column delimiter
31	Sky blue, column delimiter, reversed image
32	Yellow with column delimiter
33	Yellow, column delimiter, reversed image
34	Sky blue, underlined, with column delimiter
35	Sky blue, underlined, reversed image, with column delimiter
36	Yellow, underlined, with column delimiter
37	Nondisplay with column delimiter
38	Purple
39	Purple, reversed image
3A	Blue
3B	Blue, reversed image
3C	Purple, underlined
3D	Purple, underlined, reversed image
3E	Blue, underlined
3F	Nondisplay with column delimiter

Chapter 14. Data Transfer for PC400

This chapter explains file-description files and data conversions for the data transfer function. References in this chapter to *router sessions* mean an SNA connection to an AS/400 system (with or without a display session).

Data Transfer Function Overview

The PC400 data transfer function transfers data to or from the AS/400 system and a workstation.

PC400 can transfer data between the host and workstation. The data transfer function can be used by selecting the **Data Transfer** icon.

Notes:

1. For Personal Communications Version 5.0, the Data Transfer application is automatically invoked from a 5250 session when you choose **Send File to Host** from the **Actions** menu and **Receive File from Host** from the **Actions** menu.

2. You can change this default to invoke normal file transfer functions; to do so, click **Preferences > Transfer** from the **Edit** menu, then click the **Data Transfer** radio button on the property page with the **General** tab.

Note: Transferring data, described in this chapter, is quite different from transferring files, which is described in Chapter 15.

Table 28. Data Transfer Summary

Type of Transfer	Products required on an AS/400 system	Access Method	Sending and receiving unit	Type of connection to an AS/400 system
File Transfer	Personal Communications Tools (PCT/400 see "Chapter 5. Transferring Files" on page 31)	• Transfer menu in the session window • EHLLAPI application that invokes File Transfer • DDE application that invokes File Transfer • By playing a macro that invokes File Transfer • By clicking the Send or Receive button on the tool bar	Entire file	Display session
Data Transfer	PC Support/400 V2R2 or V2R3 or OS/400 V3R1 or later[1]	Data Transfer icon or File Transfer selections from Actions menu	Field, record, or file in a database	Any SNA link
Note 1: OS/400 provides the host transaction program for Data Transfer.				

Transferring Files from AS/400 System to a Workstation

When using a workstation, you can retrieve and use data from the following file types on an AS/400 system:
• Physical database
• Logical database
• Distributed data management (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.
- 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.
- Specify the decimal separator character.

The following output destinations can be specified:
- Display
- Disk
- Printer

Transferring Files from a Workstation to an AS/400 System

The PC-to-AS/400 transfer function enables the transfer of data from a workstation to an AS/400 physical file. Data can be transferred to any of the following destinations:
- Existing members in an existing AS/400 physical file
- New members in an existing AS/400 physical file
- New members in a new AS/400 physical file

Note: Data cannot be transferred from a workstation file to an AS/400 logical file.

Note the following considerations when transferring data.

Transferring Data to Existing Members in an Existing File

Note the following considerations when transferring data from a workstation to an existing AS/400 member.

- When data is transferred to an existing member, data in that member is replaced with that transferred from a workstation.

- When AS/400 members already contain data, a message appears, indicating that the data in the existing members will be replaced with the data that is about to be transferred.

- Consider the effect of returning data that was previously transferred from the AS/400 system (such as when an AS/400 master file is updated on a workstation).

  For example, you can transfer only the field subset of an AS/400 file by issuing a transfer request from the AS/400 system to a workstation. In this case, when returning data from the workstation to the AS/400 system, only the subset included in that AS/400 file can be transferred. Other fields that had been defined in the AS/400 file but not transferred are filled with blanks if they are character fields or, if they are numeric fields, with zeros or the values specified at file creation.

  Therefore, the data must be transferred to another AS/400 file and the transferred data must be embedded in the AS/400 file by running the AS/400 application program. Follow this procedure to control the update processing for an AS/400 master file.

  To prevent users from transferring data to a certain AS/400 file, check that the authority level for that file is defined correctly.

Transferring Data to New Members in an Existing File

You can transfer the data in a workstation file to new members in an existing AS/400 file. The transfer function automatically creates these members in the specified file in the specified library. New members are created according to the file description in the existing file.

Be particularly careful when only the field subset of the AS/400 file can be transferred from the AS/400 system to a workstation by the previous transfer request. When data is returned to the AS/400 system, new members can receive only the subset defined in that AS/400 file. Other character fields that are defined, but not transferred are filled with blanks. Numeric fields are filled with zeros or the valued specified at file creation. The date, time, and time-stamp fields use AS/400 default values.

Transferring Data to New Members in a New File

By using a transfer request from a workstation to the AS/400 system, you can transfer data to new members in a new AS/400 file. This is one of the safest transfer methods, because data already stored in the AS/400 file is not replaced with that transferred from the workstation.

There are two ways of transferring data to new members in a new AS/400 file. The method used depends on the data to be transferred.

- For data that is broken up into fields, correct conversion is achieved by transferring it in units of fields. Specify use of the workstation file-description file at data transfer. In addition, specify data as the type of the AS/400 file.

  When an AS/400 file and its members are created, the transfer function must access the description of the format of each field to be transferred in the AS/400 file. You can get this description, called a field-reference file, from the AS/400 file. To create an AS/400 file and its members, specify the name of this AS/400 field reference file, as well as the parameters for the other files and members. Note that only the fields to be transferred are defined in a new file.

- For data consisting only of text or source statement records, it is not necessary to break up the records into fields. In addition, the workstation file-description file is not required to transfer data. In other words, an AS/400 physical source file is created.

Transferring Data to an AS/400 Data File and Source File

You can transfer data to the following two types of AS/400 physical files.

- **Physical data file** The members of a physical data file can contain numeric and character data of any AS/400 data type. To transfer data to a physical data file, use the workstation file-description file to define how data is stored in a workstation data file. Besides this definition, the file description of the AS/400 file is required to ensure correct conversion of the data.

  When data is transferred to an existing AS/400 file, the file description becomes part of the AS/400 file. When data is transferred to a new AS/400 file, the file description is included in the AS/400 field-reference file.

- **Physical source file** Normally, a physical source file stores no data. It contains only text or source statements, as follows:

  - The first part (field) of a source file always contains numbers indicating the order.

  - The second part (field) of a source file always contains the date on which the file was created.

– The third part (field) of a source file contains the text of the file. This part can contain data fields of character type or zoned type only. Physical source files provide the optimum means of transferring text or source statements with a workstation.

Note the following considerations when transferring data to and from an AS/400 physical source file:

– To transfer text from the AS/400 system to a workstation, specify the name of the source file and members in **FROM**. Specify an asterisk (*) in **SELECT**. This informs the AS/400 system that only text is transferred from the source file, with the order number and date fields excluded.

– The AS/400 text must be stored in the workstation code text file. Normally, a workstation text editing program can be used to manipulate this workstation code text file.

– Specify that the file-description file is not to be stored for that workstation file. Because text is assumed to be a record consisting only of character data, it is not necessary to define fields.

– To return text from a workstation file to an AS/400 file, specify the type of the workstation file containing the text. This is almost always workstation code text. Specification of the file-description file is not required.

– To create a new AS/400 file and its members, specify a valid record length. This record length must be equal to the maximum record length of the workstation file, plus 12 bytes. This is because the transfer function automatically creates the order number and date fields when the file is transferred to the AS/400 members. The order number and date fields together occupy 12 bytes.

Preparing for Data Transfer

The following topics describe the software products required to transfer data and the points you must understand before transferring data with PC400.

Required Software Products

To use Data Transfer, IBM PC Support/400 (5738-PC1) must be installed on the AS/400 system. IBM PC Support/400 is not required with OS/400 Version 3 or later.

Before using the data transfer function, run the router of PC400 or PC Support/400.

Transfer Function

You can transfer only source programs, records, and the following information:
- Information organized for analysis
- Information used for decision making
- Information suited for computer processing

When using a spreadsheet, for example, you might want to use inventory data to create a cost analysis report. If there is no way to copy the data into the workstation, you must print the data from the AS/400 system and manually type it into a workstation file. With the transfer function, however, you can access the inventory database directly, select only the data needed for the report, process the data as required, then complete the report using that data.

Data can also be sent from the workstation to the host system for processing by AS/400 application. When a remote user is authorized to access the AS/400 system directly, he or she can access the created cost analysis report to compare with their results.

The following example outlines the joining of two files, transferring the information to the workstation, and creating a report.
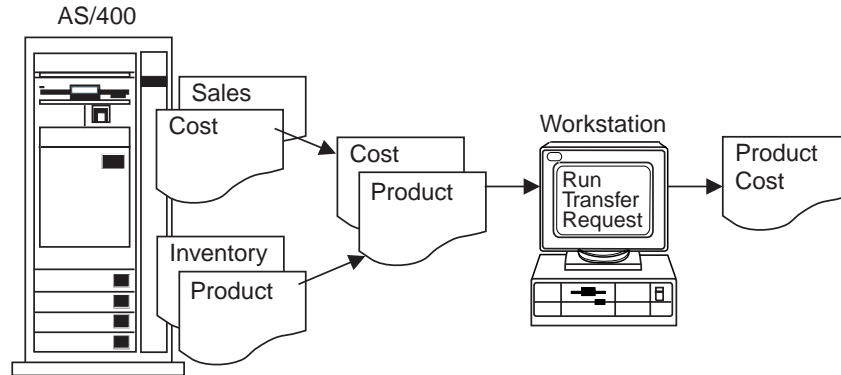
To transfer data by using PC400, you must create a transfer request. A transfer request provides the necessary information about the data you want to transfer.

Before creating a transfer request, you must have the answers to the following questions:

Where is the data located?
How much of the data do you want to transfer?
How should the data be sorted?
Where do you want data to be transferred?

When transferring data from the AS/400 system to a workstation, PC400 allows you to specify which data is to be transferred and whether the data is to be displayed or written to a workstation file.

In addition, a transfer request can be saved to a workstation file, allowing you to easily perform the same transfer at a later date. After a transfer request is saved, you can call the request to make changes or to run it again.

Data Transfer Program

PC400 data transfer is classified into two types, depending on the direction of the transfer:

- Transferring data from the workstation to the AS/400 system is called data sending.
- Transferring data from the AS/400 system to the workstation is called data receiving.

Data transfer can also be classified according to how the program is started, as follows:

- Data is transferred by interactively entering information such as *what data is transferred from which file to which file* on the screen. In this case, the interactive screen for sending is called the PC→AS/400 Transfer window, and that for receiving is called the AS/400→PC Transfer window.
- Data is transferred according to the information that has already been registered. The interactive screen is not necessary. This is called the *automatic transfer* of data.

In both cases, data transfer is performed by **PCSFT5.EXE** on the workstation and by the *PC Support/400* transfer program on the AS/400 system.

The **Data Transfer** icon is registered in the PC400 folder by installing PC400. Double-clicking on this icon displays the AS/400→PC Transfer window (for receiving). This icon includes:

```
\Personal Communications\PCSFT5.EXE
```

The PC→AS/400 Transfer window (for sending) appears when the registered contents are changed as follows:

```
\Personal Communications\PCSFT5.EXE
```

The AS/400→PC Transfer and PC→AS/400 Transfer windows have a **Switch to SEND** button and **Switch to RECEIVE** button, respectively. By clicking either of these buttons, the window for sending can be switched to the window for receiving, and vice versa.

To perform *automatic transfer*, transfer information must be created, using the interactive screen window, and then saved. Data transfer can then be performed automatically by specifying the file name in which the data was saved.

For example, if transfer information is saved to file **TENSOU.TTO**, contained in directory **C:\Personal Communications\PRIVATE**, run *automatic transfer* as follows:

```
"C:\Personal Communications\PCSFT5.EXE" "C:\Personal Communications\PRIVATE\TENSOU.TTO"
```

When you save the transfer information, register it as an icon in the PC400 folder. You can then transfer data automatically simply by double-clicking on this icon.

Data Concepts of the AS/400 System and Your Workstation

The basic components of data management are files, records, and fields. A *file* is an aggregate of records, referenced by a single name. Each record in a file contains one or more items of correlated information. Each item of information is called a *field*.

The AS/400 system and your workstation use different functions to store and group data, and to set the format.

Workstation Files

To transfer data from a workstation to the AS/400 system, the transfer function uses a special-format workstation file, called a *file-description file*. Using this file, data is stored in a valid format and converted into a valid type.

A file-description file identifies the format of a workstation data file and contains a description of the fields in the data file. The file-description file also contains a name list of all the fields in the data file. This list reflects the order, as well as the names, in which each field appears within the data file. In addition, this list includes a description of the data type, length, and decimal position of each field. Using this information, the transfer function can recognize not only how data has been modified but also where a certain field exists in a file record.

When data is transferred from the AS/400 system to a workstation, you can use the transfer function to automatically create the file-description file. In this case, the information in the file-description file depends on the file description in the AS/400 file.

You must create a file-description file with the same name as the workstation data file to transfer a workstation data file to the AS/400 system.

Distributed Data Management (DDM) Files

Distributed data management (DDM) is one of the functions supported by the AS/400 system. This function is used to access database files that are stored on remote AS/400 systems. To use the transfer function to access these database files, specify a DDM file name as the name of the AS/400 file to be transferred. Refer to the *DDM Guide* for details of how to use DDM files.

AS/400 Files

The following list provides a simple explanation of the requirements for transferring data between the AS/400 system and a workstation.

Library

The AS/400 library contains related objects that are used to generate significant groups. For example, the objects might be all the programs and files related to credit sales management. Using the library, you can group objects and find a desired file by name. The transfer function uses the library to locate an AS/400 file.

File   AS/400 files that you can manipulate consist of a file description and data stored in the file. PC400 processes an AS/400 file, called a database file. The database file can be either a physical file or a logical file.

A *physical file* is a database file that contains data stored in records. It includes a description of the record format in addition to the data itself.

A *logical file* is a database file, which you can use to access data stored in one or more physical files. Logical files, like physical files, contain a file description. However, logical files do not contain any actual data. Instead, you can access fields in one or more physical files by using the record format included in the logical file description. When a logical file is transferred from the AS/400 system to a workstation, data is obtained from one or more physical files. You need only specify a logical file as the file to be transferred. The AS/400 system recognizes which physical file contains the actual data to be transferred.

Note: Data cannot be transferred from a workstation to logical files.

Member

Data records in a database file are grouped into several members. At least one member must be included in one file.

When data is transferred to and from the AS/400 system, actual data transfer is done between file members. For example, a certain workstation file can be transferred to the AS/400 system. In this case, the file members become new members of a new or existing AS/400 file, or substitute for existing members in an existing AS/400 file.

Record format

A record format describes the fields contained in a file record and the order in which these fields appear in the record. Record formats are stored in the file description. Both physical and logical database files can have one or more record formats.

Creating a Workstation-to-AS/400 Transfer Request

To create a request for data transfer from a workstation to the AS/400 system, do as follows.

1. Using the router session, establish attachment to the AS/400 system to which data is to be transferred.
2. Select the **Data Transfer** icon.
3. When the AS/400→PC Transfer window has been displayed, select **Switch to SEND**. The display is switched to the PC→AS/400 Transfer window.

   To choose additional settings, select **Advanced**.
4. Specify each item. Refer to "Items to Be Specified" for details.

Items to Be Specified

The following section explains the items specified in the PC→AS/400 Transfer window. Those items that are specified by selecting **Advanced**, are indicated by the addition of "(Advanced)" at the end of the corresponding item name.

FROM

PC file name

This item is always required. It specifies the name of the workstation file containing the data to be transferred to the AS/400 system. Specify this item using the following format. (Items inside brackets [ ] can be omitted.)

`[d:][`*path-name*`]`*file-name*`[.ext]`

A list of workstation files can be displayed by selecting **Browse**. You can limit the number of names listed. To limit the listing, specify a combination consisting of part of a file name and a global file name character (* or ?) in the input area of the workstation file list. For example:

- When you select **OK** with **/A:** specified, the displayed listing contains the names of all files in the current directory of the diskette inserted into drive A.
- When you select **OK** with **A:\SUPPLY** specified, the displayed listing contains the names of all files under the **SUPPLY** path of the diskette inserted into drive A.
- When you select **OK** after specifying **B:*.XLS**, the displayed listing contains the names of all files having extension **XLS** in the current directory of the diskette inserted into drive B.

TO

System name

This item is always required. When the router program is active, this item specifies the default system name.

Library/File (Member)

This item is always required. It specifies the name of the AS/400 physical file that will receive the data to be transferred from the workstation. Either an existing file name or new file name can be specified.

Specify this item using the following format. (Items inside brackets [ ] can be omitted.)

`[`*library-name*`/]`*file-name*`[(`*member-name*`[,`*record-format-name*`])]`

library-name
> This is the name of the AS/400 library containing the AS/400 file to which data is to be transferred. If no library is specified, ***LIBL** is used. To create a new file to receive transferred data, specify the library name.
>
> When the input field is null and **Browse** is selected, the AS/400 system displays a list of all libraries defined in ***USRLIBL** of the AS/400 job

library list. This list can be modified by changing the job description. Run a change job description (**CHGJOBD**) command on the AS/400 system.

file-name

This is the name of an AS/400 physical database file. When data is transferred to an existing file, the data in that file is replaced with the transferred data. To create a new file to receive transferred data, specify a new file name of 1 to 10 characters.

To list the available files, do one of the following things:

- To list all files within all libraries defined in ***USRLIBL** of the AS/400 job library list, specify ***USRLIBL** followed by a slash (/), then select **Browse**. If a slash (/) is not specified after the library name, the AS/400 system displays a list of library names rather than the file names.

- To list the names of the files in a certain library, specify the library name followed by a slash (/), then select **Browse**. You can also specify a part of a file name followed by an asterisk (*), then select **Browse**. The AS/400 system lists all the files whose names begin with the specified character string.

member-name

This is the name of a member in the specified AS/400 file to which data is to be transferred. If this member name is not specified, data is transferred to the first member, ***FIRST**, in the AS/400 file. When the specified member already exists, the data in that member is replaced with the transferred data.

To transfer data to an existing file, specify the member name. The data within that file member is replaced with the transferred data.

To create a new member in an existing file or in a new file, specify a new member name of 1 to 10 characters.

By selecting **Browse** with a file name specified, the names of the members in that file are listed. When a left parenthesis, part of a member name, an asterisk (*), and a right parenthesis are specified, in this order, and then **Browse** is selected, the AS/400 system can list all member names beginning with the specified character string.

record-format-name

This is the name of the record format in the specified AS/400 file. The record format name need not be specified except when a physical file contains more than one record format. Most physical files have only one record format. Before specifying a record format name, a member name or ***FIRST** must be specified as the member name.

When you transfer data to an existing file without specifying a record format name, it is assumed that the file has only one record format (***ONLY**). Therefore, that record format is used.

When a new file is created with no record format name, **QDFTFMT** is used as the record format name.

Note: A library name, file name, member name, and record format name can be specified using up to 10 characters each. Each name must begin with one of characters A to Z, ¥, #, and @. For characters subsequent to the first, 0 to 9, underscores, and periods can also be used.

Use of File Description File (Advanced)

This item is always required. It specifies whether a file-description file is used to transfer data to the AS/400 system. The file-description file is required to transfer a workstation file, containing the data to be transferred (and converted), in fields. Such a workstation file can have either several fields or numeric data fields. To transfer a workstation file containing text (character data) only, the file-description file is not required. For details on creating a file-description file, see "File-Description Files" on page 217.

Follow either of the two procedures below.

- Do not specify this item in the following case: a workstation file having only one field (for example, PC code character) is specified in **FROM**, while the AS/400 file is a physical source file having the following record format.

  ```
     Field          Type          Length        "  " Decimal Places

  Order number   Zoned            6                      2
  Date           Zoned            6                      0
  Data           Character     1 to 4096
                 or Open
  ```

  Note: When fields contain character data or zoned data only, the data portion can be broken down into several fields. The destination AS/400 file contains the fields for order number and date. The workstation file, however, does not. This method is recommended when transferring text only between the AS/400 system and the workstation.

- Specify this item in all cases other than that described above. Two examples are:
  - Data is transferred from a workstation file having more than one field.
  - The AS/400 file that receives the data is other than a physical source file having the record format described above.

File Description File Name (Advanced)

This item appears only when item **Use of File Description File** is specified.

This item is always required. It specifies the name of the workstation file-description file that describes the data to be transferred.

Upon transferring data from the AS/400 system to a workstation, a file-description file might have been created.

A file-description file must be created when the data has not yet been transferred from the AS/400 system to a workstation or when no file-description file exists.

PC File Type (Advanced)

This item appears only when **Use of File Description File** is not specified.

This item is always required. You must specify the type of the workstation file specified in the **FROM** field. The values provided by the AS/400 system are recognized as workstation code text. If the file type of a data file is not converted, the file can include nothing other than data that does not require conversion.

AS/400 Object (Advanced)

This item is always required. It specifies whether the AS/400 member to which data is transferred is a new member or an existing member. When data is transferred to a new member, this item also specifies whether the file to contain the new member is an existing file.

Create New Member

This item specifies that a new member, to which data is transferred, is created in an existing AS/400 file.

Notes:

1. To create a new member, you must have the following authorities:
   - ***OBJOPR**, ***OBJMGT**, and ***ADD** for the file that will include the new member
   - ***READ** and ***ADD** for a library that will contain the file

   See *Security Descriptions* (SC41-8083) for details of object authorities.

2. To create a member to add to a file, the transfer function uses the AS/400 default value for the add physical file member (**ADDPFM**) command.

When you specify this item, the following item must also be specified:

Member Text

This item is optional. It is used to add an explanation of a new AS/400 member. This explanation helps remind you of the contents of the member. This explanation appears, for example, when a list of all members in a file is requested (**Browse** is selected). If this item is left blank, no explanation is added to the new AS/400 member.

To specify an apostrophe (') in the explanation, enter two apostrophes (' ').

Create New Member in New File

This item specifies that a new member, to which data is to be transferred, is created in a new AS/400 file.

Notes:

1. To create a new member in a new file, ***READ** and ***ADD** authorities are required for the library that will contain that file. Authority to use the create physical file (**CRTPF**) command of the AS/400 system is also required.

2. To create a new member in a new file, the transfer function uses the default value for the create physical file (**CRTPF**) command of the AS/400 system. It does not, however, use the following values:
   (**MAXMBRS**[*NOMAX]). This indicates that the file can contain up to 32,767 members.
   (**SIZE**[*NOMAX]). This indicates that each member of the file can contain an unlimited number of records.

When this item is specified, also specify the following item:

Member Text

This item is optional. It is used to add an explanation of a new AS/400 member. This explanation helps remind you of the contents of the member. This explanation appears, for example, when a list of all the members in a file is requested (**Browse** is selected). If this item is left blank, no explanation is added to the new AS/400 member.

To specify an apostrophe (') in the explanation, enter two apostrophes (' ').

Data Transfer for PC400

AS/400 File Type
This item is always required. It specifies the type of an AS/400 file and the members to be created (same type for both).

Specify one of the following things:
- To create an AS/400 physical source file and its members, specify **Source**. These members are created with two fields (order number and date) added to the beginning of the data transferred from the workstation file. A new AS/400 source file and its members have the following record format:

```
Field          Type        Length        Decimal Places

Order number   Zoned          6              2
Date           Zoned          6              0
Data           Character   1 to 32755
               or Open
```

Note that in an AS/400 physical source file, each record can be up to 32,755 bytes in length. But, the maximum size of a source file created using the workstation-to-AS/400 transfer function is 4,107 bytes. Also, this file must include the order and date fields. Therefore, the maximum amount of data that can be transferred is 4,096 bytes per record.

The data portions of members inherit the workstation file characteristics. In other words, when a workstation file is a workstation code text file consisting of many records containing text, the created data fields will be the same.

- To create an AS/400 physical data file and its members, specify **Data**. The file and members will contain only the data fields described in the file-description file.

The value of the **AS/400 File Type** is assumed to be **Data** when a file-description file is used to transfer data. If a file-description file is not used for data transfer, the value of this item is assumed to be **Source**.

Field Reference File Name
This item appears only when **Use of File Description File** is specified for the creation of a new file.

When **Use of File Description File** is not specified, an AS/400 physical source file is created. **AS/400 File Type** and **Field Reference File Name** are not displayed. Instead, **Record Length** appears.

This item is always required. A new AS/400 file is created using the field name in a file-description file and the field definitions in an AS/400 field-reference file.

The format of a field-reference file name is as follows. (Items inside brackets [ ] can be omitted.)

[library-name/]file-name

library-name:
This is the name of an AS/400 library containing a field-reference file. If this library name is not specified, ***LIBL** is assumed. If you cannot find the desired library, selecting **Browse** displays a list of all libraries in

*USRLIBL of the AS/400 job library list. ***USRLIBL** of the library list can be changed by modifying the job description by executing a **CHGJOBD** command on the AS/400 processor.

file-name:

This is the name of the AS/400 physical database file containing the field definitions. Always specify this file name. When a library name is specified concurrently, use a slash (/) to delimit the library name and file name. If the desired file cannot be found, enter the library name and a slash, then select **Browse**. The system displays a list of files in that library. To list all the files in the libraries defined in ***USRLIBL** of the AS/400 job library list, enter ***USRLIBL/** then select **Browse**.

If you enter part of a file name followed by an asterisk (*) and then select **Browse**, the system displays a list of available file names, each beginning with the specified part of the name.

For example, enter **ARLIB/AR*** in the **Field Reference File Name** item, then select **Browse**. The system displays a list of all physical file names beginning with AR in library ARLIB.

Note: You must have ***OBJOPR** authority for the field-reference file to be specified. To list certain files, you must also have ***OBJOPR** authority for those files.

Record Length

This item is always required. It specifies the record length of an AS/400 physical source file. When the data receiver is an AS/400 physical source file, the specified value must include the length of the order number and date fields that are added to a workstation file at transfer (the total length of these two fields is 12 bytes).

Authority

This item is always required. It specifies the authority level of a new AS/400 file.

Specify one of the following things:

- **Read/Write**. This enables other users to read from and write to the AS/400 file and allows the file name to be displayed in lists. However, users cannot delete the file (***OBJOPR**, ***READ**, ***ADD**, ***OBJMGT**, ***UPD**, and ***DLT** authorities). If other users might be transferring data from a workstation file to the AS/400 file, specify **Read/Write** or **All**.
- **Read**. This enables other users to read from the AS/400 file, and allows the file name to be displayed in lists. However, other users can neither write to the file nor delete it (***USE** authority).
- **All**. This enables other users to read from and write to the AS/400 file as well as delete it. The file name is displayed in lists (***ALL** authority).
- **None**. This prevents other users (except for the system administrator) from writing to or deleting the AS/400 file. The file name does not appear in lists (***EXCLUDE** authority).

File Text

This item is optional. It is used to add an explanation of a new AS/400 file. This explanation helps remind the user of the contents of the file. This explanation appears, for example, when a list of all files in a library is requested (**Browse** is selected). If this item is left blank, no explanation is added to the new AS/400 file.

To specify an apostrophe (') in the explanation, enter two apostrophes (' ').

Replace Existing Member

This item transfers data to an existing AS/400 member, specified in the **Library/File (Member)** item. The existing data in that AS/400 member is replaced with the transferred data.

Saving, Opening, Changing, and Executing a Transfer Request

The following section explains how to save, open, change, and execute, as a file, information (transfer request) on data to be transferred.

Saving a Transfer Request

Save a transfer request when the request is likely to be executed repeatedly. This eliminates the need to create a transfer request every time data is to be transferred. To save a transfer request, do as follows:

1. Specify the information needed for transfer, using the PC→AS/400 Transfer window.

2. After specifying the necessary information, select **Save** or **Save As** from the File menu of the menu bar.

   The Save Transfer Request File as window appears.

3. Specify each item, referring to the following explanation, then select **OK**.

   File Name

   Disk to which data is to be saved. Specify a file name or diskette file name. The default extension is **TFR**. Extension **TFR** identifies a file as a transfer request file.

   Description

   This item can be used to add an additional explanation of a transfer request, as required. The explanation can be up to 40 characters in length. This explanation is saved with the transfer request, and displayed in the list of transfer request names. It is, therefore, useful for identifying a transfer request.

4. The system asks whether the saved transfer request is to be registered in the PC400 folder.

   When **OK** is selected, the transfer request is registered as an icon. Subsequently selecting this icon transfers data according to the contents of the registered data transfer request.

Opening and Changing a Saved Transfer Request

To open and change a saved transfer request, do as follows:

1. Display the PC→AS/400 Transfer window.

2. Select **Open** from the File menu of the menu bar.

3. Specify the name of the file to be opened using the Open Transfer Request File window, then select **OK**.

The PC→AS/400 Transfer window reappears, and the transfer request information, saved to the specified file, appears for each item. This opens the saved transfer request.

4. Change the contents of the transfer request as necessary.
5. To save the changed contents, follow the procedure explained in "Saving a Transfer Request" on page 188.

Performing a Transfer Request

A transfer request can be performed in any of the following ways:
- By selecting the icon with which the transfer request has been registered
- By using the PC→AS/400 Transfer window of the Data Transfer icon

Selecting the Icon with Which the Transfer Request Has Been Registered: This method can be used only when a transfer request has been saved as an icon by using the PC→AS/400 Transfer window.

Selecting the corresponding icon starts data transfer.

Using the PC→AS/400 Transfer Window:

1. Before executing a transfer request, operations such as creating, opening, and changing a transfer request must be completed.

   Note: When data is transferred from a workstation to an existing member in an AS/400 file, the transferred data replaces the existing data in that member.

2. Select **Send** from the PC→AS/400 Transfer window.

   Data transfer starts.

3. After the transfer has been completed, select the **Cancel** push button or **Exit** from the menu bar File menu.

Conversion Errors That Can Occur during Transfer

Upon executing a transfer request, a file-description file (when specified) is read from the disk or diskette to be processed. The AS/400 system and workstation exchange information, if the data is transferable.

The workstation transfers records, one at a time, from the file specified in **FROM**. Transferred records are converted and stored in the AS/400 member specified in **TO**.

During this conversion process, conversion errors might occur. For example, the values in a workstation file might have to be rounded to fit the AS/400 fields. Another example is the case where the record length of a workstation file differs from that expected by the AS/400 system.

If such an error occurs, an error message is issued with the number of the workstation file record for which the error occurred and, sometimes, information about certain fields in that record.

If a severe error occurs, data transfer might stop. In such a case, stop the transfer request, correct the error, then rerun the transfer request.

When the error is not so severe, you can request that the system continue transferring data. By doing so, even if the same error occurs in another record, an error message does not appear and the transfer function automatically continues executing the transfer request.

Creating an AS/400-to-Workstation Transfer Request

To create a transfer request to receive data from the host, do as follows:

1. Select the **Data Transfer** icon.
2. When the PC→AS/400 Transfer window is displayed, select **Switch to RECEIVE** to switch the display to the AS/400→PC Transfer window.

   For the additional settings, select the **Advanced** button.
3. Which items are to be specified by the user vary with the data type, as follows:
   - Entire AS/400 file
   - Part of an AS/400 file
   - Data combined from several AS/400 files
   - Summary of record groups

   Before specifying each item, while referring to "Items to Be Specified" on page 192, note the following points regarding the data to be received.

Receiving an Entire AS/400 File

This is the simplest way of transferring data from the AS/400 system to a workstation. All records in a file and all the data in each record are transferred.

The **FROM** items that must be specified as follows:

System name:
> This item specifies the name of the system.

Library/File (Member):
> This item specifies the name of the AS/400 file.

SELECT:
> Specifying an asterisk (*) for this item indicates that all fields are to be transferred, or lists all the fields in the AS/400 file.

ORDER BY:
> This item is optional. It specifies how records will be grouped. When this item is left blank, records are not grouped (data is transferred in the same order it appears in the AS/400 file).

Receiving Part of an AS/400 File

Only part of an AS/400 file is transferred to the workstation.

The **FROM** items that must be specified as follows:

System name:
> This item specifies the name of the system.

Library/File (Member):
> This item specifies the name of the AS/400 file.

SELECT:
> This item specifies a field to be transferred.

WHERE:
> This item specifies the requirements that must be satisfied before records can be selected for transfer.

ORDER BY:
> This item is optional. It specifies how records will be grouped. When this item is left blank, records are not grouped (data is transferred in the same order it appears in the AS/400 file).

Receiving Data Combined from Several AS/400 Files

The data to be transferred can be stored in two or more AS/400 files. These files are assumed to be related. Based on this relationship, they can be linked or *joined*, as if all the data existed in a single file. The files can be transferred to the workstation after they have been joined. By using the AS/400-to-PC transfer function, this "join and transfer" function can be performed in a single step.

The **FROM** items that must be specified as follows:

System name:
> This item specifies the name of the system.

Library/File (Member):
> This item specifies the names of all AS/400 files from which data is to be transferred.

JOIN BY:
> This item specifies how to join or combine the data in each file.

SELECT:
> This item specifies a field to be transferred.

WHERE:
> This item specifies the requirements that must be satisfied before records can be selected for transfer.

ORDER BY:
> This item is optional. It specifies how records will be grouped. When this item is left blank, records are not grouped (data is transferred in the same order as it appears in the AS/400 file).

Receiving a Summary of Record Groups

A summary record is a single record that includes information on each set of records grouped from one or more AS/400 files.

The **FROM** items that must be specified as follows:

System name:
> This item specifies the name of the system.

Library/File (Member):
> This item specifies the names of all files from which data is to be transferred.

JOIN BY:
> This item is optional. It specifies the join conditions that must be satisfied before records can be joined.

GROUP BY:
> This item is optional. It must be specified only when the records of AS/400 files are classified into several groups. To group all records into a single group, this item need not be specified.

SELECT:
> Specifying this item causes a summary record to be created. The field names specified in **GROUP BY** can be specified.

WHERE:
> This item is optional. It specifies the requirements that each record to be grouped must satisfy. To group all records, this item need not be specified.

HAVING:

This item is optional. It specifies the summary record to be transferred. To transfer all summary records, this item need not be specified.

ORDER BY:

This item is optional. It specifies how summary records will be grouped. When this item is left blank, records are not grouped (data is transferred in the same order as it appears in the AS/400 file).

Items to Be Specified

The following section explains the items to be specified using the AS/400→PC Transfer window. Those items that can be specified by selecting **Advanced**, are identified by adding "(Advanced)" to the end of those item names.

FROM

System name

This item specifies the name of the host system that contains the data to be received. When the router program is active, this item specifies the default system name.

Library/File (Member)

This item is always required. It specifies the name or names of one or more files used to store data to be transferred. Up to 32 file names can be specified. To specify several files, delimit them with commas and use **JOIN BY**, displayed after all **FROM** items have been specified. Only the file name must be specified. Do not specify a comma as a part of a file name. When the other optional items are not specified, they are assumed automatically. For example, the library name, member name, and format name can be assumed to be ***LIBL**, ***FIRST**, and ***ONLY**, respectively. When the cursor is on the input field of **FROM**, selecting **Browse** lists libraries, files, members, and formats.

Note: To transfer data from an AS/400 physical file, you must have ***USE** authority for that file. To transfer data from an AS/400 logical file, you must have ***OBJOPR** authority for that file and ***READ** authority for each subordinate file.

Specify file names as follows. (Items inside brackets [ ]can be omitted.) To specify several file names, delimit the names with commas.

```
[library-name/]file-name[(member-name[,record-format-name])],
[library-name/]file-name[(member-name[,record-format-name])],...
```

library-name

This is the name of the AS/400 library that contains the AS/400 file to be transferred. This AS/400 file contains the data to be transferred from the AS/400 system to a workstation. If this library name is not specified, ***LIBL** is assumed. If you cannot find the desired library, selecting **Browse** displays a list of all libraries defined in ***USRLIBL** of the AS/400 job library list. ***USRLIBL** of the library list can be changed by modifying the job description by executing the **CHGJOBD** command on the AS/400 system.

file-name

This is the name of the AS/400 physical file, logical file, or DDM file from which data is transferred. This file name must always be specified. To specify a file name and library name concurrently, delimit them with a slash (/). If you cannot find the desired file name, enter the library name followed by a slash, then select **Browse**. The system then displays a list of

files contained in that library. To display a list of all the files in the libraries defined in ***USRLIBL** of the AS/400 job library list, enter ***USRLIBL/**, then select **Browse**.

member-name

This is the name of the AS/400 member containing the data to be transferred, or ***FIRST**. If this member is not specified, the system assumes ***FIRST**, and the first member of that file is used.

record-format-name

This is the name of the record format contained in the specified AS/400 file, or ***ONLY**. Before specifying the record format name, specify the member name or ***FIRST**. If the record format name is not specified, the system assumes ***ONLY**, and the only record format for that file is used. To specify a record format name, delimit the record format name and member name with a comma.

When the specified AS/400 file has several record formats, a record format name must be specified. If the file member name is not specified, a record format name cannot be specified.

Notes:

1. A library name, file name, file member name, and record format name can be specified using up to 10 characters for each. Each name must begin with A to Z, ¥, #, or @. For characters subsequent to the first, 0 to 9, underscores, and periods can also be used.

2. When the **FROM** field remains blank or a comma is entered to specify the next file name, selecting **Browse** displays a list of libraries defined in ***USRLIBL** of the AS/400 job library list.

3. Enter part of the file name, member name, or record format name, followed by an asterisk (*), then select **Browse**. The system displays a list of names beginning with the specified characters.

Example: You might want to transfer data from file member **ITEMMBR1** (first member) of file **ITEMMAST** in library **ITEMLIB**. **ITEMFMT** is the only record format of this file. The specification will be as follows:

```
ITEMLIB/ITEMMAST(ITEMMBR1,ITEMFMT)
```

Alternatively, specify:

```
ITEMLIB/ITEMMAST
```

JOIN BY (Advanced)

When several files have been specified in **FROM**, specify **JOIN BY**. When only one file has been specified in **FROM**, **JOIN BY** does not appear.

JOIN BY specifies how to link or join the records of the files specified in **FROM**. Each file specified in **FROM** must be joined with at least one other file that has been specified in **FROM**.

Use **JOIN BY** to specify one or more join conditions. The join conditions indicate the similarity of two files. Therefore, they indicate which records of one file are joined with those of another.

The join conditions are as follows:

field-name = field-name

Field name is the name of the field defined in the record format specified in **FROM**. The join conditions require two field names, one for each file to be joined.

Field names must be delimited by one of these:

=   Equal

<> or ><
      Not equal

>   Greater than

>=   Greater than or equal to

<   Less than

<=   Less than or equal to

When specifying fields in **JOIN BY**, observe the following rules.
- Join a numeric field to another numeric field. The field lengths and types do not have to be identical.
- Join a character field to another character field. The lengths do not have to be identical.

The field name to be specified might have been defined in the files specified in **FROM**. When such a field name is used in the following items, prefix the field name with the file qualifier:
- JOIN BY
- GROUP BY
- SELECT
- WHERE
- HAVING
- ORDER BY

The file qualifier is the character T (uppercase or lowercase) followed by a one- or two-digit number. Use T1 for fields defined with the first record format, T2 for fields defined with the second record format, and so on. Delimit the file qualifier and field name with a period (**.**). See "Receiving Records Using File Qualifiers" on page 211 for details of the file qualifiers.

If the field name of the file specified in **FROM** cannot be found, select **Browse** when the cursor is on the **JOIN BY** input area. Then, a list of file qualifiers and field names of the files appears.

To join three or more files, or to join two files based on two or more common fields, two or more link conditions must be used. To specify several join conditions, join the conditions with AND. For example:
```
T1.EMPNO = T2.EMPNO AND T2.EMPNO = T3.EMPNO
```

In this case, records having the same value as **EMPNO** are joined between the first and second files specified in **FROM**. Then, such records are joined between the second and third files specified in **FROM**.

Up to 32 join conditions can be specified.

After **JOIN BY** is specified, each of **SELECT**, **WHERE**, and **ORDER BY** can be completed, by following the procedure described earlier in this chapter. To browse a field name that has been defined in several files, prefix the field name with a file qualifier.

GROUP BY (Advanced)

This item is required only to classify AS/400 file records into several groups. When no value is specified in **GROUP BY**, all the records are treated as a single group.

If **GROUP BY** is not displayed, select **Group functions** at the bottom right of the screen. Then, **GROUP BY** appears. **GROUP BY** and **HAVING** are displayed concurrently. You can specify either, both, or neither.

When **GROUP BY** and **HAVING** are displayed but you do not want to specify either, select **Remove Group functions**. The two items disappear.

To classify several records into groups, specify one or more fields to act as the base for grouping. Records are grouped according to the field specified first, then by the field specified second, and so on. For example, suppose that the following groupings are specified:

```
SHIFT, DEPTNO
```

In this example, the records are first grouped by **SHIFT**. Records belonging to a single group will subsequently have the same value as **SHIFT**. Then, the records in each group are grouped by **DEPTNO**. When there is only one record having a certain **SHIFT** value, the group has only one record.

Delimit field names with commas. Blanks can be specified to improve readability. Up to 50 field names can be specified. These fields must have been defined in the record format defined in **FROM**.

If a field cannot be found, selecting **Browse** displays a list of all the fields contained in the record.

With **GROUP BY** specified, specify **SELECT** to transfer the summary record of each group.

SELECT (Advanced)

This item is always required. It specifies the field to be transferred or the function that indicates the type of summary information to be transferred.

The field to be specified must have been defined in the record format specified in **FROM**.

To transfer all the fields in the specified record, specify an asterisk (*) in this input field. (Specifying an asterisk causes all fields in the record to be transferred.)

Note: Up to 256 fields can be transferred. When more than 256 fields have been defined in a file, an asterisk cannot be used. In this case, specify the names by selecting the fields to be transferred.

To transfer fields by selecting from a record, enter the field names in the order in which the fields are arranged. One or more blanks can be placed between the field names to improve readability. However, the names must be delimited by commas, as follows:

```
ITEMNO, QONHAND, PRIC
```

Data Transfer for PC400

You can also specify:

```
ITEMNO,QONHAND,PRICE
```

When records are transferred from an AS/400 source file, specifying an asterisk (*) causes all fields in the file to be transferred, with the exception of the order number field and date field. (To transfer all the fields, including the order number field and date field, specify all the field names, including each data field name.)

A field can be specified repeatedly as required. However, bear in mind that no more than 256 fields can be selected. A list of field names can be displayed by selecting **Browse**.

Receiving a Summary of Record Groups:: The following information is necessary to receive summary records.

To transfer a summary record, do not leave this input area blank or specify an asterisk (*) (except when all the fields of the file specified at the prompt are specified in **GROUP BY**). The field names specified in **SELECT** (except for those specified in functions) must also have been specified in **GROUP BY**.

The functions and fields specified in **SELECT** return actual summary information for each group. Enter the field names and functions in **SELECT** in the order in which they are to be displayed.

Note: Null values are not included in the functions. When an entire value is null, the function output is set to null, except for **COUNT**. The **COUNT** output is 0.

The function format is as follows.

```
function (field-name)
```

This has the following meaning:

function:
This is one of the following functions:

AVG Transfers the average value of the specified fields for each record group. This function can be used only for numeric fields.

MIN Transfers the minimum or lowest value of the specified fields for each record group.

MAX Transfers the maximum or highest value of the specified fields for each record group.

SUM Transfers the total value of the specified fields for each record group. This function can be used only for numeric fields.

COUNT
Transfers the total number of records that satisfy the **WHERE** condition for each record group. Specify **COUNT(*)**.

field-name:
This is the field name defined with the record format specified in **FORM**.

Each function returns one value for each record group. In **SELECT**, several functions can be specified. To do so, delimit the functions by commas, as follows:

```
SUPPNO, AVG(PRICE), MIN(PRICE), MAX(PRICE)
```

This indicates that the average, minimum, and maximum values for **PRICE** are calculated for each supplier after **SUPPNO** has been selected. A summary record is transferred according to the function selection. Specify **SUPPNO** in **GROUP BY**, because **SUPPNO** has not been used for the functions.

WHERE (Advanced)

This item is optional. It specifies one or more conditions that records to be transferred must satisfy.

To transfer summary records, use this item to specify which records are to be grouped, then group the records. Using this item, you can specify one or more conditions that the record must satisfy to belong to a certain group. When **WHERE** is not specified, all records are grouped.

As the *conditions*, specify the test to be applied to the records in the specified file member. All the records in the specified file member are tested for the conditions specified here. Only those records that pass this test are transferred.

When **WHERE** is not specified, all records in the specified file member are transferred.

The condition format is as follows:

```
field-name  test  value
```

field-name:

This must be a field substring or field name defined in the record format.

Fields or constants can be manipulated by specifying a supported function, with the results being used for comparison. The supported functions and usage are as follows:

SUBSTR

Returns the specified part of a character string. This function contains three parameters: the field name, starting position, and length of the returned substring. The following example returns the 20 characters starting from the 10th character of the **FULLNAME** field.

SUBSTR(FULLNAME 10 20)

VALUE

Returns the first non-null value in the parameter list. (If all parameters are null, null is returned.)

VALUE(DEPOSIT WITHDRAW BALANCE)

CURRENT

Returns **DATE**, **TIME**, **TIMEZONE**, or **TIMESTAMP** for the current system.

CURRENT(TIMEZONE)

DIGITS

Returns a character string representation of a numeric field.

DIGITS(EMPLOYEE#)

CHAR

Returns a character string representation of the date field, time field, or time-stamp field. The second parameter is used to specify the format of the Systems Application Architecture (SAA) of the string to be returned (supported values are **USA**, **EUR**, **ISO**, or **JIS**).

CHAR(DATEHIRE USA)

DATE Returns the date of the time-stamp field.

> **DATE(TIMECRTD)**

TIME Returns the time of the time-stamp field.

> **TIME(TIMECRTD)**

TIMESTAMP

> Returns the time-stamp, combining the date field and time field.

> **TIMESTAMP(DATESEND TIMESEND)**

YEAR Returns the year of the date field or time-stamp field.

> **YEAR(DATEHIRE)**

MONTH

> Returns the month of the date field or time-stamp field.

> **MONTH(DATEHIRE)**

DAY Returns the date of the date field or time-stamp field.

> **DAY(DATEHIRE)**

DAYS Returns the day of the year, counted from January 1, of the date field or time-stamp field.

> **DAYS(DATEHIRE)**

HOUR

> Returns the time of the time field or time-stamp field.

> **HOUR(TIMESEND)**

MINUTE

> Returns the minute of the time field or time-stamp field.

> **MINUTE(TIMESEND)**

SECOND

> Returns the second of the time field or time-stamp field.

> **SECOND(TIMESEND)**

MICROSECOND

> Returns the microsecond of the time field or time-stamp field.

> **MICROSECOND(TIMECRTD)**

test: This is the comparison type to be applied to fields or functions.

The following tests can be used. One or more blanks can be placed before and after these tests.

= Equal

<> or ><
> Not equal (See the note.)

> Greater than

>= Greater than or equal to

< Less than

<= Less than or equal to

LIKE The field is similar to the specified value.

BETWEEN
> The field is equal to one of two constants, or to a value between them.

IN The field is the same as one of the values in the constant list.

IS   The field contains null values.

ISNOT
    The field contains no null values.

Using the **LIKE** Test

The **LIKE** test checks the field specified with the field name for a character pattern specified as a value. The field to be specified must be a character field.

The values to be tested must be character-string constants. This string can contain any characters. A percent (%) character (both SBCS% and DBCS%) indicates a character string consisting of zero or more characters. A 1-byte underscore (_) character indicates any single 1-byte character. A 2-byte underscore (_) character indicates any single 2-byte character.

The following example explains how to use the **LIKE** test:
```
NAME LIKE '%ANNE%'
```

The previous example searches for names containing character string **ANNE**, such as **ANNE**, **ANNETTE**, and **SUZANNE**.

The following example searches for names beginning with character string **ANNE**, such as **ANNE** and **ANNETTE**.
```
NAME LIKE 'ANNE%'
```

The following example searches for names ending with character string **ANNE**, such as **ANNE** and **SUZANNE**.
```
NAME LIKE '%ANNE'
```

The following example searches for all names whose second character is A.
```
NAME LIKE '_A%'
```

The following example searches for all last names beginning with character J.
```
LSTNAM LIKE 'J%'
```

This has the same effect as the following example:
```
SUBSTR (LSTNAM,1,1) = 'J'
```

When the pattern does not include a percent character (%), the length of the character string must be identical to that of the field.

Note: Values are searched according to the exact characters specified by the user. In other words, when the user's specification consists only of uppercase characters, only uppercase character strings are returned. Similarly, when the specification consists only lowercase characters, only lowercase character strings are returned.

Using the **BETWEEN** Test

The **BETWEEN** test checks the fields specified in the field name for character strings or numeric values that are equal to or between the specified constants. The values to be tested must be two character-string

constants or two numeric constants. The types of these constants must be identical to that of the field name specified by the user. Delimit the two constants with **AND**.

The following example searches for those records for which the price is between 50.35 and 75.3, inclusive:

```
PRICE BETWEEN 50.35 AND 75.3
```

The following example searches for those records for which the name begins with C:

```
NAME BETWEEN 'C' AND 'CZZZZZZZZZ'
```

The following example searches for those records for which the balance is between 0 and 5 000.

```
BALDUE BETWEEN 0 AND 5000
```

This has the same meaning as the following expression.

```
BALDUE >= 0 AND BALDUE <= 5000
```

Note: Specify the values to be tested in the form of **BETWEEN** (minimum) **AND** (maximum). For instance, **BETWEEN 1 AND 10** is a valid specification. However, **BETWEEN 10 AND 1** returns no records.

Using the **IN** Test

The **IN** test checks the fields specified in the field name for the character strings or numeric values in the list specified as the value. The value to be tested must be a list of character-string constants or numeric constants. In addition, the types of these constants must be identical to that of the specified field. Delimit the constants with blanks and enclose them in parentheses. Up to 100 constants can be specified. The following example shows how to use the **IN** test:

```
NAME IN ('SMITH' 'JONES' 'ANDERSON')
```

This example searches for those records for which the name is **SMITH**, **JONES**, or **ANDERSON**.

The following example searches for the values in the **STATE** field for which the value is other than **NY**, **MN**, or **TX**:

```
NOT STATE IN ('NY' 'MN' 'TX')
```

Note: Values are searched according to the exact characters specified by the user. In other words, when the user's specification consists of only uppercase characters, only uppercase character strings are returned. Similarly, when the specification consists of only lowercase characters, only lowercase character strings are returned.

Using the **IS** Test

The **IS** test checks the fields specified in the field name for null values.

The following example searches for those records for which the commission field contains null values:

```
COMMISSIONS IS NULL
```

Using the **ISNOT** Test

The **ISNOT** test checks the fields specified in the field name for non-null values.

The following example searches for those records for which the commission field does not contain null values:

```
COMMISSIONS ISNOT NULL
```

In the test, logical **AND** and logical **OR** can be combined. When both **AND** and **OR** are specified, **AND** comparison is performed first. Up to 50 conditions can be specified. For example:

```
MONTH=2 AND LOC='MIAMI' OR LOC='CHICAGO'
```

In this example, each record to be selected must satisfy the following condition:

```
MONTH=2 AND LOC='MIAMI'
```

or must satisfy the following condition:

```
LOC='CHICAGO'
```

This command can be modified by using parentheses. For example:

```
MONTH=2 AND (LOC='MIAMI' OR LOC='CHICAGO')
```

In this example, each record to be selected must satisfy the following condition:

```
MONTH=2
```

and it must satisfy the following condition:

```
LOC='MIAMI' OR LOC='CHICAGO'
```

NOT can also be used. The following example selects items where data is transferred not only from those records in which the **DEPT** field is not equal to 470, but also from those records for which the **DEPT** field is equal to 470 and, additionally, **STATE** is equal to **NY**.

```
NOT (DEPT = 470) OR (DEPT = 470 AND
  STATE = 'NY')
```

Comparison can start from a certain line and end at the next line. However, a field name cannot start from a certain line and end at the next line. Field names must not exceed one line.

When a value to be tested is a character string enclosed in quotation marks, the value can start from a certain line and continue to the next line.

value Field names or constants can be specified as values. To use constants, enclose number or character strings in single quotation marks.

Note: Do not specify commas between numbers, because the system recognizes these as decimal points.

When a character field or substring is specified in the field name, the character field name or character-string constant must be specified as the *value*. If the field length differs from the value length, the shorter field is extended to the value length by adding blanks on the right.

When a numeric field is specified as the field name, a numeric field name, numeric constant, or expression must be specified as the *value*. The transfer function performs decimal alignment for numeric fields prior to comparison. Zeros are added wherever numeric values are missing.

When a date field is specified in the field name, the name of a date field, character field (when the **CHAR** function is used for date field conversion), or character-string constant must be specified as the *value*. To use a character-string constant, use a value in a form other than SAA (**USA**, **EUR**, **ISO**, or **JIS**).

Using Constants

A character string, numeric value, or keyword **NULL** can be specified as a constant.

Note: **NULL** can be used only when operators **IS** or **ISNOT** are used.

Using Character Strings

A character string must be enclosed in apostrophes, as follows:
```
'JOHN'
```

Use two apostrophes to represent one apostrophe in a character string, as follows:
```
'JOHN''S'
```

Note: The transfer function supports only right apostrophes (').

When a character string is shorter than the specified field, the string is extended to the value length by adding blanks on the right. When a character string is longer than the specified field, the string is truncated.

Values are searched according to the exact characters you specified. In other words, when your specification consists only of uppercase characters, only uppercase character strings are returned. Similarly, when the specification consists only lowercase characters, only lowercase character strings are returned.

A character string can also be specified in hexadecimal format. In this case, enter X. Then, enter the character string, enclosed in apostrophes, using EBCDIC hexadecimal representation. For example:
```
X'D1D6C8D5'
```

If the character string contains any characters that cannot be represented, hexadecimal representation can be used. A character string can also be specified using DBCS representation, enclosed in apostrophes (DBCS mixed, DBCS exclusive, or DBCS alternative). In this case, first enter G. Then, enter the DBCS string, enclosed in apostrophes. For example:
```
    G' D B C S'
```

A character string can also be specified using DBCS graphic representation. In this case, first enter N. Then, enter the DBCS string, enclosed in apostrophes. An example is given below.
```
    N' D B C S'
```

<u>Using Numeric Values</u>

A numeric value can contain an optional sign (+ or −) and optional decimal point. Either a period (**.**) or comma (**,**) can be used to represent a decimal point. For example:

```
.5
-.05
1000.00
+5.00
```

Decimal alignment is performed prior to numeric field comparison. Zeros are added where numeric values are missing.

<u>Using Expressions</u>

An expression can contain numeric field names, numeric constants, parentheses, and the following arithmetic operators:

+   Plus sign (addition)

−   Minus sign (subtraction)

*   Multiplication

/   Division

For instance, the following expression searches for those records for which the number of products in stock (**ONHAND**) is at least 500 greater than the number of ordered products (**ONORDER**).

```
ONHAND > ONORDER + 500
```

The following expression searches for those records for which the price is 1 000.

```
PRICE = (10.00 * 100)
```

HAVING (Advanced)
This item is optional. It specifies which summary record is transferred.

Pay particular attention to the difference between **HAVING** and **WHERE**. **WHERE** operates on each record within a certain group. **HAVING**, on the other hand, operates only on summary records (records that contain summary information for each group).

With this item, you can specify one or more conditions that a summary record must satisfy prior to being transferred.

As the conditions, specify the tests that should be applied to the summary records. The specified test conditions are applied to all summary records, only those summary records that pass the tests are transferred. To transfer all summary records, leave the **HAVING** item blank.

The format of the conditions is as follows:

```
function (field-name)  test  value
```

This indicates:

function
> This is a function supported for **SELECT**. See the description of **SELECT** in this section for details of these functions.

field-name

This is the field defined by the record format specified in **FROM**. A field name is acceptable even when it has not been specified in **SELECT**.

Test

This is the comparison type for functions. The types are listed below.

=   Equal to

<> or ><

Not equal

>   Greater than

>=   Greater than or equal to

<   Less than

<=   Less than or equal to

value

This is a function operating on certain fields or a constant. See "WHERE (Advanced)" on page 197 for details of constants, expressions, and tests.

Note: A comma is treated as a decimal point. Therefore, do not separate numbers with commas.

Test conditions can be combined by using logical **AND** or logical **OR**. When both **AND** and **OR** are specified, **AND** comparison is performed first. Up to 50 tests can be specified. By using parentheses, the operation order can be modified, or a description can be added to an operation. For example, you can specify:

```
COUNT(*) >=2 AND MAX(PRICE) > 100
```

In this case, the following conditions are applied concurrently: groups to be transferred must contain more than one record, and the summary records in such groups are transferred only when the maximum price is greater than 100.

If the desired field cannot be found, selecting **Browse** displays a list of the names of all fields in the record.

The type, length, digit, and number of decimal places of the value returned for each function are:

```
         Type     Length    Digit      Decimal Places
SUM      Packed     16        31      (Same as tested field)
AVG      Packed     16        31       31 (Total of the digit
                                           and decimal places
                                           of the field)
COUNT    Binary      4        10        0
MAX              (Same as tested field)
MIN              (Same as tested field)
```

ORDER BY (Advanced)

This item is optional. It specifies the order in which the requested records are grouped. When **ORDER BY** is not specified, record transfer is not done according to a certain order.

Records are grouped according to the field specified first. Those records having the same value in each field specified first are grouped by the field specified second, and so on. Records containing null values are grouped after all records without null values have been grouped.

For example, you can specify:

```
DEPT,NAME,PHONE
```

In this case, records are first grouped according to **DEPT**. Then, the records having the same value for **DEPT** are grouped by **NAME**. The records with the same **DEPT** and **NAME** values are finally grouped by **PHONE**.

When a field name is specified in **ORDER BY**, it must also have been specified in **SELECT**, or **SELECT*** must have been specified.

Fields can be grouped in ascending or descending order. To do this, specify one blank after a field name then enter **ASC** or **DESC**. The default value is **ASC**. For example, specify:

```
DEPT DESC, NAME ASC
```

This indicates that the **DEPT** fields are to be grouped in descending order, after which the **NAME** fields are to be grouped in ascending (alphabetic) order.

Absolute values (**ABS**) can be specified for numeric fields. To do this, add a blank after a field name then enter **ABS**. For those fields having negative values, the negative signs are ignored and the absolute values are used.

The total length of the fields to be specified must not exceed 120 digits.

Return Record at Missing Field Value (Advanced)
When joining records from several files, joining might fail because a record is missing. This item specifies whether records with missing fields are transferred.

When you specify that records with missing fields are to be transferred, the alternative values for the missing fields are transferred. These values are normally blanks for character fields and zeros for numeric fields.

When you do not specify transfer of records with missing fields, those records are not transferred.

Specify this item to transfer data records that have alternative values for missing fields.

Do not specify this item if data records that have alternative values for missing fields are not to be specified. In this case, only those data records created from those records that exist in all files specified in **FROM** are transferred.

TO

Output device
This item specifies where received data is to be sent.

Display
> The received data is displayed on the screen.

Disk The received data is written to a workstation diskette or hard disk file.

Printer
> The received data is printed on the printer.

When **Disk** is selected as the output device, also specify the following items.

PC file

This item specifies the name of the workstation disk file or diskette file to which the data is to be written.

Replace old file

This item is always required. It specifies whether the records in the file specified by **PC File** are to be replaced with the transferred records.

The default value is **Replace old file**.

Workstation file type

This item is always required. It specifies the type of the workstation disk file or diskette file to which the transferred records are written.

The system default is PC code test.

Save transfer description

This item is always required. It specifies whether the workstation file description is written to a workstation file. This file description describes the transferred data and it is required to subsequently return data to the AS/400 system.

The system default is **Save**.

Description file name

This item is always required. It appears only when **Save Transfer Description** is selected. The **File Description File Name** specifies the name of the workstation disk file or diskette file to which the file description is written.

This item automatically sets the desired file name. This file name is the same as that specified by the user for **TO**, but to which extension **.FDF** has been added. Extension **.FDF** indicates that this file is a file-description file.

The use of extension **.FDF** is recommended when using a unique file name. To specify a file name in this item, use the same format as that in **TO**. (Items inside brackets [ ] can be omitted.)

[*d:*][*path-name*]*file-name*[*.ext*]

After **Save File Description File** is specified or a name is specified for **File Description File Name**, the AS/400→PC Transfer Request window reappears after the **Return** key is pressed. Using this screen, a transfer request can be changed, saved, or executed.

Saving, Opening, Changing, and Executing a Transfer Request

The following section explains how to save, open, change, and execute, as a file, the information (transfer request) on the data to be transferred.

Saving a Transfer Request

You should save a transfer request, especially when the request will be executed repeatedly. This eliminates the need to create a transfer request every time a request is executed. To save a transfer request, do as follows:

1. Specify the information needed for transfer, using the AS/400→PC Transfer window. See "Creating an AS/400-to-Workstation Transfer Request" on page 190 for an explanation of how to specify the required data.

2. After specifying the necessary data, select **Save** or **Save As** from the File menu of the menu bar.

   The Save Transfer Request File As window appears.

3. Specify each item, referring to the following explanation, then select **OK**.

File Name

Disk to which data is to be saved. Specify a file name or diskette file name. The default extension is TTO. Extension TTO identifies a file as a transfer request file.

Description

This item can be used to add a short explanation of a transfer request, as required. The explanation can be up to 40 characters in length. This explanation is saved with the transfer request, and displayed in the list of transfer request names. It is useful, therefore, for identifying a transfer request.

4. The system asks whether the saved transfer request is to be registered in the PC400 folder.

When **OK** is selected, the transfer request is registered as an icon. Subsequently selecting this icon transfers data according to the registered data transfer request.

Opening and Changing a Saved Transfer Request

To open and change a saved transfer request:

1. Display the AS/400→PC Transfer window.
2. Select **Open** from the File menu of the menu bar.

   The Open Transfer Request File window appears.
3. Specify the name of the file to be opened using the Open Transfer Request File window. Then select **OK**.

   The AS/400→PC Transfer window reappears, with the information specified for each item for the transfer request displayed. This completes opening of the saved transfer request.
4. Change the contents, as necessary.
5. To save the changed contents, follow the procedure given in "Saving a Transfer Request" on page 206.

Executing a Transfer Request

A transfer request can be executed in one of the following three ways:
• By selecting the icon with which the transfer request has been registered
• By using the AS/400→PC Transfer window of the Data Transfer icon

Selecting the Icon with Which the Transfer Request Has Been Registered::   This method can be used only when a transfer request has been saved as an icon by using the AS/400→PC Transfer window.

Data transfer starts as soon as you select the icon with which a transfer request has been registered.

Using the AS/400→PC Transfer Window::

1. Before attempting to execute a transfer request, all operations such as creating, opening, and changing a transfer request must have been completed.

   Note: When data is transferred from a workstation to an existing member in an AS/400 file, the transferred data replaces the existing data in the member.

2. Select **Receive** from the AS/400→PC Transfer window.

   Data transfer starts.
3. After the transfer has been completed, select the **Cancel** push button, or select **Exit** from the menu bar File menu.

Status during Transfer

Display can be specified as the **output device**, when the current transfer request is created or changed. This sends the transferred record to the screen. On the screen, each record is displayed on one line.

Each field in a transferred record is converted from the AS/400 data type to workstation code.

Note: The workstation receives the AS/400 records in order and then writes them to a temporary file of the default directory in the default drive (usually, the directory in which PC400 is installed). The maximum number of records that can be transferred is 4096 records, limited by the amount of records that can be stored in free space of the default drive.

When **Disk** is selected as the **output device**, the following actions are performed:

1. The workstation file description is written to a workstation disk file or diskette file according to the **Save File Description File** specification. (If **Save File Description File** has not been specified, this procedure is not performed.)
2. The transferred records are written to a workstation disk file or diskette file.

Limited Usage of File Names and Field Names

For a transfer request from a workstation to the AS/400 system, none of the following reserved words can be specified as a file name or field name:

CRTFILE	MBRTEXT
CRTMBR	PUBAUT
FILETEXT	RCDLEN
FILETYPE	REFFILE
INTO	

For a transfer request from the AS/400 system to a workstation, none of the following reserved words can be used as a file name or field name:

ABS	IS
AND	ISNOT
ASC	LIKE
AVG	MAX
BETWEEN	MIN
BY	NOT
COLUMNS	OPTIONS
COUNT	OR
DESC	ORDER
EXTRACT	PARTOUT
FROM	REPLACE
GROUP	SELECT
HAVING	SUBSTR
IN	SUM
INNER	TABLES
	WHERE

To use one of these reserved words as a file name or field name, use the reserved word in uppercase, enclosed in quotation marks. Here is an example:

```
TO MYLIB/"INTO"
```

Examples of Transfer Requests for Receiving

This section provides examples of transfer requests for receiving. The contents of this section provide supplementary information to help you better understand transfer requests for receiving.

This section describes how to transfer data from the AS/400 system, based on the inventory control file **INVENTORY** and supplier file **SUPPLIERS**.

The **INVENTORY** file contains information about the various parts in stock. Each part has a three-digit identification number, **PARTNUM**. The **INVENTORY** file contains the names of parts (**DESCRIPTION**) and the quantity on hand (**QONHAND**) for each part.

```
          File: INVENTORY
  Field name: PARTNUM  DESCRIPTION  QONHAND
              -------  -----------  -------
  Record 1:    209          CAM        50
        2:     221         BOLT       650
        3:     222         BOLT      1250
        4:     231          NUT       700
        5:     232          NUT      1100
        6:     207         GEAR        75
        7:     241       WASHER      6000
        8:     285        WHEEL       350
        9:     295         BELT        85
```

The **SUPPLIERS** file contains information about the suppliers of each part. Each supplier is identified by a two-digit number, **SUPPNO**. The **SUPPLIERS** file contains the number of parts delivered (**PARTNO**), their prices (**PRICE**), times of delivery (**DELIVTIME**), and ordered quantities (**QONORDER**). The parts listed in the **SUPPLIERS** file are the same as those listed in the **INVENTORY** file.

```
          File: SUPPLIERS
  Field name: SUPPNO   PARTNO   PRICE   DELIVTIME   QONORDER
              ------   ------   -----   ---------   --------
  Record 1:     51      221      .30       10          50
        2:      51      231      .10       10           0
        3:      53      222      .25       15           0
        4:      53      232      .10       15         200
        5:      53      241      .08       15           0
        6:      54      209    18.00       21           0
        7:      54      221      .10       30         150
        8:      54      231      .04       30         200
        9:      54      241      .02       30         200
       10:      57      285    21.00       14           0
       11:      57      295     8.50       21          24
       12:      61      221      .20       21           0
       13:      61      222      .20       21         200
       14:      61      241      .05       21           0
       15:      64      207    29.00       14          20
       16:      64      209    19.50        7           7
```

Receiving Part of an AS/400 File

Specify the following items:
Library/File (Member)
      INVENTORY
SELECT
      PARTNUM, QONHAND
WHERE
      QONHAND < 100

ORDER BY
        PARTNUM

In this case, only part of the **INVENTORY** file is to be transferred. Specifically, only the part number (**PARTNUM**) and quantity on hand (**QONHAND**) fields of the records for which the number of parts in stock is less than 100 (**QONHAND** < 100) are transferred. Records are transferred in ascending order of parts numbers (**PARTNUM**).

The following data is transferred:

```
     Field:   PARTNUM  QONHAND
              -------  -------
  Record 1:     207      75
         2:     209      50
         3:     295      85
```

Receiving Records Joined from Several AS/400 Files

Two AS/400 files, **INVENTORY** and **SUPPLIERS**, are assumed. Note that both files contain records including part number fields. The **INVENTORY** file contains inventory information about individual parts. The **SUPPLIERS** file contains information about purchasing and ordering.

The user might want to transfer information on part numbers, part names, and the prices of the parts to be ordered from supplier 51. The desired fields are **PARTNO** (**SUPPLIERS** file), **DESCRIPTION** (**INVENTORY** file), and **PRICE** (**SUPPLIERS** file).

By comparing the data in the **INVENTORY** file and the **SUPPLIERS** file, the user can determine that supplier 51 provides part numbers 221 and 231, called **BOLT** and **NUT**, respectively, and that their prices are 30 cents and 10 cents, respectively. The following table summarizes this information:

```
     Field:   PARTNO  DESCRIPTION   PRICE
              ------  -----------   -----
  Record 1:    221    BOLT           .30
         2:    231    NUT            .10
```

The same results are available by joining the data in these two files by using the AS/400-to-PC transfer function. To do this, specify both files (**INVENTORY** and **SUPPLIERS**) in the **FROM** item. For **SELECT**, specify which fields are to be transferred (**PARTNO**, **DESCRIPTION**, and **PRICE**). For **WHERE**, specify which records are to be transferred (records for which **SUPPNO** = 51).

Respecify the relationship between the two files in **JOIN BY**. From these results, the user can determine, by checking the **SUPPLIERS** file, that part number 221 is delivered from supplier 51 at a cost of 30 cents. In addition, to determine the part name, the user must check the **INVENTORY** file for part number 221 and its product name. In other words, the user observes that data is joined from the records in both the **SUPPLIERS** file and the **INVENTORY** file and that those records have the same part number. Therefore, to link the two records in these files, the records must have the same part number.

In short, to obtain this information, specify:
Library/File (Member)
        SUPPLIERS, INVENTORY
JOIN BY
        PARTNO = PARTNUM

```
SELECT
        PARTNO, DESCRIPTION, PRICE
WHERE
        SUPPNO = 51
ORDER BY
        PARTNO
```

Receiving Records Using File Qualifiers

To join records from several AS/400 files, fields of the same type must be joined.

For example, the part number fields in the **INVENTORY** and **SUPPLIERS** files can have the same name **PARTNO**. To specify the desired **PARTNO** fields, the user must specify which file contains those fields. To do so, file qualifiers are used.

A file qualifier is the character **T** (uppercase or lowercase) followed by a one- or two-digit number. A comma is used to delimit the file qualifier and field name. In the previous example, prefix **T1.** and **T2.** to the **PARTNO** field names. **T1.** indicates the first file of **FROM**, while **T2.** indicates the second.

To obtain the same information as in the previous example, specify:

```
Library/File (Member)
        SUPPLIERS, INVENTORY
JOIN BY
        T1.PARTNO = T2.PARTNO
SELECT
        T1.PARTNO, DESCRIPTION, PRICE
WHERE
        SUPPNO = 51
ORDER BY
        T1.PARTNO
```

T1.PARTNO indicates the **PARTNO** fields in the **SUPPLIERS** file, while **T2.PARTNO** indicates the **PARTNO** fields in the **INVENTORY** file.

Qualifiers are not needed for the names of the **DESCRIPTION**, **PRICE**, and **SUPPNO** fields, because they exist in one file only. However, the user can specify the following qualifiers for clarity:

`T2.DESCRIPTION, T1.PRICE, T1.SUPPNO`

The following examples of joining several AS/400 files describe more sophisticated techniques. You should now be familiar with the basics of how to join two files. For a more detailed explanation, refer to the following sections.

Receiving with Field Missing Records Joined

The joining of records from several files could fail because one or more records is missing. For example, the record containing part number 221 might not be found in the **INVENTORY** file. This means that the records that can be joined to the 1st, 7th, and 12th records in the **SUPPLIERS** file do not exist in the **INVENTORY** file. In this case, the **PARTNO** field and **PRICE** field for part number 221 can be determined, but the **DESCRIPTION** field cannot be determined. So, the **DESCRIPTION** field is missing.

To transfer field missing records, use **Return Record at Missing Field Value**.

When **Return Record at Missing Field Value** has been specified, the default AS/400 values are transferred instead of the missing field values. The default values for character fields are blanks, while those for numeric fields are zeros. For example, if the **INVENTORY** file does not contain the part number 221 record, the result of the previous example will be as follows:

```
    Field: PARTNO   DESCRIPTION   PRICE
           ------   -----------   -----
Record 1:    221                   .30
       2:    231   NUT             .10
```

If **Return Record at Missing Field Value** has not been specified, the field missing records are not transferred. For example, if the **INVENTORY** file does not contain the part number 221 record, the result of the previous example will be as follows:

```
    Field: PARTNO   DESCRIPTION   PRICE
           ------   -----------   -----
Record 1:    231   NUT             .10
```

Receiving with Records in a Same File Joined

Records in the same file can be joined. In other words, a file can be repeatedly specified in **FROM**. For instance, data in certain records can be compared using this function.

For example, the **SUPPLIERS** file shows that several suppliers provide the same part. The user might want to know which supplier sets a price that is double, or greater than double, that of another. To transfer the necessary information to a workstation, specify:

Library/File (Member)
        SUPPLIERS, SUPPLIERS
JOIN BY
        T1.PARTNO = T2.PARTNO
SELECT
        T1.PARTNO, T1.SUPPNO, T1.PRICE, T2.SUPPNO, T2.PRICE
WHERE
        T1.PRICE > 2 * T2.PRICE
ORDER BY
        T1.PARTNO

The same file has been specified in **FROM** twice. **JOIN BY** specifies that records having the same part number are joined. This creates a joined record containing information about two suppliers of a single part. The user can spot those records for which the price is double, or greater than double, that of another supplier.

Records in the **SUPPLIERS** file are compared, one by one, with all the records (including itself) in the **SUPPLIERS** file. When the same part number is found, the two corresponding records are linked. This processing is performed for each record in the **SUPPLIERS** file.

For each record, the first supplier's price is compared with the second supplier's price. When the first supplier's price is double, or greater than double, that of the second, only the record containing the first supplier price is kept.

The final result is as follows:

```
    Field: T1.PARTNO   T1.SUPPNO   T1.PRICE   T2.SUPPNO   T2.PRICE
           ---------   ---------   --------   ---------   --------
Record 1:       221          51        .30          54        .10
```

```
       2:      231         51        .10        54        .04
       3:      241         53        .08        54        .02
       4:      241         61        .05        54        .02
```

Specifying Records to Be Included in a Group

You might want to limit which records will be included in a group. To do so, use **WHERE**. For example, the following example transfers the average and lowest prices of each part for those records for which the delivery time (**DELIVTIME**) is less than 30 days.

Library/File (Member)
      SUPPLIERS
GROUP BY
      PARTNO
SELECT
      PARTNO, AVG(PRICE), MIN(PRICE)
WHERE
      DELIVTIME < 30

The result is as follows:

```
       Field: PARTNO   AVG(PRICE)    MIN(PRICE)
              ------   ----------    ----------
   Record 1:   221          .25           .20
          2:   231          .10           .10
          3:   222          .23           .20
          4:   232          .10           .10
          5:   241          .07           .05
          6:   209        18.75         18.00
          7:   285        21.00         21.00
          8:   295         8.50          8.50
          9:   207        29.00         29.00
```

Note that the conditions specified in **WHERE** are checked first, then the records that satisfy those conditions are included in the group.

Specifying Summary Records to Be Transferred

In some cases, you might want to transfer only summary records that satisfy certain conditions. The use of **HAVING** enables the selection of which summary records are to be transferred. **WHERE** is applied to certain records in a group, while **HAVING** is applied only to summary records.

The following example transfers the highest and lowest prices for each part. However, the summary records to be transferred are only those for which the highest price exceeds 10.00.

Library/File (Member)
      SUPPLIERS
GROUP BY
      PARTNO
SELECT
      PARTNO, MAX(PRICE), MIN(PRICE)
HAVING
      MAX(PRICE) > 10.00

The following table shows the result of removing unnecessary summary records by using **HAVING**.

Data Transfer for PC400

```
        Field: PARTNO   MAX(PRICE)   MIN(PRICE)
               ------   ----------   ----------
   Record 1:    209        19.50        18.00
          2:    285        21.00        21.00
          3:    207        29.00        29.00
```

One summary record for an entire file can be transferred. To do this, specify only the summary function in **SELECT** and nothing in **GROUP BY**. As a result, an entire file can be recognized as one group, while one summary record can be transferred for the group.

You can concurrently use the concept of summarizing groups and that of joining records from several files. To obtain the desired results, do as follows:

1. Specify a file in **FROM**, and specify the join conditions to join the records in **JOIN BY**.
2. Specify the conditions in **WHERE** to remove unnecessary records.
3. Specify the fields used for grouping the remaining records in **GROUP BY**.
4. Specify the function in **SELECT**, then create summary records.
5. Specify the conditions in **HAVING** to remove unnecessary records.
6. Specify the items for grouping the final summary records in **ORDER BY**.

Functions Available from Pull-Down Menu

The following section provides a simple explanation of the menu bar of the AS/400→PC Transfer window and PC→AS/400 Transfer window.

File

Transfer request files can be processed.

Create This item creates a transfer request file.

Open This item displays the contents of an existing transfer request file.

Save, Save As
These items save the current settings to the transfer request file being used or to a new transfer request file, respectively.

Exit This item terminates the operation started by selecting the **Data Transfer** icon.

Setup (Only for AS/400→PC Transfer)

User options
Time, date, and numeric value format for receiving can be specified. The following paragraphs describe each item.

Ignore Decimal Data Error
This item specifies whether decimal data errors found in packed or zoned decimal fields upon executing requests are to be ignored. Selecting **Yes** to ignore decimal data errors and using existing indices can considerably reduce the time needed to execute a request. If this item is not specified, the transfer request creates indices again and modifies any detected decimal data errors. This requires extra processing time.

Time Format
This item specifies a desired time format for fields of AS/400 field type having a selected time. If no time format is specified, the default value in

the workstation's national information file is assumed when the transfer request starts, and that in an existing transfer request is assumed when the request is called again.

Supported time formats are as follows:

HMS   Hours, minutes, seconds (hh:mm:ss)
ISO   International Standard Organization (hh.mm.ss)
USA   USA Standard (hh:mm AM or PM)
EUR   IBM European Standard (hh.mm.ss)
JIS   Japanese Industrial Standard (hh:mm:ss)
DDS   AS/400 DDS (Format given by AS/400 file attribute)
DFT   AS/400 default format (Host job default is used)

Time separator

This item specifies enabled delimiters when fields of the AS/400 field type for the selected <u>time</u> are of the format type for which delimiters can be specified.

When no delimiters are specified, the default value in the workstation's national information file is assumed when the transfer request starts, and that in an existing transfer request is assumed when the request is called again.

Supported time delimiters are as follows:

Colon   (:)
Period   (.)
Comma
      (,)
Blank   ( )
Null   (NULL) No Separator
Default value
      (DFT) AS/400 Default Separator

Date Format

Specifies the date format for fields of AS/400 field type for the selected date.

If this date format is not specified, the default value in the workstation's national information file is assumed.

Supported values are as follows:

MDY   Month, day, year (mm/dd/yy)

DMY   Day, month ,year (dd/mm/yy)

YMD   Year, month, day (yy/mm/dd)

JUL   Julian (yy/ddd)

ISO   International Standard Organization (yyyy-mm-dd)

USA   USA Standards (mm/dd/yyyy)

EUR   IBM European Standard (dd.mm.yyyy)

JIS   Japanese Industrial Standard (yyyy-mm-dd)

DDS   AS/400 DDS (Format given by AS/400 file attribute)

DFT   AS/400 default format (Host job default is used)

Date separator

This item specifies delimiters when fields of the AS/400 field type for the selected date are of the format type for which delimiters can be specified.

When no date delimiters are specified, the default value in the workstation's national information file is assumed when the transfer request starts, and that in an existing transfer request is assumed when the request is called again.

Supported date delimiters are as follows:
Slash (/)
Dash (-)
Period (.)
Comma
(,)
Blank ( )
Null (Null) Delimiters are not used.
DFT (DFT) AS/400 default separator

Decimal separator
This item specifies the decimal point character in an AS/400 field whose type is packed decimal or zoned decimal.

When decimal points are not specified, the default value in the workstation's national information file is assumed when the transfer request starts, and that in an existing transfer request is assumed when the request is called again.

Supported decimal point delimiters are as follows:
Period (.)
Comma
(,)
DFT (DFT) - Default decimal separator

Sort Sequence
Specify which sort sequence should be used for this transfer request.
AS/400 job default
Sort by the table identified on the AS/400 system as the job sort table.
Hexadecimal
Sort by the internal hexadecimal representation.
User specified table
Sort by the table identified by the user in a subsequent prompt.
Shared Weight Table
Sort by the shared weight table associated with the language named in a subsequent prompt.
Unique Weight Table
Sort by the shared unique table associated with the language named in a subsequent prompt.

Changing the sort sequence affects the order in which records appear *only* if the **ORDER BY** clause is being used. The sort sequence affects all character comparisons that depend on the order of the alphabet. Such comparisons can occur in the **WHERE** clause, the **GROUP BY** clause, the **HAVING** clause, the **JOIN BY** clause, the **IN** predicate, the **LIKE** predicate, the **BETWEEN** predicate, the **MAX** function, and the **MIN** function. Comparison operations are =, <>, >, >=, and >=.

Sort Sequence Table Name
Type the name of the sort sequence table that you want to use for this transfer request. The format of the table name should be *library/table*. ***LIBL** and ***CURLIB** are allowed for the library name.

Language
>
> AS/400 standard tables provide many languages. Select the user-specified languages to enter the desired language ID. Language IDs shipped with AS/400 are found in the *AS/400 NLS Guide*

Language ID
>
> Enter the language ID for the desired language.
>
> AS/400 standard tables provide many languages. Language IDs shipped with AS/400 are found in the *AS/400 NLS Guide*

Translation Table
>
> Translation tables for ASCII-to-EBCDIC translation or for EBCDIC-to-ASCII translation can be specified, created, and customized. The following paragraphs describe each item.

Current Table
>
> This item specifies whether the IBM default translation or the user-defined translation table is to be used.

Host Code Page
>
> This item specifies the host code page to be used for translation.

Workstation Code Page
>
> This item specifies the workstation code page to be used for translation.

File Name
>
> This item specifies the file name of the user-defined table to be used for translation.
> * To list all files in your workstation, select **Browse**.
> * To customize the translation table, select **Customize**.

File-Description Files

A file-description file is a workstation file that contains all field descriptions of the data in the corresponding workstation data file. Each field descriptor contains the field name, data type, and field length. There is one field descriptor for each field in the workstation file.

A file-description file defines:
* File type of the workstation file to be transferred. For an explanation of each file type, see "Creating a File-Description File" on page 218.
* Field names and order of these fields in each data record.
* Data type of each field in the workstation file.
* Size and number of decimal places of each field.

The workstation files require field definitions when the files are transferred. The field definitions describe the file as it exists on the workstation. 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 workstation 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 might 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 workstation file. Therefore, you usually do not need to worry about the contents or the format of the file-description file. However, if you transfer data that has not been previously transferred to the system, you must create a file-description file.

Creating a File-Description File

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

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 workstation file-description file. It must appear in the first line of the file, starting in column 1. 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 file in which the data is stored. It must appear only once, and must start in column 1, 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
```

Table 29 shows the valid file-type indicators.

Table 29. File-Type Indicators

Indicator	File Type
1	ASCII text
2	DOS random
3	BASIC sequential
4	BASIC random
5	Data interchange format (DIF**)
6	No-conversion file
7	Reserved
8	DOS random type 2
9	BIFF format

PCFO Entry: The PCFO entry is optional. PCFO is a keyword that identifies this record as containing information about the date and time formats, time stamp, and separator characters for applicable formats. It must appear only once and must

start in column 1, 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.

Table 30 shows the valid time formats.

Table 30. 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

Table 31 shows the valid time separators.

Table 31. 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)

Table 32 shows the valid date formats.

Table 32. Date Formats

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

Data Transfer for PC400

Table 33. 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)

Table 34 shows the valid decimal separators.

Table 34. Decimal Separators

Indicator	Separator
1	Period (.)
2	Comma (,)
*	Unspecified (workstation default used)

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 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 1. If you enter more than 256 PCFL records, you receive an error message. You cannot continue a record on 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 things:
- The keyword, PCFL, starting in column 1 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.
- The indicator for the data type. Table 35 shows the indicators that represent the data type of the data in the field. Follow the specified indicator with a space.
- The size of the field (in bytes) as it is stored in the workstation file. The length specification can be from 1 to 4 characters.

Table 35. Data Type Indicators

Indicator	Data Type
1	ASCII[1]
2	ASCII numeric
3	Hexadecimal
4	Binary
5	Zoned

*Table 35. Data Type
Indicators (continued)*

Indicator	Data Type
6	Packed
7	BASIC integer
8	BASIC single-precision floating point
9	BASIC double-precision floating point
10	EBCDIC
11	EBCDIC zoned
12	EBCDIC packed

Note:

1 Includes date, time, and time stamp 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.

Table 36 shows the valid single-byte character set (SBCS) data types for each file.

*Table 36. Valid SBCS Data Types for File
Types*

File Type	Valid Data Type
ASCII text	ASCII ASCII numeric
DOS random	ASCII Binary Hexadecimal ASCII 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
BIFF format	ASCII ASCII numeric

Note: ASCII (SBCS) includes date, time, and time stamp types if converted. EBCDIC includes date, time, and time stamp if not converted.

Data Transfer for PC400

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 data transfer function assumes a maximum length of 65 characters. This length more than covers the largest possible exponential ASCII numeric value. The size specifications for character fields must be the maximum size of any data item in that field.

Table 37 shows the allowed data length limits for each workstation data type. These are the maximum lengths you can specify for size in the PCFL entry.

Table 37. Allowable Data Length Limits for Personal Computer SBCS Data Types

Personal Computer Data Type	Data Length Limit (in Bytes)
ASCII	4093
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	4093
Hexadecimal	2048
Packed decimal (ASCII and EBCDIC)	16
Zoned decimal (ASCII and EBCDIC)	31
Time	8
HMS (see note 1)	8
USA	8
ISO, EUR, and JIS (see note 1)	8 or 10 (see note 2)
DDS, DFT	
Date	8
MDY, DMY, YMD	6 (only allowed length)
Julian	
ISO, EUR, JIS, USA (see note 1)	10
DDS, DFT	6, 8, or 10 (see note 2)
Time stamp	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 The length is determined by the format defined in the host file for DDS, or from the AS/400 job default (DFT keyword).

Table 38 shows the allowed data length limits for each AS/400 data type.

Table 38. 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)
Time stamp	26

Notes:

1   The data length limits for the workstation and the system data fields are different in some cases. For these cases, the transfer function attempts to fit the workstation data into the system field. If the data does not fit into the field, a message is displayed. Refer to "Data Conversions" on page 224 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, 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 types:
- ASCII numeric
- Binary
- Packed
- Zoned

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 data transfer function might round the number to fit it into the field. Refer to "Data Conversions" on page 224 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 data transfer function (RTOPC) do not contain a comment field.
- Precede the comment with an asterisk (*) as the first nonspace character in the line.
- 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 data transfer function needs data conversions for transferring data from the system to the workstation, and vice versa. For both types of transfers, the necessary conversion depends on the record size, the type of data being transferred, the type of workstation file being used, the system data type, and, in some cases, the data length.

Record Size

Each transferred record contains data indicating whether 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 data transfer function supports the following system data types:
- Binary data
- Character data
- Date
- Hexadecimal data
- Packed decimal data
- Time
- Time stamp
- Zoned decimal data

The data transfer function supports the following workstation data types:
- ASCII numeric data

- BASIC numeric data, including:
  - Double-precision data
  - Integer data
  - Single-precision data
- Binary data
- Character data, including:
  - ASCII
  - EBCDIC
- Hexadecimal data
- Packed decimal data
- Zoned decimal data

ASCII Numeric Data

The data transfer function defines ASCII numeric data to represent any numeric value stored in ASCII format. This is not a valid AS/400 system data type. The number −123.45 in ASCII format is:

2D 31 32 33 2E 34 35

The decimal point and sign are stored explicitly for ASCII numeric 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.

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 $−1.0 \times 10^3$ in ASCII numeric format) is:

2D 31 2E 30 45 2B 30 33

For example, the number 9.5D−15 (representing 9.5×10^{-15} in ASCII numeric format) is:

39 2E 35 44 2D 31 35

Binary Data

This data represents signed or unsigned numbers in twos complement form. Binary numbers of 1, 2, 3, or 4 bytes in length are allowed on the workstation, but the AS/400 system allows only numbers 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, whereas the workstation 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 3BF5 is equivalent to the decimal number 15349, and the binary number 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 workstation ASCII characters and AS/400 EBCDIC characters match exactly
- Values where a substitute character is chosen for a character that cannot be translated

The data 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) data includes date, time, and time stamp types if converted. EBCDIC data includes date, time, and time stamp if not converted.

Double-Precision Data

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

Date, Time, and Time-Stamp Data Types

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

Dates, times, and time stamps 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 hex 3D with the following string of bits:

0011 1101

Integer Data

Integer data is defined only for the workstation. BASIC applications use integer data. Integer data is stored in 2 bytes and represents a whole number from −32768 to 32767.

Packed Decimal Data

For both the AS/400 system and the workstation, 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 hex B or hex 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 workstation uses hex 3 to indicate a positive number or hex 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 workstation 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 workstation. The AS/400 system does not support this data type. BASIC applications use single-precision data. This data type is a positive or negative number from 2.938736×10^{-39} to 1.701412×10^{38}. 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 workstation as 37, which is the ASCII 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 workstation 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, a hex B or hex 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, as do character fields, except that the sign half-byte in the workstation field is changed to a hex 3 to indicate a positive number or a hex B to indicate a negative number.

Data Transfer for PC400

For DOS random type-2 files, zoned decimal fields from the system change from EBCDIC to ASCII, as do character fields, except that the sign half-byte in the workstation field is changed to a hex 3 to indicate a positive number or a hex 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 workstation file types are supported:
- ASCII text files
- BASIC random files
- BASIC sequential files
- DIF files
- BIFF files
- DOS random files
- DOS random type-2 files
- No-conversion files

ASCII Text Files

ASCII text files are normally used with programs that work with text (such as editors and print routines). The characteristics of an ASCII text file are as follows:

- Records consist of ASCII characters.
- A carriage return character (hex 0D) and a line feed character (hex 0A) delimit each record from the next.
- Workstation records in an ASCII file can be variable in length due to truncation of trailing blanks at the end of an AS/400 record.

Transferring Data to ASCII Text Files: When you create an ASCII text file, the data coming from the AS/400 system changes as follows:

- Hexadecimal fields change to equivalent ASCII characters for each half-byte. For example, X'D3' expands to ASCII 4433 and is written 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 characters as defined by the translation tables.
- Date, time, and time-stamp data 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 time-stamp data) or zeros (for binary, zoned, and packed,) are added to the maximum length of the field.

    Note: Some nondisplayable EBCDIC characters are translated into ASCII control characters on the workstation. If EBCDIC character fields contain nondisplayable data, you might get unexpected results and your ASCII text file might appear to be corrupted.

    For example, X'05' in an EBCDIC field is translated to an ASCII X'09', which is an ASCII control character for horizontal tab. Most workstation text editors process this tab character so that the data in your workstation text file appears to be shifted to the right when viewed.

    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 numeric. For example, X'FFD3' with no decimal position expands to ASCII 20202020202020202D3435. When displayed by an editor or printed, the string appears as −45.

  Note: The length of the ASCII field depends on the length of the binary field.

  A binary field on the AS/400 system is either 2 or 4 bytes long. The resulting ASCII field length is from 6 to 11 bytes, including the sign. Another byte is added for a decimal point.

  Table 39 shows the mapping between binary field lengths and their ASCII lengths.

Table 39. Binary-to-ASCII Field Length Mapping

Binary Length	ASCII Length	Value Range
2	6	−32768 to 32767
4	11	−2147483648 to 2147483647

- Zoned decimal fields are changed to ASCII numeric. For example, EBCDIC F0F0F9F5F2D6 with a field length that indicates two digits to the right of the decimal point expands to ASCII 20202D39352E3236. When displayed by an editor or printed, the string appears as −95.26. The resulting workstation field length is equal to the length of the system field plus 1 for the sign and 1 for the decimal point, if specified.
- Packed decimal fields change to ASCII numeric. For example, X'871D' (no decimal point) changes to ASCII 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 workstation field is equal to two times the length of the AS/400 field, plus 1 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 Text Files: When you transfer data from ASCII text files to system files, the data changes as follows:

- ASCII character data changes to EBCDIC character, date, time, or time-stamp data (based on the AS/400 field type) on a byte-to-byte basis, or to hexadecimal data by changing 2 ASCII bytes into 1 hexadecimal byte.
- ASCII 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 workstation are different because of the explicit way minus signs and decimal points are stored in ASCII numeric fields. Each field changes individually, to ensure that the resulting field length matches the specifications for that field. The data transfer function tries to fit the workstation data into the system field.
- For null-capable AS/400 fields, null values (except date, time, and time stamp) 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 Text Files: When you transfer data from a workstation ASCII text file to an AS/400 file, the following errors can occur:

- A data field in the ASCII 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 description file defines the character data as longer than the field length specified for the system file.

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 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 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 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 numeric data is found that does not match the format the file description specified. An incorrectly positioned decimal point 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 is not transferred. Data found at the end of this record and not defined by the system data definitions or workstation file-description file is not transferred to the system file, because no definitions exist to define the data and how it should change.

When you transfer data from an ASCII 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 is not transferred.

BASIC Random Files

BASIC random files are the most general-purpose BASIC file type. They contain fixed-length records with:
- 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 time-stamp 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 time-stamp data) or zeros (for binary, zoned, and packed data) are added to the maximum length of the field.
- 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 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 you transfer 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 workstation should be the same as the field lengths as stored on the system.
- ASCII character, date, time, and time-stamp data changes to EBCDIC character data byte by byte.

- For null-capable AS/400 fields, null values (except date, time, and time stamp) 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 you transfer data from a workstation 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 workstation 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 workstation file-description file defines character or hexadecimal data as longer than the field length specified on the system.

  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 workstation 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, are filled with default values and are transferred 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 you transfer 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 workstation file.

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 double quotation marks (X'22') that surround them.

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

- Fields are delimited by ASCII 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 in 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 characters for each half-byte. Double quotation marks surround them.

  For example, X'F3' expands to ASCII 22443322 and is written to the file.

- EBCDIC character, date, time, and time-stamp fields change byte by byte and are mapped into ASCII characters as defined by the translation tables. ASCII 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 is converted to the AS/400 variable-length format.

- 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.

  For example, X'FFD3' appears as ASCII 2D3435. When displayed on an ASCII device, the string appears as –45.

- 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 expands to ASCII 2D39352E3236. The string appears as –95.26 when an editor displays it or it prints.

- 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'871F' (no decimal point) changes to ASCII 383731. The string appears as 871 when an editor displays it or it prints.

Data Transfer for PC400

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

- ASCII character, date, time, and time-stamp data changes to EBCDIC character data on a byte by byte basis and to hexadecimal by changing 2 ASCII bytes into 1 hexadecimal byte.

- ASCII 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 workstation data might be different because the minus signs and decimal points are stored in ASCII numeric fields, and leading and trailing spaces are stripped away.

  BASIC might create exponential numbers in these files. The data 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 data transfer function tries to fit the workstation data into the system field.

Errors When Transferring Data from BASIC Sequential Files: When you transfer data from a 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 file-description file defines character data as longer than the field length specified for the file.

  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 workstation 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 workstation field is greater than the number of decimal positions specified on the system. The extra bytes are significant, because the data is rounded up if the first extraneous digit is 5 or greater, and is rounded down if it is less than 5.

- Data in this field is incorrect or does not match the workstation data type. This error occurs when a field defined as numeric by the file description 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, are filled with default values and transferred 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 the workstation field size. The data is lost. This error occurs when extra data, not defined by the file-description file, is found at the end of a character field. The extra bytes are truncated and are not transferred to the system file.

- Extra data found at the end of the record. The extra data is not transferred. 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 workstation file-description file. This extra data is not transferred 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).

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

The data 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 data transfer function consists of a decimal number that can contain a minus sign or a decimal point or both. The data 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 time-stamp data changes byte by byte and is mapped into ASCII characters as defined by the translation tables. 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 expands 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 is transferred from a DIF file to the AS/400 system, one of the following things can occur, depending on the type of data in the file:

- If the system field is a character (EBCDIC) field, it is filled with untranslatable characters (hexadecimal zeros) and is transferred 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 was not transferred to the system.

When you transfer 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 you transfer 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 is processed 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, ensure that the file is 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 you transfer the data to the AS/400 system, the file must be saved in the same format as originally created by the data transfer function.

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

- ASCII character, date, time, and time-stamp data is changed to EBCDIC character data or to hexadecimal data. ASCII-to-EBCDIC conversion is done byte by byte. ASCII-to-hexadecimal conversion 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 workstation can be different, because of the explicit way minus signs and decimal points are stored 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 data transfer function tries to fit the workstation data into the system field.
- In null fields, a NULL DIF character field results 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 is converted to the AS/400 variable-length format.

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

- Data in this workstation file is not valid, or the version of this workstation file is not supported. The DIF file does not follow the standard DIF format. Processing ends, and no more records are transferred.
- Data in this field is too long for the AS/400 field. The data is truncated. The workstation 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 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 workstation 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 workstation field is greater than the number of decimal positions specified on the system. The data is rounded up if the first extraneous digit is 5 or greater, and is rounded down if it is less than 5.

- Data in this field is incorrect or does not match the workstation data type. One of the following things 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 field size. The data is lost. This error occurs when extra data, not defined by the file-description file, is found at the end of a character field. The extra bytes are truncated and are not transferred to the system file.

- Extra data is found at the end of this record. The extra data is not transferred. This error occurs when there is extra data at the end of the record, and the AS/400 data definitions or file-description file have not defined it. This extra data is not transferred to the system, because no definitions exist to define the data and how it should change.

BIFF Files

In a BIFF file, data is expressed in lines and columns. A BIFF file contains character and numeric data (positive and negative decimal values).

The BIFF file format is used in EXCEL. The supported BIFF version is 4 (EXCEL 4.0).

Data Transfer for PC400

The transfer facility supports only the following two data types for a BIFF file:
- Character data
- Numeric data

Transferring Data to BIFF Files: When a BIFF file is created, the system data is converted to equivalent EXCEL cell data.

If untranslatable data is found, the entire field is treated as an error cell.

Transferring Data from BIFF Files: If an error cell is found during data transfer from a BIFF file to the AS/400 system, either of the following things can occur depending on the data type of the file:
- If the system field is a character (EBCDIC) field, the error cell containing untranslatable characters (hexadecimal zeros) is transferred to the system. A message indicating how many bytes of untranslatable data were transferred is displayed.
- If the system field is a hexadecimal, zoned decimal, packed decimal, or binary field, an error message indicating that the data in this cell is not valid and thus has not been transferred to the system is displayed.

When data is transferred from a system file to a BIFF file, the first record contains field names, which can be treated as column headers.

To return a BIFF file to the system, the first line must contain these field names (as defined in the system) or data. If the first line does not contain field names, the file is regarded as containing data only.

When a file is sent to the AS/400 system, cell information (such as the character size and font information) is ignored. This means that cell information is lost, even if the contents of a BIFF file that have been sent to the AS/400 system are retransmitted to a workstation.

When data is transferred from a BIFF file to an AS/400 file, the data is converted as follows:
- ASCII character cell data is converted to EBCDIC character data or hexadecimal data; 1-byte ASCII data is converted to 1-byte EBCDIC data.
- ASCII numeric cell data is converted to a binary number, or a zoned or packed decimal number, depending on the data type specified in the system.

When data is transferred from a BIFF file to the AS/400 system, the following specific processing is performed:
- When data is transferred to a BIFF file, the first record contains the names of the fields to be transferred, which can be treated as column headers. To return a BIFF file to the AS/400 system, the first line must contain the same field names (as defined in the AS/400 system) or data. If the first line or the first set does not contain a character field that exactly matches the AS/400 field, the file is treated as being a file with no column headers, and only data is processed.
- When a BIFF file is transferred to the AS/400 system, header information is not used.
- To ensure correct transfer of a BIFF file to the AS/400 system, the file format must be valid (lines and columns). Data for each set or line must correspond to one record in the AS/400 file.

Errors When Transferring Data from BIFF Files: When data is transferred from a BIFF file on a workstation to the system file with the data definition, the following errors can occur:

- Data in this workstation file is not valid, or the version of this workstation file is not supported. The BIFF file does not conform to the standard BIFF format. Processing terminates, and no more records are transferred.

- Data in this field is too long for the corresponding AS/400 field. The data is truncated. A file-description file defines character or numeric data that is longer than the field specified in the system file.

  - For conversion from ASCII to EBCDIC, this error occurs if a file-description file defines ASCII data that is longer than the field specified in AS/400 system.

    During conversion from ASCII to hexadecimal, this error will occur if a file-description file defines ASCII data that is twice as long as the field specified in AS/400 system. This is because 2-byte ASCII data is converted to one hexadecimal character.

  - A truncation error only occurs if excess bytes are other than blanks (X'20') during conversion from ASCII to EBCDIC, or other than zeros (X'30') during conversion from ASCII to hexadecimal. Truncating these excess bytes enables data to fit into the specified AS/400 fields.

- Numeric data is too long to fit into the corresponding AS/400 field. The maximum value is assumed. This error occurs under either of the following conditions:

  - Numeric data in a workstation field is too long to fit into the number of bytes specified for the AS/400 field.

  - The number of decimal digits in a numeric field exceeds the number of digits specified for the AS/400 field.

  The field value is set to the maximum value that can be specified for the number of bytes, and that for the number of digits, specified for the AS/400 system.

- Data in this field contains too many decimal places. The data is rounded off. The number of decimal places in a workstation field is greater than the number of decimal places specified for the system. If the first excess digit is 5 or more, the data is rounded up. Otherwise, it is rounded down.

- Data in this field is not correct, or its type does not match the type of workstation data. One of the following things has occurred:

  - A numeric field contains other than numeric data.

  - A character field or a hexadecimal field contains a number, or a numeric (zoned or packed decimal, or binary) field contains characters.

  - A hexadecimal field or a numeric (zoned or packed decimal, or binary) field for the AS/400 contains a BIFF error cell.

  If this error occurs, the transfer request terminates to avoid transferring incorrect data to the system file.

- Data for this field is missing. This error occurs if the data field is defined, but the file does not contain any data. This means that the end of the record is reached before all defined data has been found.

  If this error occurs (that is, if data is defined for one or more fields, but it is not found there) the fields containing the default value are transferred to the system file. The default value is EBCDIC spaces for a character field and zeros for a numeric field.

To specify a user-specific default value, use the default value (DFT) keyword in DDS for the file.

- Data in this field exceeds the size of a workstation field. Data is lost. This error occurs if excess data, not defined in the workstation file-description file, is found at the end of the field. For character data, excess bytes are truncated, and not transferred to the system file. For numeric data, the entire field is converted to zeros and transferred to the system file.

- Excess data is found at the end of this record. The excess data is not transferred. This error occurs if such excess data is not defined in the AS/400 data definition or in the workstation file-description file. This excess data is not transferred to the system, because the data and the conversion method are not defined.

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 workstation are represented as two-complement numbers, so it is unnecessary to change individual bytes. The workstation uses the convention of storing numeric values with the least significant byte in the left-hand byte position. The data transfer function then reverses the order of the bytes in the 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 time-stamp 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 length, and trailing blanks (for character, hex, date, time, and time stamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- Hexadecimal fields do not change.
- Packed decimal fields do not change except for the last half-byte, which contains the sign. The workstation 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, as do character fields, except that the sign half-byte in the workstation 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 you transfer data from DOS random files to AS/400 files, the data changes as follows:

- ASCII character, date, time, and time-stamp data changes to EBCDIC character data on a byte by byte basis.
- Binary fields in the workstation 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 workstation.
- 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 workstation change from ASCII to EBCDIC (ASCII to EBCDIC for DBCS), as do character fields. The last half-byte (the half-byte that contains the sign) in the workstation 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 time stamp) 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 you transfer 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 workstation file contains character or hexadecimal data shorter than the specified field length. It also occurs if the length of the workstation 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 workstation file-description file defines character or hexadecimal 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. 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 workstation 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 workstation field is greater than the number of decimal positions specified on the system. The 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 workstation 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 you transfer 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 workstation 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 workstation 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 workstation 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 workstation applications.

Transferring Data to DOS Random Type-2 Files: When you create DOS random type-2 file data definitions, system data changes as follows:

- Binary fields on the AS/400 system and the workstation are represented as twos complement numbers, so it is unnecessary to change individual bytes. The workstation uses the convention of storing numeric values with the least significant byte in the left-hand byte position. The data 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 time-stamp 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 length, and trailing blanks (for character, hex, date, time, and time stamp) or zeros (for binary, zoned, and packed) are added to the maximum length of the field.
- Hexadecimal fields do not change.
- Packed decimal fields do not change. The sign convention used on the workstation 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 workstation.

  For example, EBCDIC X'F0F1F2F5F2D6' appears as ASCII X'303132353276'.

Transferring Data from DOS Random Type-2 Files: When you transfer data from DOS random type-2 files to AS/400 files, the data changes as follows:

- ASCII character data, date, time, and time stamp data change to EBCDIC character data on a byte by byte basis.
- Binary fields in the workstation 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 workstation.
- 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 workstation change from ASCII to EBCDIC, as do character fields. However, the sign half-byte is changed to an 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 time stamp) 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 you transfer 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 workstation file contains character or hexadecimal data shorter than the specified field length. It also occurs if the length of the workstation 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 workstation file-description file defines character or hexadecimal 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. 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 workstation 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 workstation 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 workstation 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 you transfer 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 workstation file.

No-Conversion Files

No-conversion files, defined by the data transfer function, consist of data that has not changed. For example, when data transfers from the system to a workstation no-conversion file, the data transfers exactly as it is stored on the AS/400 system. Date, time, and time-stamp data transfers to EBCDIC character data on the workstation.

Transferring Data to No-Conversion Files: When you transfer 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 time-stamp data is converted to EBCDIC character data.

Variable-length and null fields are converted to fixed length, and trailing EBCDIC blanks (for character, hex, date, time, and time stamp) 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 data transfer function performs no data change or translation. Date, time, and time-stamp data transfers to EBCDIC character data on the workstation.

However, the data 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.

Errors When Transferring Data from No-Conversion Files: When you transfer data from a workstation no-conversion file to a system file, the following errors can occur:

- Data sizes are not equal. When you transfer no-conversion files, the length and decimal position specifications for the system and the workstation 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 workstation 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 workstation 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 workstation 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.

AS/400 System-to-Personal Computer Performance Considerations

Transferring data from the AS/400 system to the workstation depends on the following performance considerations:

- The system workload.
- 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.

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-workstation data 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, the AS/400-to-workstation data transfer function uses the existing access paths whenever possible to improve performance.

Data Transfer for PC400

For the AS/400-to-workstation data 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 things are true of the WHERE field:
  - It is compared with 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 if 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 when:
  - 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.

Chapter 15. File Transfer for PC400

File transfer is designed so that you can use it in the following cases:

- To store a workstation file on the AS/400 system for a backup
- To edit a source file of an AS/400 program with a workstation editor, and send the file edited on the workstation to the AS/400 system.
- To distribute workstation documents and programs to the AS/400 users

PC File Transfer with the CRLF Option

If the CRLF option is specified, the transfer program checks for new-line characters. If the record length is reached before a new-line character is found, the record is divided at this point; one sentence of a workstation file will become two or more records. Particularly, specify a sufficiently long record length when retransmitting a workstation file containing 2-byte characters.

By default, the message `records segmented.` is not displayed. To display the message:

Look for the profile for the session you will use. Normally, this will be in the **\Personal Communications\Private** directory under the name *filename*.**WS** (*filename* is a user-specified file name).

Use an editor to insert the following sentence into the [Transfer] section. If there is no [Transfer] section, first enter [Transfer]. Be careful to enter it correctly.

```
DisplayTruncateMessage=Y

or

[Transfer]
DisplayTruncateMessage=Y
```

The next time the session is started, this specification becomes active.

Transfer to a Physical Source File

An AS/400 physical source file contains 12 bytes of information for each record as internal information: 6 bytes are for a record number, the other 6 bytes are for a date. When you transfer a file from a workstation using file transfer, the date field contains 000000. If the APPEND option is not specified, the record number is incremented by 1, up to a maximum of 9999. Otherwise, it is incremented from the nearest integer, greater than the number of the last record in the original file (for example, 24 for 23.1). If the number of records exceeds 9999, the next and all subsequent record numbers are 9999.

Use the source specifications input utility (SIU) to renumber records when saving the file after editing.

Transfer to a Physical File

A file, such as a PC program, that does not require the processing of the contents of an AS/400 file or the reading of data, should be transferred to a physical file with the BINARY transfer type. Because data is not converted, if the data is subsequently retransmitted from the AS/400 system to a workstation, the original

workstation file can be re-created exactly. If the data is converted, however, data might not be restored to its original form, depending on the contents of the conversion table.

For the maximum number of members (MAXMBRS), a physical file attribute, the default value is 1. When a physical file is created during file transfer, MAXMBRS is 1.

When a file is transferred from a workstation to a physical file, the default file name **xxxBIN** is assumed (**xxx** is a workstation file extension.) If you transfer more than one file, an error occurs when the second and subsequent files are transferred: `The TRANS58 file or member cannot be created.` File transfer terminates. A file should be created with the expected file attribute before it is transferred from a workstation to the AS/400 system.

Use of the DSPMBRLST Command

For file transfer from the AS/400 system to a workstation, the Paste function can be used. If the name of the `Library/File(Member)` to be transferred is copied with the Copy function of the Edit menu, it can be displayed as the host file candidate to be transferred on the transfer request screen by clicking **Paste**. This is particularly convenient when transferring more than one file at a time.

Use the DSPMBRLST command to list AS/400 files or members. The command format is as follows:

`DSPMBRLST LIB(lib-name) FILE(file-name)`

LIB parameter

The LIB parameter contains the target library name. The default value is ***USRLIBL**. Extensive specification, such as ***ALL**, ***** for generic name, is possible, but is time consuming. AS/400 files or members are listed more efficiently if a specific name is specified.

FILE parameter

The FILE parameter contains the target file name. There is no default value. The parameter must be specified. ***ALL** and ***** for generic name can be specified.

Executing this command lists `Library/File(Member)` on the screen. If they cannot be listed on one screen, **MORE...** is displayed in the lower right corner of the screen. Use the next page or the preceding page key to scroll the screen. Create a list for Paste with the Copy or the CopyAppend function of the Edit menu, as required.

Restrictions for Transferred File Size

A file that is more than 1,040,000 bytes cannot be transferred correctly.

DBCS File-Transfer Option

For Japanese DBCS sessions, specifying the JISCII file-transfer option does the following when sending a file:
- Converts 1-byte workstation codes to EBCDIC codes
- Converts 2-byte codes to IBM kanji codes
- Inserts SO (hex 0E) and SI (hex 0F) before and after a kanji field
- Converts RS (hex 1E) and US (hex 1F) to SO (hex 0E) and SI (hex 0F)

Specifying the JISCII option when receiving a file does the following:

- Converts EBCDIC codes to 1-byte workstation codes
- Converts IBM kanji codes to 2-byte codes
- Removes SO (hex 0E) and SI (hex 0F) before and after a kanji field

For other DBCS sessions, such as Korean, Chinese, or Taiwanese, specify the ASCII file-transfer option.

Chapter 16. Considerations for the Use of PC400 Sessions

This chapter describes considerations and restrictions for PC400 functions. First-time users of PC400 or users who want only to perform simple operations using the PC400-supplied default values need not read this chapter.

Refer to this chapter:
- If a problem occurs during operation
- When you want to obtain more detailed information about the system
- When you want to know the restrictions imposed on the system's use

Considerations and restrictions on the use of PC400 and supplementary information other than the items described in this chapter may be included in README.TXT in the Personal Communications directory. Double-click the **Readme - Please!** icon to open this file.

Scroll Bar

When you select **Font** from the Appearance menu in the host session window and select **Fixed Size** from the Select Display Font window, the entire operator information area might not appear on the screen; the session-window size is restricted to be smaller than the screen size. If you specify **With Scroll Bar**, the OIA will not scroll.

Print Processing

Following are some additional considerations when printing with PC400.

Printing Bar Codes

This function requires OS/400 Version 4.2.

CPI/LPI of Device Fonts

If the printer driver cannot print with device fonts associated with the user-specified CPI/LPI, the print output can be generated with incorrect CPI/LPI values.

PCSERR999 Error Messages

Message PCSERR999 - Personal Communications internal error:*module-name* - *xxxx* might appear if there is insufficient memory. If any print jobs are queued in the print manager, delete those print jobs.

Disconnect in Testrequest to AS/400 on Telnet 5250

Executing a Testrequest function when connected to an AS/400 may cause the session to be disconnected. If you experience this problem then please make sure that OS/400 APAR MA15053 has been applied on the AS/400.

AS/400 Host Print Problem

If you attempt to use the host print function (mapped to **CTRL-Pause** by default) while viewing a spooled print file, you may notice that the ends of some of the lines are wrapped incorrectly in the second generated spool file. This problem

occurs with both 24X80 and 27X132 display modes. This problem has been fixed by a PTF on OS/400. The APAR number is SA57195 and is available on PTF MF13596 for OS/400 V3R1.

Printable Area

Depending on the printer driver used, it might not be possible to use the entire surface of the paper for printing.

If the printing position is beyond the printable area, the page is automatically changed. When using a printer driver that allows you to set the margins, specify the minimum margins, thus maximizing the printable area.

PDT Mode

Printing using a PDT file is restricted as follows:
- Only the fonts specific to the printer being used are supported.
- Graphics are printed via the Windows printer driver selected in **Printer Setting**, regardless of the PDT mode.
- Postscript printers are not supported. There are no PDF files for Postscript printers.

Setting the Code Page

The host code page, which is set in the Configuration panel, is used as the default. The Set Initial Condition (SIC) command is used to set the host code page.

You can change the code page by using Set GCGID Through GCID (SCG) command or Set CGCS Through Local ID (SCGL) command. The same code pages for the display session are available.

Chapter 17. Displaying Grid Lines (DBCS Only)

Grid lines can be displayed using the screen format data. PC400 supports the following two methods of displaying grid lines:

1. Building the data of the grid-line screen using the grid-line control code by a user.

2. Utilizing the screen grid-line support provided by DDS. This is available on OS/400 V3R1, or later.

   The grid lines defined by using this support can be displayed with the grid lines defined by using the first method.

   However, with this method, only one color of the grid lines is available. DDS uses the color specified for a divider line by the color-mapping function. Nothing to the right of the 80th column not be displayed. As for line types, only a solid line is available. The grid lines defined by using this support can be printed on host printers.

This chapter describes how to build the data of the grid-line screen (method 1).

For information about the screen grid lines provided by DDS (method 2), please refer to the AS/400 publications.

Limits for Displaying the Grid Lines

The top of each line and the left of each row on the screen are the limits for displaying the grid lines on the screen.

The host system always sends the grid-line designation data, which specifies any one of the four types of the grid lines (a), (b), (c), (d) (no grid line), for the 1920 (24 lines by 80 columns) positions respectively.

As a result, a grid line, such as (e) is displayed by gathering these vertical lines and horizontal lines.
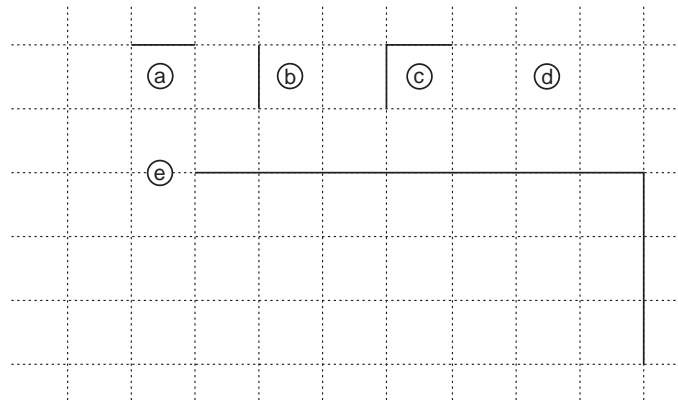
Grid Lines

The limits for displaying the grid lines are as follows:
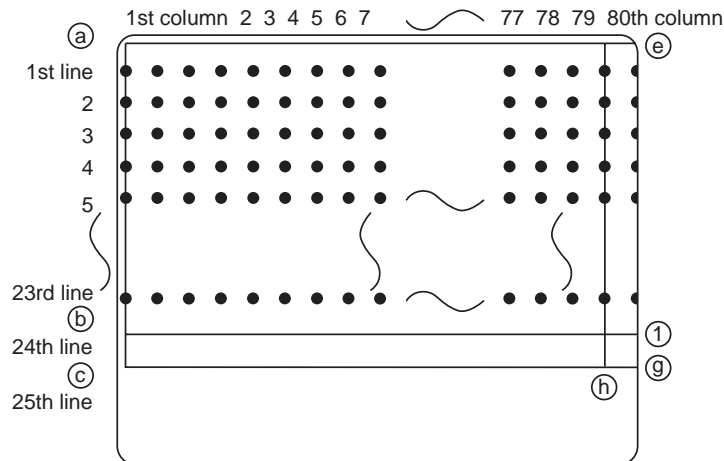
1. A horizontal grid line can be drawn between any character-displaying position within the limits of (a), (b), (f), (e).

   The horizontal line above the 25th row, (c)–(g), is always displayed by the system.

2. A vertical grid line can be drawn between any character-displaying position within the limits of (a), (c), (h), (d).

   The vertical grid line (e)–(g) cannot be drawn on the right.

3. The grid-line screen format cannot be updated partially.

4. Note that the grid lines are removed when displaying the characters from the first column to the fourth column.

Controlling the Grid-Line Format

Up to two sets of grid-line screen-format data can be stored within the PC400.

To store the data of the grid-line screen format, PC400 provides the grid-line information buffers "F" and "B". When PC400 receives data which contains the identification "F" or "B" from a host system, PC400 displays the data of the grid-line screen format and the character data concurrently. The following figure shows the relation between the designated grid-line information buffer and the

character data to be displayed.
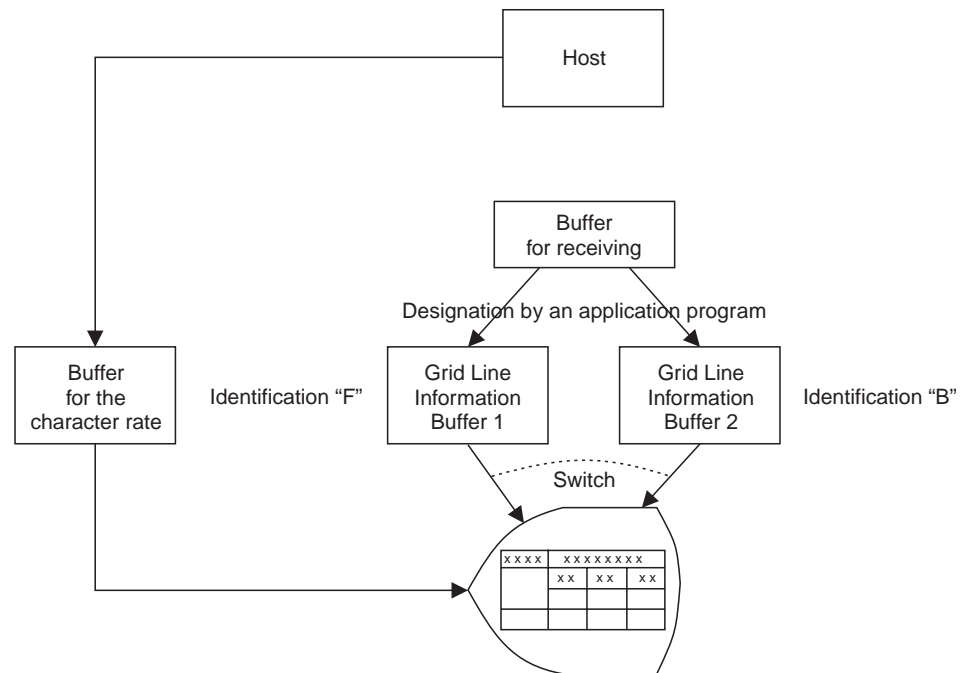
Displaying the Grid-Line Screen Format

To store the grid-line screen format into the grid-line information buffer, the grid-line draw control code is used. The application program on the host system is required to specify the following information about the grid-line information buffers:

- Which buffer is used to store the grid-line screen definition data
- Which buffer is used to display the grid-line screen definition data

To specify which buffer to be used, the character data "F" or "B" with the non-display attribute should be set at the designated position in the grid-line screen-format data and the character data.

The grid-line screen format is not displayed when its format gets stored into the buffer. It is displayed when the character data that contains the "F" or "B" with the non-display attribute is received at the specified position.

Refer to "Grid-Line Screen-Format Displaying Mechanism" on page 256.

Deleting the Grid-Line Screen Format

The contents in the grid-line information buffer are deleted by using the deletion control code. By using this control code, you can prevent grid lines from being displayed unexpectedly.

Refer to "Grid-Line Screen-Format Displaying Mechanism" on page 256.

Grid-Line Screen-Format Displaying Mechanism

This section describes the control for the grid-line displaying function and how to specify the grid line and the character data.

There are three ways to control the grid-line screen format:

- Grid-line draw control
- Deletion control in the grid-line information buffer
- A combination of grid-line screen format and a user-defined screen format

Note: The control codes should be defined to be set in a different field from the character data. If these control codes are set in the same field as the character data, the grid line will not be displayed correctly or unexpected data will be displayed.

Hexadecimal numbers are used for the control codes and the data in the following description, unless otherwise specified.

Grid-Line Draw Control

Grid-line draw control is used to store the grid-line screen format in the PC400. The format of the control data is as follows:
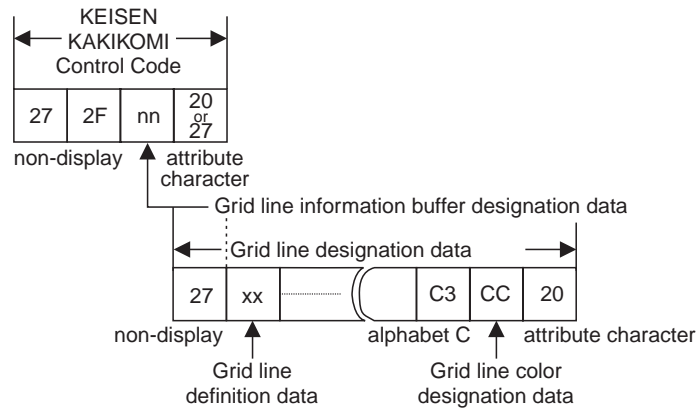
When sending this control data from an application program on the host system, this control data is sent as normal display data. The starting position for displaying this data should be specified as follows:

1. Grid-line draw control code: first line, first column
2. Grid-line designation data: first line, fourth column

The last byte of the control code should be the attribute X'20' or X'27', which indicates the end of the field. This attribute character should be sent from the host so that it can be overridden with the non-display attribute X'27', which is the first byte of the grid-line designation data. Items 1 and 2 should be defined as different fields. When it receives the grid-line draw control code and the grid-line designation data, PC400 stores the grid-line designation data in the specified grid-line information buffer.

However, the grid-line screen format is displayed when receiving character data that contains the information to display the grid-line screen format.

Grid-Line Draw Control Code

Function

When receiving a grid-line draw control code from the host system specifying the first row and first column, PC400 assumes the following data are for writing the grid-line screen format.

27	2F	nn	20 or 27

non-display   attribute

Parameter	Meaning	Valid Value
nn	Grid line information buffer designation data	• X'C6' ("F") • Grid-line definition data to be stored into the grid-line information buffer 1. • X'C2' ("B") • Grid-line definition data to be stored into grid-line information buffer 2.

Note: When PC400 receives the grid-line draw control code, the current contents of the specified grid-line information buffer are deleted to prepare the buffer to store other grid-line definition data.

Grid Line Designation Data

Function

After receiving the grid-line draw control code, the data starting with the first line and fourth column is assumed to be the grid line designation data.

It specifies on which character positions (24 rows by 80 columns) the vertical and horizontal grid lines should be displayed.

27	xx	· ·	· ·	· ·	· ·	xx	C3	CC	20

non-display ◄――――― 640 bytes ―――――► alphabet C   attribute character
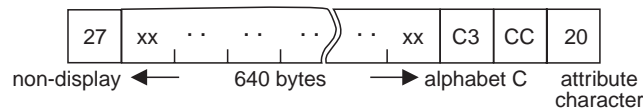
It defines the grid-line information for each screen position from the first line, first column to the 24th line, 80th column, respectively. It defines the grid lines for three columns using a byte, so the length of this data is fixed-length 640 bytes as follows:

```
24 rows * 80 columns / 3 = 640
```

Grid Lines

The following illustration shows the relation between the actual grid lines on the display screen and the grid-line definition data.

Grid line on the workstation screen
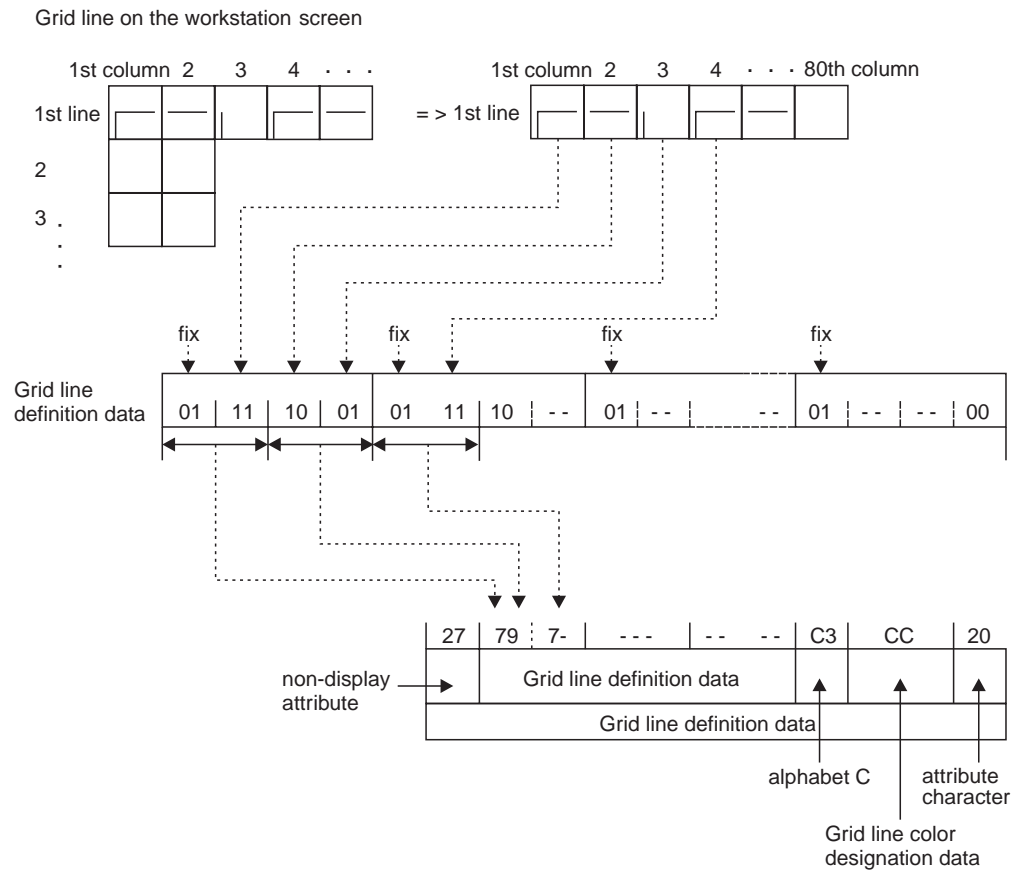
The error process for the grid-line designation data is as follows:

- If the first byte of the grid-line designation data is not X'27', the grid-line definition data is ignored and the contents of the grid-line information buffer are deleted.

- Incorrect designation of the grid-line definition data:

  If there is a code other than X'40' to X'7F' in the grid-line definition data with a length of 640 bytes, it is assumed to be an incorrect code and the grid-line definition data after this code is ignored. The correct portion of the grid-line definition data is displayed as the grid line on the screen normally.

Control Code for Deletion of the Contents of the Grid Line

Function

The data is assumed to be the control code for deletion of the contents of the grid line information buffer when receiving the following data starting with the first line and first column.

It is used to delete the contents in the grid-line information buffer specified by the parameter.

29	2F	nn	20

non-display   attribute

reverse image

Parameter	Meaning
xx ... xx	Grid-line definition data
CC	Reserved

Combination Display of Grid-Line Screen Format and User Screen Format

To display the grid-line screen format on the screen, an application program on the host system should send the following character screen format to PC400. Character data longer than 1 byte should be sent with it so that the grid line can be displayed.

Function

Displays the grid-line screen format with the user-defined screen format.

Parameter	Meaning	Valid Value
nn	Grid-line information buffer designation data	X'C6' ("F") Grid-line definition data in the grid-line information buffer 1 to be displayed X'C2' ("B") Grid-line definition data in grid-line information buffer 2 to be displayed
Note: The attribute character for the character data can be replaced with X'20' at the first line and fourth column.		

The position from the first to the fourth column on the first line of the user screen format can not be used as character data when displaying the grid line in the user-defined screen format. In addition, the character "F" or "B" should be present at the first row and third column position. If it does not exist, the grid-line screen format is not displayed and the user screen format which contains the only character data is displayed.

Part 4. Appendixes

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM documentation or non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those documents or Web sites. The materials for those documents or Web sites are not part of the materials for this IBM product and use of those documents or Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

263

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department T01
Building 062
P.O. Box 12195
RTP, NC 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries, or both:

400
ACF/VTAM
AFP
AIX
AIXwindows
APL2
APPN
AS/400
AT
Advanced Peer-to-Peer Networking
AnyNet
Application System/400
BookManager
CICS
CICS/MVS
CICS/VSE
COBOL/2
CUA
Common User Access
Client Access
Client Access/400
Current
DB2
DB2/400
DRDA
DisplayWrite
Distributed Relational Database Architecture
ExecJet
GDDM
Global Network
IBM
IBMLink
IIN
IMS
InfoWindow
MVS/ESA
MVS/XA

Microsoft, Windows, Windows NT©, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Tivoli is a trademark of Tivoli Systems Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

PC Direct is a registered trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Trademarks

C-bus is a trademark of Corollary, Inc.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary includes terms and definitions from:
- The *American National Standard Dictionary for Information Systems* , ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology* Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing* , New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*
- Internet Request for Comments: 1392, *Internet Users' Glossary*
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines* , Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with:
> This refers to a term that has an opposed or substantively different meaning.

Synonym for:
> This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with:
> This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also:
> This refers the reader to terms that have a related, but not synonymous, meaning.

Deprecated term for:
> This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

The complete *IBM Dictionary of Computing* is available on the World Wide Web at **www.networking.ibm.com/nsg/nsgmain.htm**.

A

accept.   (1) In a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macroinstruction, or a VTAM operator issues a command. See also *acquire*. (2) An SMP process that moves distributed code and MVS-type programs to the distribution libraries.

ACCESS. In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

access method. (1) A technique, implemented in software, that controls the flow of information through a network. (2) A technique for moving data between main storage and input/output devices.

ACDI port. In Communications Manager/2, a serial port (such as COM1, COM2, or COM3) that can be programmed for asynchronous communications through the Asynchronous Communications Device Interface (ACDI).

ACF. Advanced Communications Function.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

acquire. (1) In VTAM, to take over resources that were formerly controlled by an access method in another domain or to resume control of resources that were controlled by that domain but released. Contrast with *release*. See also *resource takeover*. (2) In a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*.

action. (1) An operation on a managed object, the semantics of which are defined as part of the managed object class definition. (2) In the AIX operating system, a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way.

activate. To make a resource ready to perform its function. Contrast with *deactivate*.

active. (1) Operational. (2) Pertaining to a node or device that is connected or is available for connection to another node or device. (3) The state of a resource when it has been activated and is operational.

ACTPU. Activate physical unit. In SNA, a command used to start a session on a physical unit.

adapter. (1) A hardware component that must be installed in the personal computer to connect to the SDLC, LAN, asynchronous, DFT, or other communication attachment (possibly connecting through a modem). (2) A part that electrically or physically connects a device to a computer or to another device.

adaptive pacing. Synonym for *adaptive session-level pacing*.

adaptive session-level pacing. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission within a network to adapt dynamically to variations in availability and demand of buffers on a session-by-session basis. Session-level pacing occurs within independent stages along the session path according to local congestion at the intermediate and endpoint nodes. Synonymous with *adaptive pacing* and *adaptive session pacing*. Contrast with *fixed session-level pacing*.

adaptive session pacing. Synonym for *adaptive session-level pacing*.

address. In data communication, the unique code assigned to each device, workstation, or user connected to a network.

adjacent link station (ALS). (1) In SNA, a link station directly connected to a given node by a link connection over which network traffic can be carried.

Note: Several secondary link stations that share a link connection do not exchange data with each other and therefore are not adjacent to each other.
(2) With respect to a specific node, a link station partner in an adjacent node.

adjacent nodes. Two nodes connected together by at least one path that connects no other node. (T)

Advanced Communications Function (ACF). A group of IBM licensed programs, principally VTAM, TCAM, NCP, and SSP, that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

Advanced Peer-to-Peer Networking (APPN). An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection;

(c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Peer-to-Peer Networking (APPN) end node.   A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to other end nodes.

Advanced Peer-to-Peer Networking (APPN) network.   A collection of interconnected network nodes and their client end nodes.

Advanced Peer-to-Peer Networking (APPN) network node.   A node that offers a broad range of end-user services and that can provide the following:
- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

Advanced Peer-to-Peer Networking (APPN) node.   An APPN network node or an APPN end node.

advanced program-to-program communication (APPC).   (1) (2) An LU 6.2 logical unit protocol implementation of SNA that lets interconnected systems share programming tasks. The general facility characterizing the LU 6.2 architecture and its various implementations in products. (3) Sometimes used to refer to the LU 6.2 architecture and its product implementations as a whole, or to an LU 6.2 product feature in particular, such as an APPC application programming interface.

AID.   Attention identifier.

alert.   (1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In SNA management services (SNA/MS), a high priority event that warrants immediate attention.

allocate.   (1) An LU 6.2 application programming interface (API) verb used to assign a session to a conversation for the conversation's use. (2) Contrast with *deallocate*.

all points addressable (APA).   In computer graphics, pertaining to the ability to address and display or not display each picture element (pel) on a display surface.

ALS.   Adjacent link station.

American National Standards Institute (ANSI).   An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

AND operation.   Synonym for *conjunction*.

ANR.   Automatic network routing.

ANSI.   American National Standards Institute.

APA.   All points addressable.

APAR.   Authorized program analysis report.

API.   Application programming interface.

APL.   A programming language requiring the use of a special keyboard to represent various operator symbols.

APPC.   Advanced program-to-program communication.

application.   A collection of software components used to perform specific types of user-oriented work on a computer.

application program.   (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application programming interface (API).   (1) (2) A defined programming language interface between an IBM system control program or an IBM-licensed program and the program user. The set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program. (3) In VTAM, the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

Apply.   A push button that carries out the selected choices in a window without closing the window.

APPN.   Advanced Peer-to-Peer Networking.

APPN end node.   See *Advanced Peer-to-Peer Networking (APPN) end node*.

APPN network.   See *Advanced Peer-to-Peer Networking (APPN) network*.

argument.   A parameter passed between a calling program and a called program.

ASCII (American National Standard Code for Information Interchange).   The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

ASYNC.   Asynchronous.

asynchronous (ASYNC).   (1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

attach.   (1) In programming, to create a task that can be executed asynchronously with the execution of the mainline code. (2) To connect a device logically to a ring network.

attachment.   A communication link used by Personal Communications to connect a personal computer to a host system, consisting of an adapter and controlling software.

attention identifier (AID).   A character in a data stream indicating that the user has pressed a key, such as the Enter key, that requests an action by the system.

attribute.   Variable data that is logically a part of an object and that represents a property of the object. For example, a serial number is an attribute of an equipment object.

authorized program analysis report (APAR).   A report of a problem caused by a suspected defect in a current unaltered release of a program.

automatic network routing (ANR).   In High-Performance Routing (HPR), a highly efficient routing protocol that minimizes cycles and storage requirements for routing network layer packets through intermediate nodes on the route.

autotask.   (1) An unattended NetView operator station task that does not require a terminal or a logged-on user. Autotasks can run independently of VTAM and are typically used for automated console operations. (2) Contrast with *logged-on operator*.

available.   In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

B

Basic Input/Output System (BIOS).   Code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard.

batch.   (1) An accumulation of data to be processed. (2) A group of records or data processing jobs brought together for processing or transmission. (3) Pertaining to activity involving little or no user action. Contrast with *interactive*.

begin bracket. In SNA, the value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *end bracket*. See also *bracket*.

binary. Pertaining to the base two system of numbers. The binary digits are 0 and 1. Executable files are generally in binary format rather than the character string format that text files are composed of.

BIND. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

BIOS. (1) Basic Input/Output System. (2) See also *NetBIOS*.

bis. A suffix used to designate the second version of a standard, as in ITU-T Recommendation V.25 bis.

bit. Either of the digits 0 or 1 when used in the binary numeration system. (T)

bitmap. A representation of an image by an array of bits.

BIU segment. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either (a) a request/response header (RH) followed by all or a part of a request/response unit (RU) or (b) a part of an RU. Synonymous with *segment*.

block. A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or physical records. (T)

bps. Bits per second.

bracket. In SNA, one or more chains of request units and their responses that are exchanged between two session partners and that represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are database inquiries/replies, update transactions, and remote job entry output sequences to workstations.

bracket protocol. In SNA, a data flow control protocol in which exchanges between two session partners are achieved through the use of brackets, with one partner designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules.

bridge. (1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with *gateway* and *router*.

browse. (1) To look at records in a file. (2) In the NetView Graphic Monitor Facility, to open a view that cannot receive status changes from the NetView program. Contrast with *monitor*.

buffer. (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) A portion of storage used to hold input or output data temporarily.

bus. (1) A facility for transferring data between several devices located between two end points, only one device being able to transmit at a given moment. (T) (2) A computer configuration in which processors are interconnected in series.

button. See *mouse button*, *push button*, *radio button*, and *spin button*.

byte. (1) A string that consists of a number of bits, treated as a unit, and representing a character. (T) (2) A binary character operated upon as a unit and usually shorter than a computer word. (A) (3) A group of 8 adjacent binary digits that represent one EBCDIC character.

C

call. (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (I) (A) (2) In data communication, the actions necessary to make

a connection between two stations on a switched line. (3) In communications, a conversation between two users. (4) To transfer control to a procedure, program, routine, or subroutine. (5) To attempt to contact a user, regardless of whether the attempt is successful.

calling. (1) The process of transmitting selection signals in order to establish a connection between data stations. (I) (A) (2) In X.25 communications, pertaining to the location or user that makes a call.

Cancel. A push button that removes a window without applying any changes made in that window.

Caps Lock. The personal computer keyboard mode entered after the Caps Lock key is pressed. This mode is indicated by a capital A in the operator information area (OIA). When the Shift key is pressed while in this mode, Caps Lock is temporarily discontinued and the OIA indicator changes. Pressing the Shift key again returns the Caps Lock mode.

card. (1) An electronic circuit board that is plugged into a slot in a system unit. (2) A plug-in circuit assembly. (3) See also *adapter*. (4) In NetView for AIX, see *event card*.

carrier. (1) An electric or electromagnetic wave or pulse train that may be varied by a signal bearing information to be transmitted over a communication system. (T) (2) In data communication, a continuous frequency that can be modulated or impressed with an information carrying signal.

carrier detect. Synonym for *received line signal detector (RLSD)*.

CD. Compact disc.

CD-ROM. High-capacity read-only memory in the form of an optically read compact disc.

CDS. (1) Control data set. (2) Configuration data set. (3) Central directory server.

central directory server (CDS). A network node that provides a repository for information on network resource locations; it also reduces the number of network searches by providing a focal point for queries and broadcast searches and by caching the results of network searches to avoid later broadcasts for the same information.

chain. (1) A group of logically linked user data records processed by LU 6.2. (2) A group of request units delimited by begin-chain and end-chain. Responses are always single-unit chains. See *RU chain*.

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) In data communication, a means of one-way transmission. (3) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel-attached. (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. (3) Contrast with *link-attached*. (4) Synonymous with *local*.

character cell. The maximum physical boundary of a character on a display screen.

character set. A finite group of characters defined for a keyboard or output device.

check box. A square box with associated text that represents a choice. When a user selects the choice, the check box is filled to indicate that the choice is selected. The user can clear the check box by selecting the choice again, thereby deselecting the choice.

CICS. Customer Information Control System.

CICS/VS. Customer Information Control System for Virtual Storage.

circuit. (1) One or more conductors through which an electric current can flow. See *physical circuit* and *virtual circuit*. (2) A logic device.

class. (1) In object-oriented design or programming, a group of objects that share a common definition and that therefore share common properties, operations, and behavior. Members of the group are called instances of the class. (2) In the AIX operating system, pertaining to the I/O characteristics of a device. System devices are classified as block or character devices.

class of service (COS). A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

CLI. Command line interface.

click. To press and release a button on a pointing device without moving the pointer off of the object or choice.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user. (3) In an AIX distributed file system environment, a system that is dependent on a server to provide it with programs or access to programs. (4) Synonymous with *requester*.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

clipboard. An area of storage provided by the system to hold data temporarily.

Close. A choice that removes a window and all of the windows associated with it from the workplace. For example, if a user is performing a task in a window and a message appears, or the user asks for help, both the message and the help windows disappear when the user closes the original window.

cluster. A station that consists of a control unit (a cluster controller) and the terminals attached to it.

CMS. Conversational monitor system.

CNOS. Change number of sessions.

coaxial cable. A cable consisting of one conductor, usually a small copper tube or wire, within and insulated from another conductor of larger diameter, usually copper tubing or copper braid.

code page. (1) A table that defines a coded character set by assignment of a character meaning to each code point in the table for a language or a country. (2) A mapping between characters and their internal binary representation. (3) An assignment of graphic characters and control function meanings to all code points; for example, assignment of characters and meanings to 256 code points for an 8-bit code, assignment of characters and meanings to 128 code points for a 7-bit code. (4) In the Print Management Facility, a font library member that associates code points and character identifiers. A code page also identifies invalid code points. (5) A particular assignment of hexadecimal identifiers to graphic characters. (6) In AFP support, a font file that associates code points and graphic character identifiers.

COM. See *ACDI port*.

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in an FID4 transmission header.

command area. In Basic CUA architecture, the area on a panel that contains the command entry field.

command line. (1) On a display screen, a display line usually at the bottom of the screen, in which only commands can be entered. (2) In CUA architecture, deprecated term for *command area*.

command line interface (CLI). A type of computer interface in which the input command is a string of text. Contrast with *graphical user interface (GUI)*.

command name. The first term in a command, usually followed by operands.

command prompt. A displayed character or string of characters that indicates that a user may enter a command to be processed.

common operations services (COS). The portion of SNA management services that pertains to the major vectors for limited remote operations control.

Common Programming Interface for Communications (CPI-C). An evolving application programming interface (API), embracing functions to meet the growing demands from different application environments and to achieve openness as an industry standard for communications programming. CPI-C provides access to interprogram services such as (a) sending and receiving data, (b) synchronizing processing between programs, and (c) notifying a partner of errors in the communication.

communication adapter. (1) A circuit card with associated software that enables a processor, controller, or other device to be connected to a network. (2) A mechanism that enables communication facilities to be attached to host processors..

compact disc (CD). (1) A disc, usually 4.75 inches in diameter, from which data is read optically by means of a laser. (2) A disc with information stored in the form of pits along a spiral track. The information is decoded by a compact-disc player and interpreted as digital audio data, which most computers can process.

compile. (1) To translate all or part of a program expressed in a high-level language into a computer program expressed in an intermediate language, an assembly language, or a machine language. (T) (2) To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one computer instruction for each symbolic statement, or both, as well as performing the function of an assembler. (A) (3) To translate a source program into an executable program (an object program). (4) To translate a program written in a high-level programming language into a machine language program.

compiler. (1) A program that translates a source program into an executable program (an object program). (2) A program that decodes instructions written as pseudo codes and produces a machine language program to be executed at a later time.

component. Hardware or software that is part of a functional unit.

computer. A functional unit that can perform substantial computations, including numerous arithmetic operations and logic operations without human intervention during a run. In information processing, the term computer usually describes a digital computer. A computer may consist of a stand-alone unit or may consist of several interconnected units. (T)

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network. (3) In Personal Communications, the arrangement of personal computers connected to one or more host systems by one or more attachment types. Examples are: SDLC, LAN, ASYNCH, X.25, or DFT.

configuration file. A file that specifies the characteristics of a system device or network.

configuration services. One of the types of network services in a control point (SSCP, NNCP, or ENCP). Configuration services activates, deactivates, and records the status of physical units, links, and link stations.

configure. To describe to a system the devices, optional features, and programs installed on the system.

congestion. See *network congestion*.

conjunction. The Boolean operation whose result has the Boolean value 1 if and only if each operand has the Boolean value 1. (I) (A) Synonymous with *AND operation*.

connected. In VTAM, the state of a physical unit (PU) or a logical unit (LU) that has an active physical path to the host processor containing the system services control point (SSCP) that controls the respective PU or LU.

connection. (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In SNA, the network path that links together two logical units (LUs) in different nodes to enable them to establish communications. (3) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system.

connectivity. (1) The capability of a system or device to be attached to other systems or devices without modification. (T) (2) The capability to attach a variety of functional units without modifying them.

control character. A character whose occurrence in a particular context specifies a control function. (T)

control data set (CDS). In NPM, an SMP data set used in the NPM installation process.

controller. A device that coordinates and controls the operation of one or more input/output devices, such as workstations, and synchronizes the operation of such devices with the operation of the system as a whole.

control point (CP). (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

Control Program (CP). In VM/ESA, a component that manages the resources of a single computer so multiple computing systems appear to exist. Each of these apparent systems, or virtual machines, is the functional equivalent of an IBM System/370, 370-XA, or ESA computer.

control unit terminal (CUT) mode. (1) An IBM protocol used for communications with an IBM 3174 or 3274 Control Unit or other appropriate interface unit. In this protocol, a program in the workstation emulates a 3278 or 3879 terminal for a user at a virtual terminal, and the interface unit is responsible for enforcing the protocol. (2) Contrast with *distributed function terminal (DFT)*.

conversation. A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under control of the VM control program.

coordinated universal time (UTC). The time scale, based on the Système International (SI) second, as defined and recommended by the Comité Consultatif International de la Radio (CCIR) and maintained (using an atomic clock) by the Bureau International des Poids et Mesures (BIPM).

Note: The Système International is based on three fundamental units of measure—the meter, the kilogram, and the second—and is sometimes called the "MKS system" because of these units.

For most practical purposes, coordinated universal time (UTC) is equivalent to the mean solar time at the prime meridian (0 degrees longitude) of Greenwich, England, known as *Greenwich mean time (GMT)*. Synonymous with *Z time* and *Zulu time*.

Copy. A choice that places a copy of a selected object onto the clipboard.

COS. Class of service.

country code. In X.25 communications, the 3-digit number that precedes the national terminal number in the network user address for public networks.

CP. (1) Control point. (2) In VM, Control Program.

CPI-C. Common Programming Interface for Communications.

CP name. A network-qualified name of a control point (CP), consisting of a network ID qualifier identifying the network (or name space) to which the CP's node belongs, and a unique name within the scope of that network ID identifying the CP. Each APPN or LEN node has one CP name, assigned to it at system-definition time.

CSD. Corrective service diskette.

cursor. (1) A movable, visible mark used to indicate a position of interest on a display surface. (A) (2) A visible indication of the position where user interaction with the keyboard will appear. The keyboard cursors are the selection cursor and the text cursor.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

Customer Information Control System for Virtual Storage (CICS/VS). (1) An IBM licensed program that can be used in a communications network. (2) A general-purpose licensed program that controls online communication between terminal users and a database.

customization. The process of configuring one or more personal computers connected to a host system by the SDLC, LAN, ASYNCH, X.25, DFT, or other attachment. Customization precedes Personal Communications installation and includes the definition of the sessions to be carried out by Personal Communications.

Cut. A choice that moves a selected object and places it onto the clipboard. The space it occupied is usually filled by the remaining object or objects in the window.

D

data. (1) A re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. Operations can be performed upon data by humans or by automatic means. (T)   (2) Any representations such as characters or analog quantities to which meaning is or might be assigned. (A)   (3) A representation of facts or instructions in a form suitable for communication, interpretation, or processing by human or automatic means. Data include constants, variables, arrays, and character strings.

Note: Programmers make a distinction between instructions and the data they operate on; however, in the usual sense of the word, data includes programs and program instructions.

database. (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) A collection of interrelated data organized according to a database schema to serve one or more applications. (T) (3) A collection of data fundamental to a system. (A) (4) A collection of data fundamental to an enterprise. (A)

data carrier detect (DCD). Synonym for *received line signal detector (RLSD).*

data circuit. (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (I) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

Notes:

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

data flow control (DFC). In SNA, a request/response unit (RU) category used for requests and responses exchanged between the data flow control layer in one half-session and the data flow control layer in the session partner.

data link. In SNA, synonym for *link.*

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

Note: The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reaches the higher layers.

data packet. In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

data set. (1) Synonym for *file*. (2) Deprecated term for *modem*.

data stream. (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

data terminal ready (DTR). A signal to the modem used with the EIA 232 protocol.

data types. In the NetView program, a description of the organization of panels. Data types are alerts, events, and statistics. Data types are combined with resource types and display types to describe the NetView program's display organization. See also *display types* and *resource types*.

DBCS. Double-byte character set.

DCD. Data carrier detect.

DCE. (1) Data circuit-terminating equipment. (2) Distributed Computing Environment.

deactivate. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

deallocate. An LU 6.2 application programming interface (API) verb that terminates a conversation, thereby freeing the session for a future conversation. Contrast with *allocate*.

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

definite response (DR). In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request chain. Contrast with *exception response* and *no response*.

definition file. A file loaded by default that contains keyboard and screen color characteristics that can be adapted for a particular session. The lines of a definition file must be coded according to a strict layout.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. (3) See also *macroinstruction*. (4) See Figure 7, Figure 8 on page 278, and Figure 9 on page 278.
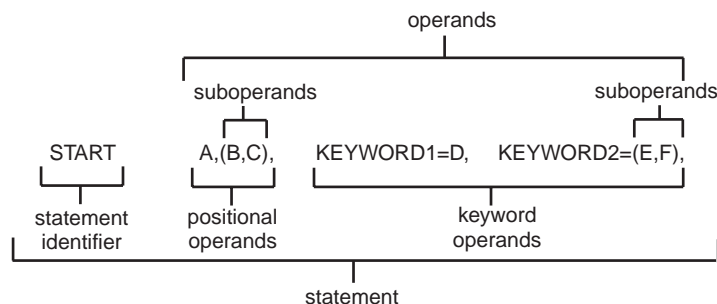
Figure 7. Example of a Language Statement

```
definition              keyword operand
statement          ┌─────────┴─────────┐
identifier               suboperands
                       ┌──────┴──────┐
  ┌──┴──┐
   LINE           AUTO=(YES,32)
  └──┬──────────────────┬──┘
              │
      definition statement
```
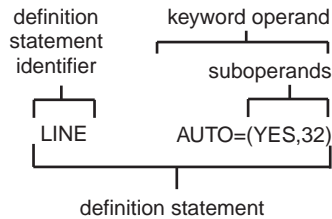
Figure 8. Example of an NCP Definition Statement

```
definition              keyword operand
statement          ┌─────────┴─────────┐
identifier               suboperands
                       ┌──────┴──────┐
  ┌─┴─┐
   PU            DISCNT=(YES,NF)
  └─┬──────────────────┬──┘
              │
      definition statement
```
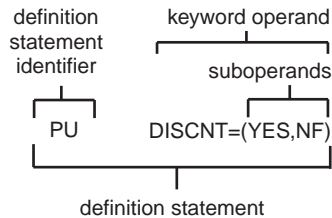
Figure 9. Example of a VTAM Definition Statement

Delete. A choice that removes a selected object. The space it occupied is usually filled by the remaining object or objects in the window.

delimiter. (1) A character used to indicate the beginning and end of a character string. (T) (2) A flag that separates and organizes items of data. (A) (3) A character that groups or separates words or values in a line of input. (4) In the IBM Token-Ring Network, a bit pattern that defines the limits of a frame or token.

dependent LU. See *SSCP-dependent LU*.

dependent LU requester (DLUR). An APPN end node or an APPN network node that owns dependent LUs, but requests that a dependent LU server provide the SSCP services for those dependent LUs..

dependent LU server (DLUS). An APPN network node that provides SSCP services for a dependent LU in its own or another APPN network. Contrast with *dependent LU requester*..

destination. (1) Any point or location, such as a node, station, or a particular terminal, to which information is to be sent. (2) An external logical unit (LU) or application program to which messages or other data are directed.

destination address. A code that identifies the location to which information is to be sent.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

device driver. (1) A file that contains the code needed to use an attached device. (2) A program that enables a computer to communicate with a specific peripheral device; for example, a printer, a videodisc player, or a CD drive. (3) A collection of subroutines that control the interface between I/O device adapters and the processor.

DFT. Distributed function terminal.

dialog. (1) The interaction between a user and a computer. (2) In an interactive system, a series of related inquiries and responses similar to a conversation between two people. (3) In the AIXwindows Toolkit, a two-way text interface between an application and its user. The interface takes the form of a collection of widgets and gadgets, including a DialogShell widget, a BulletinBoard widget (or a subclass of a BulletinBoard widget or some other container widget), plus various children, including Label, PushButton, and Text widgets.

directory. (1) A table of identifiers and references to the corresponding items of data. (I) (A) (2) A named hierarchical grouping of files in a file system. (3) A database in an APPN node that lists names of resources (in particular, logical units) and records the CP name of the node where each resource is located. See *distributed directory database* and *local directory database*.

directory services (DS). A control point component of an APPN node that maintains knowledge of the location of network resources.

diskette. (1) A small magnetic disk enclosed in a jacket. (T) (2) A thin, flexible magnetic disk and a semi-rigid protective jacket, in which the disk is permanently enclosed.

display. A visual presentation of data. (I) (A)

display levels. Synonym for *display types*.

display types. In the NetView program, a concept to describe the organization of panels. Display types are defined as total, most recent, user action, and detail. Display types are combined with resource types and data types to describe NetView's panel organization. See also *data types* and *resource types*. Synonymous with *display levels*.

Distributed Computing Environment (DCE). The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

distributed directory database. The complete listing of all the resources in the network as maintained in the individual directories scattered throughout an APPN network. Each node has a piece of the complete directory, but it is not necessary for any one node to have the entire list. Entries are created, modified, and deleted through system definition, operator action, automatic registration, and ongoing network search procedures. Synonymous with *distributed network directory* and *network directory database*.

distributed function terminal (DFT). (1) A protocol used for communication between a terminal and an IBM 3274 or IBM 3174 control unit that allows multiple concurrent logical terminal sessions. (2) A mode of operation that uses this protocol. (3) A terminal that supports multiple concurrent logical terminal sessions. (4) Contrast with *control unit terminal (CUT) mode*.

distributed network directory. Synonym for *distributed directory database*.

distributed processing. Processing that takes place across two or more linked systems.

DLC. Data link control.

DLL. Dynamic link library.

DLUR. Dependent LU requester..

domain operator. In a multiple-domain network, the person or program that controls operation of resources controlled by one system services control point (SSCP). See also *network operator*.

DOS. Disk Operating System. See *IBM Disk Operating System*.

DOS session. A session in which a personal computer operates as a stand-alone computer, running under Disk Operating System (DOS). See *host session*.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set (SBCS)*.

double-click. To press and release a button on a pointing device twice while a pointer is within the limits that the user has specified for the operating environment.

download. (1) To transfer programs or data from a computer to a connected device, typically a personal computer. (T) (2) To transfer data from a computer to a connected device, such as a workstation or microcomputer. Contrast with *upload*.

downstream. In the direction of data flow from the host to the user. Contrast with *upstream*.

drive. A peripheral device, especially one that has addressed storage media.

DS. Directory services.

DTE. Data terminal equipment. (A)

DTR. Data terminal ready.

duplex. Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full-duplex*. Contrast with *half-duplex*.

dynamic. (1) In programming languages, pertaining to properties that can only be established during the execution of a program; for example, the length of a variable-length data object is dynamic. (I) (2) Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time. (3) Contrast with *static*.

dynamic data exchange (DDE). A protocol that allows applications to exchange data.

dynamic link library (DLL). A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

EC. Engineering change.

Edit. A choice on a menu bar that provides access to other choices that enable a user to modify data.

EGA. Enhanced graphics adapter.

EHLLAPI. Emulator High-Level Language Application Programming Interface.

element. (1) A field in the network address. (2) In SNA, the particular resource within a subarea that is identified by an element address. See also *subarea*.

emulator. A program that allows a device to operate as if it were a different type of device. Personal Communications, for example, allows supported personal computers and printers to operate as if they were 3270-series workstations.

Emulator High-Level Language Application Programming Interface (EHLLAPI). In Communications Manager/2, an application programming interface that provides programming access to the area in computer memory that corresponds to the user's screen image (this area in memory is known as the "presentation space").

enabled. (1) Pertaining to a state of the processing unit that allows the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line.

end bracket. In SNA, the value (binary 1) of the end bracket indicator in the request header (RH) of the first request of the last chain of a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*. See also *bracket*.

end node (EN). (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

enhanced graphics adapter (EGA). An adapter, such as the IBM Enhanced Graphics Adapter, that provides high-resolution graphics, allowing the use of a color display for text processing as well as graphics applications.

entry field. An area into which a user types or places text. Its boundaries are usually indicated.

environment variable. A variable that specifies (a) how an operating system or another program will run or (b) the devices that the operating system will recognize.

EOF. End of Field

EOT. End of Text

ER. Explicit route.

error. A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. (I) (A)

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

event. An occurrence of significance to a task; for example, an SNMP trap, the opening of a window or a submap, or the completion of an asynchronous operation.

event card. In NetView for AIX, a graphical representation, resembling a card, of the information contained in an event sent by an agent to a manager reflecting a change in the status of one of the agent's managed nodes.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file.

exception response (ER). In SNA, a protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, can be returned. Contrast with *definite response* and *no response*.

execute. To perform the actions specified by a program or a portion of a program. (T)

exit. To execute an instruction within a portion of a computer program in order to terminate the execution of that portion. Such portions of computer programs include loops, subroutines, modules, and so on. (T)

explicit route (ER). In SNA, a series of one or more transmission groups that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*.

EXT. External trace file.

extended binary-coded decimal interchange code (EBCDIC). The standard code, using a character set consisting of 8-bit coded characters, used by Personal Communications for information interchange between personal computers and a host system. See also *American National Standard Code for Information Interchange*.

E1. See *T1*.

F

fault. An accidental condition that causes a functional unit to fail to perform its required function. (I) (A)

feature. A part of an IBM product that may be ordered separately by the customer.

FID. Format identification.

field. (1) An area in a record or panel used to contain data. (2) In the IBM 3270 data stream, a group of consecutive positions on a presentation space having similar characteristics that are defined by a field attribute byte at the beginning of the field. (3) An identifiable area in a window. Examples of fields are: an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice.

file. A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

file mode. In VM/CMS, the third field of the file identifier, which indicates the assigned virtual disk wherein the file resides.

file specification. In Communications Manager/2, the full identifier for a file, which includes its drive, path, file name, and file extension. The format is dependent on the storage medium of a file; for example, `C:\PATH\FILENAME.EXT`.

file transfer. The transfer of one or more files from one system to another over a data link.

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

file type. In VM/CMS, the second field of the file identifier. The file type can be arbitrary, or of significance to the system (for example, SCRIPT, LIST3270, and NOTEBOOK file types).

fixed pacing. Synonym for *fixed session-level pacing*.

fixed session-level pacing. A form of session-level pacing in which the data transfer rate is controlled using fixed pacing-window sizes, which are initialized at session-activation time. Synonymous with *fixed pacing*. Contrast with *adaptive session-level pacing*.

flow. In NetDA/2, the amount of traffic that can pass through a node, connection, or route in both directions during a given period of time.

flow control. (1) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. (2) See also *pacing*.

focal point (FP). See *management services focal point (MSFP)*.

folder. A container used to organize objects.

format identification (FID) field. In SNA, a field in each transmission header (TH) that indicates the format of the TH; that is, the presence or absence of certain fields. TH formats differ in accordance with the types of nodes between which they pass. Following are the six FID types:

- FID0, used for traffic involving non-SNA devices between adjacent subarea nodes when either or both nodes do not support explicit route and virtual route protocols
- FID1, used for traffic involving SNA devices between adjacent subarea nodes when either or both nodes do not support explicit route and virtual route protocols
- FID2, used for traffic between a subarea node and an adjacent type 2 peripheral node
- FID3, used for traffic between a subarea node and an adjacent type 1 peripheral node
- FID4, used for traffic between adjacent subarea nodes when both nodes support explicit route and virtual route protocols
- FIDF, used for certain commands (for example, for transmission group control) sent between adjacent subarea nodes when both nodes support explicit route and virtual route protocols.

FP. Focal point.

frame. (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. (4) A data structure (data frame) composed of fields meeting the field specifications of a type of communication protocol. Frames are used to control data transfer across a data link. (5) In SDLC, a sequence of bits delimited by an opening and closing flag. In X.25 packet switching data networks, frames are composed of 8-bit byte sequences delimited by beginning and ending flags; the frames in X.25 control various functions, data transfer, and transmission checking.

FTP. File Transfer Protocol.

full-duplex (FDX). Synonym for *duplex*.

function management data (FMD). An RU category used for end-user data exchanged between logical units (LUs) and for requests and responses exchanged between network services components of LUs, PUs, and control points.

function management header (FMH). One or more headers, optionally present in the leading request units (RUs) of an RU chain, that allow one LU to (a) select a transaction program or device at the session partner and control the

way in which the end-user data it sends is handled at the destination, (b) change the destination or the characteristics of the data during the session, and (c) transmit between session partners status or user information about the destination (for example, a program or device). Function management headers can be used with LU type 1, 4, and 6.2 protocols.

G

gateway.   (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T)   (2) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM. (3) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (4) In the AIX operating system, an entity that operates above the link layer and translates, when required, the interface and protocol used by one network into those used by another distinct network. (5) In TCP/IP, synonym for *router.*. (6) A station in a local area network through which a connection to a host system or a separate network is established.

GDDM.   Graphical Data Display Manager.

generation.   The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

generic unbind.   Synonym for *session deactivation request*.

GMT.   Greenwich mean time.

Graphical Data Display Manager (GDDM).   (1) An IBM host function that processes text and graphics for output to a display, printer, or graphic output device. (2) In the NetView Performance Monitor (NPM), an IBM licensed program used in conjunction with the Presentation Graphics Feature (PGF) to generate online graphs in the NPM Graphic Subsystem.

graphical user interface (GUI).   (1) A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, representing actual objects, that the user can access and manipulate with a pointing device. (2) Contrast with *command line interface (CLI)*.

Greenwich mean time (GMT).   The mean solar time at the prime meridian (0 degrees longitude) of Greenwich, England. Greenwich mean time is sometimes called *Z time* or *Zulu time*.

Note: Although *Greenwich mean time (GMT)* and *coordinated universal time (UTC)* are sometimes used interchangeably, they are not synonyms. Greenwich mean time is an approximate time. Because the second is no longer defined in terms of astronomical phenomena, the preferred name for this time scale is *coordinated universal time (UTC)*.

H

half-duplex (HD, HDX).   In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*.

handle.   In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened.

hard disk.   A rigid-magnetic disk such as the internal disks used in the system units of personal computers and in external hard disk drives (also known as fixed disks).

HD.   Half-duplex.

HDLC.   High-level data link control.

header.   (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

Help. A choice that gives a user access to helpful information about objects, choices, tasks, and products. A Help choice can appear on a menu bar or as a push button.

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

high-level language (HLL). A programming language that does not reflect the structure of any particular computer or operating system.

high-level language application programming interface (HLLAPI). (1) A software product that supports interaction between a host program and an application program running on a personal computer in terminal emulation (usually 3270 terminal emulation). (2) A programming interface that usually operates in conjunction with an emulator, such as 3270 emulation, and allows interaction using 3270 data stream between a host and a remote application program.

highlighting. Emphasizing a display element or segment by modifying its visual attributes. (I) (A)

High-Performance Routing (HPR). An addition to APPN that enhances data-routing performance and session reliability..

High-Performance Routing (HPR) node. An APPN end node or network node that includes High-Performance Routing support.

hiragana. One of the two common Japanese phonetic alphabets (the other is katakana). In hiragana, each character is represented by 1 byte. See also *kanji*..

host. (1) In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe. (2) See *host processor*.

host code page. The character code page used by the host system to which a gateway connects.

host ID. In the Internet suite of protocols, that part of the IP address that defines the host system on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

host print. A printer session on a local personal computer directed by the host system.

host processor. (1) A processor that controls all or part of a user application network. (T) (2) In a network, the processing unit in which the data communication access method resides.

host session. A logical connection that enables a personal computer to communicate with a host system. A session can be identified by LU address, LT number, or session ID. See *DOS session*. See also *logical terminal*.

host system. In Personal Communications, the computer linked to one or more personal computers by the SDLC, LAN, ASYNCH, X.25, or DFT attachment.

HPR. High-Performance Routing..

I

IBM Disk Operating System (DOS). A disk operating system based on MS-DOS that operates with all IBM personal computers.

IBM OS/2 Presentation Manager. The front-end data manager and user interface for the IBM OS/2 operating system; an example of a graphical user interface.

icon. A graphical representation of an object, consisting of an image, image background, and a label.

ICP. Internet Control Protocol.

ID. (1) Identifier. (2) Identification.

IEEE. Institute of Electrical and Electronics Engineers.

IEEE 802.2. An IEEE standard describing how data is formatted into frames for LAN transmission. See also *Type 1 frame* and *Type 2 frame*.

inactive. (1) Not operational. (2) Pertaining to a node or device not connected or not available for connection to another node or device. (3) Contrast with *active*.

inbound. In communications, data that is received from the network.

information (I) format. A format used for information transfer.

information (I) frame. A frame in I format used for numbered information transfer.

input field. (1) In computer graphics, an unprotected field on a display surface in which data can be entered, modified, or erased. (2) A field in a display file into which a user can key in data. An input field is passed from the device to the program when the program reads the record containing that field.

input/output (I/O). (1) Pertaining to input, output, or both. (A)   (2) Pertaining to a device, process, or channel involved in data input, data output, or both.

installation. (1) In Personal Communications, the process of loading microcode from the Personal Communications diskettes. (2) In system development, preparing and placing a functional unit in position for use. (T)   (3) A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

instance. In the AIX operating system, a concrete realization of an abstract object class. An instance of a widget or a gadget is a specific data structure that contains detailed appearance and behavioral information that is used to generate a specific graphical object on-screen at run time.

Institute of Electrical and Electronics Engineers (IEEE). An organization that establishes and promotes standards in related industries.

interactive. Pertaining to the exchange of information between a user and a computer.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T)   (2) Hardware, software, or both, that links systems, programs, or devices.

intermediate node. A node that is at the end of more than one branch. (T)

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

International Telecommunication Union (ITU). The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

International Telecommunication Union - Telecommunication Standardization Sector (ITU-TS). The part of the International Telecommunication Union (ITU) that is reponsible for developing recommendations for telecommunications.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet Control Protocol (ICP). The VIrtual NEtworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internetwork Packet Exchange (IPX). The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology.

I/O. Input/output.

IP. Internet Protocol..

IPX. Internetwork Packet Exchange.

ISO. International Organization for Standardization.

ITU. International Telecommunication Union.

ITU-T. See *ITU-TS*.

ITU-TS. International Telecommunication Union - Telecommunication Standardization Sector.

J

JISCII. Japanese Industry Standard Code for Information Interchange. A code used by IBM Personal Computers in Japan. It contains Japanese Industry Standard (JIS) C 6226 "Kanji code for exchanging information" (JIS Kanji set) and user unique characters.

jump. To switch to the logical terminal session specified.

K

kanji. A Japanese ideographic alphabet. In kanji, each character is represented by 2 bytes. See also *hiragana* and *katakana*..

katakana. One of the two common Japanese phonetic alphabets (the other is hiragana). In katakana, each character is represented by 1 byte. Katakana is primarily used to write foreign words phonetically. See also *kanji*..

kbps. Kilobits per second.

kernel. The part of an operating system that performs basic functions such as allocating hardware resources.

keyboard template. An overlay placed on a keyboard to explain the function of each key.

keylock. In Communications Manager/2, a function that controls access to configuration data and some advanced functions. See also *lock*, *master key*, and *service key*.

keyword. (1) In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; for example, in some contexts, IF characterizes an if-statement. A keyword normally has the form of an identifier. (I) (2) One of the predefined words of an artificial language. (A) (3) A significant and informative word in a title or document that describes the content of that document.

L

LAN. Local area network.

line. (1) The portion of a data circuit external to data circuit-terminating equipment (DCE), that connects the DCE to a data switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. (I) (2) Synonymous with *channel* and *circuit*.

line control discipline. Synonym for *link protocol* and *protocol*.

line discipline. Synonym for *link protocol* and *protocol*.

line speed. The number of binary digits that can be sent over a telecommunication line in one second, expressed in bits per second (bps).

link. (1) The combination of the link connection (the transmission medium) and two link stations, one at each end of the link connection. A link connection can be shared among multiple links in a multipoint or token-ring configuration. (2) To interconnect items of data or portions of one or more computer programs: for example, the linking of object programs by a linkage editor, linking of data items by pointers. (T) (3) In SNA, synonymous with *data link*.

link-attached. (1) Pertaining to devices that are connected to a controlling unit by a data link. (2) Contrast with *channel-attached*. (3) Synonymous with *remote*.

link connection. (1) The physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE). (2) In SNA, synonymous with *data circuit*.

link connection segment. A portion of the configuration that is located between two resources listed consecutively in the service point command service (SPCS) query link configuration request list.

link protocol. (1) The rules for sending and receiving data at the link level. (2) Synonymous with *line control discipline* and *line discipline*.

link station. (1) The hardware and software components within a node representing a connection to an adjacent node over a specific link. For example, if node A is the primary end of a multipoint line that connects to three adjacent nodes, node A will have three link stations representing the connections to the adjacent nodes. (2) In VTAM, a named resource within an APPN or a subarea node that represents the connection to another APPN or subarea node that is attached by an APPN or a subarea link. In the resource hierarchy in a subarea network, the link station is subordinate to the subarea link. (3) See also *adjacent link station (ALS)*.

link status (LS). Information maintained by local and remote modems.

load. (1) To bring all or part of a computer program into memory from auxiliary storage so that the computer can run the program. (2) To place a diskette into a diskette drive.

local. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Contrast with *remote*. (3) Synonym for *channel-attached*.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. (3) See also *Ethernet* and *token ring*. (4) Contrast with *metropolitan area network (MAN)* and *wide area network (WAN)*.

local copy. A printer session initiated by a personal computer user or host that reproduces a presentation space with all supported graphics.

local directory database. That set of resources (LUs) in the network known at a particular node. The resources included are all those in the node's domain as well as any cache entries.

local LU. A logical unit not distributed over the LAN, but controlled by a gateway personal computer. This is normally a physical device such as a workstation, printer, or terminal.

Locate. Synonym for *Locate/CD-Initiate*.

Locate/CD-Initiate. (1) An abbreviated term for a message exchanged between APPN nodes that contains one of the following sets of general data stream (GDS) variables:
- A Locate, a Find Resource, and a Cross-Domain Initiate GDS variable used for a network search request
- A Locate, a Found Resource, and a Cross-Domain Initiate GDS variable used for a search reply when a network resource has been located

These message structures correspond to the CP components that perform the search of the distributed network directory and establish the session. The Locate GDS variable contains information used to control the delivery of the search messages in the network. The Find and Found GDS variables contain information used in the directories: origin cache data (control point information) and search arguments (destination LU name), and located resource

information, respectively. The Cross-Domain Initiate GDS variable contains endpoint TG vector information to be used in selecting the route for the session. The length of the Locate/CD-Initiate message is limited to 1024 bytes. (2) Synonymous with *Locate* and *Locate search message*.

Locate search message. Synonym for *Locate/CD-Initiate.*

lock. (1) The means by which integrity of data is ensured by preventing more than one user from accessing or changing the same data or object at the same time. (2) In Communications Manager/2, a password-protection system that can be used to prevent access to some advanced functions. See also *keylock*.

logged-on operator. (1) A NetView operator station task that requires a terminal and a logged-on user. (2) Contrast with *autotask*.

logical link. (1) A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network. (2) In APPNTAM, the unidirectional representation in a node of a link..

logical record. A set of related data or words considered to be a record from a logical viewpoint. (T)

logical terminal. (1) A destination with a name that is related to one or more physical terminals. (2) The definition of a specific 3270 or 5250 emulation session.

logical unit (LU). A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

low-entry networking (LEN) end node. A LEN node receiving network services from an adjacent APPN network node.

LS. Link status.

LU. Logical unit.

LU-LU session. A logical connection between two logical units (LUs) in an SNA network that typically provides communication between two users.

LUS. Logical unit services.

LU type. The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:
- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU 2. A type of LU for an application program that communicates with a single display workstation in an interactive environment, using the SNA 3270 data stream.

LU 3. A type of LU for an application program that communicates with a single printer, using the SNA 3270 data stream.

LU 6.2. (1) A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application programming interface (API) consisting of structured verbs that are mapped into a product implementation. (2) A type of LU that supports sessions between two applications in a distributed data processing environment using the SNA general data stream, which is a structured-field data stream, or a user-defined data stream.

LU 6.2 verb. A syntactical unit in the LU 6.2 application programming interface representing an operation.

M

macroinstruction. (1) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that may also specify values for parameters in the replaced instructions. (T) (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. (3) See also *definition statement*.

maintenance analysis procedure (MAP). A maintenance document that gives an IBM service representative a step-by-step procedure for tracing a symptom to the cause of a failure.

maintenance services. In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). Maintenance services provide facilities for testing links and nodes and for collecting and recording error information.

MAN. Metropolitan area network.

management services (MS). (1) One of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management, and change management. (2) Services that assist in the management of systems and networks in areas such as problem management, performance management, business management, operations management, configuration management, and change management.

management services focal point (MSFP). For any given management services discipline (for example, problem determination or response time monitoring), the control point that is responsible for that type of network management data for a sphere of control. This responsibility may include collecting, storing, or displaying the data, or all of these. (For example, a problem determination focal point is a control point that collects, and that may store or display, problem determination data.)

manager. (1) In systems management, a user that, for a particular interaction, has assumed a manager role. (2) An entity that monitors or controls one or more managed objects by (a) receiving notifications regarding the objects and (b) requesting management operations to modify or query the objects. (3) A system that assumes a manager role.

mantissa. The decimal part of a common logarithm.

map. In NetView for AIX, a database represented by a set of related submaps that provide a graphical and hierarchical presentation of a network and its systems.

MAP. Maintenance analysis procedure.

mapping. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

master key. In Communications Manager/2, the access password used to unlock the keylock for Communications Manager/2. A user who has access to the master key can also change the service key.

matching. In MPTN architecture, pertaining to the relationship between peer transport users or peer transport providers that use the same user protocols or the same transport protocols.

Maximize. A choice that enlarges a window to its largest possible size.

MB. Megabyte.

medium. (1) A physical carrier of electrical energy. (2) A physical material in or on which data may be represented.

megabyte (MB). (1) For processor storage, real and virtual storage, and channel volume, 220 or 1 048 576 bytes. (2) For disk storage capacity and communications volume, 1 000 000 bytes.

memory. All of the addressable storage space in a processing unit and other internal storages that is used to execute instructions. (T)

menu. (1) A list of options displayed to the user by a data processing system, from which the user can select an action to be initiated. (T)　(2) In text processing, a list of choices displayed to the user by a text processor from which the user can select an action to be initiated. (T)　(3) A list of choices that can be applied to an object. A menu can contain choices that are not available for selection in certain contexts. Those choices are indicated by reduced contrast.

menu bar. The area near the top of a window, below the title bar and above the rest of the window, that contains choices that provide access to other menus.

message. (1) An assembly of characters and sometimes control codes that is transferred as an entity from an originator to one or more recipients. A message consists of two parts: envelope and content. (T)　(2) A communication sent from a person or program to another person or program.

metropolitan area network (MAN). A network formed by the interconnection of two or more networks which may operate at higher speed than those networks, may cross administrative boundaries, and may use multiple access methods. (T) Contrast with *local area network (LAN)* and *wide area network (WAN)*.

MID. Machine identifier.

migration. The installation of a new version or release of a program to replace an earlier version or release.

mode. See *mode name*.

modem (modulator/demodulator). (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

mode name. The name used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service within the transport network.

module. A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine. (A)

monitor. (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T)　(2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A)　(3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations. (4) In the NetView Graphic Monitor Facility, to open a view that can receive status changes from the NetView program. Problem determination and correction can be performed directly from the view. Contrast with *browse*.

mouse. A commonly used pointing device, containing one or more buttons, with which a user can interact with a product or the operating environment.

mouse button. A mechanism on a mouse pointing device used to select objects or choices, initiate actions, or directly manipulate objects; a user presses a mouse button to interact with a computer system. The button makes a "clicking" sound when pressed and released.

MS. Management services.

MSFP. Management services focal point.

Multiple Virtual Storage (MVS). See *MVS*.

MVS. Multiple Virtual Storage. Implies MVS/390, MVS/XA, and MVS/ESA.

N

national language support (NLS). The modification or conversion of a United States English product to conform to the requirements of another language or country. This can include the enabling or retrofitting of a product and the translation of nomenclature, machine-readable information (MRI), or documentation of a product. NLS is also known as "internationalization."

native. In MPTN architecture, pertaining to the relationship between a transport user and a transport provider that are both based on the same transport protocol.

NC. Network control.

negative response (NR). In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*.

negotiation. The process of deciding what packet size to transmit between a network and a 3710 Network Controller.

NetBIOS. (1) Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols. (2) See also *BIOS*.

NetView-NetView task (NNT). The task under which a cross-domain NetView operator session runs. See *operator station task*.

network. (1) An arrangement of nodes and connecting branches. (T) (2) A configuration of data processing devices and software connected for information interchange. (3) A group of nodes and the links interconnecting them.

network architecture. The logical structure and operating principles of a computer network. (T)

Note: The operating principles of a network include those of services, functions, and protocols.

network congestion. An undesirable overload condition caused by traffic in excess of what a network can handle.

network control (NC). In SNA, a request/response unit (RU) category used for requests and responses exchanged between physical units (PUs) for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjust peripheral nodes. See also *data flow control*, *function management data*, and *session control*.

network directory database. Synonym for *distributed directory database*.

network management. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network node (NN). See *Advanced Peer-to-Peer Networking (APPN) network node*.

network node server. An APPN network node that provides network services for its local LUs and client end nodes.

network operator. (1) A person who controls the operation of all or part of a network. (2) In a multiple-domain network, a person or program responsible for controlling all domains. (3) See also *domain operator*.

NLS. National language support.

NMVT. Network management vector transport.

NN. Network node.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) Any device, attached to a network, that transmits and receives data. (3) An endpoint of a link or a junction common

to two or more links in a network. Nodes can be processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities.

no response. In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request not to return any response, regardless of whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

NUL. A transmission control character used as a filler or for sequence control in network communication.

O

object. (1) In object-oriented design or programming, an abstraction consisting of data and the operations associated with that data. See also *class*. (2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both.

Off. A choice that appears in the cascaded menu from the Refresh choice. It sets the refresh function to off.

offline. (1) Pertaining to the operation of a functional unit that takes place either independently of, or in parallel with, the main operation of a computer. (T) (2) Neither controlled by, nor communicating with, a computer. Contrast with *online*.

offset. The number of measuring units from an arbitrary starting point in a record, area, or control block, to some other point.

OIA. Operator information area.

OK. A push button that accepts the information in a window and closes it. If the window contains changed information, those changes are applied before the window is closed.

On. A choice that appears in a cascaded menu from the Refresh choice. It immediately refreshes the view in a window.

online. (1) Pertaining to the operation of a functional unit when under the direct control of the computer. (T) (2) Pertaining to a user's ability to interact with a computer. (A) (3) Pertaining to a user's access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer. (5) Contrast with *offline*.

open. (1) A break in an electrical circuit. (2) To make an adapter ready for use.

Open. A choice that leads to a window in which users can select the object they want to open.

operand. (1) An entity on which an operation is performed. (I)　(2) That which is operated upon. An operand is usually identified by an address part of an instruction. (A)　(3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

operating system (OS). Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

operation. In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object.

operator. (1) In a language statement, the lexical entity that indicates the action to be performed on operands. See also *definition statement*. (2) A person or program responsible for managing activities controlled by a given piece of software such as MVS, the NetView program, or IMS. (3) A person who operates a device. (4) A person who keeps a system running. (5) See also *autotask*, *logged-on operator*, *network operator*, and *operator station task*.

operator information area (OIA). The area near the bottom of the display area where terminal or system status information is displayed.

operator station task (OST). The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

origin. An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

OS. Operating system.

outbound. In communications, data that is transmitted to the network.

output. Pertaining to a device, process, or channel involved in an output process, or to the associated data or states. The word "output" may be used in place of "output data," "output signal", "output process", when such a usage is clear in a given context. (T)

P

pacing. (1) A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. (2) See *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*. (3) See also *flow control*.

pacing group. Synonym for *pacing window*.

pacing window. (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests. (3) Synonymous with *pacing group*.

packet. (1) In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)   (2) A discrete unit of information including addressing and sequence information that can be sent interleaved with other packets of the same and other transmission sequences. Packets sent over X.25 virtual circuits always arrive in the order they were sent.

page. (1) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between real storage and auxiliary storage. (I)   (A)   (2) The information displayed at the same time on the screen of a display device. (3) To replace the information displayed on the screen with prior or subsequent information from the same file.

panel. See *window*.

parallel. (1) Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus. (T)   (2) Pertaining to concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device. (A)   (3) Pertaining to concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. (A)   (4) Pertaining to the simultaneity of two or more processes. (A)   (5) Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. (A)   (6) Contrast with *serial*.

parallel sessions. Two or more concurrently active sessions between the same two network accessible units (NAUs) using different pairs of network addresses or local-form session identifiers. Each session can have independent session parameters.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I)   (A)   (2) In Basic CUA architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

password. (1) A value used in authentication or a value used to establish membership in a set of people having specific privileges. (2) A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it. (3) In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it.

path. (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension (REX)*, and *virtual route (VR)*.

path control (PC). The function that routes message units between network accessible units in the network and provides the paths between them. It converts the basic information units (BIUs) from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units containing one or more PIUs with data link control. Path control differs by node type: some nodes (APPN nodes, for example) use locally generated session identifiers for routing, and others (subarea nodes) use network addresses for routing.

path control network. Synonym for *transport network*.

PC. (1) Personal computer. (2) Path control. (3) Personal Communications.

PCSDIAL. A program that automatically dials a host number, permitting a fast connection.

PDF. (1) Printer Definition File — A file determining the transfer of characters and control codes to a printer, as well as the output format. (2) Package Definition File — A text file that contains predefined property settings for a package. (3) Portable Document Format — Adobe's © proprietary document format.

peer. In network architecture, any functional unit that is in the same layer as another entity. (T)

peripheral LU. In SNA, a logical unit in a peripheral node. Contrast with *subarea LU*.

peripheral node. A node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node can be a type 1, 2.0, or 2.1 node connected to a subarea boundary node.

peripheral PU. In SNA, a physical unit in a peripheral node. Contrast with *subarea PU*.

personal computer (PC). (1) A microcomputer primarily intended for stand-alone use by an individual. (T) (2) A desktop, floor-standing, or portable microcomputer that usually consists of a system unit, a display monitor, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily for stand-alone operation but may be connected to mainframes or networks.

physical circuit. A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

physical connection. (1) A connection that establishes an electrical circuit. (2) A point-to-point or multipoint connection. (3) Synonymous with *connection*.

physical unit (PU). (1) The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. (2) See also *peripheral PU* and *subarea PU*.

physical unit (PU) services. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions.

PM. Presentation Manager.

pop. To remove an item from the top of a pushdown list. Contrast with *push*.

POP. Post Office Protocol.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

positive response. In SNA, a response indicating that a request was received and processed. Contrast with *negative response*.

Post Office Protocol (POP). A protocol used for exchanging network mail and accessing mailboxes.

Prepare. A presentation services header that flows as part of commit processing, indicating the partner has begun the first phase of the two-phase commit process.

Presentation Manager (PM). See *IBM OS/2 Presentation Manager*.

presentation space. (1) A conceptual two-dimensional surface in storage on which data for a portion of the display surface is represented. (2) The space that contains the device-independent definition of a picture. (3) In the context of the Emulator High-Level Language Application Programming Interface (EHLLAPI), the area in computer memory that corresponds to the user's screen image. (4) The host display space for the window. It is the entire window of data, which can exceed the boundaries of a display screen's viewing area.

presentation space ID (PSID). In Communications Manager/2, synonym for *short name*.

primitive. Synonym for *service primitive*. (T)

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

procedure. (1) In a programming language, a block, with or without formal parameters, whose execution is invoked by means of a procedure call. (I)   (2) The description of the course of action taken for the solution of a problem. (A)

process. (1) To perform operations on data in a process. (I)   (A)   (2) In data processing, the course of events that occurs during the execution of all or part of a program. (T)   (3) A course of the events defined by its purpose or by its effect, achieved under given conditions. (4) Any operation or combination of operations on data. (5) A function being performed or waiting to be performed.

processor. In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

product-set identification (PSID). (1) In SNA, a technique for identifying the hardware and software products that implement a network component. (2) A management services common subvector that transports the information described in definition (1).

profile. Data that describes the significant characteristics of a user, a group of users, or one or more computer resources.

program. (1) A sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to prepare the program for execution, as well as to execute it. (I)   (2) In programming languages, a logical assembly of one or more interrelated modules. (I)   (3) To design, write, and test computer programs. (I)   (A)

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I)   (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T)   (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

PSID. Presentation space ID.

PTF. Program temporary fix.

PU. Physical unit.

push. To add an item to the top of a pushdown list. Contrast with *pop*.

push button. A button, labeled with text, graphics, or both, that represents an action that will be initiated when a user selects it.

PUT. Program update tape.

Q

queue. (1) A list constructed and maintained so that the next data element to be retrieved is the one stored first. (T) (2) A line or list of items waiting to be processed; for example, work to be performed or messages to be displayed. (3) To arrange in or form a queue.

R

radio button. A circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which the user can select one. The circle becomes partially filled when a choice is selected.

Rapid Transport Protocol (RTP). A connection-oriented, full-duplex transport protocol for carrying session traffic over High-Performance Routing (HPR) routes. See also *automatic network routing (ANR)* and *Rapid Transport Protocol (RTP) connection*.

Rapid Transport Protocol (RTP) connection. A connection between two High-Performance Routing (HPR) nodes that may traverse one or more intermediate HPR nodes and links. The connection endpoints provide error recovery and adaptive rate-based flow control for the connection traffic, and nondisruptive switching of the underlying physical path in the case of route outage. The intermediate HPR nodes minimize their routing overhead using automatic network routing (ANR) protocols, which rely on header information to permit efficient source routing and prioritized transmission along the RTP connection.

read-only memory (ROM). Memory in which stored data cannot be modified by the user except under special conditions.

received line signal detector (RLSD). In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that it is receiving a signal from the remote data circuit-terminating equipment (DCE). Synonymous with *carrier detect* and *data carrier detect (DCD)*.

receive pacing. In SNA, the pacing of message units that a component is receiving. Contrast with *send pacing*.

record. A set of data treated as a unit. (T)

release. (1) A distribution of a new product or new function and APAR fixes for an existing product. Normally, programming support for the prior release is discontinued after some specified period of time following availability of a new release. The first version of a product is announced as Release 1, Modification Level 0. (2) In VTAM, to relinquish control of resources (communication controllers or physical units). See also *resource takeover*. Contrast with *acquire*.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

request. A message unit that signals initiation of a particular action or protocol. For example, Initiate-Self is a request for activation of an LU-LU session.

requester. A computer that accesses shared network resources through a server. Synonym for *client*.

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

resource. Any facility of a computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

resource registration. The process of identifying names of resources, such as LUs, to a network node server or a central directory server.

resource takeover. In VTAM, an action initiated by a network operator to transfer control of resources from one domain to another without breaking the connections or disrupting existing LU-LU sessions on the connection. See also *acquire* and *release*.

resource types. In the NetView program, a concept to describe the organization of panels. Resource types are defined as central processing unit, channel, control unit, and I/O device for one category; and communication controller, adapter, link, cluster controller, and terminal for another category. Resource types are combined with data types and display types to describe display organization. See also *data types* and *display types*.

response. (1) In data communication, a reply represented in the control field of a response frame. It advises the primary or combined station of the action taken by the secondary or other combined station to one or more commands. (2) See also *command*.

return code. (1) A code used to influence the execution of succeeding instructions. (A)   (2) A value returned to a program to indicate the results of an operation requested by that program.

REX. Route extension.

ring. See *ring network*.

ring network. A network configuration in which devices are connected by unidirectional transmission links to form a closed path.

RLSD. Received line signal detector.

ROM. Read-only memory. (A)

route. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *explicit route (ER)*, *path*, and *virtual route (VR)*.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*.. (5) Contrast with *bridge*.

routine. A program, or part of a program, that may have some general or frequent use. (T)

routing. (1) The process of determining the path to be used for transmission of a message over a network. (T)   (2) The assignment of the path by which a message is to reach its destination. (3) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

RouTing update Protocol (RTP). The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes. See also *Internet Control Protocol (ICP)*.

RTP. Rapid Transport Protocol..

RTP connection. See *Rapid Transport Protocol (RTP) connection*..

RU chain. In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated by means of control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

S

SAP. (1) Service access point. (2) Service Advertising Protocol.

SBCS. Single-byte character set.

scroll. To move a display image vertically or horizontally to view data that otherwise cannot be observed within the boundaries of the display screen.

scroll bar. A window component that shows a user that more information is available in a particular direction and can be scrolled into view. Scroll bars can be either horizontal or vertical.

SCS. SNA character string.

SDLC. Synchronous Data Link Control.

segment. (1) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data. (3) Synonym for *BIU segment*. (4) See also *link connection segment*.

select. To explicitly identify one or more objects to which a subsequent choice will apply.

selection. The process of explicitly identifying one or more objects to which a subsequent choice will apply.

send pacing. In SNA, the pacing of message units that a component is sending. Contrast with *receive pacing*.

serial. (1) Pertaining to a process in which all events occur one after the other; for example, serial transmission of the bits of a character according to V24 CCITT protocol. (T) (2) Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (A) (3) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (A) (4) Contrast with *parallel*.

server. (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T) (2) In a network, a data station that provides facilities to other stations; for example, a file server, a print server, a mail server. (A) (3) In the AIX operating system, an application program that usually runs in the background and is controlled by the system program controller. (4) In the Enhanced X-Windows Toolkit, a program that provides the basic windowing mechanism. It handles interprocess communication (IPC) connections from clients, demultiplexes graphics requests onto screens, and multiplexes input back to clients.

Server-Requester Programming Interface (SRPI). An IBM application programming interface (API) used by requester and server programs to communicate with the personal computer or host routers.

service access point (SAP). (1) In Open Systems Interconnection (OSI) architecture, the point at which the services of a layer are provided by an entity of that layer to an entity of the next higher layer. (T) (2) A logical point made available by an adapter where information can be received and transmitted. A single service access point can have many links terminating in it. (3) The gateway address of the controller. A SAP provides a point to link the controller to the host system.

Service Advertising Protocol (SAP). In Internetwork Packet Exchange (IPX), a protocol that provides the following:
- A mechanism that allows IPX servers on an internet to advertise their services by name and type. Servers using this protocol have their name, service type, and IP address recorded in all file servers running NetWare.
- A mechanism that allows a workstation to broadcast a query to discover the identities of all servers of all types, all servers of a specific type, or the nearest server of a specific type.
- A mechanism that allows a workstation to query any file server running NetWare to discover the names and addresses of all servers of a specific type.

service key. In Communications Manager/2, one of two access passwords (the other is the master key) used to unlock the keylock for Communications Manager/2.

service point (SP). An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry

point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

Service Point Application (SPA) Router. In Communications Manager/2, a program that receives commands issued from the NetView program and sends the commands to an application program to be processed on an OS/2 workstation.

service primitive. (1) In Open Systems Interconnection architecture, an abstract description of an interaction between a service user and a service provider.

Note: A service user is usually an entity. A service provider, in a given layer, usually comprises entities of that layer and a lower layer service (except at the Physical Layer). Therefore, a service primitive is also an abstract description of an interaction between two adjacent entities. (T)
(2) Synonymous with *primitive*.

session. (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (T)   (2) A logical connection between two network accessible units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) accompanying any transmissions exchanged during the session. (3) A logical connection between a server and a requester that was initiated by a successful request for a shared resource. See also *host session* and *DOS session*.

session activation request. In SNA, a request that activates a session between two network accessible units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Contrast with *session deactivation request*.

session control (SC). In SNA, either of the following:
- One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification.
- A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

session data. Data about a session, collected by the NetView program, that consists of session awareness data, session trace data, and session response time data.

session deactivation request. In SNA, a request that deactivates a session between two network accessible units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

session ID. An alphabetic ID (*a* through *h*) assigned by Personal Communications to each session or screen. This applies to all types of host sessions and is used in file transfers. See also *short name*.

session-level pacing. (1) A flow control technique that permits a receiving half-session or session connector to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. (2) See also *adaptive session-level pacing*, *fixed session-level pacing*, and *virtual route (VR) pacing*.

session parameters. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network accessible units. See also *logon mode*.

session services (SS). One of the types of network services in the control point (CP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that a control point (an ENCP, NNCP, or SSCP) assist with initiating or terminating sessions between logical units. Assistance with session termination is needed only by SSCP-dependent LUs. See *configuration services*, *maintenance services*, and *management services*.

shared. Pertaining to the availability of a resource for more than one use at the same time.

shift-in character (SI). A code extension character used to terminate a sequence that has been introduced by the shift-out character to make effective the graphic characters of the standard character set. (I)   (A)

Shift Lock. The personal computer keyboard mode entered after the Shift Lock key is pressed. In this mode, the keyboard performs the function printed on the upper half of the keytop. Press the Shift key to cancel the Shift Lock mode.

shift-out character (SO). A code extension character that substitutes for the graphic characters of the standard character set an alternative set of graphic characters upon which an agreement has been arrived at or that has been designated using code extension procedures. (I) (A)

short name. (1) In Personal Communications, a character displayed in column 7 of the operator information area that shows the session ID. See also *session ID* and *operator information area*. (2) In Communications Manager/2, the one-letter name (A through Z) of the presentation space or emulation session. Synonymous with *presentation space ID (PSID)* and *short-session ID*.

short-session ID. In Communications Manager/2, synonym for *short name*.

SI. The shift-in character. (I) (A)

simplex. Pertaining to printing on only one side of a sheet of paper. Contrast with *duplex* and *tumble*.

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set (DBCS)*.

SNA. Systems Network Architecture.

SNA character string (SCS). In SNA, a character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network accessible units (NAUs), boundary function, gateway function, and intermediate session routing function components; and the transport network.

SO. The shift-out character. (I) (A)

socket. An endpoint for communication between processes or application programs.

SP. Service point.

spin button. A component used to display, in sequence, a ring of related but mutually exclusive choices. A user can accept the value displayed in the entry field or can type a valid choice into the entry field.

SRPI. Server-Requester Programming Interface.

SS. Session services.

SSCP. System services control point.

SSCP-dependent LU. An LU that requires assistance from a system services control point (SSCP) in order to initiate an LU-LU session. It requires an SSCP-LU session.

SSCP-LU session. In SNA, a session between a system services control point (SSCP) and a logical unit (LU). The session enables the LU to request the SSCP to help initiate LU-LU sessions.

stage. A program that processes messages in a NetView pipeline. Stages send messages to each other serially.

statement. A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

static. (1) In programming languages, pertaining to properties that can be established before execution of a program; for example, the length of a fixed length variable is static. (I) (2) Pertaining to an operation that occurs at a predetermined or fixed time. (3) Contrast with *dynamic*.

station. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

status. The condition or state of hardware or software, usually represented by a status code.

storage. (1) A functional unit into which data can be placed, in which they can be retained and from which they can be retrieved. (T) (2) The action of placing data into a storage device. (I) (A) (3) A storage device. (A)

stream. (1) To send data from one device to another. (2) See *data stream*.

subarea. A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea LU. In SNA, a logical unit in a subarea node. Contrast with *peripheral LU*.

subarea node (SN). A node that uses network addresses for routing and maintains routing tables that reflect the configuration of the network. Subarea nodes can provide gateway function to connect multiple subarea networks, intermediate routing function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes can be subarea nodes.

subarea PU. In SNA, a physical unit in a subarea node. Contrast with *peripheral PU*.

subdirectory. A directory contained within another directory in a file system hierarchy.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

subsystem management. In Communications Manager/2, a group of advanced functions for diagnosis and problem resolution. These functions are typically used by system administrators and application programmers.

synchronization point. Synonym for *sync point*.

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I)

sync point. An intermediate or end point during processing of a transaction at which an update or modification to one or more of the transaction's protected resources is logically complete and error free. Synonymous with *synchronization point*.

system. In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A)

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

system services control point (SSCP) domain. The system services control point, the physical units (PUs), the logical units (LUs), the links, the link stations, and all the resources that the SSCP has the ability to control by means of activation and deactivation requests.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

table. A repository for data that NetDA/2 uses to design a network. Each table contains information related to the network.

task. In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (I) (A)

Task-Related User Exit (TRUE). A CICS module used for invoking resource managers that are outside of CICS. A TRUE is one of the components of CICS sockets.

TCP. Transmission Control Protocol..

TCP/IP. Transmission Control Protocol/Internet Protocol.

template. See *keyboard template*.

TERMINATE. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

timeout. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

Time Sharing Option (TSO). An option of the MVS operating system that provides interactive time sharing from remote terminals.

time stamp. (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the history of the object. (3) In query, the identification of the day and time when a query report was created that query automatically provides on each report.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

token ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

topology. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

TP. Transaction program.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A) (2) For data links, a record of the frames and bytes transmitted or received.

transaction program (TP). A program that processes transactions in an SNA network. There are two kinds of transaction programs: application transaction programs and service transaction programs. See also *conversation*.

translation table. A table used to replace one or more characters with alternative characters; for example, to translate characters representing a virtual address to those representing a real address, characters representing an event to those representing a procedure call, characters of a national character set to those of another national language, or characters representing a relocated address to those representing an absolute address.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol..

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transport network. (1) The part of the SNA network that includes the data link control and path control layers. Synonymous with *path control network*. (2) An implementation of transport networking. Examples are parts of SNA, TCP/IP, OSI, IPX, NetBIOS, DECnet, and Appletalk.

transport protocol. A specification of the rules governing the exchange of information between components of a transport network.

TRUE. Task-Related User Exit.

TSO. Time Sharing Option.

tumble. Pertaining to printing on both sides of a sheet of paper such that the bottom of the printed image on one side of the paper is juxtaposed against the *top* of the printed image on the other side of the paper. Contrast with *duplex* and *simplex*.

Type 1 frame. In the IEEE 802.2 LAN standard, a frame sent without the establishment of a link-level connection. It is used for IP and IPX traffic as well as NetBIOS datagrams. Contrast with *Type 2 frame*.

Type 2 frame. In the IEEE 802.2 LAN standard, a frame sent over an established link-level connection. It is used for NetBIOS and SNA links. Contrast with *Type 1 frame*.

T1. In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-kbps channels. The European version (E1) transmits 2.048 Mbps. The Japanese version (J1) transmits 1.544 Mbps.

U

UNBIND. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

upload. (1) To transfer programs or data from a connected device, typically a personal computer, to a computer with greater resources. (T) (2) To transfer data from a device, such as a workstation or a microcomputer, to a computer. Contrast with *download*.

upstream. In the direction of data flow from the user to the host. Contrast with *downstream*.

user. (1) Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T) (2) Anyone who requires the services of a computing system.

UTC. Coordinated universal time.

V

value. (1) A specific occurrence of an attribute; for example, "blue" for the attribute "color." (T) (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

variable. (1) In programming languages, a language object that may take different values, one at a time. The values of a variable are usually restricted to a certain data type. (I) (2) A quantity that can assume any of a given set of values. (A) (3) A name used to represent a data item whose value can be changed while the program is running.

vector. The MAC frame information field.

verb. See *LU 6.2 verb*.

version. A separately licensed program that usually has significant new code or new functions.

VGA. Video graphics adapter.

video graphics adapter (VGA). A computer adapter that provides high-resolution graphics and a total of 256 colors. See also *enhanced graphics adapter (EGA)*.

virtual circuit. (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

virtual machine (VM). (1) A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system. (T) (2) In VM/ESA, the virtual processors, virtual storage, virtual devices, and virtual channel subsystem allocated to a single user. A virtual machine also includes any expanded storage dedicated to it..

Virtual Machine/Conversational Monitoring System (VM/CMS). A time sharing system control program that manages the resources of an IBM System/370 or System/390 such that multiple remote terminal users have a functioning simulation of a computing system at their disposal. The Conversational Monitoring System provides general time sharing, development, and problem-solving facilities.

Virtual Machine/Enterprise Systems Architecture (VM/ESA). An IBM licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a real machine.

virtual route (VR). (1) In SNA, either (a) a logical connection between two subarea nodes that is physically realized as a particular explicit route or (b) a logical connection that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units (PIUs). (2) Contrast with *explicit route (ER)*. See also *path* and *route extension (REX)*.

virtual route (VR) pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *session-level pacing*.

virtual storage. The storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (I) (A)

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VM. Virtual machine.

VM/CMS. Virtual Machine/Conversational Monitoring System.

VM/ESA. Virtual Machine/Enterprise Systems Architecture.

VR. Virtual route.

VS. Virtual storage.

VTAM. (1) Virtual Telecommunications Access Method. (2) Synonymous with *ACF/VTAM*.

V.25. In data communication, a specification of the CCITT that defines the automatic answering equipment and parallel automatic calling equipment on the General Switched Telephone Network, including procedures for disabling of echo controlled devices for both manually and automatically established calls.

V.25 bis. A procedure defined by the CCITT that allows call establishment and data transfer to take place over the same link. The support eliminates the need for two physical lines or ports when automatic calling units (ACUs) are employed in a switched connection.

W

WAN. Wide area network.

weight. For route selection purposes, the degree to which resources (such as nodes and transmission groups) meet the criteria specified by a particular class of service. In APPN route selection, routes of minimum weight are chosen..

wide area network (WAN). (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communication network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks. (3) Contrast with *local area network (LAN)* and *metropolitan area network (MAN)*.

window. (1) A portion of a display surface in which display images pertaining to a particular application can be presented. Different applications can be displayed simultaneously in different windows. (A) (2) An area with visible boundaries that presents a view of an object or with which a user conducts a dialog with a computer system. (3) In data communication, the number of data packets a data terminal equipment (DTE) or data circuit-terminating equipment (DCE) can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets. (4) See *pacing window*.

window size. The specified number of frames of information that can be sent before receiving an acknowledgment response.

WinSock application programming interface (API). A socket-style transport interface developed for the Windows family of operating systems.

workstation. (1) A functional unit at which a user works. A workstation often has some processing capability. (T) (2) One or more programmable or nonprogrammable devices that allow a user to do work. (3) A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

Z

Z time. Abbreviation for *Zulu time*. Synonym for *coordinated universal time (UTC)*.

Zulu time (Z). Synonym for *coordinated universal time (UTC)*.

Special Characters

Numerics

A

B

C

D

E

F

Readers' Comments — We'd Like to Hear from You

Personal Communications Version 5.0
for Windows® 95, Windows 98,
Windows NT®, and Windows 2000
Reference Volume I

Publication No. SC31-8680-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?   ☐ Yes   ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____   Address _____

Company or Organization _____

Phone No. _____

IBM®

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

IBM Corporation
Information Development
Department CGMD / Bldg 500
P.O. Box 12195
Research Triangle Park, NC
 27709-9990

Fold and Tape

Please do not staple

Fold and Tape

IBM ®