Personal Communications for Windows, Version 5.7

5250 Emulator User’s Reference

SC31-8837-02

Personal Communications for Windows, Version 5.7

5250 Emulator User’s Reference

SC31-8837-02

Note

Before using this information and the product it supports, read the information in "Notices", on page 129.

Third Edition (September 2003)

This edition applies to Version 5.7 of Personal Communications (program number: 5639–I70) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 6. Printing 107

Chapter 7. Displaying Grid Lines (DBCS Only) 119

Part 3. Appendixes 127

Appendix. Notices 129

Index 133

Figures

Tables

About This Book

IBM® Personal Communications for Windows® reference books are comprised of four volumes: a 3270 Emulator User's Reference, a 5250 Emulator User's Reference, a VT Emulator User's Reference and an Administrator's Guide and Reference. These volumes provide information for using IBM Personal Communications iSeries™ for Windows operating systems (hereafter called PC400) and IBM Personal Communications for Windows operating systems. In this book, *Windows* refers to Windows 95, Windows 98, Windows NT®, Windows Me, Windows 2000, and Windows XP. When information is applies only to a specific operating systems, this is 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 uses Personal Communications on a workstation to access hosts using 5250 terminal emulation.

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.

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 can be typed in either uppercase or lowercase, unless otherwise indicated.

Part 1. General Information

Chapter 1. Personal Communications Highlights

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:

- **zSeries™ Connections**

  LAN LAN via IEEE 802.2
  Communications Server for Windows
  Telnet3270
  3270 via iSeries
  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

  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 iSeries
Dependent Logical Unit Requester (DLUR)
SDLC Synchronous Data Link Control
3270 via iSeries (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 iSeries (passthru)
APPC 3270 via SDLC
Dependent Logical Unit Requester (DLUR)
X.25 Qualified Logical Link Control
X.25 3270 via iSeries
APPC 3270 via X.25
X.25 DLUR
Communications Server Client
Communications Server
IBM-EEDLC
LU (0, 1, 2, 3) via DLUR
APPC 3270
OEM LU (0, 1, 2, 3)
3270 via iSeries (passthru)
APPC 3270
Dependent LU Requester (DLUR)

- **iSeries 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.
- **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 iSeries Advanced Print Support Utility
- **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)

Chapter 2. 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 *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, go to the following Internet address:

  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, 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 clicking 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, click **Log Viewer**; or, from an active session, click **Actions** → **Launch** → **Log Viewer**.
2. From the list of logged messages, double-click a message to display the message text.

For more information about log viewer functions, refer to *Administrator's Guide and Reference*.

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:

1. From the **Administrative and PD Aids** folder, click **Trace Facility**; or, from an active session, click **Actions** → **Launch** → **Trace Facility**. 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 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 default path for the formatted trace file, the Information Bundler will not find the trace information. Copy the trace files to the system-class application data directory.
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 self-extracting .EXE file. This .EXE file is sent to support personnel using 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, click **Actions → Launch → Information Bundler**.
2. A .EXE file containing system and Personal Communications information is created in the Personal Communications system-class application data directory. By default this file is called X12345.EXE. Refer to the installation documentation for the location of the system-class application data directory for each Windows operating system.

Internet Service

The Internet Service utility enables you to send the .EXE 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, click **Actions** → **Launch** → **Internet Service**.

   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. Type your e–mail address in the field provided.
4. In the Problem Determination Filename field, type the file name and path of the .EXE file created with the Information Bundler. This file is located in the Personal Communications installation directory.
5. In the PMR Number field, type 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.

Part 2. Personal Communications iSeries

Chapter 3. Considerations for Using PC400 Sessions

This chapter contains hints and tips for using PC400 sessions. Supplementary information other than the items described in this book may be included in the Readme HTML file in the Personal Communications directory.

Power Management

Personal Communications complies with Windows 2000 Power Management requirements for handling sleep events. This support minimizes session interruptions due to network disconnections caused by sleep on Windows 2000 and subsequent versions.

Note: In this context, the term *sleep* means that the system is on standby or is in hibernation. To applications such as Personal Communications, standby and hibernation are the same.

The benefits of this power management system include the following:

- Power consumption is reduced using the Advanced Configuration and Power Interface (ACPI). The system is able to enter a lower power state (or sleep mode) that appears to be off but is still powered enough to enable the system to power up (or wake) to handle timed events or device-related needs such as receiving a fax.
- The PC is instantly available to the user because it can rapidly return from a low power state to a fully-functional state.
- Customers can rely on their PCs to power down and up in a way that is easily understood and predictable.

The following Personal Communications components are affected by this Power Management arrangement:

- Emulator sessions
- Transfers that utilize an emulator session

Sleep Permission

Before entering a sleep state (stand by or hibernate), Windows 2000 normally requests permission from the applications that are running. When one or more emulator sessions are connected and Windows signals that the user is available for interaction, Personal Communications asks the user to grant or deny sleep permission. If the user grants permission, Personal Communications logs the event and then notifies Windows. When user interaction is not possible, sleep permission is denied without permission from the user.

When Personal Communications is not in the connected state, Windows 2000 may automatically sleep, without prompting the user for permission.

You can specify a setting in the User Preference Manager that allows the system to standby or hibernate without prompting. In default mode (unchecked), if there is at least one connected session, you will be prompted to allow the system to standby or hibernate. If there are no connected sessions, Personal Communications allows the system to standby or hibernate without prompting. Refer to *Quick Beginnings* for information on using the User Preference Manager.

Critical Sleep

When Windows 2000 resumes after an emergency suspension, Personal Communications might display and log a warning message.

Scroll Bar

When you click **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 iSeries on Telnet 5250

Executing a Testrequest function when connected to an iSeries might cause the session to be disconnected. If you experience this problem, make sure that OS/400 APAR MA15053 has been applied on the iSeries.

iSeries Host Print Problem

If you attempt to use the host print function (mapped to CTRL-Pause by default) while viewing a spooled print file, the ends of some of the lines might be 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 using 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. Use the Set Initial Condition (SIC) command 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.

Default Key Function Assignments

This section lists the functions assigned, by default, to each key on your keyboard.

For more information about each function, refer to the **Keyboard** choice on the **Help** menu.

You can change the default key assignments to the following default function tables, by selecting **Keyboard Setup** from the **Assist** menu.

When the Keyboard Setup window appears, select one of the following choices:
- 3270 for a 3270 keyboard layout
- 5250 for a 5250 keyboard layout
- 3270+5250 for a combined keyboard layout
- VT for a DEC VT220 keyboard layout

Personal Communications includes two .KMP keyboard map files that map the standard Win32 hotkeys for Cut, Copy, and Paste to Ctrl+X, Ctrl+C and Ctrl+V, respectively. The key values for the following functions have also been changed: Enter and New Line. You can utilize these keyboard map files, or can add these key values to an existing map file.

For 5250 sessions, the .KMP file provided is pcswinkb5.kmp. The remapping is given in the following table.

Table 1. Win32 Keyboard Map Functions for a 5250 Layout

Function of Key	Key
Edit Cut	Ctrl+X
Edit Copy	Ctrl+C
Edit Paste	Ctrl+V
Enter	Enter
New Line	Right Ctrl

The PF7 and PF8 remain mapped to Roll Up and Roll Down, respectively.

Setting the 5250 Keyboard Layout Default

To make the 5250 keyboard layout defaults available, do the following:

1. Select **Preferences → Keyboard** from the **Edit** menu. The Keyboard dialog box is displayed.

2. Select the **IBM Default** radio button next to Current Keyboard.
3. Click **OK**.

Default Key Functions for a 5250 Layout

Table 2 shows the default key functions for iSeries. The key used is the same for all the supported keyboard types.

Notes:

1. If you use iSeries from the combined package, see “Default Key Functions for the Combined Package” on page 20.
2. The default key functions for a 5250 layout are not available by default. To make these functions available, perform the procedures in “Setting the 5250 Keyboard Layout Default” on page 17.

Table 2. Default Key Functions for a 5250 Layout

Function of Key	**Key**
Alternate Cursor	Ctrl+F11
Attention	Esc
Backspace	← (Backspace)
Backtab	Shift+→|
Backtab Word	Alt+←
Begin Bold*	Ctrl+B
Begin of line*	Ctrl+4(pad)
Begin Underscore*	Ctrl+U
Bottom of Page*	Ctrl+2(pad)
Carrier Return	Ctrl+Enter or Ctrl+− (pad) or Ctrl++ (pad)
Center Text*	Ctrl+C
Clear	Pause
Cursor Blink	Ctrl+F10
Cursor Down	↓ or 2(pad)
Cursor Left	← or 4(pad)
Cursor Right	→ or 6(pad)
Cursor Up	↑ or 8(pad)
Delete Character	Delete or .(pad)
Delete Word	Ctrl+Delete or Ctrl+.(pad)
Display Text Code	Alt+Insert
Dup	Shift+Insert
Edit Copy	Ctrl+Insert
Edit Cut	Shift+Delete
Edit Paste	Shift+PageDown or Ctrl+Shift+Insert
Edit Undo	Alt+← (Backspace)
End Bold/Underscore*	Ctrl+J

Table 2. Default Key Functions for a 5250 Layout (continued)

Function of Key	Key
End of line*	Ctrl+6(pad)
End of page*	Ctrl+P
Enter/Control	Right Ctrl
Erase EOF	End or 1(pad)
Erase Input	Alt+End
Fast Cursor Down	Alt+↓ or Alt+2(pad)
Fast Cursor Up	Alt+↑ or Alt+8(pad)
Field Exit	Enter(pad) or ↵ (Enter)
Field Mark	Shift+Home
Field Minus (–)	–(pad)
Field Plus (+)	+ (pad)
Half Index Down*	Ctrl+H
Half Index Up*	Ctrl+Y
Help	Alt+F1
Home	Home or 7(pad)
Host Print	Ctrl+Pause
Insert	Insert or 0 (pad)
Insert Symbol*	Ctrl+A
Jump Next	Alt+PageUp
Mark Down	Shift+↓
Mark Left	Shift+←
Mark Right	Shift+→
Mark Up	Shift+↑
Move Mark Down	Ctrl+↓
Move Mark Left	Ctrl+←
Move Mark Right	Ctrl+→
Move Mark Up	Ctrl+↑
Next Column*	Ctrl+D
Next Stop*	Ctrl+N
Pause	Ctrl+F7
PF1 to PF12	F1 to F12
PF13 to PF24	Shift+F1 to F12
Play	Ctrl+F6
Quit	Alt+Left Ctrl
Record	Ctrl+F5
Required Backspace	Ctrl+← (Backspace)
Required Space*	Ctrl+Space

Table 2. Default Key Functions for a 5250 Layout (continued)

Function of Key	Key
Required Tab*	Ctrl+→|
Reset/Control	Left Ctrl
Roll Down	9(pad) or PageUp
Roll Up	3(pad) or PageDown
Rule	Ctrl+Home
Stop Code*	Ctrl+S
System Request	Shift+Esc
Tab Field	→|
Tab Word	Alt+→
Test Request	Alt+Pause
Top of Page*	Ctrl+8(pad)
Word Underscore*	Ctrl+W
Note: **(***pad***)** indicates a key on the numeric keypad. * indicates a Text Assist Key (SBCS only).	

Default Key Functions for the Combined Package

Table 3 shows the default key functions for the combined package. The key used is the same for all the supported keyboard types.

When you use the 3270+5250 keyboard layout, the key definitions for the 3270 and 5250 layouts are combined with those listed here.

Table 3. Default Key Functions for the Combined Package

Function of Key	Key
Change Screen	Not assigned
Character Advance	Shift+BackSpace
Help	Not assigned
Host Print	Not assigned
PA3	Not assigned
Roll Down	PageUp
Roll Up	PageDown
Printer Setup	Not assigned

Keyboard Setup

You can use Keyboard Setup to modify the function defined for each key on the keyboard, except some reserved keys.

You can define the following functions for the keys:
- Performing a key function
- Playing a macro
- Entering characters

Keyboard File

When you specify a key, you can save the new keyboard layout in a file (.KMP). If you create two or more keyboard files, you can alternate between them as required.

To assign a function to a key on the keyboard:

1. Click **Preferences → Keyboard** from the **Edit** menu
2. When the Keyboard Setup window appears, select **Customize**.

   Note: Select **Spain** from the **Language** menu during keyboard setup if you want Catalan support.
3. Assign the key functions, referring to the online help for detailed instructions.
4. Save your changes and exit the Customize Keyboard window.
5. Click **OK** after completing the setup.

You can reset either the entire keyboard or specific keys to defaults:

- To reset the entire keyboard, set the current keyboard to **IBM Default** in the Keyboard Setup window.
- To reset specific keys, select a key in the Customize Keyboard window and then select **Default** from the Current Actions for Selected Key box.

Note: There are seven keys that you cannot redefine and they are gray or dimmed in Keyboard Setup; these keys are as follows: Alt, AltGr, Print Screen, Scroll Lock, CapsLock, NumLock, and Shift.

Win32 Cut, Copy, and Paste Hotkeys

Personal Communications includes two .KMP keyboard map files that maps the standard Win32 hotkeys for Cut, Copy, and Paste to Ctrl+X, Ctrl+C and Ctrl+V, respectively. The key values Enter and New Line have also been changed. You can use the new keyboard map files, or can add these new key values to an existing map file.

For 5250 sessions, the default .KMP file is pcswinkb5.kmp. The remapping is given in Table 4.

Table 4. New Keyboard Map Functions for a 5250 Layout

Function of Key	**Key**
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Enter	Enter
New Line	Right Ctrl

The PF7 and PF8 remains mapped to Roll Up and Roll Down, respectively.

Usage Notes for Sessions in OLE Documents

Changing Fonts

If you are using an In-Place embedded session, then changing the font face name, switching between automatic sizing and fixed size, or changing the size for a fixed size font can result in an incorrect display on the screen. To correct the display, adjust the size of the session object window slightly.

Initial Selection of Font

The initial font selection for a embedded or linked session is determined by its Session ID (a letter A through Z) just like a regular session. Therefore, the initial font may change if other sessions are already active. Further, to prevent In-Place embedded sessions from having adverse effects on subsequent sessions, font changes made during use of In-Place embedded sessions are not saved.

WordPad

Personal Communications session objects created in Microsoft WordPad documents by the drag-and-drop method cannot be used after the document has been saved and closed. This limitation might be fixed by some future version of WordPad. For the present, you should only create Personal Communications objects in WordPad by using the Insert-Object menu item.

Some versions of Microsoft Word and Microsoft WordPad incorrectly save the state of embedded objects that are displayed as icons. If you open a document that contains a Personal Communications session object that was created to display as an icon, and the object is activated, then it may activate In-Place instead of as a separate window. This problem might be fixed in future versions of these containers.

WordPro

If you attempt to open a link to a Personal Communications session in a Lotus® WordPro document, current versions of WordPro may give unpredictable results. This limitation may be fixed by some future version of WordPro. For the present, you should only use embedded Personal Communications objects in WordPro documents. You can use the “Display as Icon” option if a separate window is desired.

Updating Linked Files

Files that are linked into Word 97 or Excel 97 do not update automatically. You must manually save the linked file before your edits are reflected in the container window.

Inactivity Timeout for Communication Links

The Inactivity Timeout automatically disconnects a link after it has been idle for a specified length of time. Its purpose is to avoid excessive charges on dial-up links, such as switched-line connections with SNA/A, Asynchronous IIN, Hayes AutoSync, or SDLC. Inactivity Timeout is not recommended for other types of connections.

To enable Inactivity Timeout, add the following statement to the PU section of your workstation profile (.WS file):

```
[PU]
  InactiveTimeout=xx
```

The value *xx*, ranging from 1 to 999, is the number of minutes a link remains connected when there is no activity over it. The default value, InactiveTimeout=0 disables Inactivity Timeout.

Note: The Inactivity Timeout function monitors only attention keys (that is, the Enter, Clear, and PFx keys). We recommend that you set a comparably longer value for *xx* if, for example, you expect to key in large amounts of data on the screen before pressing the Enter key.

Considerations

The following are environmental considerations for Personal Communications.

Virtual Memory

If you receive a message stating that the system is low on virtual memory, increase the virtual memory paging file. If you get this message as you are trying to open new host sessions or starting some Personal Communications function such as File Transfer, the virtual memory settings should be increased. Refer the operating system documentation for instructions on how to increase the size of the paging file.

Emulator Session Icons

Emulator session icons that were not migrated during installation of Personal Communications Version 5.7 will not function correctly if they were not created in the application data directory specified during installation of Version 5.7. The icons can be updated by using the **File → Import** option from the Session Manager. This option will not copy the prior icons to the application data directory specified during installation; the icons must be moved manually

Disabling CDRA Tables

Personal Communications uses the standard IBM CDRA translation tables when converting between ASCII and EBCDIC. From some code page pairs, the standard tables differ from those that Personal Communications has used in the past. For code pages that were supported in prior releases, you can configure Personal Communications to use the old tables. A switch is available in PCSWIN.INI to disable the use of CDRA tables. This switch is located in the Translate section and is named UseOnlyPCOMM. This switch takes a binary value and is defaulted to FALSE (except for Japan, Korea, Taiwan, and PRC where it defaults to TRUE). For the code pages that are new to Version 5.7, you must use the standard tables. Setting the switch will apply to all sessions, as well as data transfer and command line file transfer.

Chapter 4. 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 iSeries system (with or without a display session).

Data Transfer Function Overview

The PC400 data transfer function transfers data to or from the iSeries system and a workstation.

PC400 can transfer data between the host and workstation. The data transfer function can be used by clicking the **Data Transfer** icon.

Notes:

1. For Personal Communications Version 5.7, the Data Transfer application is automatically invoked from a 5250 session when you click **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 "File Transfer for PC400" on page 104. The main differences are listed in the following table.

Table 5. Data Transfer Summary

Type of Transfer	Products required on an iSeries system	Access Method	Sending and receiving unit	Type of connection to an iSeries system
File Transfer	Personal Communications Tools (PCT/400 see Chapter 5, "Transferring Files", on page 95)	• Transfer menu in the session window • EHLLAPI application that invokes File Transfer • DDE application that invokes File Transfer • Playing a macro that invokes File Transfer • 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
[1]OS/400 provides the host transaction program for Data Transfer.				

Transferring Files from an iSeries System to a Workstation

When using a workstation, you can retrieve and use data from the following file types on an iSeries 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.

You can specify the following output destinations:

- Display
- Disk
- Printer

Transferring Files from a Workstation to an iSeries System

The PC→iSeries Transfer function enables the transfer of data from a workstation to an iSeries physical file. Data can be transferred to any of the following destinations:

- Existing members in an existing iSeries physical file
- New members in an existing iSeries physical file
- New members in a new iSeries physical file

Note: Data cannot be transferred from a workstation file to an iSeries logical file.

Transferring Data to Existing Members in an Existing File

Note the following considerations when transferring data from a workstation to an existing iSeries member.

- When data is transferred to an existing member, data in that member is replaced with that transferred from a workstation.
- When iSeries 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 iSeries system (such as when an iSeries master file is updated on a workstation).

  For example, you can transfer only the field subset of an iSeries file by issuing a transfer request from the iSeries system to a workstation. In this case, when returning data from the workstation to the iSeries system, only the subset included in that iSeries file can be transferred. Other fields that had been defined in the iSeries 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 iSeries file and the transferred data must be embedded in the iSeries file by running the iSeries application program. Follow this procedure to control the update processing for an iSeries master file.

  To prevent users from transferring data to a certain iSeries 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 iSeries 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 iSeries file can be transferred from the iSeries system to a workstation by the previous transfer request. When data is returned to the iSeries system, new members can receive only the subset defined in that iSeries 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 iSeries default values.

Transferring Data to New Members in a New File

By using a transfer request from a workstation to the iSeries system, you can transfer data to new members in a new iSeries file. This is one of the safest transfer methods, because data already stored in the iSeries file is not replaced with that transferred from the workstation.

There are two ways of transferring data to new members in a new iSeries 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 iSeries file.

  When an iSeries file and its members are created, the transfer function must access the description of the format of each field to be transferred in the iSeries file. You can get this description, called a field-reference file, from the iSeries file. To create an iSeries file and its members, specify the name of this iSeries 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 iSeries physical source file is created.

Transferring Data to an iSeries Data File and Source File

You can transfer data to the following two types of iSeries physical files.

Physical data file

The members of a physical data file can contain numeric and character data of any iSeries 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 iSeries file is required to ensure correct conversion of the data.

When data is transferred to an existing iSeries file, the file description becomes part of the iSeries file. When data is transferred to a new iSeries file, the file description is included in the iSeries 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 iSeries physical source file:

- To transfer text from the iSeries system to a workstation, specify the name of the source file and members in **FROM**. Specify an asterisk (*) in **SELECT**. This informs the iSeries system that only text is transferred from the source file, with the order number and date fields excluded.
- The iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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.

You can also send data from the workstation to the host system for processing by iSeries application. When a remote user is authorized to access the iSeries system directly, he or she can access the created cost analysis report to compare with their results.

Figure 1 outlines the joining of two files, transferring the information to the workstation, and creating a report.

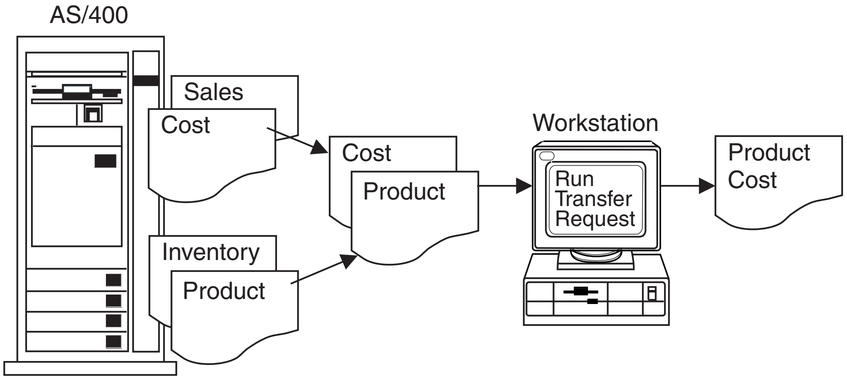


Figure 1. Data Transfer Example

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 iSeries 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 iSeries system is called *data sending*.
- Transferring data from the iSeries 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→iSeries Transfer window, and that for receiving is called the iSeries→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 iSeries system.

The **Data Transfer** icon is registered in the PC400 folder by installing PC400. Double-clicking on this icon displays the iSeries→PC Transfer window (for receiving). This icon includes:

```
\Personal Communications\PCSFT5.EXE
```

The PC→iSeries Transfer window (for sending) appears when the registered contents are changed as follows:

```
\Personal Communications\PCSFT5.EXE
```

The iSeries→PC Transfer and PC→iSeries 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*, you must create transfer information, using the interactive screen window, and then save the information. You can then perform data transfer automatically by specifying the file name in which the data was saved.

For example, if you save transfer information 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries system.

Distributed Data Management (DDM) Files

Distributed data management (DDM) is one of the functions supported by the iSeries system. This function is used to access database files that are stored on remote iSeries systems. To use the transfer function to access these database files, specify a DDM file name as the name of the iSeries file to be transferred. Refer to the *DDM Guide* for details of how to use DDM files.

iSeries Files

The following list provides a simple explanation of the requirements for transferring data between the iSeries system and a workstation.

Library

The iSeries 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 iSeries file.

File

iSeries files that you can manipulate consist of a file description and data stored in the file. PC400 processes an iSeries 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, that 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 iSeries 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 iSeries 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 iSeries system, actual data transfer is done between file members. For example, a certain workstation file can be transferred to the iSeries system. In this case, the file members become new members of a new or existing iSeries file, or substitute for existing members in an existing iSeries 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-iSeries Transfer Request

To create a request for data transfer from a workstation to the iSeries system, do as follows.

1. Using the router session, establish attachment to the iSeries system to which data is to be transferred.
2. Click the **Data Transfer** icon.
3. When the iSeries→PC Transfer window displays, select **Switch to SEND**. The display is switched to the PC→iSeries Transfer window.

   To choose additional settings, select **Advanced**.
4. Specify each item. See “Items to Be Specified” for details.

Items to Be Specified

The following section explains the items that you specify in the PC→iSeries 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 iSeries 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 click **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 click **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 click **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 iSeries physical file that will receive the data to be transferred from the workstation. You can specify either an existing file name or new file name.

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 iSeries library containing the iSeries 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 iSeries system displays a list of all libraries defined in *USRLIBL of the iSeries job library list. You can modify this list by changing the job description. Run a change job description (CHGJOBD) command on the iSeries system.

file-name

This is the name of an iSeries 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 iSeries job library list, specify ***USRLIBL** followed by a slash (/), then select **Browse**. If a slash (/) is not specified after the library name, the iSeries 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 iSeries 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 iSeries 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 iSeries file.

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 iSeries 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 iSeries 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 the following characters: A to Z, ¥, #, or@. For characters subsequent to the first, the numbers 0 to 9, underscores, and periods can also be used.

Use of File Description File (Advanced)

This item specifies whether a file-description file is used to transfer data to the iSeries 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 65.

- 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 iSeries 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 iSeries 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 iSeries system and the workstation.

- Specify this item in all other cases. Two examples are:
  - Data is transferred from a workstation file having more than one field.
  - The iSeries 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 iSeries 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 iSeries 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 iSeries 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.

iSeries Object (Advanced)

This item is always required. It specifies whether the iSeries 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 iSeries 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 iSeries 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 used to add an explanation of a new iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries member.

To specify an apostrophe (′) in the explanation, enter two apostrophes (′ ′).

iSeries File Type

This item is always required. It specifies the type of iSeries file and the members to be created (same type for both).

Specify one of the following things:

- To create an iSeries 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 iSeries 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 iSeries 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-iSeries 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 iSeries 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 **iSeries 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 iSeries physical source file is created. **iSeries File Type** and **Field Reference File Name** are not displayed. Instead, **Record Length** appears.

This item is always required. A new iSeries file is created using the field name in a file-description file and the field definitions in an iSeries 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 iSeries 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 iSeries job library list. *USRLIBL of the library list can be changed by modifying the job description by executing a CHGJOBD command on the iSeries processor.

file-name

This is the name of the iSeries 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 iSeries 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 iSeries physical source file. When the data receiver is an iSeries 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 iSeries file.

Specify one of the following things:

- **Read/Write**. This enables other users to read from and write to the iSeries 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 iSeries file, specify **Read/Write** or **All**.
- **Read**. This enables other users to read from the iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries file.

To specify an apostrophe (′) in the explanation, enter two apostrophes (′ ′).

Replace Existing Member

This item transfers data to an existing iSeries member, specified in the **Library/File (Member)** item. The existing data in that iSeries 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→iSeries Transfer window.
2. After specifying the necessary information, click **Save** or **Save As** from the **File** menu.

   The Save Transfer Request File as window appears.
3. Specify each item, referring to the following explanation, then click **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 you click **OK**, 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→iSeries Transfer window.
2. Click **Open** from the **File** menu.
3. Specify the name of the file to be opened using the Open Transfer Request File window, then click **OK**.

   The PC→iSeries 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 38.

Performing a Transfer Request

A transfer request can be performed in any of the following ways:

- By clicking the icon with which the transfer request has been registered
- By using the PC→iSeries Transfer window of the Data Transfer icon

Clicking 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→iSeries Transfer window.

Clicking the corresponding icon starts data transfer.

Using the PC→iSeries 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 iSeries file, the transferred data replaces the existing data in that member.
2. Select **Send** from the PC→iSeries Transfer window.

   Data transfer starts.
3. After the transfer has been completed, click **Cancel** or **Exit** from the **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 iSeries 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 iSeriesmember 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 iSeries fields. Another example is the case where the record length of a workstation file differs from that expected by the iSeries 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 iSeries-to-Workstation Transfer Request

To create a transfer request to receive data from the host, do as follows:

1. Click the **Data Transfer** icon.

2. When the PC→iSeries Transfer window is displayed, select **Switch to RECEIVE** to switch the display to the iSeries→PC Transfer window.

   For the additional settings, click the **Advanced** button.
3. Which items are to be specified by the user vary with the data type, as follows:
   - Entire iSeries file
   - Part of an iSeries file
   - Data combined from several iSeries files
   - Summary of record groups

   Before specifying each item, while referring to “Items to Be Specified” on page 42, note the following points regarding the data to be received.

Receiving an Entire iSeries File

This is the simplest way of transferring data from the iSeries 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 are as follows:

System name
This item specifies the name of the system.

Library/File (Member)
This item specifies the name of the iSeries file.

SELECT
Specifying an asterisk (*) for this item indicates that all fields are to be transferred, or lists all the fields in the iSeries file.

ORDER BY
This item is optional. It specifies how records are grouped. When this item is left blank, records are not grouped (data is transferred in the same order it appears in the iSeries file).

Receiving Part of an iSeries File

Only part of an iSeries file is transferred to the workstation.

The **FROM** items that must be specified are as follows:

System name
This item specifies the name of the system.

Library/File (Member)
This item specifies the name of the iSeries 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 are grouped. When this item is left blank, records are not grouped (data is transferred in the same order it appears in the iSeries file).

Receiving Data Combined from Several iSeries Files

The data to be transferred can be stored in two or more iSeries 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 iSeries→PC Transfer function, this “join and transfer” function can be performed in a single step.

The **FROM** items that must be specified are as follows:

System name
This item specifies the name of the system.

Library/File (Member)
This item specifies the names of all iSeries 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 are grouped. When this item is left blank, records are not grouped (data is transferred in the same order as it appears in the iSeries 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 iSeries files.

The **FROM** items that must be specified are 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 iSeries files are classified into several groups. To group all records into a single group, this item need not be specified.

SELECT
Specifying this item creates a summary record. 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 iSeries file).

Items to Be Specified

The following section explains the items to be specified using the iSeries→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 iSeries physical file, you must have *USE authority for that file. To transfer data from an iSeries 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 iSeries library that contains the iSeries file to be transferred. This iSeries file contains the data to be transferred from the iSeries 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 iSeries job library list. *USRLIBL of the library list can be changed by modifying the job description by executing the CHGJOBD command on the iSeries system.

file-name

This is the name of the iSeries 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 iSeries job library list, enter *USRLIBL/, then select **Browse**.

member-name

This is the name of the iSeries 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 iSeries 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 iSeries 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 iSeries 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.

For 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 59 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 iSeries 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
```

You can also specify:

```
ITEMNO,QONHAND,PRICE
```

When records are transferred from an iSeries 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 **FROM**.

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.

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.

= Equal

<> or ><
Not equal

> 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.

Test usage is as follows:

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.

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.

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 46 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 iSeries 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 iSeries→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 iSeries→PC Transfer window. See “Creating an iSeries-to-Workstation Transfer Request” on page 39 for an explanation of how to specify the required data.

2. After specifying the necessary data, click **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 click **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 you click **OK**, 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 iSeries→PC Transfer window.
2. Select **Open** from the **File** menu.

   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 click **OK**.

   The iSeries→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 54.

Executing a Transfer Request

You can execute a file transfer request in one of the following two ways:

- By selecting the icon with which the transfer request has been registered
- By using the iSeries→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 iSeries→PC Transfer window.

Data transfer starts as soon as you select the icon with which a transfer request has been registered.

Using the iSeries→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 iSeries file, the transferred data replaces the existing data in the member.

2. Select **Receive** from the iSeries→PC Transfer window.

   Data transfer starts.
3. After the transfer has been completed, click **Cancel**, or click **Exit** from the **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 iSeries data type to workstation code.

Note: The workstation receives the iSeries 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 iSeries system, none of the following reserved words can be specified as a file name or field name:

CRTFILE
CRTMBR
FILETEXT
FILETYPE
INTO
MBRTEXT
PUBAUT
RCDLEN
REFFILE

For a transfer request from the iSeriessystem to a workstation, none of the following reserved words can be used as a file name or field name:

ABS
AND
ASC
AVG
BETWEEN
BY
COLUMNS
COUNT
DESC
EXTRACT
FROM
GROUP
HAVING
IN
INNER
IS
ISNOT
LIKE
MAX
MIN
NOT
OPTIONS
OR
ORDER
PARTOUT
REPLACE
SELECT
SUBSTR
SUM
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:

```
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 iSeries 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 iSeries 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 iSeries Files

Two iSeries 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.

You 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 iSeries→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 iSeries 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, you 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. Use a comma 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 iSeries 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 iSeries 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**. 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**

```
   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 the Pull-Down Menu

The following section provides a simple explanation of the menu bar of the iSeries→PC Transfer window and PC→iSeries Transfer window.

File

Transfer request files can be processed.

Create Creates a transfer request file

Open Displays the contents of an existing transfer request file

Save, Save As
Save the current settings to the transfer request file being used or to a new transfer request file, respectively

Exit Terminates the operation started by selecting the **Data Transfer** icon

Setup (Only for iSeries→PC Transfer)

User Options

Time, date, and numeric value format for receiving can be specified.

Ignore Decimal Data Error
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
Specifies a desired time format for fields of iSeries field type having a selected time. If no time format is specified, the default value in the workstation's national information file is used 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	iSeries DDS (Format given by iSeries file attribute)
DFT	iSeries default format (Host job default is used)

Time separator

Specifies enabled delimiters. The fields of the iSeries field type for the selected time must be in a format that supports delimiters.

When no delimiters are specified, the default value in the workstation's national information file is used 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) iSeries Default Separator

Date Format

Specifies the date format for fields of iSeries field type for the selected date.

If this date format is not specified, the default value in the workstation's national information file is used.

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 iSeries DDS (Format given by iSeries file attribute)

DFT iSeriesdefault format (Host job default is used)

Date separator

Specifies delimiters. The fields of the iSeries field type for the selected date must be in a format that supports delimiters.

When no date delimiters are specified, the default value in the workstation's national information file is used when the transfer request starts, and that in an existing transfer request is used 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) iSeries default separator

Decimal separator
Specifies the decimal point character in an iSeries 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 used when the transfer request starts, and that in an existing transfer request is used when the request is called again.

Supported decimal point delimiters are as follows:

Period (.)

Comma
(,)

DFT (DFT) - Default decimal separator

Sort Sequence

Specifies which sort sequence should be used for this transfer request.

iSeries job default
Sort by the table identified on the iSeries 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

iSeries standard tables provide many languages. Select the user-specified languages to enter the desired language ID. Language IDs shipped with iSeries are found in *AS/400 NLS Guide*.

Language ID

Enter the language ID for the desired language iSeries standard tables provide many languages. Language IDs shipped with iSeries are found in *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.

Current Table
Specifies whether the IBM default translation or the user-defined translation table is to be used.

Host Code Page
Specifies the host code page to be used for translation.

Workstation Code Page
Specifies the workstation code page to be used for translation.

File Name
Specifies the file name of the user-defined table to be used for translation.
- To list all files in your workstation, click **Browse**.
- To customize the translation table, click **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 the following:
- The file type of the workstation file to be transferred. For an explanation of each file type, see “Creating a File-Description File”.
- The field names and order of these fields in each data record.
- The data type of each field in the workstation file.
- The 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 iSeries system files. The data must be defined for both the iSeries system and the workstation files, because the field names from each file are needed to send the data to the iSeries 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 iSeries 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 6 shows the valid file-type indicators.

Table 6. 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 7 shows the valid time formats.

Table 7. 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 iSeries file attribute
7	DFT	Host job default is used
*	Unspecified	Host job default is used

Table 8 shows the valid time separators.

Table 8. 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 9 shows the valid date formats.

Table 9. 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 iSeries file attribute
10	DFT	Host job default is used
*	Unspecified	Host job default is used

Table 10 shows the valid date separators.

Table 10. 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 11 shows the valid decimal separators.

Table 11. 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 iSeries system and can be from 1 to 10 characters.
- The indicator for the data type. Table 12 on page 69 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 12. Data Type Indicators

Indicator	Data Type
1	ASCII[1]
2	ASCII numeric
3	Hexadecimal
4	Binary
5	Zoned
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 iSeries system.

Table 13 shows the valid single-byte character set (SBCS) data types for each file.

Table 13. 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.

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 14 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 14. 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 HMS (see note 1) USA ISO, EUR, and JIS (see note 1) DDS, DFT	 • 8 • 8 • 8 • 8 or 10 [2]
Date MDY, DMY, YMD Julian ISO, EUR, JIS, USA (see note 1) DDS, DFT	 • 8 • 6 (only allowed length) • 10 • 6, 8, or 10 [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 iSeries job default (DFT keyword).

Table 15 shows the allowed data length limits for each iSeries data type.

Table 15. Allowable Data Length Limits for iSeries Data Types

iSeries Data Type	Data Length Limit in Bytes [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 [2]
Date	
MDY, DMY, YMD	8
Julian	6 (only allowed length)
ISO, EUR, JIS, USA	10
DDS, DFT	6, 8, or 10 [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 72 for more details.

2 The length is determined by the format defined in the host file for DDS, or from the iSeries 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 72 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
```

File-Description File 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 iSeries 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:

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

The data transfer function supports the following workstation data types:

- 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

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.

BASIC Numeric Data

Double-Precision Data: Double-precision data is defined only for the workstation. The iSeries 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.

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.

Single-Precision Data: Single-precision data is defined only for the workstation. The iSeries 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.

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 iSeries 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 iSeries 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.

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

Packed Decimal Data

For both the iSeries 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 iSeries 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.

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 iSeries 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 iSeries 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 iSeries 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.

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.

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 iSeries 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 -1.0×10^3 in ASCII numeric format) is:

2D 31 2E 30 45 2B 30 33

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

39 2E 35 44 2D 31 35

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 iSeries record.

Transferring Data to ASCII Text Files: When you create an ASCII text file, the data coming from the iSeries 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 iSeries 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 16 shows the mapping between binary field lengths and their ASCII lengths.

Table 16. 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 iSeries 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 iSeries 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 iSeries binary, zoned decimal, or packed decimal data, depending on the specified data type.

  The field lengths on the iSeries 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 iSeries fields, null values (except date, time, and time stamp) cannot be reliably detected and are not uploaded. For variable-length iSeries 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 iSeries file, the following errors can occur:

- A data field in the ASCII text file is too long for a field in the iSeries-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 iSeries 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 iSeries 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 iSeries 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 iSeries fields, null values (except date, time, and time stamp) cannot be reliably detected and are not uploaded. For variable-length iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries system if the field is null-capable.
- In variable-length fields, if the iSeries field is variable length, the field is converted to the iSeries 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.

Transferring Data from BASIC Sequential Files: When you transfer data from BASIC sequential files to iSeries 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 iSeries-defined file, the following errors can occur:

- Data in this field is too long for the iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 two ASCII bytes to one 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 iSeries field if the field is null-capable.
- If the iSeries field is variable-length, the field is converted to the iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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).

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 iSeries 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 you transfer data 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 iSeries 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 iSeries system are retransmitted to a workstation.

When you transfer data from a BIFF file to an iSeries 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 you transfer data from a BIFF file to the iSeries system, the following specific processing is performed:

- When you transfer data 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 iSeriessystem, the first line must contain the same field names (as defined in the iSeries system) or data. If the first line or the first set does not contain a character field that exactly matches the iSeries field, the file is treated as being a file with no column headers, and only data is processed.
- When you transfer a BIFF file to the iSeries system, header information is not used.
- To ensure correct transfer of a BIFF file to the iSeries system, the file format must be valid (lines and columns). Data for each set or line must correspond to one record in the iSeries file.

Errors When Transferring Data from BIFF Files: When you transfer data 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 iSeries 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 iSeries 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 iSeries 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 iSeries fields.
- Numeric data is too long to fit into the corresponding iSeries 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 iSeries field.
  - The number of decimal digits in a numeric field exceeds the number of digits specified for the iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries 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 iSeries fields, null values (except date, time, and time stamp) cannot be reliably detected and are not uploaded. For variable-length iSeries 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 iSeries 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 iSeries field.
  - The decimal value of a numeric field contains more digits than were specified for the iSeries 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 iSeries 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 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 iSeries 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 iSeries fields, null values (except date, time, and time stamp) cannot be reliably detected and are not uploaded. For variable-length iSeries 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 iSeries 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 iSeries field.
  - The decimal value of a numeric field contains more digits than were specified for the iSeries 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 iSeries 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 iSeries system to a no-conversion file, the data transfers exactly as it is stored on the system.

Variable-length iSeries 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.

iSeries System-to-PC Performance Considerations

Transferring data from the iSeries 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 iSeries 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 iSeries-to-workstation data transfer function uses many functions within the iSeries system to determine the fastest method of selectively retrieving records. When it selects a smaller group of records to transfer, the iSeries-to-workstation data transfer function uses the existing access paths whenever possible to improve performance.

For the iSeries-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 5. Transferring Files

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

Host Requirements

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
- 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. Click **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 "Creating List Files" 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. Click **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 "Creating List Files" 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. Click **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 in “Sending Files to the Host System” on page 95 and “Receiving Files from the Host System” on page 96, 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. Click **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 99.)
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. Click **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 “Items to Be Specified”).

   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**.

Items to Be Specified

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 17 on page 100 lists the mode values for the file transfer options. “File Transfer for PC400” on page 104 lists transfer options.

Table 17. 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

Logical Record Length (LRECL)

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
```

Additional Options

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

Setting General 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.

Data Transfer

You can choose whether the Data Transfer function (see Chapter 4, "Data Transfer for PC400", on page 25) is to be used instead of the normal file transfer function.

Bidi Options

The following options apply if the session is configured for an Arabic or Hebrew host code page.

- Host File Orientation
- PC File Orientation
- Lam-Alef Expansion
- Lam-Alef Compression
- Symmetric Swapping
- Round Trip
- Numeral Shape

Refer to *Quick Beginnings* or the online help for information about these options.

Host Command

You can specify host command to be called when file transfer starts. If nothing is entered in this text box, 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 Library

You can specify the iSeries 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, 806, 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.

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 timeout 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.

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:

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
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
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')

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 iSeries system for a backup
- To edit a source file of an iSeries program with a workstation editor, and send the file edited on the workstation to the iSeries system.
- To distribute workstation documents and programs to the iSeries 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 application data 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 iSeries 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 iSeries 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 iSeries 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 iSeries system.

Using the DSPMBRLST Command

For file transfer from the iSeries 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 iSeries 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. iSeries 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 6. Printing

You can use Personal Communications to print from display or printer sessions. For considerations and restrictions on printing, see Chapter 3, “Considerations for Using PC400 Sessions”, on page 15.

Display 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.

Using the **File → Print Screen Collection → Collect Screen** feature, you can add a capture of all or part of the screen to a collection of captures. You can then print all the collected captures at the same time, using the **File → Print Screen Collection → Print Collection** feature. After printing, the collected captures are deleted.

All the collected screens can be deleted without printing by clicking **File → Print Screen Collection → Purge Collection**. An individual screen or part of the collection cannot be deleted.

The **File → Print Screen Collection → Print Collection on Exit** option ensures that the collected screens are printed before you close or disconnect the session. This option is enabled by default. To end the session without printing the collected screen, clear the Print Collection on Exit option. All the collected screens are then deleted when you close or disconnect the session.

Note: The Collect Screen feature works independently of the normal **Print Screen** function. You can still use Print Screen to print individual screens, while collecting multiple screens.

You can add the **Collect Screen** and **Print Collection** functions to the toolbar, a popup keypad, a custom keyboard map, or a mouse customization, using the **Edit → Preferences** menu in the session window. The settings in the Page Setup dialog are used (shared with the normal Print Screen function).

In PDT mode, there is an option available for printing more than one screen in a page. Refer to the *Administrator’s Guide and Reference* for more information.

Printer Sessions

From printer sessions, you can direct printing from a zSeries or iSeries 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.

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 *Administrator's Guide and Reference*.

You can map a key sequence to bring up the Printer Setup dialog. There is no default key combination for this function. See "Default Key Function Assignments" on page 17 for more information about keyboard functions.

Defining a Printer for a Session

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

1. Click **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 check box **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. Do one of the following:

   •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 named for the printer language, not the printer model.

Older SBCS PDFs and PDTs are no longer shipped with Personal Communications, but are available at the following website: http://www.ibm.com/software/network/pcomm. If you already have modified PDFs, any PDF and PDTs with different names than those in Table 18 are retained during an install.

Note: All DBCS PDFs and PDTs are still shipped with Personal Communications.

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.

Table 18. Old Printer Definition Files

Printer Definition File (PDF) Name	Remarks
`ibm5577`	(No FFF) (DBCS only)
`ibm5577a`	(No FFF)
`ibm5577b`	(No FFF)
`lbp4`	

Table 18. Old Printer Definition Files (continued)

Printer Definition File (PDF) Name	Remarks
lips3a4	(No FFF) (DBCS only)
lips3b4	(No FFF) (DBCS only)

The supplied basic_ascii PDF does not contain any printer commands, which results in only ASCII text being sent to a printer or file. An accompanying PDT is also shipped. This PDF is for SBCS only.

PFT Migration

You can migrate a PC Support/400 Workstation Feature Printer Function Table (PFT) to a PDF for PC400. Refer to the *Administrator's Guide and Reference* for more information.

Double-Byte Character Support

The PDTs provided by Personal Communications and the corresponding printers are listed in this section. The DBCS PDTs are as follows:

- IBM5577.PDT
- IBM5577B.PDT
- IBM5585.PDT
- IBM5587.PDT
- ESC_P.PDT
- LIPS3A4.PDT
- LIPS3B4.PDT
- 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 printers to use with these PDTs are as follows:

```
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)
```

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 19. 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:

Table 20. Page Setup Parameters — Text Options

Group	Options	Setting
Print Options The following options can be selected 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.

Table 20. Page Setup Parameters — Text Options (continued)

Group	Options	Setting
	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.
	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.

Table 20. Page Setup Parameters — Text Options (continued)

Group	Options	Setting
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.
	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.

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

Table 21. Page Setup 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.
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 iSeries 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 22. Page Setup Parameters — Orientation (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 formatted 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 23. 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.

Table 24. 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. Click **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 Printer 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.

Image Print Transform

Personal Communications Version 5.7 allows use of Image Print Transform in 5250 print sessions, when using Host Print Transform. Refer to the most recent IBM iSeries printing reference for more information on this feature.

Workstation Profile Parameter for Code Page

Occasionally a font does not support the desired code page. The wrong characters may be printed within the specific character set (Latin 2, for example). Personal Communications has a workstation profile parameter that allows the program to use a different code page that is supported by the desired font.

You can use the **PrinterFontCodePage** parameter if the following conditions are met:

- If you can specify the printer font code page with which the desired font is encoded.
- If Personal Communications provides the translation table for the host code page and the printer font code page.

However, since some Personal Communications releases may require manual adjustment of the workstation profile, try using different fonts before altering the .ws file. Fonts are listed in the Personal Communications Page Setup panel for all display sessions and 3270 host print sessions. For 5250 print sessions, the PCSPD.DAT file can be manually changed to control the fonts used. The Courier New font should support most languages and corresponding code pages.

To edit the .ws file, you must change the PrinterFontCodePage parameter to the value of the supported code page you wish to use. This option must be put in the [printers] section, and is case-sensitive. See the following example for the proper parameter syntax. The parameter does not need to be placed immediately after the [printers] section label.

```
[printers]
PrinterFontCodePage=852
```

In this case, the desired font is encoded with code page 852. Personal Communications uses a different, existing translation table to translate data from EBCDIC to 852, versus using the standard Windows code page.

This option is on the Page Setup panel for Personal Communications Version 5.7 5250 print sessions.

Chapter 7. 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 is 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 iSeries 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.

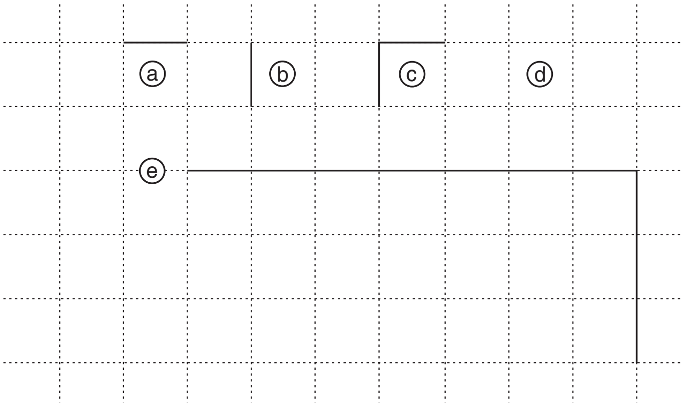


Figure 2. Grid Line Types

The limits for displaying the grid lines are as follows:

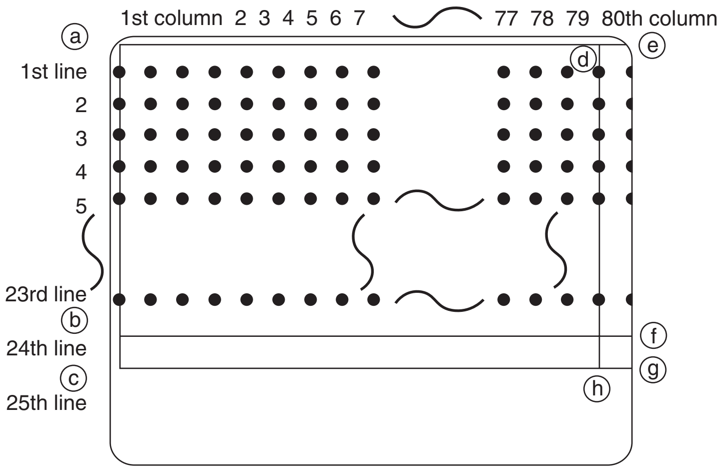


Figure 3. Grid-Line Display Limits

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.

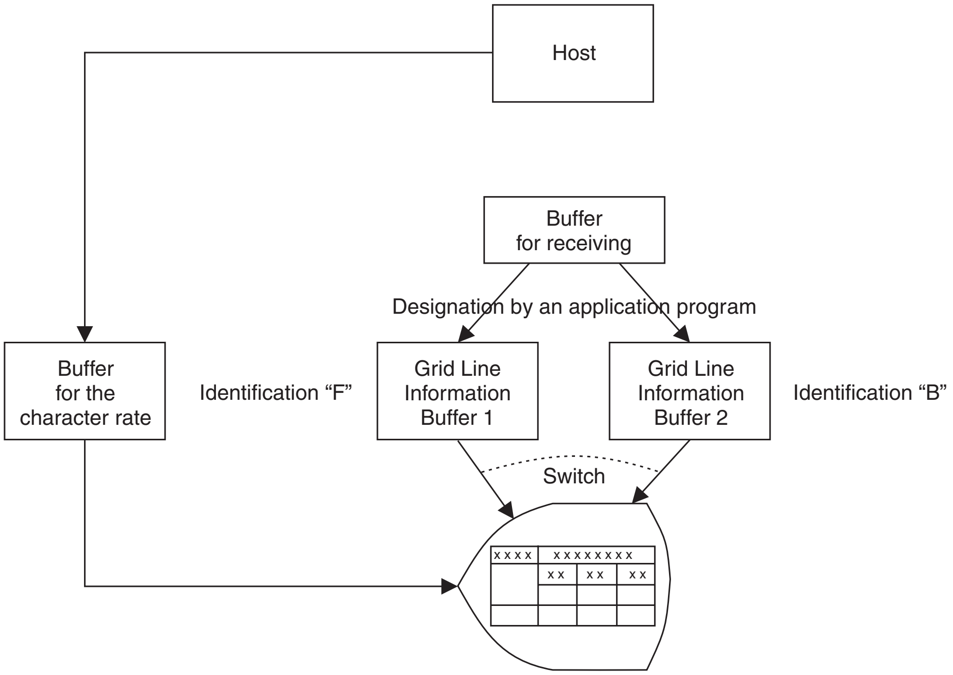

Host
Buffer for receiving
Designation by an application program
Buffer for the character rate
Identification “F”
Grid Line Information Buffer 1
Grid Line Information Buffer 2
Identification “B”
Switch
xxxx
xxxxxxxx
xx
xx
xx

Figure 4. Grid-Line Information Buffer and Displayed Character Data

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 122.

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 122.

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:

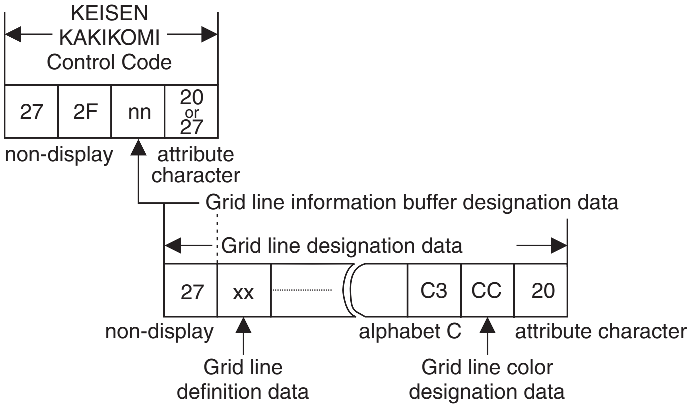


Figure 5. Grid-Line Control Data Format

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:

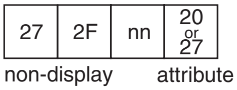
1. Grid-line draw control code: first line, first column
2. Grid-line designation data: first line, fourth column

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

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.



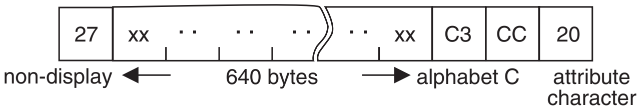
Parameter	Meaning	Valid Value
nn	Grid line information buffer designation data	• X'C6' (F) • Grid-line definition data to be stored into the grid-line information buffer 1. • X'C2' ("B") • Grid-line definition data to be stored into grid-line information buffer 2.

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

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.



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
```

The following illustration shows the relation between the actual grid lines on the display screen and the grid-line definition data.

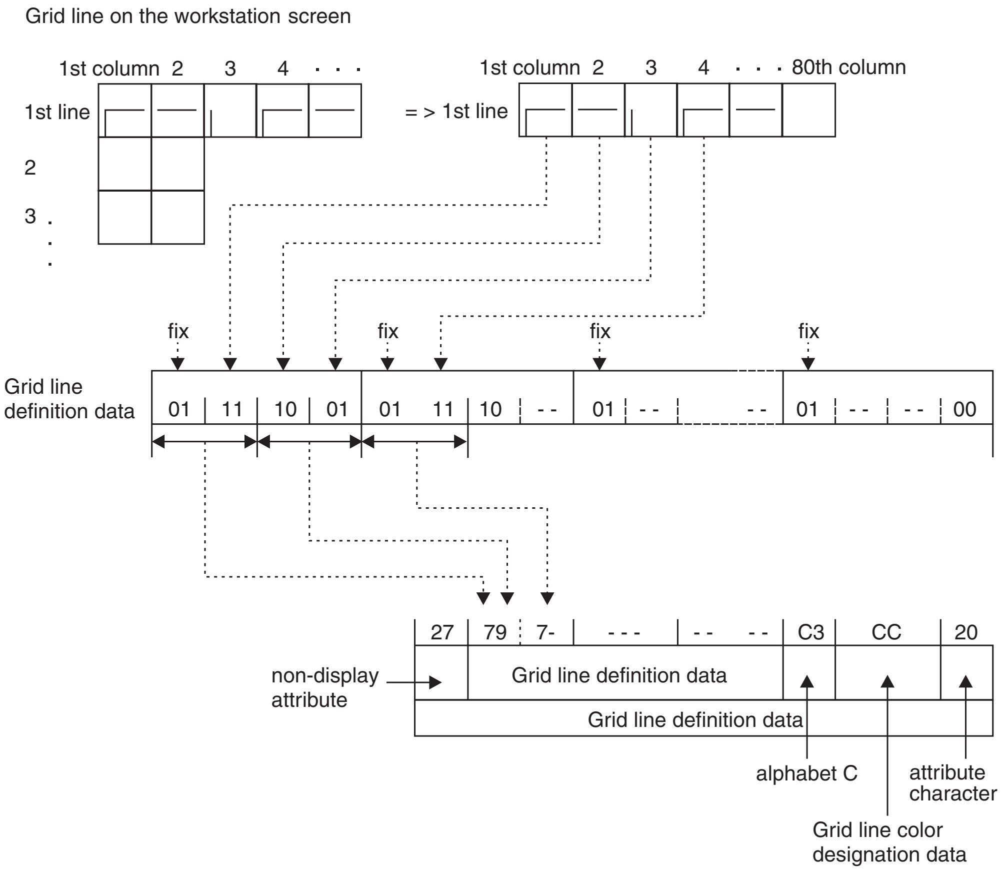

Grid line on the workstation screen
1st column 2 3 4 . . .
1st line
2
3
1st column 2 3 4 . . . 80th column
= > 1st line
fix
fix
fix
fix
Grid line definition data
01 11 10 01 01 11 10 - - 01 - - - - 01 - - - - 00
27 79 7- - - - - - - - C3 CC 20
non-display attribute
Grid line definition data
Grid line definition data
alphabet C
attribute character
Grid line color designation data

Figure 6. Grid-Line Definition Data and Displayed Grid Lines

The error process for the grid-line designation data is as follows:

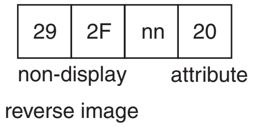
- If the first byte of the grid-line designation data is not X'27', the grid-line definition data is ignored and the contents of the grid-line information buffer are deleted.
- Incorrect designation of the grid-line definition data:

  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

PC400 assumes that the data is the control code for deletion of the contents of the grid line information buffer when it receives the following data, starting with the first line and first column.

This data is used to delete the contents in the grid-line information buffer specified by the parameter.

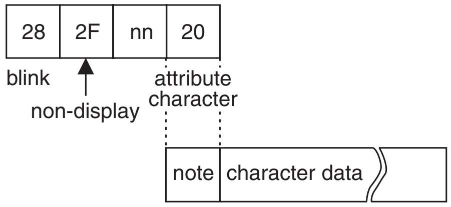


Parameter	Meaning
xx ... xx	Grid-line definition data
CC	Reserved

Combination Display of Grid-Line Screen Format and User Screen Format

Displays the grid-line screen format with the user-defined 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.



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 3. 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 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:

AnyNet
APPN
CICS
COBOL/2
GDDM
IBM
IBM Global Network

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, JavaBeans™, 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.

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.

Index

Readers' Comments — We'd Like to Hear from You

Personal Communications for Windows, Version 5.7
5250 Emulator User's Reference

Publication No. SC31-8837-02

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.

Readers' Comments — We'd Like to Hear from You

SC31-8837-02

Cut or Fold
Along Line

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

Cut or Fold
Along Line

Program Number: 5639–I70

Printed in U.S.A.

SC31-8837-02