

MUSIC/SP

Version 3

Release 1

Personal Computer Workstation User's Guide

Fifth Edition (May 1994)

This edition applies to Release 1 of Multi-User System for Interactive Computing / System Product (MUSIC/SP) Version 3, and to all releases of this product until otherwise indicated in new editions or Technical Newsletters. MUSIC/SP Version 3 is published and licensed by McGill University Systems Inc.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: MUSIC Product Group, McGill University Systems Inc., 550 Sherbrooke St. West, Suite 1650, Montreal, Quebec, Canada H3A 1B9. Fax: (514) 398-4488.

About This Guide

This guide describes the components of the Personal Computer Workstation (PCWS) software. It is intended for the user of a personal computer, who wishes to connect to MUSIC/SP or another host system. MUSIC/SP (Multi-User System for Interactive Computing/System Product) is a multi-user, multi-function, interactive system complete with a collection of application programs, compiler interfaces, and utilities.

The Personal Computer WorkStation (PCWS) software is an asynchronous communications package which runs on the IBM PC and PS/2 families of computers. It allows you to use your IBM PC as an IBM 3270 terminal, a DEC VT100 terminal or as a TTY terminal. The PCWS software supports these models of the IBM Personal Computer: PC, PC/XT, PC/AT, PC Convertible, and the line of Personal System/2 computers.

Prerequisite Knowledge

PCWS users need to be familiar with DOS (Disk Operating System) on their personal computer and the operating system on the host computer. (For example, MUSIC/SP is an operating system for a host computer.) For information about DOS, refer to the IBM publication *Disk Operating System Guide*. If the host system is MUSIC/SP, the *MUSIC/SP User's Reference Guide* provides information on the use of MUSIC. Other MUSIC/SP publications are listed later in this preface.

PCWS for Windows

This guide only covers the original PCWS designed to run under the DOS operating system. Another program called *PCWS for Windows* also exists which comes with its own Windows formatted help files.

Terminology

The term MUSIC/SP is an acronym for the phrase *Multi-User System for Interactive Computing / System Product*. Where the term MUSIC is used in this publication, it refers to MUSIC/SP.

A glossary is provided at the end of this guide to define the terms used throughout this publication.

Trademarks

- MUSIC/SP, NET3270 are trademarks of McGill University, Montreal, Canada
- IBM, Micro Channel, AS/400, Personal System/2, PS/1, PS/2, and AIX are registered trademarks of International Business Machines Corporation.
- SAA, ES/9370, ES/9000, PROFS, PR/SM, MVS/ESA, VM/ESA, PC-DOS, DisplayWrite, OfficeVision, and GDDM are trademarks of International Business Machines Corporation.
- DEC VT-100 is a registered trademark of Digital Equipment Corporation.
- MINITAB is a registered trademark of MINITAB Inc.

- SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.
- UNIX is a trademark of AT&T Bell Laboratories.
- Other names may be trademarks or registered trademarks of their respective companies.

Chapter Outlines

Chapter 1. Introduction

Describes the basic concepts of the Personal Computer Workstation program and what it can do.

Chapter 2. The PCWS Menu

Describes how to use the PCWS menu for selecting terminal type, personal computer settings, etc.

Chapter 3. Running PCWS

Describes how to connect to a host system as well as various PCWS features.

Chapter 4. The PCWS Terminal Type

Describes the PCWS terminal type used to connect through an asynchronous controller. This terminal type has 2 modes of operation: PAGE mode and 3270 mode.

Chapter 5. The VT100 Terminal Type

Describes using your PC as a VT100 terminal through a protocol converter.

Chapter 6. File Transfer

Describes how to transfer files between MUSIC and your PC.

Chapter 7. MUSIC and PC Interfacing

Describes the MUSIC programs specially designed to interface with PCWS.

Chapter 8. The PCWS API Library

Describes the Application Programming Interface (API) library for PCWS.

Appendix A. Keystrokes

The functions and keys are listed for PAGE mode, 3270 mode, and VT100 mode.

Appendix B. The PCWS Configuration File

Describes the optional PCWS.CFG file.

Glossary

Defines the common terms used throughout this guide.

MUSIC/SP Publications

The following is a list of all the current MUSIC/SP publications. These hardcopy publications can be ordered through IBM. Online versions (softcopy) of the user publications can be accessed with the MUSIC/SP command called "MAN".

- *MUSIC/SP General Information Manual* (May 1994), presents the highlights of MUSIC/SP.
- *MUSIC/SP Administrator's Guide* (May 1994), describes how to install and operate MUSIC/SP.
- *MUSIC/SP Administrator's Reference* (May 1994), describes the internals of MUSIC/SP; utility programs and supervisory commands; gives detailed storage estimates; and documents console messages.
- *MUSIC/SP User's Reference Guide* (May 1994), describes how to use MUSIC/SP; its command language; terminal and batch set up; and job processing using the various language processors.
- *MUSIC/SP Guide for New Users* (May 1994), introduces new users to the use of MUSIC/SP via an IBM 3270-type workstation. It describes the FSI (Full Screen Interface) menu facility. New users learn how to use many programs on MUSIC/SP for such tasks as editing and running programs.
- *MUSIC/SP Mail and Office Applications Guide* (May 1994), describes the features of electronic mail and the TODO (Time, Office, and Documentation Organizer) facility. This includes the scheduling function, spell checking, and MUSIC/SCRIPT (text processing).
- *MUSIC/SP Personal Computer Workstation User's Guide* (May 1994), describes the components of the Personal Computer Workstation (PCWS). It is intended for the novice or experienced user of a personal computer, who wishes to connect to MUSIC/SP or another host system.
- *MUSIC/SP Communications Guide* (May 1994), describes the programs available on MUSIC/SP that provide communication between users through electronic conferencing and discussion lists. Emphasis is placed on access to the Internet with programs such as TELNET (logging on other computers), FTP (File Transfer Protocol), RN (Newsreader), and GOPHER (document search and retrieval protocol).
- *MUSIC/SP Teacher's Guide* (May 1994), describes various MUSIC/SP facilities related to the academic environment. Emphasis is placed on communication between teacher and student and easy methods for learning how to use MUSIC applications.
- *MUSIC/SP Campus-Wide Information Systems (CWIS) Guide* (May 1994), describes how to create and maintain a Campus-Wide Information System, Help facility, or Classified Ads facility; how to do full-text searching; and how to provide gopher access. MUSIC/SP's resources are used to provide online distribution of information to a wide audience.

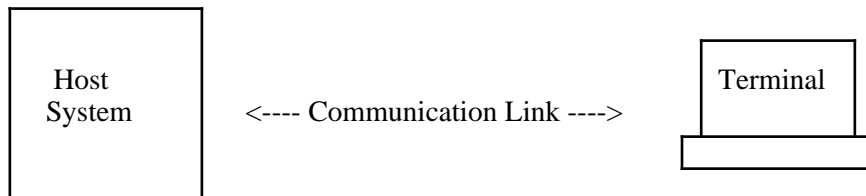
Chapter 1. Introduction

Chapter 1. Introduction

Basic Concepts

Often people use computer terminals to communicate with multi-user host systems. (MUSIC/SP is an example of a host system). Most terminals today are display terminals. They consist of 2 major elements: a keyboard and a video screen (also known as the standard input and standard output devices). It is through the keyboard that users can send their commands to the mainframe (host system). Results are then sent back from the host to the terminal where they are displayed on the screen.

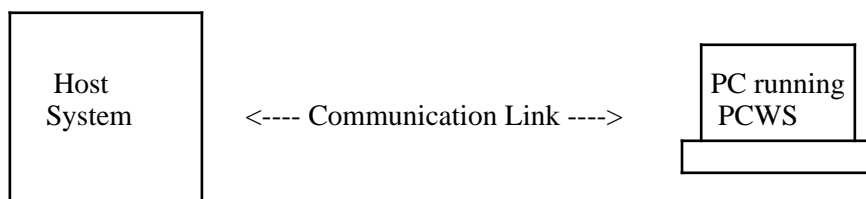
All terminals have to be connected to the mainframe via some sort of wire. A local terminal is one which is connected by a cable going directly to the computer room. On the other hand, a remote terminal uses a telephone line and modems to form the communication link.



If you encounter unfamiliar terminology in what follows, you can refer to the glossary at the back of this manual. Some common terms in computer communications are explained there.

What Does PCWS Do?

PCWS is a Terminal Emulator Program designed to run under the DOS operating system on computers of the IBM PC and PS/2 families. PCWS allows you to use your PC to communicate with a host system as if it was a traditional computer terminal.



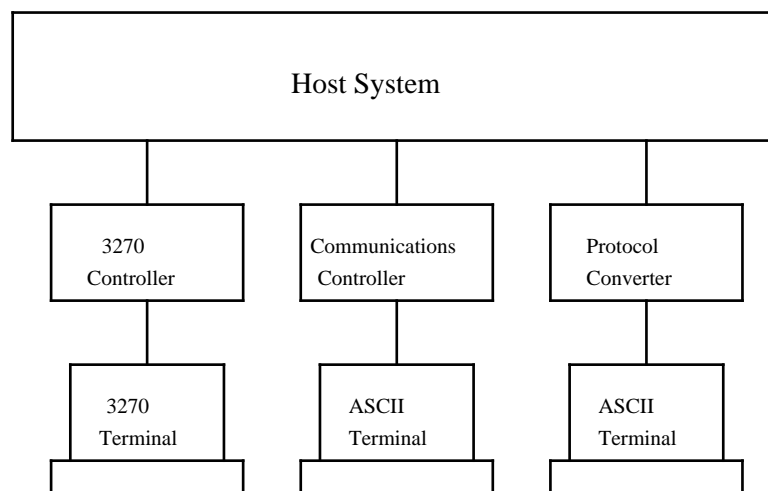
PCWS was designed specifically for use with the MUSIC/SP operating system. It can also connect to other systems that support asynchronous ASCII communications. Throughout this manual, a system such as MUSIC is referred to generically as the *host system*.

PCWS allows the PC to communicate with the outside world through one of its serial ports. A remote PC can use a regular telephone line to communicate with a host system if a modem is connected to its serial ports. A modem translates the digital information it receives from the PC into signals which can be transmitted over the telephone line, and vice versa. Modems transfer information at a rate of 2,400 bits per second (bps), 9,600 bps, or higher.

A PC can also be connected directly to the host system, avoiding the use of a telephone line. This permits a transmission speed of 19,200 bps, allowing a PC to effectively emulate a locally attached terminal. This speed approximates the performance of a locally attached 3270 terminal.

How are Terminals Attached?

To understand the different ways a PC can physically be connected to a host, it is helpful to take a quick look at how regular terminals are attached. There are two major classes of computer terminals. These are the 3270 and ASCII terminals types. The following figure shows how both of these normally connect to IBM mainframes.



Most IBM host software is written to support the 3270 type of terminal. These are devices which connect via a coax line to a 3270 controller. They use the EBCDIC encoding conventions.

ASCII terminals are connected either via an Asynchronous Communications Controller or a Protocol Converter such as the 7171 ASCII Control Unit or the 9370 ASCII Subsystem. If connected to a communications controller, the ASCII terminal operates as a simple line by line TTY terminal. On the other hand, when connected to a protocol converter the ASCII terminal is treated as if it was a 3270 terminal.

PCWS allows a PC to act as an ASCII terminal and therefore be connected to a host system in one of two ways:

1. Via an Asynchronous Communications Controller
2. Via a Protocol converter

PCWS Terminal Types

The PCWS program makes it possible for a PC to emulate one of the following two terminal types:

1. PCWS Terminal Type

The PCWS terminal type is a special one, designed specifically for communicating via an asynchronous line with the MUSIC/SP host system. It allows a PC to emulate a 3270 terminal over a regular asynchronous ASCII line without requiring a protocol converter. This is made possible by the cooperation of

the software running at both ends of the connection. MUSIC/SP takes care of translating the 3270 data streams from EBCDIC to ASCII before they are sent to the PC. On the PC side, PCWS handles the interpretation of the data stream and the management of the screen. For more information, refer to *Chapter 4. The PCWS Terminal Type*.

2. VT100 Terminal Type

The VT100 terminal type of PCWS allows a PC to communicate to a host system via a protocol converter. An ASCII terminal connected to a protocol converter appears to the host as a 3270 terminal. It is hence the converter's role to translate the EBCDIC 3270 data streams, sent by the host, into ASCII control sequences the terminal recognizes. The VT100 terminal type of PCWS enables the PC to recognize and interpret the same control sequences the protocol converter would send to a real VT100 terminal. For more information, refer to *Chapter 5. The VT100 Terminal Type*.

System Requirements

PCWS is designed to run under DOS on computers of the IBM PC and PS/2 families. The PC should have a minimum of 128Kb of memory, be running DOS version 3.0 or higher and have a serial port. These are standard features on all IBM PS/2 Models.

Installing PCWS

An install program called INSTALL.EXE is distributed with PCWS. This program will automatically install PCWS on your PC's hard disk. Alternatively, you could also install PCWS by following the 3 steps listed below.

The following changes to your AUTOEXEC.BAT file insure that PCWS will run properly. Note that after editing the AUTOEXEC.BAT file, your PC must be re-booted so that the changes take effect.

For additional information on the AUTOEXEC.BAT file, the PATH statement and the COMSPEC variable, please refer to your DOS manual.

1. Place the PCWS directory in the PATH

PCWS is distributed on a diskette which you receive from your installation. If your computer has a hard disk, create a dedicated PCWS directory (ex. C:\PCWS) and copy all the files from the PCWS diskette to it.

Include the name of the directory containing the PCWS files in the PATH statement of your AUTOEXEC.BAT file. This makes it possible to start PCWS from the DOS prompt regardless of the current directory.

2. Include SET PCWS=C:\PCWS (or other directory)

The PCWS environment variable must be set to a directory where you wish to save your private PCWS data files. These are optional data files the PCWS program may use. The purpose of these files, which are listed below, is covered in later chapters.

PCWS.STP:	(SetUp file)
PCWS.CFG:	(Configuration file)
PCWS.SCR:	(Screen Capture file)
*.CON:	(Connection files)

If the PCWS cannot be found, PCWS uses the current directory to read and write the various data file.

Note: If installing PCWS on a LAN, every user should have this environment variable pointing to their own private directory.

Example: SET PCWS=F:\USR\MARY\PCWS

3. Include SET COMSPEC=C:\DOS\COMMAND.COM (or other full path)

The COMSPEC environment variable must be set to the full path name (Disk, Path & Filename) of COMMAND.COM.

Chapter 2. The PCWS Menu

Chapter 2. The PCWS Menu

Menu Mode

To start the PCWS program, enter "PCWS" at the DOS prompt. The program begins in *menu mode*, meaning, the PCWS menu is on the screen.

```
PCWS x.xx (c) Copyright McGill University
Go Terminal Settings Functions Help Exit

Emulation Mode
Auto Connect
Back to DOS
```

The second line of the menu is called the *menu line*. When the menu first shows up, the Go item of the menu line is highlighted by the selection bar. The left and right arrow keys can be used to move this selection bar from one menu item to the next. In all cases, except Exit, setting the selection bar on a menu item causes a pull-down window to appear just below that item. In the diagram above, the pull-down window for the Go item is displayed. All items on the menu line are described in this chapter.

Menu Keystrokes

When the PCWS menu is on the screen, a small set of keystrokes can be used to navigate through it. These keys always behave in the same manner while in menu mode. The table below describes the functions of these keys:

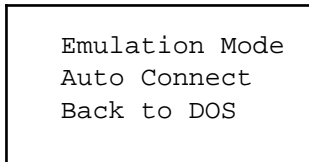
Cursor Keys	The cursor keys are used to move the selection bar right, left, up, and down.
Enter	The ENTER key selects the highlighted item for further processing.
Esc	The ESC key is used to exit from the current window.
Alt-M	The Alt-M combination is a hotkey to switch from Menu mode to Emulation mode and vice versa. (Emulation mode is described in the section below.)

Mouse Support

If the Microsoft Mouse driver is loaded, a mouse can be used to select items from the PCWS menu. Moving the mouse right, left, up and down simulates pressing the corresponding cursor key. Pressing the left button simulates an ENTER key. Pressing the right button simulates pressing the ESC key.

The Go Menu Item

When the *Go* item of the menu line is selected, the following window is displayed just below it:



Emulation Mode

Selecting the *Emulation Mode* will make PCWS switch from menu mode to emulation mode. When in emulation mode the PC is ready to communicate with the host system. In this mode the PC acts as the terminal it is set to emulate. The data coming in at the serial port, as well as the data typed at the keyboard is processed accordingly.

Note: When you first start up PCWS you must establish a communication link to the host system. This is done by entering emulation mode and typing in the required commands. Optionally, you can use Auto Connect if you have previously set up an automated connection procedure. For more information about logically connecting your PC, refer to *Chapter 3 - Running PCWS*

Emulation mode can also be entered, no matter where you are within the PCWS menu, by pressing Alt-M. The same Alt-M key sequence is to be used to switch back to the menu from emulation mode.

In emulation mode, the last line of the screen is known as the status line. It is used to display useful information such as the current terminal type and COM port number.

Auto Connect

The *Auto Connect* item allows the user to start up an automatic connection process to a host system. For more information, please refer to the "Automatic Connection Feature" section of *Chapter 3 - Running PCWS*.

Back to DOS

The *Back to DOS* item allows the user to temporarily exit from the terminal emulation mode and be presented with the DOS prompt. Any PC application can then be invoked. The link with the host system is maintained by PCWS so that it can be resumed at any time.

When ready to resume communication with the host, enter "PCWS" at the DOS prompt. The terminal session is then re-entered at the same point where you left off.

The Terminal Menu Item

The *Terminal* item of the menu line allows the user to view and change the terminal type the PC is to emulate. When this item is selected, a list of the possible terminal types is displayed in a window. The current terminal type is the one highlighted by the selection bar.

```
PCWS
VT100
```

The selection of the terminal type depends on how the PC is connected. If connected via an asynchronous line, the PCWS terminal type must be used. On the other hand, if the PC is connected via a protocol converter, it is the VT100 terminal type which must be selected. For more details on how a PC can be connected to a host, please refer to the "How are terminals attached?" section of *Chapter 1 - Introduction*.

To change the terminal type, use the up and down arrow keys to highlight the correct type and then press the ENTER key.

The Settings Menu Item

A number of PCWS settings can be changed to suit your needs and preferences. The current values of these PCWS settings can be saved in a file. In turn, such a "settings file" can later be loaded to override the current settings. When PCWS is started, it searches for the optional PCWS.STP settings file. If found, the values stored in this file are used to override the PCWS default settings.

When the Settings item of the menu line is selected, the pull-down window below is displayed on the screen:

```
View/Change
File to Disk
Load from Disk
```

View/Change The first item allows you to view and change various PCWS settings. You must use the "File to Disk" option to save these settings if you want PCWS to remember them in the future. See "Viewing and Changing PCWS Settings" below for more information.

File to Disk When the *File* option is selected, PCWS prompts you for a file name. While the program does suggest that you store the values in PCWS.STP, you can enter any valid DOS file name.

Load from Disk When the *Load* option is selected, PCWS prompts you for a file name. While the program does suggest that you load PCWS.STP, you can enter the name of any file previously saved with the *File* option.

Viewing and Changing PCWS Settings

When the *View/Change* option is selected, a second pop-up window displays the different categories of PCWS settings:


```

Communication
Color
PAGE Mode
VT100 Mode
Miscellaneous

```

Communication Settings

A number of communication parameters must be set properly in order for PCWS to be able to communicate with the host computer. When the communication category is selected, the window below pops-up on the screen and all the current values are highlighted:

```

Port          :   COM1  COM2  COM3  COM4
Speed         :   300  1200  2400  4800  9600  19.2
Parity        :   None  Odd   Even  Space  Mark
Stop Bits     :   1 Bit  2 Bits
Word Length   :   5 Bits  6 Bits  7 Bits  8 Bits
XON/XOFF      :   OFF  ON
Local Echo    :   OFF  ON
Parity Check  :   OFF  ON

```

The default PCWS values for the communication settings are: COM1, 1200 bps, even parity, 1 stop bit & 7 data bits. The following table gives a brief description of these parameters.

Port	Selects which communication port to use.
Speed	Selects the speed in bits per second (bps) at which PCWS will communicate to the host computer.
Parity	Specifies how the parity bit (if any) is to be set.
Stop Bits	Selects the number of stop bits to be used.
Word Len.	Selects the size of the data bytes. When communicating with standard ASCII systems, the word length should be 7 data bits with a parity bit, or 8 data bits with no parity bit.
XON/XOFF	Enables or disables the XON/XOFF flow control system.
Local Echo	Specifies whether PCWS is to echo the keyboard data to the screen or not. This option should be set to OFF if the host system already echos what is typed in.
Parity Check	Specifies if PCWS is to check for parity errors on incoming bytes. If set to OFF, the errors are ignored and the parity bit is set to 0.

Color Settings

The user can change a number of default colors used by PCWS. When the *color* category is selected, the window below pops-up displaying all current colors:

```
Low Unp. Field   :
High Unp. Field  :
Low Pro. Field   :
High Pro. Field  :
Page Mode Output :
Page Mode Input  :
VT100 Base Color :
Status Line      :
```

Low Unp. Field
High Unp.
Low Pro.
High Pro.

PCWS allows the user to define the colors of four different fields used in full-screen applications. A field can be protected or not. Unprotected fields are the areas where the user can type information on the screen. A field is also set up to use either low or high intensity. High intensity is used to highlight areas on the screen.

Page Mode Output

When in page mode, this color is used for the top part of the screen where results are displayed.

Page Mode Input

When in page mode, this color is used for items entered on the command line.

VT100 Base Color

This color is used when the PC is emulating a VT100 terminal and no full-screen application is running.

Status Line

This is the status line color. The status line is the last line of the screen when the PC is in emulation mode. It is used to display useful information such as the current terminal type and COM port number.

PAGE Mode Settings

A number of PCWS settings allow the user to customize some of the PAGE Mode functions. When the *PAGE Mode* category is selected, the window below pops-up on the screen:

```
Input Area Mode  :   Char  Line
Autopage         :   OFF  ON
```

Input Area Mode

Specifies whether the input area in Page Mode works in Character or Line mode. When in Line mode the data entered is only transmitted to the host when the ENTER key is pressed. This allows the use of editing keys such as Insert, Delete, Backspace, etc. When character mode is selected, the characters typed in the input area are immediately sent to the host. The input area is cleared when the ENTER key is

pressed. The default is Line Mode.

Autopage By default, Page Mode always stops after displaying a full page of data (Autopage OFF). Setting Autopage to ON, causes PCWS to display all the output without stopping every time the screen fills up.

VT100 Mode Settings

A number of PCWS settings define how the program is to handle some of the VT100 functions. Changing these settings is similar to entering SET-UP mode on a real VT100 terminal. When the VT100 Mode category is selected, the window below pops-up on the screen with all current values highlighted:

```
Wrap Around   :   OFF  ON
Margin Bell   :   OFF  ON
New Line      :   LF  CR/LF
Backspace Key :   08h  7Fh
```

Wrap Around Specifies whether the cursor can wrap around to the next line when it reaches the 80th column of a line. If Wrap Around is set to OFF, the 81st character and all following characters are overwritten into the last position of the current line.

Margin Bell Specifies whether or not a beep should go off when the cursor is 8 characters from the end of line.

New Line A line feed (LF) can be set to be interpreted as a simple LF (the default) or as a CR-LF combination. As a side effect, selecting the later option causes the ENTER key to send a CR-LF sequence rather than a single CR.

Backspace key Specifies what hex character PCWS sends to the host when the Backspace key is pressed. The default is 08h (backspace), but a 7Fh (delete) can be sent instead.

Miscellaneous Settings

Selecting this category pops-up the window below which displays a number of miscellaneous PCWS setting.

```
Printer ID    :   LPT1  LPT2  LPT3
Print Filter  :   OFF  ON
```

Printer ID Indicates to which printer PCWS should direct output when the printer function is toggled on. (For more details refer to the "Functions Menu Item" section below.)

Print Filter Specifies whether control characters should be filtered when printing data. When filtering is on, only the following control characters are interpreted: BS, LF, CR, & FF. If filtering is off, then all data received is printed.

The Functions Menu Item

When the Functions item of the menu line is selected, the following pull-down window is displayed:

```
Reset Port
Send Break
Toggle Printer
```

Reset Port	Resets the communications port. The Data Terminal Ready (DTR) signal is dropped during this time. Also, the OUT1 line is enabled on the asynchronous communications port which drops most internal modems.
Send Break	Sends a BREAK signal to the mainframe system. This signal sets the line to an all SPACE (logical 0) state for a second or so. This is normally used to interrupt the mainframe application.
Toggle Printer	Toggles the printer on and off. When the printer is toggled on, all data received from then on is sent to the printer. Also, when the printer function is turned on, an LPT1 indicator appears on the emulation mode status line.

The Help Menu Item

When the *Help* item of the menu line is selected, the following pull-down window is displayed:

```
Install
Terminal Types
Keystrokes
Mouse Support
```

Items in this window are topics on which help is available. To make a selection, use the up and down arrow keys to highlight the desired topic and press the ENTER key. PCWS searches for its help files in the same directory from which it was loaded. This should not be a problem if the correct procedures were followed in the topic "Installing PCWS" in *Chapter 1 - Introduction*.

When in *help mode* (viewing a help topic) the following keys are available:

PgUp:	View previous page
PgDn:	View next page
ESC:	Return to PCWS menu mode

The Exit Menu Item

Selecting the Exit item from the menu line terminates the PCWS program. The communication line is dropped and the program is removed from memory.

Chapter 3. Running PCWS

Chapter 3. Running PCWS

Communicating with Host

In connecting to any host, there are usually very specific details that you must know. The check lists below should be taken as guides. They provide an overview of the steps involved in the two different ways of connecting to MUSIC/SP:

1. Via an Asynchronous Communications Controller
2. Via a Protocol Converter

To obtain more information on some of the details not specified in the following sections, refer to local support personnel and other documentation relating to your equipment.

Connecting Via an Asynchronous Communications Controller

1. To start the PCWS program, enter "PCWS" at the DOS prompt. The program starts up in menu mode, meaning, the PCWS menu is displayed on the screen.

When connecting via an asynchronous communications controller, the PCWS program must be set to use the *PCWS* terminal type. This can be checked by moving the selection bar to the Terminal item of the menu. Communication settings (COM port number, transmission speed,...) should also be checked at this stage and changed if necessary. For more information on how to change settings while in menu mode please refer to *Chapter 2 - The PCWS Menu*.

2. You are now ready to exit the menu and enter emulation mode. To do so, move the selection bar to the GO item and select "Emulation Mode" from the window. Switching to emulation mode can also be done by pressing the Alt-M hotkey.

When entering the Emulation Mode for the first time, PCWS initializes the selected COM port. You are now ready to make the physical connection to the host system. In the case of a remote PC, this involves issuing special commands to the modem such as "ATDT" followed by a phone number. Once the modem has dialed the number and the host has answered, there will be some indication that the physical connection has been made.

3. At this point, press the ENTER key. The computer at the host site may respond with some message, prompt, or perhaps nothing. The next step is to make the logical connection to MUSIC/SP host system. Here, there are a number of possibilities, depending upon the hardware or software at the host site. For example, with PACX units, you must enter the class (a number which is associated with the system you want). If you reach VM, you must issue the "Dial MUSIC" command.
4. Once you reach the MUSIC/SP logo, press ENTER to go to the sign-on screen. Enter your userid and the PCWS Terminal type as follows:

```
/ID userid;PCWS
```

By specifying the "PCWS" terminal type you are informing MUSIC on how to communicate with your PC. This enables you to use special features such as the 3270 mode and the XTMUS/XTPC file transfer.

Connecting Via a Protocol Converter

1. To start the PCWS program, enter "PCWS" at the DOS prompt. The program starts up in menu mode, meaning, the PCWS menu is displayed on the screen.

When connecting via a protocol converter, the PCWS program must be set to use the VT100 terminal type. This can be checked by moving the selection bar to the Terminal item of the menu. Communication settings (COM port number, transmission speed,...) should also be checked at this stage and changed if necessary. For more information on how to change settings while in menu mode please refer to *Chapter 2 - The PCWS Menu*.

2. You are now ready to exit the menu and enter Emulation Mode. To do so, move the selection bar to the GO item and then select "Emulation Mode" from the window. Switching to emulation mode can also be done by pressing the Alt-M hotkey.

When entering the emulation mode for the first time, PCWS initializes the selected COM port. You are now ready to make the physical connection to the host system. In the case of a remote PC, this involves issuing special commands to the modem such as "ATDT" followed by a phone number. Once the modem has dialed the number and the host has answered, there will be some indication that the physical connection has been made.

3. At this point, press the ENTER key. The computer at the host site may respond with some message, prompt, or perhaps nothing. The next step is to make the logical connection to MUSIC/SP host system. Here, there are a number of possibilities, depending upon the hardware or software at the host site. For example, with PACX units, you must enter the class (a number which is associated with the system you want).

You can tell when you reach the protocol converter when a prompt similar to the one below appears:

```
"Please enter terminal type:"
```

4. Enter "VT100P". This is a terminal type especially designed with enhanced key definitions and screen colors. (VT100P keystrokes are defined in Appendix A.)

If the VT100P terminal type is not available on your protocol converter, you can use the VT100 terminal type. In this case, however, the key definitions and screen colors will be the defaults set in the protocol converter.

5. At the VM logo, press ENTER and issue the "Dial MUSIC" command. When the MUSIC/SP Logo appears, press ENTER to go to the MUSIC/SP sign-on screen. Sign on by entering your userid and password.

Automatic Connection Feature

The .CON Files

PCWS provides a way to automate the process of connecting a PC to a host system. Special files called *connection scripts* make this automatic connection feature possible. The PCWS connection scripts are ASCII text files. They can be created or modified using any PC editor.

PCWS connection scripts all have the '.CON' file name extension. These files consist of a list of instructions. Every .CON file has the same basic format: one instruction per line. The purpose of these files is to describe every step involved in getting a particular host connection.

Getting a PC logically connected to a host system involves number of specific steps. The number of these steps directly depends on the complexity of the communication link. Although connection procedures can vary in length they all basically consist of sending instructions and waiting for responses. The design of .CON files revolves around the following pattern:

```
Send a 1st instruction
Wait for proper response
Send a 2nd instruction
Wait for proper response
Send a 3rd instruction
...
```

See the topic "The .CON Commands" below for more information about creating a .CON file.

Getting Started

To start up the automatic connection process, you must specify which .CON file PCWS is to process. This can be done in two ways:

- a) Through a command line argument:

When PCWS is first invoked, the name of a .CON file can be specified as a command line argument. This informs PCWS to bypass the main menu and immediately start processing the connect file. For example,

```
PCWS pcwshome
```

Entering the above at the DOS command line will have PCWS automatically start processing the PCWSHOME.CON file.

- b) Through the PCWS menu:

The second way of specifying a .CON file is through the main menu. To do so, select the Go item of the menu line to display the following pull-down window.

```
Emulation Mode
Auto Connect
Back to DOS
```

Selecting the Auto Connect item instructs PCWS to start searching for connection scripts. The directory PCWS searches depends on if the PCWS environment variable is defined or not. (The PCWS environment variable is described in the topic "Installing PCWS" in *Chapter 1 - Introduction*.) If defined, PCWS searches the directory the environment variable specifies. Otherwise, the program searches the current directory.

If its search is successful, PCWS then pops-up a second window displaying the file names of all .CON files it found. The up and down arrow keys can be used to move the selection bar to the desired file name. Pressing the ENTER key selects the highlighted file.

Once a .CON file is selected, PCWS starts processing its instructions. These get executed one after the other just as if they were typed in by you. A properly written .CON file allows your PC to automatically establish a logical host connection.

The .CON Commands

All .CON files have the same format of one instruction per line. Furthermore, every line must start with a valid .CON command. Only the commands listed below can be part of a PCWS connection script.

BREAK Command

The BREAK command sends a BREAK signal on the line.

COMMENTS

A line starting with a semicolon (;) is interpreted as a comment line and is therefore ignored by PCWS.

DELAY Command

```
DELAY [sec]
```

The DELAY command tells the PC the PC to pause before executing the next .CON command. The optional argument specifies the number of seconds to delay. The default is 1 second. The DELAY command can be used in front of a WAIT command to increase its timeout. It can also be used right before a SEND command if dialing in from home (i.e. if using a slower speed).

FLUSH Command

The FLUSH command instructs PCWS to ignore the data received so far. PCWS does so by clearing the buffer where it saves all characters arriving at the serial port. To prevent the overflow of this buffer, the FLUSH command should be used about every second "screenfull" you get.

PROCESS Command

The PROCESS command instructs PCWS to process the data received so far. Note however that most of the data PCWS receives (and saves in an internal buffer) while attempting to connect can be ignored. The one string you want to PROCESS, and not FLUSH, is the one the protocol converter sends after you entered the terminal type. This is the important "initialization" string.

```
;---- Send VT100P as the terminal type
send VT100P
;---- Wait for VM logo
wait VM
;---- Process the "initialization" str
process
```

SAY Command

```
SAY string
```

The SAY command simply displays the specified string on the screen. It is normally used to inform you that a particular step has just been completed. The SAY command allows the user to keep track of where the processing of the .CON file is at. This can be particularly useful in longer .CON files.

SEND Command

Format 1: SEND [/N] string

In this first format, the SEND command sends the specified string followed by an ENTER key to the host system. The optional /N switch is used to send the string without the ending ENTER key.

Format 2: SEND @x

The SEND command can also be used to send special keys to MUSIC. To do so, use the SEND @x format where @x is a code representing a specific key. The following table indicates which @x codes correspond to which keys:

<u>Code</u>	<u>Key</u>	<u>Code</u>	<u>Key</u>
@E	Enter	@T	Tab
@1-9	PF1-PF9	@x	PA1
@a-c	PF10-PF12	@y	PA2

WAIT Command

WAIT string

The WAIT command usually follows a SEND command. It forces the PC to wait until it receives the specified string. This string must be part of the screen image we are expecting. Once the string is received, PCWS then continues processing the .CON file.

The WAIT command will timeout after 10 seconds. To increment this value, use a "DELAY x" command just before the WAIT. For example, the following commands allow a total of 20 seconds to get the CONNECT message back from the modem:

```
;--- Send Phone Number and wait for CONNECT
send   ATDT3988111
delay  10
wait   CONNECT
```

Example of .CON File

The following example shows a typical .CON file that can be used to connect a remote PC to MUSIC.

```
;-----
; PCWSHOME.CON
; Example of a PCWS .CON file.
; Connects to MUSICM system via Modem.
;-----
delay
send   ATZ
wait   OK
say    Connected to Modem
;
delay
send   ATDT8111
delay  10
wait   CONNECT
say    Connected to PACX
```

```

;
delay
send @E
wait class
;
delay
send 13
wait start
say Connected to MPG CPU
;
delay
send @E
wait CONNECT
;
delay
send @E
wait type
flush
;
delay
send VT100P
wait PP
say Connected to 7171
process
;
delay
send @E
wait DIAL
;
delay
send D MUSIC
wait MMM
say Connected to MUSIC
flush
;
delay
send @E
wait ID
;
;-----
; Note: You can have this script file Sign-On for you by adding
; the following 3 lines: (id= Your ID, & ps= Your Password).
; send /N id
; send @T
; send ps
;-----

```

PCWS Program Control Keys

A number of Emulation Mode keystrokes are referred to as the PCWS Program Control Keys. These always behave the same way whether using the native PCWS or the VT100 terminal type. The list below describes the functions associated with these control keys.

Back to DOS

The Alt-D combination can be used to temporarily exit from your host session and run another PC application. The Alt-D key combination puts DOS back in control and leaves only the PCWS resident part in memory. You can return to your communication session by entering PCWS at the DOS prompt.

Help

The Atl-H combination selects the HELP item of the main menu. A help topic can then be selected of the pull-down window.

Capture Screen

The Alt-I combination saves the screen contents to a file. This file is called PCWS.SCR and is located in the directory specified by the PCWS environment variable. If this environment variable doesn't exist the file is created in the current directory. If the file already exists, the data is simply appended to it. For more details on the PCWS environment variable please refer to "Installing PCWS" in Chapter 1 - Introduction.

Main Menu

The Alt-M combination selects the GO item of the main menu. This combination is in fact a hotkey to switch from Emulation mode to Menu mode and vice-versa.

Exit PCWS

The Atl-X combination selects the EXIT item of the main menu. This provides for a fast way of exiting PCWS.

Mouse Support

If the Microsoft Mouse driver is loaded, a mouse can be used to move the cursor and to detect *hotspots* in 3270 full-screen applications.

If PCWS finds the mouse driver properly installed, it displays a left pointing arrow at the last position of the status line. The status line is displayed at the bottom of the screen when PCWS is in emulation mode. Clicking any mouse button on the "Alt-M:Menu" string of the status line is a fast way to switch into PCWS's menu mode.

Moving The Cursor

Pressing the left button once causes PCWS to move the hardware cursor to the location of the highlighted mouse cursor. Pressing the right button also moves the cursor plus simulates pressing the ENTER key.

```
Left Button (click) : Move Cursor
Right Button (click) : Move Cursor & press ENTER key
```

HotSpots

The PCWS hotspot support enables you to use the mouse to simulate pressing function keys. This is accomplished by first moving the mouse cursor to a hotspot location and then double-clicking the left button.

PCWS recognizes, as a hotspot, any string on the screen starting with one of the following patterns:

```
n:, nn:, Fn, Fnn, PFn, PFnn, An, PAn
```

(The colons in the above are placeholder for any non-blank char.)

Many applications have a function key legend at the bottom of the screen. As an example say the string "F3:Exit" is displayed on the screen. Double-clicking the left mouse button anywhere on this text causes PCWS to automatically press the F3 key.

```
Left Button (double-click) : On HotSpot : Press Function key
                          : On Blank    : Press ENTER key
                          : Otherwise   : Enter Block Mode
```

Block Mode

The PCWS mouse support includes a Block Mode which allows you to use the mouse to enter an alphanumeric key in any input area of the screen.

To enter Block Mode, double-click the left button on a character which isn't part of a HotSpot. The blinking mouse cursor indicates you are in Block Mode. At this point, you can drag the blocked character around the screen and copy it to any number of input areas. To do so, first move the blocked character to where you want to copy it and then click left button once. Notice that while in Block Mode, the following string is displayed on the status line (last line of screen):

```
[Block Mode: <Cancel> <Process>].
```

To exit Block Mode, place the mouse cursor on either the "Cancel" or "Process" word. Clicking on "Process" causes the host to process the data entered while in Block Mode. On the other hand, clicking on "Cancel" clears the screen and causes the data entered in Block Mode to be ignored.

Chapter 4. The PCWS Terminal Type

Chapter 4. The PCWS Terminal Type

Introduction

The PCWS terminal type is a special one, designed specifically for communicating over an asynchronous line with the MUSIC/SP host system. This terminal type has two different modes of operation:

1. PAGE Mode
2. 3270 Mode

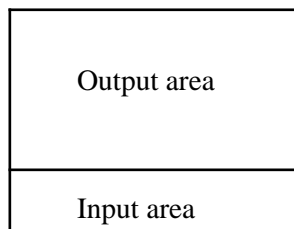
The initial mode of the PCWS terminal type is known as the PAGE mode. In this mode, the PC basically emulates a standard line-by-line TTY terminal. PAGE mode remains the normal mode of operation until a MUSIC/SP full-screen application is started. The PCWS terminal type then automatically switches into 3270 mode. This second mode allows full-screen applications (such as the Editor, Mail, etc.) to run on the PC as if it was a real 3270 terminal. Upon terminating the 3270 application, PCWS automatically switches back into PAGE mode.

PAGE Mode

PAGE mode is the initial terminal emulation mode of the PCWS terminal type. It allows the PC to operate as a line-by-line TTY terminal with several unique features. One such feature is that every printable character received at the COM port is saved in a circular buffer. A limited number of control characters get interpreted in the standard way. These are: BS (Backspace), LF (Line Feed), CR (Carriage Return) and FF (Form Feed).

The buffer, where incoming characters are saved, is called the PAGE buffer. The size of this buffer is set in terms of pages. A PCWS *page* is defined as 21 lines of data. By default, the PAGE buffer is set to contain 5 pages, numbered from 00 to 04.

The idea of the PAGE buffer is to provide the ability to view previously received data. By saving the incoming bytes, PAGE mode allows you to conveniently go back and view data which has scrolled off the screen. Hence, PAGE mode refers to the capability to page back and forth through the output the PC has received.



The figure above illustrates the PAGE mode screen format. A line across the bottom divides the screen in two parts called the output and input areas.

PAGE Mode Input Area

The *input area* consists of the line just below the separation line. Text to be sent to the communications port can be entered in this area. Typing in text results in commands being sent directly to the modem and/or the host system. By default, the input area operates in *Line* mode. In this mode, the data is only sent to the

COM port when the ENTER key is pressed. You can modify a line of data before sending it. The local editing functions include insert, delete, backspace and cursor positioning.

The input area can also be set to operate in *Character* mode. In this mode, the characters are sent as they are typed. Line editing is now impossible. Characters appear in the input area as they would in Line mode. Pressing the Enter key sends a CR (carriage return) and clears the input area. The mode in which the input area operates is one of the PAGE mode settings you can change. For more information please refer to the "Settings Item" section of *Chapter 2 - The PCWS Menu*.

PAGE Mode Output Area

The *output area* can be considered as a window where a page from the PCWS PAGE buffer is viewed. When first entering PAGE mode, you are presented with the first page of the buffer, PAGE 00. This page is where PCWS starts saving the incoming data bytes. Displaying PAGE 00 allows you to view the first 21 lines of data.

Note: Since PAGE 00 starts out empty, the output area remains blank until the COM port actually receives data.

One of the advantages of PAGE mode, is the ability to page up and down in the PAGE buffer. When the current page fills up, a "More..." message appears in the bottom right-hand corner of the screen. Press PgDn to see the next 21 lines of data. Pressing the PgUp key allows you to view the previous page of data. Moving up and down through the PAGE buffer is analogous to editing a file and paging up and down.

Pressing the Home key displays the *TOP* logical page while pressing the End key allows the user to display the *BOTTOM* one. The Up and Down arrow keys can be used to move up or down 1 segment. Every buffer page is divided into 3 segments of 7 lines each. These segments are numbered from 0 to 2. The Status Line (the last line of the screen) always indicates the page and segment currently displayed. As an example, when first entering PAGE mode the string 'PAGE 00s0' is displayed on the Status Line. This indicates that you are currently viewing the first segment of the first page.

PAGE Mode Line Calldown

The *Line Calldown* feature allows the user to point to a line displayed in the top part of the screen and retrieve it into the input area. The line can then be modified using the local editing functions and re-transmitted to MUSIC/SP. This feature is only enabled when the input area is in line mode.

Pressing the "+" key on the keypad toggles the Line Calldown mode on and off. When activating this feature, a highlighted bar marks the top line of the screen. The up and down arrow keys can then be used to move this bar to the desired line. Once this step is completed, pressing the ENTER key copies the marked line to the input area.

PAGE Mode Keyboard Functions

This section lists the PAGE mode keyboard functions and specifies the default keystrokes associated with each of these.

Page Up & Page Down

The PgUp key moves up 1 page in the PAGE buffer and thus displays the previous 21 lines of data to the screen. The PgDn key reverses this direction displaying the next logical page of the buffer.

Top & Bottom

The Home and End keys respectively display the logical TOP and BOTTOM page of the PAGE buffer.

Up & Down Arrow

The Up arrow key moves up 1 segment in the PAGE buffer and thus displays the 7 previous lines of data to the screen. The Down arrow key reverses this direction displaying the next logical segment of the buffer. If Line calldown is active, the up and down arrow keys are used to move the highlighted bar to the desired line.

Backspace

The Backspace key moves the cursor left one location and deletes the character at that position. If the cursor is in column 1, just the delete function is performed. When in Character mode, the Backspace key sends out a x'08', but does not provide editing in the input area.

Enter

The ENTER key sends the data typed in the input area to the host system. The input area is then cleared and the cursor is returned to column 1. If the input line is in character mode, then only a carriage return is sent and the input area is cleared. If Line Calldown is active, then the line marked by the highlighted bar is copied to the input area.

Line Calldown

The "+" key on the keypad toggles the Line Calldown feature on and off.

The PCWS Program Control Keys

<u>Functions</u>	<u>Key</u>	<u>Description</u>
Back to DOS	Alt-D	Back to DOS. Type PCWS to return.
Help	Alt-H	Select HELP item of Main Menu.
Capture Screen	Alt-I	Save screen to file PCWS.SCR.
Main Menu	Alt-M	Select GO item of Main Menu.
Exit PCWS	Alt-X	Select EXIT item of Main Menu.

For more details please refer to the "PCWS Program Control Keys" section of *Chapter 3 - Running PCWS*.

3270 Mode

When a MUSIC/SP full-screen application is started, the PCWS terminal type automatically switches from PAGE mode to 3270 mode. 3270 mode allows full-screen applications (such as the Editor, Mail, etc.) to run on the PC as if it was a real 3270 terminal. Upon terminating the 3270 application, PCWS automatically switches back into PAGE mode.

The Field Concept

Full-screen applications are programs which use the capability of 3270 terminals to divide the screen in a number of input and output areas. Using 3270 terminology, these screen areas are called *fields*.

A full-screen application can define as many fields as it needs. It does so by specifying the position, the length and the characteristics of every field it creates. Some of the characteristics a field might have are listed below:

High or Low Intensity

The intensity of a field affects the brightness of the displayed data. A high intensity field displays data at a brighter level than a low intensity (normal) one. A program normally sets a field to high intensity as a means of highlighting it.

Protected or Unprotected

Protected fields define program output areas. Programs use these fields to display results, instructions, messages,... Unprotected fields correspond to user input areas. You can only modify or enter data in unprotected fields.

Non-Display

Data typed in a non-display field does not get displayed. For example, passwords are often entered in non-display fields.

Autoskip

Once an autoskip field has been filled with data, the cursor automatically moves to the next unprotected field.

Numeric-Only

A numeric-only field will only accept numbers as input.

3270 Colors

PCWS allows you to change the colors used in full-screen applications. There are four PCWS settings which define the color of four different 3270 field types. These settings control the colors mapped to the following four field combinations:

- | | |
|-------------------|-------------|
| - Low Intensity, | Unprotected |
| - High Intensity, | Unprotected |
| - Low Intensity, | Protected |
| - High intensity, | Protected |

For more details on how to change the 3270 field colors, please refer to the "Settings Item" section of *Chapter 2 - The PCWS Menu*.

3270 Mode Keyboard Functions

This section lists the 3270 mode keyboard functions and specifies the default keystrokes associated with each of these. The 3270 functions keys can basically be divided into 2 groups: *local editing* and *action* keys.

The *local editing* keys allow you to move the cursor around the screen and to modify data in unprotected fields. All local editing keys are processed locally by the PC. Pressing a local editing key doesn't create any interaction between the PC and the host system. To inform the host of the latest screen changes you must press one of the action keys.

The *action* keys are the ones which cause the PC and the host computer to exchange information. Pressing an action key sends a unique sequence to the host which always starts with a byte known as the AID (Attention Identification) byte. This AID byte specifically indicates which action keys was pressed.

The 3270 Local Editing Keys

Tab

The Tab key performs the 3270 tab function. Pressing this key advances the cursor to the next unprotected field.

Back Tab

The Shift-Tab key performs the 3270 backtab function. Pressing this key moves the cursor to the beginning of the previous unprotected field.

Home

The Home key performs the 3270 home function. Pressing this key moves the cursor to the first unprotected field on the screen.

Cursor Keys

The up, down, left and right arrow keys move the cursor around the screen. The cursor wraps around when reaching any edge of the screen.

The 3270 Action Keys

Enter

The ENTER key performs the 3270 enter function. Pressing this key sends the latest user input to the host.

PA1

The "-" key on the keypad performs the 3270 PA1 function. The PA1 key transmits a unique code which is usually interpreted by the application program as an attention signal.

PA2

The "+" key on the keypad performs the 3270 PA2 function. The PA2 key transmits a unique code which is usually interpreted by the application program as a request to clear the screen and display additional information.

Clear

The "Esc c" sequence performs the 3270 clear function. This function clears the first 24 lines of the screen.

Fnn

The Program Function keys (F1 - F12) are defined by the application program that is running. For more information about what the function keys do, refer to the documentation for the full-screen application that is running.

The PCWS Program Control Keys

<u>Functions</u>	<u>Key</u>	<u>Description</u>
Back to DOS	Alt-D	Back to DOS. Type PCWS to return.
Help	Alt-H	Select HELP item of Main Menu.
Capture Screen	Alt-I	Save screen to file PCWS.SCR.
Main Menu	Alt-M	Select GO item of Main Menu.
Exit PCWS	Alt-X	Select EXIT item of Main Menu.

For more details please refer to the "PCWS Program Control Keys" section of *Chapter 3 - Running PCWS*.

Switch to PAGE Mode

The Alt-P combination make PCWS temporarily re-enter PAGE mode. This allows you to view the latest *pages*. At this point, the only valid keys are PgUp, PgDn, Up Arrow & Down Arrow. Any other keystrokes make PCWS switch back into 3270 Mode.

Chapter 5. The VT100 Terminal Type

Chapter 5. The VT100 Terminal Type

Introduction

The VT100 terminal type of PCWS allows a PC to emulate the DEC (Digital Equipment Corporation) VT100 terminal. This VT100 emulation capability enables a PC to be connected via a protocol converter.

A protocol converter is a device allowing ASCII terminals to connect to a host computer as if they were IBM 3270 terminals. A PC using the VT100 terminal type of PCWS appears to the host as a regular 3270 terminal. The converter's role is to translate the 3270 data streams it receives from the host into ASCII control sequences the terminal recognizes, and vice versa. As for PCWS, it makes sure the PC processes incoming and outgoing data just as a real VT100 terminal would.

PCWS allows you to modify a number of VT100 mode settings. These settings define how the program is to handle some of the VT100 functions. For more information please refer to "VT100 Settings" in *Chapter 2 - The PCWS Menu*.

Device Definition Table

A protocol converter holds a 'Device Definition Table' (DDT) for every ASCII terminal it supports (including VT100). Converters such as the 7171 and the 9370 ASCII Subsystem are shipped with default DDTs for a number of terminals. Each of these supplied DDTs allows a different terminal type to be connected to the converter.

A DDT is a list of all control sequences the terminal recognizes along with the respective functions to which they correspond. It is thus the DDT which instructs the protocol converter on how to interpret the control sequences it receives. During the connection procedure, the protocol converter must find out which DDT to use. To do so, it issues the following prompt:

```
"ENTER TERMINAL TYPE:"
```

Your answer tells the converter which table to use for that particular connection. From then on, the converter searches this table for every control sequence it receives and then performs the desired function.

A special DDT called 'VT100P' is distributed with MUSIC/SP and should be installed on your site's converter by a system's administrator. The entries of this table are specially designed so that the key definitions and screen colors are compatible with the 3270 mode of PCWS. The keystrokes for the default 'VT100' and the special 'VT100P' terminal types are specified in Appendix A.

VT100 Mode Keypad

On a VT100 terminal, the numeric keypad can be set in 2 different states: Numeric and Application Mode. The PCWS VT100 terminal type supports these 2 keypad modes and allows the user to toggle between them by pressing the 'NumLock' key.

The PC keypad operates in VT100 Application mode when 'NumLock' is off. In this case, keys are defined to send escape sequences that may be used by the host as user defined functions. It is the internal DDT that indicates which escape sequence corresponds to which function. Thus, with a properly set DDT the keypad

keys can be used to access some of the 3270 functions such as INSERT (the '0' key), HOME (the '7' key),...

One difficulty in implementing an emulator of a VT100 terminal on a PC is the handling of the up, down, left and right arrow keys. The problem arises from the fact that on a real VT100 terminal the cursor movement keys are located off the keypad. On a PC however, these cursor keys are imbedded in the keypad. For this reason, the '2','4','6' and '8' keypad keys do not send out user definable escape sequences but are rather used as the cursor keys.

The PC keypad operates in VT100 Numeric mode when 'NumLock' is on. In this case, pressing a key on the keypad simply sends the ASCII code corresponding to the numeric character on the key.

VT100 Mode Keyboard Functions

This section lists the VT100 mode keyboard functions and specifies the default keystrokes associated with each of these.

Return	By default, the ENTER key simply sends a CR (x'0D') character. If however the VT100 New Line option is set to 'CR/LF', the ENTER key then sends a CR/LF sequence. See "VT100 Settings" in <i>Chapter 2 - The PCWS Menu</i> .
Line Feed	Pressing the "*" key on the keypad sends a LF (x'0A') character.
Tab	The Tab key sends a tab character (x'09') to the host.
Esc	The Esc key sends an escape character (x'1B').
Backspace	By default, the Backspace key sends a backspace character (x'08') to the host. If however the VT100 Backspace parameter is set to '7Fh', the backspace key sends a Del character (x'7F'). See "VT100 Settings" in <i>Chapter 2 - The PCWS Menu</i> .
NumLock	The NumLock key toggles the keypad from Application mode to Numeric mode and vice versa.

The PCWS Program Control Keys

<u>Functions</u>	<u>Key</u>	<u>Description</u>
Back to DOS	Alt-D	Back to DOS. Type PCWS to return.
Help	Alt-H	Select HELP item of Main Menu.
Capture Screen	Alt-I	Save screen to file PCWS.SCR.
Main Menu	Alt-M	Select GO item of Main Menu.
Exit PCWS	Alt-X	Select EXIT item of Main Menu.

For more details please refer to the "PCWS Program Control Keys" section of *Chapter 3 - Running PCWS*.

Chapter 6. File Transfer

Chapter 6. File Transfer

Introduction

File transfer between the MUSIC/SP host system and the PC is supported by PCWS. Files can be transferred in both directions: from the PC to MUSIC and vice-versa. The PCWS file transfer facility is designed to be both powerful and easy to use.

A file transfer is initiated by entering the XTMUS or XTPC command from the MUSIC/SP session. The command XTMUS (transfer to MUSIC) is used to transfer files from the PC to MUSIC. The command XTPC (transfer to the PC) on the other hand is used to transfer files from MUSIC to the PC. Once the command has been entered, PCWS automatically displays a file transfer monitor window to indicate the progress of the file transfer.

Text and Binary Modes

Personal computers use a code called ASCII (American National Standard Code for Information Interchange) to represent characters internally, but IBM host systems use a code called EBCDIC (Extended Binary-Coded Decimal Interchange Code). This difference must be considered when transferring files between systems.

By default, PCWS transfers files in *text* mode. This mode will correctly translate alphabetic, numeric, and punctuation characters from one code system to the other. PCWS uses the CR-LF sequence to separate records on a PC. When transferring files in text mode from the PC to MUSIC, data bytes higher than X'7F' lose the high order bit and get translated.

Some files, such as program files, are only useful on the system that it was designed for. Therefore, when transferring these files, you want to preserve the exact format and contents of the file. The -Bin option allows you to transfer a file in *binary* mode without any translation (preserve all data bytes).

File Names

XTPC and XTMUS use the following file name conventions:

- PC file names are given in their standard DOS format. This allows for a 1 character drive followed by a colon, a 50 character path name, and an 8 character file name followed by a period and then the 3 character file type. The path separator character is the back slash. Thus the maximum length of a PC file name is 64 characters. Examples of valid names are: FILE1, C:\PCWS\FILE2, A:TEST.COM.
- MUSIC file names can have a userid prefix. For more information about file names, refer to the *MUSIC/SP User's Reference Guide*, or type "HELP FILES" in *Go mode.

File names can contain any number of wildcard characters. The only restriction is that if a *dest* is specified, then it must contain exactly the same number of wildcard characters as the *src*. The two wildcard characters allowed are the "*" (asterisk) which maps to any string, including the null string, and the "?" (question mark) which maps to exactly any one character.

When transferring files to the PC, the *dest*, which is the PC destination file name may be a drive prefix alone,

or a path ending with a back slash. This transfers a file from MUSIC to the PC using the same file name.

Similarly, when transferring files to MUSIC, the *dest* may be a userid alone ending with a colon (":"). This transfers a file from the PC to MUSIC using the same file name.

The "*" file name specification on the PC matches **all** files, but the "*.*" specification **only** matches file names that have file name extensions. If you want to transfer all files that do not have file name extensions, use the "*." specification.

XTMUS and XTPC Commands

Format 1:

```
XTMUS src [dest] [-BIN] [-REPL] [-APP] [-COMP] [-Fx] [-n]
                [-B ] [-R  ] [-A  ] [-C  ]

                [-MON] [-SWITCH] [-NOMSGS]
                [-M  ] [-S      ] [-N      ]
```

XTMUS transfers a file from the PC to MUSIC. *src* is the name of the PC file to be transferred. The optional *dest* is the name of the file to be created on MUSIC. If the *dest* is omitted then the *src* file name is used as the receiving file name.

```
XTPC src [dest] [-BIN] [-REPL] [-APP] [-COMP]
                [-B  ] [-R   ] [-A  ] [-C  ]

                [-MON] [-SWITCH] [-NOMSGS]
                [-M  ] [-S      ] [-N      ]
```

XTPC transfers a file from MUSIC to the PC. *src* is the name of the MUSIC file to be transferred. The optional *dest* is the name of the file to be created on the PC. If the *dest* is omitted then the *src* file name is used as the receiving file name.

Optional Parameters

Note that abbreviations for the following parameters are shown in upper case (i.e. -Bin = -B).

- Bin allows the user to transfer files without translating the data between ASCII and EBCDIC. See the topic "Text and Binary Modes" above for more information.
- Repl informs the destination system to replace the file if it already exists. If the -Repl option is not specified, and the file already exists on the destination system, then the file transfer is aborted.
- App informs the destination system to append the information to the end of a file that already exists.
- Comp turns on compression to reduce the amount of data transmitted between the two systems.
- Mon disables the file transfer monitor and turns off printing of messages.
- Switch automatically turns off the transfer monitor when the file transfer is complete. Normally, you need to press the Esc key to switch it off.

-Nomsgs turns off printing of all messages including error messages.

The following options are available only with the XTMUS command:

-Fx changes the record format of the receiving file on MUSIC. "x" can be F, FC, V, or VC. The default is VC record format.

-n changes the record length of the receiving file but is not normally required. Text files are uploaded by default as VC (Variable Compressed) records. The record length is automatically adjusted to match text being uploaded. Binary files uploaded using the -Bin option are stored by default as 80 byte FC (Fixed Compressed) records.

Format 2:

```
XTMUS <listname
```

```
XTPC <listname
```

This format of the XTMUS and XTPC commands makes it possible to transfer a list of files. *listname* is the name of a MUSIC file which contains the list of files to be transferred. All lines of the listname file must have the following format:

```
src [dest] [optional parameters]
```

Files are transferred one after the other. Before transferring the next file in the list, the optional parameters are reset. This format of the XTMUS and XTPC commands does not support the [-M] and [-N] parameters.

Return Codes

At the end of the file transfer, both the XTMUS and XTPC programs set an end of job (EOJ) return code. These codes can be useful when using XTMUS or XTPC in REXX programs. The return codes are:

- 0 - file transfer okay and ended normally
- 1 - can not run file transfer on this terminal
- 2 - invalid parameter
- 3 - can not open MUSIC file
- 4 - can not open PC file
- 5 - an error occurred during file transfer
- 6 - file already exists, use -REPL or -APP option
- 7 - (XTMUS only) can not save MUSIC file at end of file transfer

Note: XTPC can not distinguish between PC errors on opening a file, so return code 6 is not used, but is seen as return code 4.

Examples

Example 1:

```
XTPC *.OBJ -B
```

copies all of the user's MUSIC object modules to the PC, and the files on the PC will have the same names as

the MUSIC files. The binary option is used since the files contain many non-printable characters.

Example 2:

```
XTMUS A:\DIR1\TEST.1 ABCD:TEMP.1
```

copies the file TEST.1 from directory \DIR1 on drive A on the PC to the MUSIC file TEMP.1 under the userid ABCD.

Example 3:

In the following example the MUSIC file called SAMPLE contains the following:

```
A:file1.txt file1.txt
B:file2.txt file2.txt
```

The command "XTMUS < SAMPLE" causes the PC files A:file1.txt and B:file2.txt to be transferred to MUSIC.

Example 4:

```
XTPC file1 C:\DIR1\DIR2\
```

transfers *file1* from MUSIC to the PC. The PC file will be called *file1* in the directory C:\DIR1\DIR2.

K-PCWS : PCWS Kermit Support

KERMIT is a popular file transfer protocol designed at Columbia University. K-PCWS is a program distributed as part of PCWS. It is a PC implementation of the Kermit protocol allowing PCWS users to do file transfer with a wide variety of systems including MUSIC/SP.

In order to transfer files using a Kermit protocol, a Kermit program must be running on each end of the communication line. To start the Kermit program on the host system side, first use PCWS to establish the logical connection. Then at the host's system prompt (*GO for MUSIC/SP) enter KERMIT.

At this point you can use the Kermit SET command to set various protocol parameters. You can also place the remote Kermit into server mode by entering the SERVER command. This step is required for the SEND and GET commands of K-PCWS to work.

To start the Kermit on the PC end, first press the Alt-D combination to temporarily get back to the DOS prompt. There, enter the K-PCWS command. The "Kermit-PCWS>" prompt indicates that you are now ready to initiate a file transfer.

Example of a KERMIT session

1. Start the KERMIT on the host side of the connection. To do so, enter the KERMIT command at the host's system prompt (ex: *GO for MUSIC/SP). You are then presented with the host's KERMIT prompt (ex: Kermit-MUSIC>).
2. If the file(s) to be transferred are binary files, enter the following command: SET File Type Bin. Optionally, you can then issue a SHOW command to verify the settings of the other SET parameters. Note that if the file(s) to be transferred are plain text file(s), the file type should be specified as TEXT. To do so, enter the following command: SET File Type Text.

3. Enter the SERVER command.
4. Press the Alt-D combination to temporarily set yourself back to the DOS prompt.
5. Start the KERMIT on the PC side of the connection by entering the K-PCWS command. This will display the "Kermit-PCWS>" prompt.
6. You are now ready to specify the file(s) you wish to transfer. Use the SEND command to transfer files from the PC to the host. Use the GET command to transfer files in the other direction. The SEND and GET commands are documented in the section below.
7. Enter the FIN command to end the remote Kermit Server mode.
8. Enter the EXIT command to end K-PCWS. At the DOS prompt, enter PCWS to return to the host session.

K-PCWS Commands

This section describes the various commands that the PCWS version of KERMIT supports. These commands can all be entered at the "Kermit-PCWS >" prompt.

Entering a '?' character will cause K-PCWS to display the following summary of the commands it supports:

bye	Shutdown remote Kermit server and exit from K-PCWS.
exit	Exit from K-PCWS and return to DOS.
fin	Shutdown server (without logging out), don't exit K-PCWS.
get	Get a file from the host (host must be in server mode).
local	Issue a DOS command.
quit	Exit from K-PCWS and return to DOS.
remote	Ask the server to perform a cmd, send results to your screen.
send	Send a file to the other KERMIT.
set	Establish various non-standard settings.
show	Display the values of SET options.

BYE Command

Shutdown remote Kermit Server and exit from K-PCWS. At the DOS prompt, enter PCWS to return to the host session.

EXIT Command

The EXIT command ends the K-PCWS program and returns you to the DOS prompt. There, enter PCWS to return to the host session.

FIN Command

End the remote Kermit Server mode. Remain at the "Kermit-PCWS" prompt.

The GET Command : GET HostFile [PcFile]

The GET command requests a remote Kermit server to send the file(s) specified by "HostFile". The "Host-File" may contain wildcard characters. If "HostFile" specifies a single file, you can have K-PCWS save it under another name on the PC. This can be done by specifying the optional "PcFile" argument.

```
Ex.1: Get *.c          (Host sends all files having a '.c' extension)
Ex.2: Get abc.c a.c   (Host sends file 'abc.c' to be saved as 'a.c')
```

The LOCAL Command : LOCAL [DOS command or program]

The LOCAL command allows a DOS command or program to be executed from within K-PCWS.

```
Ex: LOCAL dir /p
```

The QUIT Command

The QUIT command ends the K-PCWS program and returns you to the DOS prompt. QUIT is the same as EXIT.

The REMOTE command

The REMOTE command indicates to the remote Kermit server that we want it to perform a command. The following are the REMOTE commands K-PCWS supports:

REMOTE DElete filespec

Ask server to delete the specified file(s) on the remote system.

REMOTE DIrectory [filespec]

Ask server to display a directory listing of specified files.

REMOTE Host [command]

Ask server to send command to the remote system's command processor.

REMOTE Kermit command

Send a Kermit command to be executed by the remote Kermit. For example:

```
Remote Kermit show
```

REMOTE Space

Ask server to provide a brief summary of disk usage.

The SEND Command : SEND PcFile [HostFile]

The SEND command causes a file or file group to be sent from the PC to the host system. The "PcFile" may contain the DOS '*' and '?' wildcard characters. If "PCFile" specifies a single file, you can have Kermit save it under another name on the host system. This can be done by specifying the optional "HostFile" argument. Examples:

```
Send *.c          (Sends all files having a '.c' extension)
Send abc.c a.c   (Send file 'abc.c' and save it as 'a.c')
```

The SET Command

The SET command allows K-PCWS to set various parameters:

SET Block-check [1|2|3]

Default is 1. Specify the block check type to use.

SET Debug [0|1]

Default is 0. Setting Debug to 1 will cause K-PCWS to save the packets being exchanged in the KERMIT.DEB file.

SET Tab expand [0|1]

Default is 0. Setting it to 1 could be useful when sending certain text files to the remote host. When on, this feature will use spaces to expand tab characters to every 8th column.

The following SET commands specify various SEND (S) or RECEIVE (R) parameters.

SET [S|R] End of packet Nbr

To specify character that ends packets. Normally 13 (Carriage Return).

SET [S|R] PACket-Length Nbr

To specify packet length to use.

SET [S|R] PADChar Nbr

To specify padding character. Normally 0.

SET [S|R] PADDing Nbr

To specify number of padding characters. Normally 0.

SET [S|R] Start of packet Nbr

To specify control character to start the packets. Normally 1.

The SHOW Command

The SHOW command will display the current values of the parameters which can be changed by the SET command.

Chapter 7. MUSIC and PC Interfacing

Chapter 7. MUSIC and PC Interfacing

Overview of PCWS Interfaces

This chapter first describes MUSIC programs specially designed to interface with PCWS. Additionally, examples are given of PC interfacing applications which can be created using the different PCWS features.

TOTTY and TO3270

When connected via a protocol converter (7171 or 9370 ASCII subsystem), the TOTTY and TO3270 programs allow you to dynamically turn on and off the transparent mode support. For more information about transparent mode support, refer to the *MUSIC/SP Administrator's Reference*.

TOTTY places the user in ASCII transparent mode. This means although connected via the 7171, the terminal will behave as if it is directly connected to MUSIC via an ASCII line. The syntax is:

```
TOTTY [PCWS] [NOMSGS]
```

The first parameter is the terminal type. If PCWS is not specified, the standard TTY mode will be used. NOMSGS can be used to turn off the program messages.

TO3270 disables the transparency support and places the user in 3270 emulation mode. The syntax is:

```
TO3270 [NOMSGS]
```

Both TOTTY and TO3270 terminate by setting an end of job (EOJ) return code. These return codes are:

- 0 - Okay, did not do switching
- 1 - Okay, did switching
- 2 - Error, can not do switching on this terminal
- 3 - Error, invalid parameter

PCEXEC

```
PCEXEC command [-Hold] [-Direct]  
               [-H   ] [-D   ]
```

PCEXEC executes the specified DOS command or PC program as though it had been entered at the DOS prompt. The -H option is used to hold the DOS screen before returning to MUSIC. For example,

```
PCEXEC dir /p -h
```

displays the PC current directory, pausing at every page before returning to MUSIC.

If it is a PC program that is to be executed, as opposed to a DOS internal command, the -D option can be used so that information on whether the execution was successful or not is returned. This can be useful when using PCEXEC in REXX programs.

PCEXEC end of job (EOJ) return codes are:

- 0 - Okay
- 1 - Can not run PCEXEC on this terminal (must use PCWS)
- 2 - Invalid value for len (0<len<64)
- 3 - Write/read error while talking to the PC
(Codes 4,5 and 6 are only supported when -D is used)
- 4 - Program not found on PC
- 5 - Could not execute program on PC
- 6 - Error while executing program on PC.

Examples of MUSIC and PC Interfacing

Example 1 - Sending a PC file to a MUSIC printer.

This REXX program, uses XTMUS to send a PC file to a MUSIC printer. If the printer is "MUSIC" then the OUTPUT program is invoked.

```
/INC REXX
/* Print a PC file at a MUSIC printer.
   XTMUS is used and it assumes the record length of the PC file.
   When we print a file on MUSIC we can only print
   the first 132 characters of each record in the file.
*/

PARSE UPPER ARG PCFILENAME DESTINATION
if DESTINATION = '' then DESTINATION = 'MUSIC'

/* make sure we have a good destination */

CALL ROUTE '2' , 'DESTINATION' , 'RC'
if RC <> 0 then
  do
    say '0Unknown destination 'DESTINATION '.'
    say ' Transfer from the PC to MUSIC not initiated.'
    exit
  end

/* transfer the pc file to MUSIC */

SAY '0Transferring 'PCFILENAME' from the PC to MUSIC.'
'$PGM:XTMUS 'PCFILENAME ' @TEMP -R -M'
IF RC <> 0 THEN
  DO
    SAY '0PC file was not transferred to MUSIC successfully.'
    SAY ' Return code 'RC'.'
    'PURGE @TEMP'
  EXIT
END
SAY '0PC file was transferred to MUSIC.'
SAY '1'

/* print our MUSIC file and delete it */
```

```
'EDIT @TEMP;MSG$ OFF;PRINT @TEMP R('DESTINATION');PURGE @TEMP;QQ'

/*  if we printed our PC file at MUSIC*
    then invoke the OUTPUT program for the user  */

IF SUBSTR(DESTINATION,1,5) = 'MUSIC' THEN '*COM:OUTPUT'
EXIT
```

Example 2 - Invoking a PC editor from MUSIC.

The following is another sample REXX program. It allows the user to invoke his favorite PC editor to create a text file. This file however will be saved under his MUSIC code.

Note how TOTTY, PCEXEC, XTMUS and TO3270 are used in turn.

```
/INC REXX
/* Exec to create a file using an editor on the PC.
   Once created and saved on the PC,
   transfer the file to MUSIC.

   The exec is called as follows

   'PCEDIT' pceditorname musicfilename

   where pceditorname is the specifications giving the drive,
       filespec, and extension for the editor to use on the PC
   where musicfilename is the MUSIC filename where the file created
       on the PC is to be transferred
*/

parse arg pceditorname musicfilename
if pceditorname = '' or musicfilename = '' then exit 10

/*  switch terminal modes from 3270 to tty if required.
    we'll want to do this throughout the whole exec.  */

'$PGM:TOTTY PCWS NOMSGS'
if rc = 2 or rc = 3 then exit 1
terminalswitch=0
if rc = 1 then terminalswitch=1

/*  do the actual edit  */

'$PGM:PCEXEC '||pceditorname||' .\PC#TXT.TMP -D'
if rc <> 0 then
do
    irc=rc
    if terminalswitch=1 then '$PGM:TO3270 NOMSGS'
    exit irc
end

/*  PCEDIT finished okay.
    Now transfer the editor file to MUSIC.  */
```

```

say '0Transferring the PC file to MUSIC...'
'$PGM:XTMUS .\PC#TXT.TMP '||musicfilename||' -R -C -M'
if rc <> 0 then
  do
    '$PGM:PCEXEC ERASE .\PC#TXT.TMP'
    if terminalswitch=1 then '$PGM:TO3270 NOMSGS'
    exit 11
  end

/*  The transfer to MUSIC was okay.
    Now erase the editor file on the PC.  */

'$PGM:PCEXEC ERASE .\PC#TXT.TMP'
irc=rc
if terminalswitch=1 then '$PGM:TO3270 NOMSGS'
exit irc

```

PCWS Interface to MUSIC MAIL

Certain features included in the MUSIC/SP Mail program facilitate its usage when connected with PCWS.

The PCWS Mail interface can be used to:

- 1) Send an existing PC File
- 2) Save the mail text in a PC File
- 3) Create the mail text on the PC

Using PC File Names

Wherever a file name can be specified in the MAIL program, a PC file name can be entered. The MAIL program recognizes a PC file name when a prefix of the form D: (where D is a PC drive) is included. For example, MAIL will process

```
x:filename
```

as a PC file if *x* is a letter associated to a PC drive and *filename* is a valid DOS file name. For additional information on using the PCWS Mail interface, please refer to the *MUSIC/SP Mail and Office Applications Guide*.

Chapter 8. The PCWS API Library

Chapter 8. The PCWS API Library

Introduction to the PCWS API Library

This chapter describes a special feature of PCWS called the PCWS Application Programming Interface (API) library. Knowledge of this feature isn't necessary for PCWS to run, and hence, is irrelevant for everyday PCWS end-users.

The PCWS API library is intended for experienced PC software developers who wish to build a PCWS interface into their programs.

Using The API Library

The PCWS API functions described in this chapter are distributed in the library file called PCWSLIB.LIB. All functions from this library are written in assembler using the 'small' memory model. Therefore, they can be invoked from assembler or any high-level language with near calls. As the following list shows, every function in this API library starts with the 'Pcws' prefix:

extern unsigned	pascal PcwsFind(void);
extern void	pascal PcwsFlush(void);
extern unsigned	pascal PcwsGetCh(void);
extern void	pascal PcwsInitCom(void);
extern unsigned	pascal PcwsLoad(void);
extern char	pascal PcwsProcess(void);
extern void	pascal PcwsReset(void);
extern unsigned	pascal PcwsScan(char *ptr, int x);
extern void	pascal PcwsSend(char *ptr);
extern void	pascal PcwsSendC(unsigned);
extern void	pascal PcwsSendK(unsigned);
extern void	pascal PcwsSetCom(struct comstruc *);
extern unsigned	pascal PcwsWait(char *ptr, int x);

The above list is in fact part of PCWSLIB.H, a 'C' header file also distributed with PCWS. If you are writing your application in 'C', you should include this file.

The first PCWS API function your application must call is 'PcwsFind'. This function tries to find the resident part of PCWS. This resident part must be present in memory for your application to interface with PCWS properly. If 'PcwsFind' isn't successful, the 'PcwsLoad' function must then be invoked. 'PcwsLoad' loads the PCWS resident part in memory allowing your application to use the other API functions.

The API Functions

The following is a complete list, in alphabetical order, of the PCWS API functions.

PcwsFind

Description: Attempt to find the PCWS resident part in memory
C declaration: unsigned pascal PcwsFind(void);
Parameters: None
Return Value: 00: Didn't find resident part
Otherwise: Returns the PSP address of the resident part

The 'PcwsFind' should always be the first one called by your application. If the resident part of PCWS isn't found, your application must then call 'PcwsLoad' to enable the use of other PCWS API functions.

PcwsFlush

Description: Make PCWS ignore all characters received so far
C declaration: void pascal PcwsFlush(void);
Parameters: None
Return Value: None

This function instructs PCWS not to process the characters which the selected COM port has received so far. It is normally used in connection scripts to ignore the various logo screen which normally show-up when establishing a connection.

PcwsGetCh

Description: Get a character from the PCWS main buffer
C declaration: unsigned pascal PcwsGetCh(void);
Parameters: None
Return Value: MSB: -1 if no character is ready
Otherwise: LSB: Character from buffer

This function gets a character from the 'main' buffer where PCWS saves incoming characters. If no character is available, the Most Significant Byte (MSB) of the returned word (2 bytes) is set to -1. Otherwise, the Least Significant Byte (LSB) holds the character to be processed by your application. In the latter case, PCWS updates its internal pointers in readiness for the next PCWSGETCH function call.

PcwsInitCom

Description: Initialize the selected COM port.
C declaration: void pascal PcwsInitCom(void);
Parameters: None
Return Value: None

By default, this function uses the communication settings data found in the PCWS.STP setup file to initialize a COM port. Typically, applications which automate connections to host systems use this function before sending any data to the serial port. To use a different COM port, speed, etc., the application must first invoke 'PcwsSetCom' before calling 'PcwsInitCom'.

PcwsLoad

Description: Load the PCWS resident part (PCWSHDL) in memory
C declaration: unsigned pascal PcwsLoad(void);
Parameters: None

Return Value: 00: PCWS resident part is loaded
Otherwise: Problems occurred installing the resident part.

This function loads the PCWS resident part in memory allowing your application to use other API functions. This function should be called only if a previous call to 'PcwsFind' was unsuccessful.

PcwsProcess

Description: Instruct PCWS to process the characters received so far
C declaration: char pascal PcwsProcess(void);
Parameters: None
Return Value: 00: OK
Otherwise: Can't load COMMAND.COM

PCWS will process all the data which the selected COM port received and stored in a buffer. The screen stays intact as the processing is done in the background. 'PcwsProcess' is used rather than 'PcwsFlush' in cases where the latest data to arrive is important. One such case is right after sending the terminal type to a protocol converter. Indeed, at this point the converter sends an important initialization string back to the PC. A non-zero return value indicates the absence of or an invalid COMSPEC environment variable.

PcwsReset

Description: Reset the selected COM port
C declaration: void pascal PcwsReset(void);
Parameters: None
Return Value: None

The Data Terminal Ready (DTR) line is dropped for a second and then set back on. The PCWS main input buffer is also reset to ignore all characters received up to that point.

PcwsScan

Description: Scan main input buffer for the specified string
C declaration: unsigned pascal PcwsWait(char *ptr, int x);
Parameters: 1. Offset (Word) of string to wait for
2. Length of string to wait for
Return Value: 00: Success (String found)
Otherwise: Failure

PCWS stores characters incoming through the serial port in a 'main' buffer. Your application can use the 'PcwsScan' function to scan this buffer and verify if a certain string was received.

PcwsSend

Description: Send a string to the host
C declaration: void pascal PcwsSend(char *ptr);
Parameters: 1. Offset (Word) of ASCII string to send
Return Value: None

Send the specified string. The only parameter of this function is a pointer to a null terminated (ASCII) string.

PcwsSendC

Description: Send a character to the host
C declaration: void pascal PcwsSendC(unsigned);
Parameters: 1. Least significant byte of word is the character to send
Return Value: None

Sends the code of the ASCII control or printable character. The valid range of codes is from 0 to 7F hex.

PcwsSendK

Description: Send a function key to host.
C declaration: void pascal PcwsSendK(unsigned);
Parameters: Numbers from 1 to 24 specify keys PF1 through PF24.
Numbers 25 and 26 respectively specify PA1 and PA2.
Return Value: None

The parameter is a number specifying the function key to send. The valid numbers range from 1 to 26.

PcwsSetCom

Description: Set the communications settings.
C declaration: void pascal PcwsSetCom(struct comstruc *);
Parameters: Ptr to a 'comstruc' structure
Return Value: None

Indicates the communication settings PCWS is to use. The only parameter of this function is a pointer to a structure documented in the next in the next section "The Communication Structure". After using 'PcwsSetCom', the application must call 'PcwsInitCom'.

PcwsWait

Description: Wait until the PC receives the specified string.
C declaration: unsigned pascal PcwsWait(char *ptr, int x);
Parameters: 1. Offset (Word) of string to wait for
2. Length of string to wait for
Return Value: 00: Success, 01: Timeout, 02: User pressed the ESC key.

The first parameter is a pointer to the string to wait for. The second is its length. On return, the calling program should make sure the specified string was received successfully. A timeout occurs after 10 seconds.

The Communication Structure

The 'PcwsSetCom' function takes on, as its only parameter, a pointer to a structure describing the communication settings. The following 'C' structure declaration is taken from the PCWSLIB.H header file:

```

struct comstruc
[
char port;
char speed;
char par;
char stop;
char len;
char xon;
char echo;
char parchk;
];

```

The table below indicates the possible values the fields of this structure can take on, as well as the default communication setting values of PCWS.

<u>Field</u>		<u>Default</u>	<u>Possible Values</u>
port	(Port)	0	0-COM1, 1-COM2, 2-COM3, 3-COM4
speed	(Speed)	2	0-300, 1-1200, 2-2400, 3-4800, 4-9600, 5-19.2
par	(Parity)	2	0-None, 1-Odd, 2-Even, 3-Space, 4-Mark
stop	(Stop Bits)	0	0-'1 bit', 1-'2 bits'
len	(Word Len)	2	0-'5 bit', 1-'6 bit', 2-'7 bit', 3-'8 bit'
xon	(Xon/Xoff)	1	0-Off, 1-On
echo	(Local echo)	0	0-Off, 1-On
parchk	(Parity Chk)	0	0-Off, 1-On

(For more information on the Communication Settings please refer to "The Settings Menu Item" section of Chapter 2. - The Pcws Menu.)

Appendixes and Glossary

Appendix A. Keystrokes

PCWS Program Control Keys

The PCWS Control keys behave the same way in all PCWS emulation modes:

<u>Functions</u>	<u>Key</u>	<u>Description</u>
Back to DOS	Alt-D	Back to DOS. Type PCWS to return.
Help	Alt-H	Select HELP item of Main Menu.
Capture Screen	Alt-I	Save screen to file PCWS.SCR.
Main Menu	Alt-M	Select GO item of Main Menu.
Exit PCWS	Alt-X	Select EXIT item of Main Menu.

PAGE Mode Keystrokes

<u>Functions</u>	<u>PAGE Mode Key</u>
Go to TOP Page	Home
Go to BOT Page	End
Go Up 1 Page	PgUp
Go Dn 1 Page	PgDn
Go Up 7 Lines	Up Arrow
Go Dn 7 Lines	Dn Arrow
Start of Line	Ctrl-Home
End of Line	Ctrl-End

3270 Mode Keystrokes

<u>Functions</u>	<u>3270 Mode Key</u>
Clear	Alt-c
Enter	Enter
Home	Home
Tab	Tab
BackTab	Shift-Tab
NewLine	Keypad *
Page Up (PF7)	PgUp
Page Dn (PF8)	PgDn
PA1	Keypad -
PA2	Keypad +
PF1 to PF12	F1 to F12
PF13 to PF24	Sh-F1 to Sh-F12
3270 -> PAGE Mode	Alt-p

VT100P/VT100 Mode Keystrokes

<u>Functions</u>	<u>VT100P Mode</u>	<u>VT100 Mode (7171 defaults)</u>
Clear	Alt-c	Esc O M
Enter	Enter	Enter
Home	Home	N/A
Tab	Tab	Tab
BackTab	Shift-Tab	Esc Tab
NewLine	Keypad *	Ctrl-Enter
Page Up (PF7)	PgUp	PgUp
Page Dn (PF8)	PgDn	PgDn
PA1	Keypad -	Esc ,
PA2	Keypad +	Esc .
PF1 to PF12	F1 to F12	Esc 1 to Esc =
PF13 to PF24	Sh-F1 to Sh-F12	Esc Sh-1 to Esc Sh-=
Last Char Of Line	End	End
Erase EOF	Ctrl-End	Esc Del
Char Err Reset	Ctrl-r	Ctrl-r
Disconnect	Esc d	Esc d

Appendix B. The PCWS Configuration File

The PCWS.CFG File

The optional configuration file for PCWS is called PCWS.CFG. It allows the PCWS program to adjust itself to a number of machine dependent variables. The PCWS.CFG file isn't required for PCWS to run. It only needs to be created if the default values used by PCWS don't match with those of your PC.

As the other data files, the PCWS.CFG file should be placed in the directory specified by the PCWS environment variable. It can be created or modified using any PC editor.

The Keywords

Every line of PCWS.CFG must start with a keyword. The currently supported keywords are described in this section.

COM3 & COM4

For a program to be able to communicate through one of the computer's COM ports, it needs to know two things about it. These are:

1. The Base Address of the chip which runs the COM Port (the UART)
2. The IRQ (Interrupt Request Line) the UART is connected to.

The table below indicates the default values (in hex) used by PCWS:

<u>Port</u>	<u>Base Addr.</u>	<u>IRQ Line</u>
COM1	03F8	4
COM2	02F8	3
COM3	03E8	4
COM4	02E8	3

While the settings for COM1 and COM2 are standards set by the original IBM PC, the ones for COM3 and COM4 can vary from one computer to another. The COM3 and COM4 addresses and interrupt request lines can usually be somehow configured either through hardware, software or both. The trick in using the non-standard COM ports, is to get the software and hardware to match.

The default values PCWS uses for the COM3 or COM4 settings can be changed. To do so, the user simply needs to insert lines to the PCWS.CFG file. The inserted line(s) must follow the format of those in the table above and hold the valid settings.

Format:	COMx Base Irq
Example:	COM4 02E0 4

This example would tell PCWS to use 02E0 hex as the base address for the COM4 port which uses IRQ 4.

DEFINE

The DEFINE keyword allows the redefinition of keystrokes. DEFINE can translate a particular keystroke into any other. Optionally, DEFINE can also be used to let a single keystroke be interpreted as a sequence of others. The format is as follows:

```
DEFINE oldkeyname newkeyname [newkeyname ...]
```

The DEFINE arguments are key names. These can be any alphanumeric key or any of the following strings:

- CR, ESC, TAB, BKSP, SPC
- UP, DOWN, LEFT, RIGHT, HOME, END, PGUP, PGDN, INS, DEL, PAD-, PAD+
- F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12
- ALT-A to ALT-Z, and CTRL-A to CTRL-Z
- ALT-F1 to ALT-F12, and CTRL-F1 to CTRL-F12
- CTRL-HOME, CTRL-END, CTRL-PGUP, CTRL-PGDN

The first argument represents the key to be translated. The subsequent ones define how the key should be interpreted.

Example:

```
DEFINE F1 ESC 1
```

This line indicates that the F1 key should be redefined as an ESC followed by a '1'. This is particularly useful for those that don't have access to the VT100P terminal type on their protocol converter. For more information please refer to the "Device Definition Table" section in Chapter 5 - The VT100 Terminal Type.

Example:

```
DEFINE ESC F3
```

This line indicates that we want the ESC key to act as the F3 key. If you do redefine the ESC, it is recommended that you use another DEFINE comma such as: 'DEFINE alt-e ESC' to match a key combination to the ESC key. You will need this second DEFINE to exit from the file transfer monitor.

Example:

```
DEFINE alt-a PAD+ F4
```

This line indicates that the "alt-a" combination is to be used to add a new MUSIC/SP session regardless of if you are at the *GO prompt or in a full-screen application such as EDITOR or MAIL.

Example:

```
DEFINE alt-j F12 j o i n CR
```

This line indicates you wish to use the "alt-j" combination to first press the F12 key then send the string "join" followed by the ENTER key. This is the procedure for joining 2 lines in the MUSIC/SP editor.

RETRACE

The RETRACE keyword will enable video retrace when writing to the screen. It should only be specified if "snow" appears on the screen. Typically, this will happen if the PC monitor is an older CGA. If used, the RETRACE keyword must appear all alone on one line.

This glossary describes the important terms and abbreviations used in this guide. Included are technical terms unique to this publication, terms that can be found generally throughout MUSIC/SP publications, and general data processing terms. If you do not find the term you are looking for, refer to the IBM publication *Vocabulary for Data Processing, Telecommunications, and Office Systems*.

ASCII. American National Standard Code for Information Interchange. This is a standard 7-bit code for 128 characters used to store and transmit information in conjunction with computer systems.

asynchronous communication. A physical transfer of data in which both the sender and the receiver do not share a common clock. Therefore, reception of data is essentially unpredictable. There need not be any regularity in the transmission of data.

baud rate. A unit of signaling speed equal to the number of state changes per second. For example, 600 baud means that there are 600 state changes per second. If each change of state conveys 4 bits of information, then the signaling speed is 2400 bps (see bps below).

bit. Binary digit. One of the characters 0 or 1. In computers, the finest granularity of storage.

bps. Bits per second. In data transmission, the number of bits transmitted per second. Dividing the bps by 10 gives you the approximate transmission speed in cps (characters per second). For example, a 2400 bps modem transmits at approximately 240 cps.

COM port. One of the serial communication ports available in the architecture of the IBM PC.

cursor. A moveable marker on a display screen that indicates where the next character will be placed.

default setting. A value that some option is initially assigned, without any specification required on your part. Usually, default settings are chosen so that they need not be altered in most circumstances.

data transparency. The ability to transfer data in blocks without restricting the contents of the block. Special care must be taken, because any escape sequences used to control the data transfer can always occur within a block of information.

disable. Usually used in conjunction with a parameter that can have two possible values. To *disable* a parameter means to turn off the function that the parameter controls. The actual value required to do this may vary.

download. The process of transferring information (usually a sizeable quantity) from a host system to your PC. This term is often used in conjunction with file transfer on PCWS.

EBCDIC. Extended Binary-Coded Decimal Interchange Code. This is an 8-bit code for 256 characters used to store and transmit information in conjunction with computer systems.

echo. The automatic re-display of characters received or transmitted on your screen. Both your modem and the host system possess the ability to *echo* characters onto your screen. The modem may echo characters that it transmits, while the host system may echo characters that it receives. If both echo characters, then characters appear in duplicate upon your screen. (Also referred to as *echoplex*).

enable. Usually used in conjunction with a parameter that can have two possible values. To *enable* a parameter means to turn on the function that the parameter controls. The actual value required to do this may vary.

Enter key. In 3270 type terminals, the *Enter key* is used to transmit a screen to the computer, after local editing has been performed. In terminal emulation, the Return key is usually taken to mean the Enter key.

error checking. The process of detecting and sometimes correcting transmission errors. Although characters are sent correctly, noise on telephone lines can sometimes garble characters. Error checking is a protection against unknowingly receiving such garbled characters and accepting them as valid.

full duplex. Two-way transmission of information in which both sides can send and receive simultaneously. Thus A can send a message to B at the same time as B is sending a message to A. An everyday example of full duplex communications is a telephone conversation.

full screen mode. A mode of operation (3270 emulation mode in PCWS) in which an entire screen of data is displayed. The operator is allowed to make changes to all or some of the fields (areas) in the screen (depending upon the application program). Then all of the changes are transmitted to the host system by pressing the Enter key.

half duplex. Two-way transmission of information in which only one side can transmit at a time. Thus, the other side automatically becomes the receiver. Thus, when A is sending a message to B, B must wait until A finishes the message before transmitting. An everyday example of half duplex communications is the radio communications in a taxi.

hand-shaking. The initial exchange of information between communicating computers that establishes the rules under which subsequent data communications are to take place. Thus, software on both machines running at different levels can agree to use a common protocol.

host system. Computer software that runs on a mainframe computer. MUSIC/SP is a host system. Examples of a mainframe computer are the IBM computers in the ES/9000 series.

keypad. The keypad is a cluster of keys on the right-hand side of some keyboards, usually consisting of numbers.

menu. A *menu* is a list of choices or functions to be performed. The user has the option of selecting one of

the choices, either by number or by function key, or of exiting the menu and returning to the previous environment.

modem. Modulator-Demodulator. An electronic device that is used to translate digital signals from a computer into audible signals that can be transmitted over a telephone connection, and vice versa. PCs may use an *on-board* modem (a modem placed on a board and installed within the PC chassis itself) or an outside modem connected via a cable to the PC.

offline. The state of being disconnected from a computer system. In conjunction with PCWS, the state of being disconnected from the host system.

online. The state of being connected to a computer system. In conjunction with PCWS, the state of being connected to the host system.

parity checking. A check on the accuracy of transmitted information, achieved by adding an extra bit that is either 0 or 1 depending upon the number of bits in the character being transmitted that are 1. There are two flavors: even parity and odd parity. The parity bit is chosen such that for even parity, the total number of bits that are 1, including the parity bit forms an even number. For odd parity, the total number of bits that are 1, including the parity bit forms an odd number.

password. A secret character sequence assigned to you by the support personnel of the host system to which you are connecting. In conjunction with a sign-on code (logon id) this allows you to identify yourself to the host system.

path. On the PC, a list of directories through which DOS searches in order to locate a program that you have requested to be run.

protocol. A set of rules (including formats and timing) governing the communication process. All participants in a communications process must agree to use the same rules at the start of communications, usually through some form of *hand-shaking* procedure. Unless all participants use the same communications protocol, communication is not possible.

RS-232C. Electronic Industries Association (EIA) 25 pin physical interface standard, equivalent to the CCITT V.24 interface. This is the standard connection between a modem and a terminal, or PC.

scrolling. The process of moving the image presented on a display screen such that new data is displayed at one edge of the screen. Depending upon the type of terminal being emulated, data that is scrolled off the screen may or may not be retrievable.

serial communications. Refers to the width of a communications path. In serial communications, data is transmitted 1 bit at a time. Modems operate in this manner.

serial port. The physical connection on the PC through which serial data is transmitted and received. The cable from the modem is attached to this connector.

snow. The brief appearance of bright light spots on a PC display screen when the screen is refreshed. This is usually related to a mis-match in the screen display rate and the refresh rate. (In PCWS, refer to the RETRACE configuration keyword).

start bit. In serial communications, it is the first bit transmitted by the sender, informing the receiver that transmission has started and allowing the receiver to start timing procedures for individual bits. This bit is a control bit and is not part of the character being transmitted.

stop bit. In serial communications, it is the last bit sent to the receiver, indicating that the character has been completely transmitted. This bit is a control bit and is not part of the character being transmitted.

toggle. Parameters with only two possible values can be thought of as electrical switches, and thus can be *toggled* between one value and the other.

upload. The process of transferring information (usually a sizeable quantity) from your PC to a host system. Usually used in conjunction with file transfer on PCWS.

transparent mode. see data transparency

VM. VM is short for IBM's VM/SP (Virtual Machine/System Product) operating system.

word length. In serial communications, the number of data bits used to transmit a character, excluding the start, stop and parity bits. Common word lengths are 7 and 8.

XON/XOFF. A particular protocol used for controlling the flow of data. XOFF is transmitted to the host system to indicate that the flow of data should be suspended, and XON is transmitted to the host system to indicate that the flow of data can be resumed. Control-S is usually identified with XOFF and Control-Q is usually identified with XON.

Index



.CON File, 20



Action Keys, 30
Alt-M Hotkey, 8
API Functions
 PcwsFind, 53
 PcwsFlush, 53
 PcwsGetCh, 53
 PcwsInitCom, 53
 PcwsLoad, 53
 PcwsProcess, 54
 PcwsReset, 54
 PcwsScan, 54
 PcwsSend, 54
 PcwsSendC, 55
 PcwsSendK, 55
 PcwsSetCom, 55
 PcwsWait, 55
API Library, 52
App Parameter, 39
Application Programming Interface Library, 52
ASCII, 38
ASCII, Translating to EBCDIC, 39
Asynchronous Communications Controller, 16
Asynchronous Package, iii
Attaching Terminals, 3
Auto Connect, 9, 18
AUTOEXEC.BAT, 4
Automatic Connection Feature, 17
Autopage, 13
Autoskip Field, 29



Back Tab Key, 30
Back to DOS, 9, 22
Back to DOS Key, 28, 35
Backspace Key, 13, 28, 35
Bin Parameter, 39
Binary Mode, 38
Bits per second, 11
Block Mode for the Mouse, 23
BREAK Command, 19
Break Signal, 14

BYE K-PCWS Command, 42



Capture Screen, 22, 28, 35
Changing PCWS Settings, 10
Clear Key, 31
Color Settings, 12
Colors 3270, 29
COM Configuration Keywords, 60
Command Line Argument, 18
COMMAND.COM, 5
Comments for Auto Connect, 19
Communicating with Host, 16
Comp Parameter, 39
Compressing Data, 39
COMSPEC variable, 4
CON Commands, 19
CON Files, 17
Concepts of PCWS, 2
Configuration File, 4, 60
Configuration Keywords, 60
Connecting Terminals, 3
Connection Files, 4
Connection Scripts, 17
Communication Settings, 11
Cursor Keys, 8, 30



Data Compression, 39
DEFINE Configuration Keyword, 61
DELAY Command, 19
Device Definition Table, 34
DOS, Back to, 9



EBCDIC, 38
EBCDIC, Translating from ASCII, 39
Editing with PC, 48
Emulation Mode, 9, 16-17
End Key, 28
Enter Key, 8, 28, 30
Environment Variable, 4
Esc Key, 8, 35
Example of .CON File, 20

EXIT K-PCWS Command, 42
Exit Menu Item, 14
Exit PCWS Key, 22, 28, 35

F

Field Concept, 29
File Menu Item, 10
File Names, 49
File Names for File Transfer, 38
File Transfer, 38
 -App, 39
 -Bin, 39
 -Comp, 39
 -Fx, 40
 -Mon, 39
 -Nomsgs, 40
 -Repl, 39
 -Switch, 39
File Names, 38
Record Length, 40
Return Codes, 40
FIN Command, 42
FIN K-PCWS Command, 42
FLUSH Command, 19
Format Parameter, 40
Functions Menu Item, 14

G

GET K-PCWS Command, 43
Go Menu Item, 9

H

Help Key, 22, 28, 35
Help Menu Item, 14
High Intensity Field, 29
High Pro., 12
High Unp., 12
Home Key, 28, 30
Host Communication, 16
HotSpots, 22
HotSpots for the Mouse, 23

I

Input Area Mode, 12
Input Area, PAGE Mode, 26
Installing PCWS, 4
Interfacing, MUSIC and PCs, 46
Interrupt Request Line, 60
Invoking a PC Editor, 48

K

K-PCWS, 41
K-PCWS Commands, 42
 BYE, 42
 EXIT, 42
 FIN, 42
 GET, 43
 LOCAL, 43
 QUIT, 43
 REMOTE, 43
 SEND, 43
 SET, 44
 SHOW, 44
Kermit Support, 41
Keystrokes
 Defining, 61
 PAGE Mode, 58
 VT100 Mode, 59
 3270 Mode, 58
Keystrokes for Menu, 8

L

LAN, Installing PCWS, 5
Line Calldown, 27-28
Line Feed, 35
Load Menu Item, 10
Local Echo, 11
Local Editing Keys, 30
LOCAL K-PCWS Command, 43
Low Intensity Field, 29
Low Pro., 12
LPT1, 14

M

Mail Interface, 49
Main Menu Key, 22, 28, 35
Manuals, v

Margin Bell, 13
Menu
 Exit Item, 14
 Functions Item, 14
 Go Item, 9
 Help Item, 14
 Keystrokes, 8
 Settings Item, 10
 Terminal Item, 9
Menu Mode, 8
Mon Parameter, 39
Mouse Block Mode, 23
Mouse Support, 8, 22
MUSIC
 Publications, v
MUSIC and PC Interfacing, 46

N

New Line, 13
Nomsgs Parameter, 40
Non-Display Field, 29
Numeric-Only Field, 29
NumLock Key, 35

O

Output Area, PAGE Mode, 27

P

PACX Units, 16
PAGE Mode, 26
 Input, 12
 Input Area, 26
 Keyboard Functions, 27
 Output, 12
 Output Area, 27
Page Mode Autopage, 12
PAGE Mode Keystrokes, 58
Page Mode Line Calldown, 27-28
PAGE Mode Settings, 12
Parity, 11
Parity Check, 11
PATH Statement, 4
PA1 Key, 30
PA2 Key, 31
PC

 Supported Models, iii
PC and MUSIC Interfacing, 46
PC File Names, 49
PCEXEC Command, 46
PCWS
 Concepts, 2
 Directory, 4
 Environment variable, 4
 Installing, 4
 Terminal Types, 3
PCWS API Library, 52
PCWS Command, 18
PCWS Configuration File, 60
PCWS Interface to Mail, 49
PCWS Kermit Support, 41
PCWS Terminal Type, 16, 26
PCWS.CFG Configuration File, 4
PCWS.SCR Screen Capture File, 4
PCWS.STP SetUp File, 4, 10
PcwsFind, 53
PcwsFlush, 53
PcwsGetCh, 53
PcwsInitCom, 53
PcwsLoad, 53
PcwsProcess, 54
PcwsReset, 54
PcwsScan, 54
PcwsSend, 54
PcwsSendC, 55
PcwsSendK, 55
PcwsSetCom, 55
PcwsWait, 55
PgDn Key, 27
PgUp Key, 27
Port Menu Item, 11
Prerequisite Knowledge, iii
Print Filter, 13
Printer ID, 13
Printer Toggle, 14
Printing a PC File, 47
PROCESS Command, 19
Program Control Keys, 21
Protected Field, 29
Protocol Converter, 17, 34
Publications, v

Q

QUIT K-PCWS Command, 43

R

Record Length Parameter, 40
 REMOTE K-PCWS Command, 43
 Repl Parameter, 39
 Requirements, System, 4
 Reset Port, 14
 RETRACE Configuration Keyword, 61
 Return Codes, File Transfer, 40
 Return Key, 35

S

SAY Command, 19
 Screen Capture File, 4
 Send Break, 14
 SEND Command, 20
 SEND K-PCWS Command, 43
 SERVER Command, 42
 SET K-PCWS Command, 44
 Settings
 Color, 12
 Communication, 11
 Miscellaneous, 13
 PAGE Mode, 12
 Viewing and Changing, 10
 VT100 Mode, 13
 Settings Menu Item, 10
 SetUp File, 4
 SHOW K-PCWS Command, 44
 Speed Menu Item, 11
 Status Line, 9, 12, 14, 22
 Stop Bits, 11
 Storage Requirements, 4
 Supported PC Models, iii
 Switch Parameter, 39
 System Requirements, 4

T

Tab Key, 30, 35
 Terminal Menu Item, 9
 Terminal Types, 3
 Terminals, Attaching to MUSIC, 3
 Text Mode, 38
 Toggle Printer, 14
 TOTTY Program, 46
 TO3270 Program, 46
 Trademarks, iii
 Transferring Files, 38
 Translating Data, ASCII - EBCDIC, 39

Transparent Mode, 46

U

UART, 60
 Unprotected Field, 29

V

View/Change Menu Item, 10
 VT100
 Base Color, 12
 VT100 Mode
 Settings, 13
 VT100 Mode Keyboard, 35
 VT100 Mode Keypad, 34
 VT100 Terminal Type, 34
 VT100P Terminal Type, 17
 VT100P/VT100 Mode Keystrokes, 59

W

WAIT Command, 20
 Word Len., 11
 Wrap Around, 13

X

XON/XOFF, 11
 XON/XOFF Menu Item, 11
 XTMUS Command, 39
 XTMUS Parameters, 39
 XTPC
 Example, 41
 XTPC Command, 39
 XTPC Parameters, 39

3

3270 Action Keys, 30
 3270 Colors, 29
 3270 Local Editing Keys, 30
 3270 Mode, 29
 Keyboard Functions, 30
 3270 Mode Keystrokes, 58

Table of Contents

PCWS for Windows	iii
Chapter 1. Introduction	1
Basic Concepts	2
What Does PCWS Do?	2
How are Terminals Attached?	3
PCWS Terminal Types	3
System Requirements	4
Installing PCWS	4
Chapter 2. The PCWS Menu	7
Menu Mode	8
Menu Keystrokes	8
The Go Menu Item	9
The Terminal Menu Item	9
The Settings Menu Item	10
The Functions Menu Item	14
The Help Menu Item	14
The Exit Menu Item	14
Chapter 3. Running PCWS	15
Communicating with Host	16
Connecting Via an Asynchronous Communications Controller	16
Connecting Via a Protocol Converter	17
Automatic Connection Feature	17
PCWS Program Control Keys	21
Mouse Support	22
Chapter 4. The PCWS Terminal Type	25
Introduction	26
PAGE Mode	26
3270 Mode	29
Chapter 5. The VT100 Terminal Type	33
Introduction	34
Device Definition Table	34
VT100 Mode Keypad	34
VT100 Mode Keyboard Functions	35
Chapter 6. File Transfer	37
Introduction	38
Text and Binary Modes	38
File Names	38
XTMUS and XTPC Commands	39
Return Codes	40
Examples	40
K-PCWS : PCWS Kermit Support	41

Chapter 7. MUSIC and PC Interfacing	45
Overview of PCWS Interfaces	46
TOTTY and TO3270	46
PCEXEC	46
PCWS Interface to MUSIC MAIL	49
Chapter 8. The PCWS API Library	51
Introduction to the PCWS API Library	52
Using The API Library	52
The API Functions	52
The Communication Structure	55
Appendixes and Glossary	57
Appendix A. Keystrokes	58
PCWS Program Control Keys	58
PAGE Mode Keystrokes	58
3270 Mode Keystrokes	58
VT100P/VT100 Mode Keystrokes	59
Appendix B. The PCWS Configuration File	60
The PCWS.CFG File	60
The Keywords	60
Glossary	63
Index	67

**MUSIC/SP
PCWS User's Guide
May 1994**

**READER'S
COMMENT
FORM**

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that the MUSIC Product Group may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, institution, mailing address and date:

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation.

MUSIC/SP PCWS User's Guide (May 1994)

Reader's Comment Form

fold and tape

please do not staple

fold and tape

**Place
stamp
here**

**MUSIC Product Group
McGill University Systems Inc.
550 Sherbrooke St. West
Suite 1650, West Tower
Montreal, Quebec H3A 1B9
CANADA**

fold and tape

please do not staple

fold and tape