



*The Personal Computer
Personal Series*

Asynchronous Communications Support

First Edition (December 1981)

Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.

Products are not stocked at the address below. Requests for copies of this product and for technical information about the system should be made to your authorized IBM Personal Computer Dealer.

A Product Comment Form is provided at the back of this publication. If this form has been removed, address comment to: IBM Corp., Personal Computer, P.O. Box 1328, Boca Raton, Florida 33432. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligations whatever.

© Copyright International Business Machines Corporation
1981

CHAPTER 1. INTRODUCTION

The Asynchronous Communications Support program (or Communications Program) turns an IBM Personal Computer with an Asynchronous Communications Adapter into a terminal that can communicate with a wide variety of computers that support the American Standard Code for Information Interchange (ASCII). It permits access to the VM/370 and TSO operating systems operating on IBM computers equipped with appropriate interface hardware. Access to these systems can be either by local cable connection or over remote common carrier lines. The Communications Program also can be used for communication between two IBM Personal Computers, again through local connection or a remote hookup. The Upload/Download Facility in the program permits the transfer of data or program files (in a textual format) between an IBM Personal Computer and a VM/370 or TSO system or between two Personal Computers.

Phases of Operation

The Communications Program operates in two phases, (1) *Terminal Selection* and (2) *Terminal Operation*. After the program is loaded, the *Terminal Selection Phase* begins. You are first presented with a series of menus for the choice of terminal parameters. In the simplest case (for example, connecting two IBM Personal Computers) you need to specify only the line bit (character transmission) rate. In other cases, you may wish to define a special purpose terminal by selecting from the menus a variety of other parameters such as parity and line turnaround characters. Once a terminal is specified for a particular system, its definition can be stored as a file on diskette and recalled for subsequent terminal sessions.

After the terminal parameters are specified (or loaded from a diskette file), the *Terminal Operation Phase* of the program is started. In this phase a communication link is established with another computer (called the "host" computer). Once this link is established, the lines of text you type and those received from the host computer are displayed on the screen of your IBM Personal Computer. Transmission and other errors are reported to you on the bottom line of the screen. A control key permits you to go into Upload/Download mode so files can be uploaded (sent) from the IBM Personal Computer to the host or downloaded (received) from the host.

In addition to being a ready-to-use terminal communication system, the Communications Program makes it easy for you to add your own extensions. Both portions of the program that sets up the terminal parameters and that which runs as a terminal are written in BASIC. If you are familiar with BASIC, you can easily modify these portions of the program to add, for example, new defined

PREFACE

This reference manual explains how to use the IBM Personal Computer Asynchronous Communications Support program. It provides you with the following information about the program:

- Requirements
- Selecting and running terminals
- Creating and running your own specified terminal
- Communicating with host systems
- Communicating with another IBM Personal Computer or another personal computer
- Sending files back and forth between computers

Assumptions

This manual assumes that you are already familiar with the terminal operation for the particular host system that you will be connected to. You should refer to the host system manual to understand how to use that system. Also, this program uses some of the facilities of the IBM Personal Computer DOS and BASIC. Rather than repeat selected portions of the DOS and BASIC manuals, at times we refer you to those manuals.

Organization of This Manual

This manual has seven chapters and one appendix.

Chapter 1 has some introductory information about the program as well as directions on how to use the keyboard and how to configure and copy the program diskette.

Chapter 2 describes how to load and run the program, selecting terminals, starting terminals, and communicating with other computers.

Chapter 3 discusses transferring files between the IBM Personal Computer and host systems or another IBM personal Computer and also describes how to recover portions of transferred files should a disconnect occur.

Chapter 4 contains troubleshooting and debugging information.

Chapter 5 describes how the program works, the terminal protocols, and the operation of transferring files.

Chapter 6 tells you how to change the program for specified terminals and how to create your own specified terminal.

Chapter 7 describes how to talk to the Base Program that controls the terminal interface with the various communications lines.

Appendix A lists the error messages that you may encounter when using this program.

CONTENTS

CHAPTER 1. INTRODUCTION	1-1
Phases of Operation	1-2
Explanation of Diagram	1-3
Required Hardware and Software	1-6
Using the Keyboard When Running	
as a Terminal	1-7
Typewriter Keyboard	1-9
The Enter Key	1-9
Special Symbols	1-9
Uppercase	1-9
Backspace	1-10
PrtSc	1-10
Other Shifts	1-11
Using Function Keys When Running	
as a Terminal	1-11
F1 - Communications Line	
Attention	1-11
F2 - Go to Function Selection	
Menu	1-12
F3 - Display Error Messages	1-12
F4 - Enable/Disable Receive Error	
Checking	1-12
F5 - Force to Sending State	1-13
F6 - Enable/Disable Hex Listing ...	1-13
Using the Numeric Keypad	1-13
Special Key Combinations	1-14
Ctrl-Break	1-14
Alt-Ctrl-Del	1-14

What You Must Do First	1-15
Putting the DOS Programs onto the Program Diskette	1-15
Procedure for a Single Diskette Drive System	1-15
Procedure for a Two Diskette Drive System	1-18
CHAPTER 2. GETTING STARTED	2-1
Loading the Communications Program ...	2-2
Selecting a Terminal	2-4
Using Menus during Terminal Selection	2-4
Half or Full Duplex Terminal Protocols	2-4
Terminal Selection Menu	2-5
VM/370 Terminal	2-6
TSO Terminal	2-6
User Specified Half Duplex Term.	2-6
User Specified Full Duplex Term....	2-6
Personal Computer Communications	2-6
Terminal Description Stored on Disk	2-7
Terminal Parameter Selection Menu — Half Duplex	2-7
Line Bit Rate	2-8
Type of Parity Checking	2-10
Number of Stop Bits	2-11
Line Turnaround Char. Sent to Host	2-12
Line Turnaround Char. Sent by Host	2-13
First, Second, and Third Characters to be Deleted	2-14
Line End Character Sent by Host	2-15
Start up of Selected Terminal	2-15

CHAPTER 4. TROUBLESHOOTING PROBLEMS AND UNDERSTANDING ERROR MESSAGES	4-1
Getting Connected to a Host System	4-1
Troubleshooting a Cable Connected System	4-2
Troubleshooting a Modem Connected System	4-3
I/O Signals Used by the Communications Program	4-4
Providing Control Signals for Program Operation	4-6
How Messages Are Displayed	4-8
Dynamic Messages, at the Bottom of the Screen	4-8
Enabling and Disabling Receive Errors	4-9
Static Messages	4-10
Messages during Upload, Download, and Compare	4-11
CHAPTER 5. HOW TERMINAL COMMUNICATIONS WORKS	5-1
Initialization Phase	5-2
Half Duplex Terminal Protocol	5-3
Full Duplex Terminal Protocol	5-5
Operation of the Upload, Download and Compare Programs	5-7
Downloading	5-7
Comparing	5-9
Uploading	5-9
Subroutine Error Indication Protocol ...	5-10
Adding Access to Other Systems	5-10

CHAPTER 6. MODIFYING THE BASIC PROGRAM	6-1
Changing Terminal Selection Default Parameters	6-2
Adding Additional Terminals to the Selection Menu	6-4
Adding Additional Parameters	6-5
Modifying the Terminal Protocols	6-7
 CHAPTER 7. USING THE BASE PROGRAM	 7-1
General Overview	7-1
Loading the Base Program	7-2
Specifying CALL Name Offsets	7-3
Entry Points	7-4
SETSENSE (CHAN%,TYPE%,DAT%,ERFLG%)	7-4
BREAK (CHAN%,ERFLG%)	7-10
BUFSIZE (CHAN%,TYPE%,DAT%,ERFLG%)	7-10
RECEIVE (CHAN%,LN%,ST\$,ERFLG%)	7-10
SEND (CHAN%,ST\$,ERFLG%)	7-11
Effect of Control Signals on Data Transmission and Reception	7-12
Operation of Base Program	7-13
APPENDIX A. MESSAGES	A-1
INDEX	X-1

Return to Terminal Selection	
Menu	2-16
Save Current Terminal	
Specification	2-16
Terminal Parameter Selection Menu —	
Full Duplex	2-17
XON/XOFF Support Present or	
Absent	2-17
Local or Host Character	
Echoing	2-19
Format of Parameters Stored for	
Terminals — A Summary	2-21
Default Parameters for Various	
Terminals	2-25
Starting Up as a Terminal	2-28
Cable Connections	2-28
Connecting to a Host Computer	2-29
Function Keys Used in Terminal	
Operation	2-32
Escaping from and Returning to	
Terminal Operation	2-32
Return to Terminal Operation	2-32
Return to Terminal Selection	2-33
Return to BASIC	2-33
Using VM/370	2-34
Logging onto VM/370 —	
An Example	2-37
Using TSO	2-42
Logging onto TSO —	
An Example	2-44
Operating as a Full Duplex Terminal	2-49
Communicating between IBM Personal	
Computers	2-51
Establishing Communication	2-51
Changing Parameters Other than	
Line Bit Rate	2-53

CHAPTER 3. TRANSFERRING FILES 3-1

Transferring Files between an IBM Personal Computer and a Host Computer 3-2

- General Rules for Operation 3-3
- Transferring Files with VM/370 3-5
 - VM/370 EDIT EXEC 3-7
- Transferring Files with TSO 3-7
 - TSO Profile Settings 3-7
- Uploading a File 3-9
- Downloading a File 3-12
- Comparing Files 3-14
- Uploading and Downloading BASIC Programs 3-15

Transferring Files between IBM Personal Computers 3-18

- Transmitting an IBM Personal Computer File 3-19
- Receiving a File on Your IBM Personal Computer 3-21

Personal Computer File Transfer Protocol 3-23

- Transmitting a File From a Non-IBM Personal Computer 3-24
- Receiving a File Transmitted by an IBM Personal Computer 3-25

Recovering from a Disconnect during File Transfer 3-27

- Recovering a File after Disconnecting during an Upload 3-27
- Recovering a File after Disconnecting during a Download 3-28
- Recovering a File after Disconnect between Two IBM Personal Computers 3-28

terminals to the displayed menus or to change the logic of a terminal protocol. The Base Program portion of the Communications Program has a well-documented interface so you can write your own communications application programs entirely in BASIC.

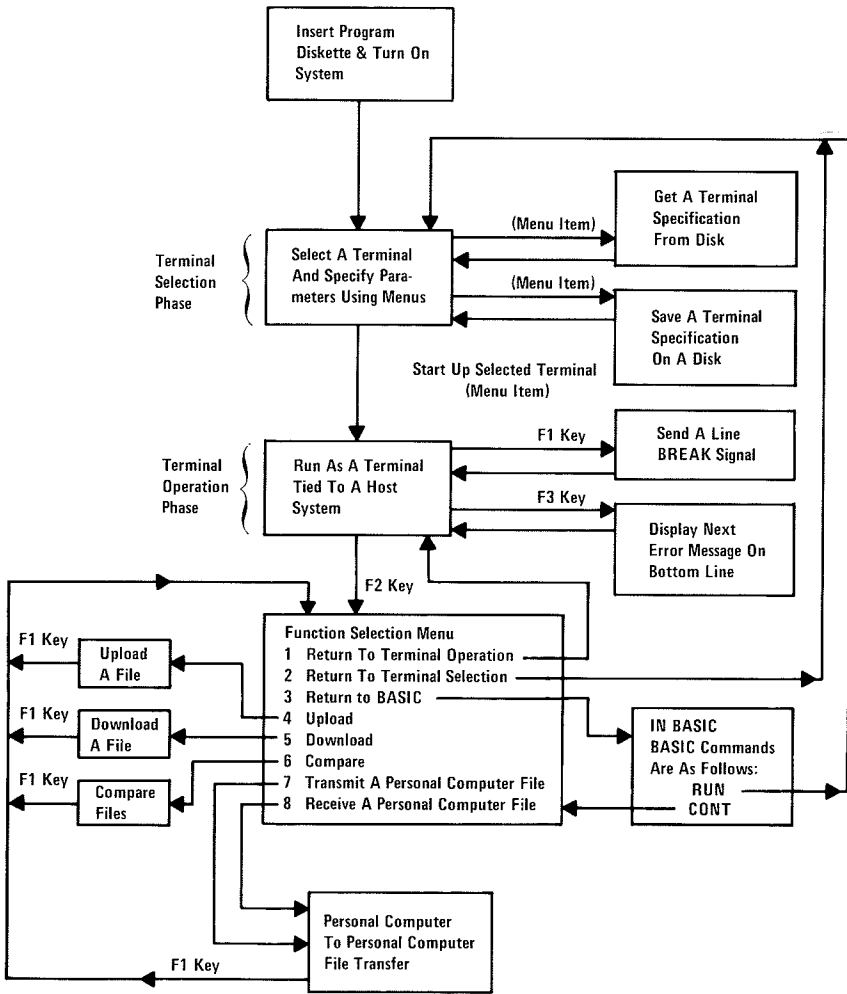
Explanation of Diagram

The diagram, Functions of the Communications Program, shows the various phases of operation of the Communications Program. A quick inspection of the diagram gives you an overview of the Communications Program. At the top is the Terminal Selection Phase. In this phase you may select a terminal and specify its parameters using selection menus. Note that two menu items permit you to save a terminal specification on a diskette file and to retrieve such a specification for later use. The terminal Selection Phase is described in detail in the first part of Chapter 2.

In the middle of the diagram is the Terminal Operation Phase. You enter this phase from the Terminal Selection Phase by selecting the **Start Up Selected Terminal** menu item. In this phase you operate as a terminal tied to a host system. There are three useful function keys that operate when you are in the Terminal Operation Phase.

The F1 key sends a BREAK signal on the communications line. The F3 key displays the next error message on the bottom line of the screen. The F2 key takes you to the Function Selection Menu. You can find the details of the operation of these keys later in this chapter.

At the bottom of the diagram is the Function Selection Menu, your access to more function in the Communication Program. The first three items on this menu give you different ways of escaping from



Functions of the Communications Program

(or returning to) operation as a terminal. They are described in detail in Chapter 2 in the section “Escaping from and Returning to Terminal Operation.”

By picking one of items 4, 5, or 6 on the Function Selection Menu you may upload, download or compare files. These menu items and their operation are described in the first part of Chapter 3.

Finally, if you are in communication with another IBM Personal Computer, you can exchange files with that computer by selecting either Item 7 or Item 8 on the Function Selection Menu. The operation of these functions is described in Chapter 3 in the section “Transferring Files between IBM Personal Computers”.

We suggest that you use this diagram both as a guide when reading the rest of this manual and as a reference index to the manual when you are using the Communications Program.

Required Hardware and Software

You need the following to operate the Asynchronous Communications Support Program (called the Communications Program):

- An IBM Personal Computer with at least 64 KB of memory and at least one diskette drive.
- An Asynchronous Communications Adapter
- A full duplex telephone modem (typically with an acoustic coupler) or a direct cable connection to a local host computer. (Communications using half duplex modems is not supported.)
- A cable to connect the Asynchronous Communications Adapter to the modem or to the direct cable connection.
- Disk Operating System (DOS) and Disk BASIC language.
- The Asynchronous Communications Support diskette containing the following files:
 - The Communications BASIC program (TERMINAL.BAS)
 - The Communications Base Program (RS232INT.BAS)
 - Terminal Specification files—VMMOD.TER, TSOMOD.TER and PCMOD.TER
 - Batch programs AUTOEXEC.BAT and UPDATE.BAT and the file MESSAGE

Using The Keyboard When Running as a Terminal

This section describes the use of the keys on the IBM Personal Computer keyboard when you are operating the Communications Program as a terminal.

The keyboard is divided into three areas, (1) the *typewriter* area in the middle, (2) ten *function keys* on the left side of the keyboard, and (3) the *numeric keypad* on the right side.

Typewriter Keyboard

The typewriter area of the keyboard behaves much like a typewriter with the letter, numbers and special characters. Capital letters, and the special characters shown above the numbers on the number keys, are typed by holding down either of the Shift keys (⇧) and pressing the desired key. These characters are transmitted to the host computer as they are typed.

The Enter Key

The key with the ↵ symbol on it is the carriage return key. It is referred to as the *Enter* key in this manual.

When operating as a half duplex terminal (for example, a VM/370 terminal), the Enter key signals the host computer that you have completed transmission of a line of input and switches the terminal from SENDING to RECEIVING mode. When operating as a full duplex terminal, pressing the Enter key transmits a carriage return (or other line end character) to the host computer.

Special Symbols

There are some special symbols on this keyboard that you won't find on a regular typewriter, such as [and]. And some characters are not where you might expect them to be if you're used to using a typewriter. For example, the uppershift comma (,) is not a comma, but the < symbol.

Uppercase

There is no normal Shift Lock key on this keyboard. The Caps Lock key, to the right of the Space Bar, is similar to a Shift Lock key, but it only gives you

capital letters. It does not give uppershift characters on any other keys. After you press this key, you continue to get capital letters until you press it again. You can get lowercase letters when in Caps Lock state by pressing and holding one of the Shift keys. When you release the Shift key, you'll go back to Caps Lock state.

Backspace

The Backspace key, with the symbol ←, on the upper row of the typewriter area, behaves somewhat differently from the Backspace key on a typewriter. It not only backspaces, it erases from the screen what you've typed. It also transmits the backspace character (Hex 08) to the host computer. As the characters that you delete on the screen with the backspace were already sent to the host computer, the host computer should be programmed to discard one character for each backspace received.

PrtSc

Below the Enter key is a key labeled PrtSc on top and * on the bottom. *PrtSc* stands for *Print Screen*. When the keyboard is in lowershift, pressing this key causes an asterisk to be typed. In uppershift, however, this is a special key that causes a copy of what is on the screen to be printed on the printer.

IMPORTANT: Use the PrtSc Key only when you are not receiving output from the host computer. The operation of the IBM Personal Computer is inhibited during the printing of the screen and a receive buffer overflow could result.

Other Shifts

In addition to the Shift keys which change the keyboard from lowershift to uppershift, there is one other shift key you can use on the typewriter keyboard, the Ctrl (Control) key. To use it, hold the Ctrl key, then press the desired key. Then you can release both keys.

The Ctrl key is used to transmit to the host computer certain codes and characters not otherwise available from the keyboard. For example, Ctrl-G (obtained by pressing Ctrl and G simultaneously) is the "bell" character. When this character is received by the IBM Personal Computer, the speaker beeps. You could transmit this character to the host system as a character that would be embedded in a text file. When that file is sent back to the IBM Personal Computer the speaker beeps when the Ctrl-G character is encountered.

Many computer systems use these control characters for special purposes. You should understand how they are handled by your host system before using them.

Using Function Keys When Running as a Terminal

There are three keys that you need to know about when running as a terminal. They are the Function Keys *F1*, *F2*, and *F3* on the upper left corner of the keyboard. These keys operate as follows:

F1 – Communications Line Attention

Pressing this key produces a BREAK signal on the communications line. This signal is generally used to interrupt processing on the host system. The F1 key is also used to interrupt file transfer operations (like Upload or Download).

F2 – Go to Function Selection Menu

When this key is pressed, you are presented with a menu that permits selection of the Upload, Download or Compare functions. Also, you may return to Terminal Mode, to the Terminal Selection Phase of operation, or to BASIC. The section, “Escaping from and Returning to Terminal Operation,” discusses this menu in more detail.

F3 – Display Error Message

Error messages are displayed on the bottom line of this screen. If another message is produced while one is being displayed, it is kept in a list. Pressing F3 replaces the displayed message with the next one on the list.

If there is a message on the list waiting to be displayed, an * is displayed to the left of the current message. You hear a tone (beep) each time a message is displayed or added to the list. See the section “Dynamic Messages at the Bottom of the Screen” in Chapter 4 for further details. These three function keys are all you need for most terminal operations.

Keys F4, F5, and F6 provide additional function.

F4 – Enable/Disable Receive Error Checking

As described in Chapter 4, under “Enabling/Disabling Receive Errors,” certain dynamic error messages may be either enabled or disabled. If these messages are disabled, pressing the F4 key enables them. If these messages are enabled, pressing the F4 key disables them. At system startup, these messages are *disabled*. When a change in enable/disable status is made, an appropriate message is displayed at the bottom of the screen.

F5 – Force to Sending State

When operating as a half duplex terminal in the RECEIVING state, pressing this key forces the program to SENDING state. This key should not normally be needed as a Communications Attention (F1) usually performs the same function.

F6 – Enable/Disable Hex Listing

When operating as a terminal, pressing F6 sets the Communications Program so each line of text transmitted to the IBM Personal Computer from the host system is displayed as normal text followed by the hexadecimal codes for each of the characters. Characters that were specified for deletion (see “First, Second and Third Characters to be Deleted” in Chapter 2) are not displayed. To turn off this mode of operation, press the F6 key again.

IMPORTANT: The BASIC Terminal Program takes significantly longer to process a line in Hex Listing mode than in normal mode. When using this mode for bit rates of greater than 300 bits/sec., buffer overflow may occur.

Function Keys F7 to F10 are disabled, not used.

Using The Numeric Keypad

You may use the Num Lock key to set the numeric keypad so it works like a calculator keypad. Pressing the *Num Lock* key shifts the numeric keypad into its own “uppershift” mode, so that you get the numbers 0 through 9 and the decimal point, as indicated on the keytops. The numeric keypad is disabled except when in this numeric mode.

Special Key Combinations

You should be aware of the special functions of the following combinations of keys:

Ctrl-Break

This combination interrupts program execution and returns to BASIC. In general, this function should only be used when there is no other way of recovering from a problem with the Communications Program.

Alt-Ctrl-Del

This combination performs a *System Reset*. In other words, it is the same as switching the computer from off to on. All three keys must be pressed at the same time. After a System Reset, the Communications Program is reloaded. Thus, this function should be used only if no other means of problem recovery works.

What You Must Do First

We mentioned that you would need DOS. The Communications Program diskette that you purchased contains only a maintenance program and the program itself. Before you can run the program you must add some DOS programs to the Communications Program diskette.

Also, to download during your session you must use diskettes formatted by DOS.

Putting the DOS Programs onto the Program Diskette

The procedure you are about to follow assumes you are starting with your system turned off. There are two different procedures depending on whether your system has one diskette drive or two. In either procedure you will need to press the Enter key when you are finished typing an entry. The procedure for initially putting the DOS programs on the program diskette and for updating the diskette (if you get a new version of DOS someday) is the same.

Procedure for a Single Diskette Drive System

1. Put the DOS diskette in the drive.
2. Turn on the system.

The system responds:

Enter today's date (m-d-y):

3. Enter the date. For example if today were July 6, 1982 you would type: 7-6-82

The system responds:

The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981

A>

4. Type:

b:update

The system responds:

Insert diskette for drive B: and strike
any key when ready

5. Remove the DOS diskette and insert the Communications Program diskette.
6. From this point, you are alternately told to insert either the DOS diskette or the program diskette several times until all the needed DOS programs are put onto the program diskette. When the system tells you to "Insert diskette for drive A: and strike any key when ready", insert the DOS diskette. When it says "Insert diskette for drive B: and strike any key when ready", insert the program diskette. When all the DOS programs are finally on the Communications Program diskette, the system responds:

Your program diskette contains the needed DOS programs. While you still have your DOS diskette available, use the DOS DISKCOPY command and make a backup copy of your program diskette. Also, if you intend to use the Download

function, you may need to format some blank diskettes which you can do now too with the DOS FORMAT command.

A>

The following shows what you see on the screen when you put the DOS programs on the Communications Program diskette:

```
Enter today's date (m-d-y): 7-6-82
IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981
```

A>b:update

Insert diskette for drive B: and strike any key when ready

A>a:sys b:

Insert diskette for drive A: and strike any key when ready

Insert diskette for drive B: and strike any key when ready

System transferred

A>copy a:command.com b:

Insert diskette for drive A: and strike any key when ready

Insert diskette for drive B: and strike any key when ready

1 File(s) copied

A>copy a:date.com b:

Insert diskette for drive A: and strike any key when ready

Insert diskette for drive B: and strike any key when ready

```
1 (File(s) copied  
A>copy a:basic.com b:
```

Insert diskette for drive A: and strike any key when ready

Insert diskette for drive B: and strike any key when ready

```
1 File(s) copied  
A>type b:message
```

Your program diskette contains the needed DOS programs. While you still have your DOS diskette available, use the DOS DISKCOPY command and make a backup copy of your program diskette. Also, if you intent to use the Download function, you may need to format some blank diskettes which you can do now too with the DOS FORMAT command.

A>

Procedure for a Two Diskette Drive System

1. Put the DOS diskette in Drive A.
2. Put the Communications Program diskette in Drive B:

3. Turn on the system.

The system responds:

Enter today's date (m-d-y):

4. Enter the date. For example, if today were July 6, 1982, you would type: 7-6-82

The system responds:

The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981

A>

5. Type:

b:update

When all the DOS programs are copied to the Communications Program diskette, the system responds:

Your program diskette contains the needed DOS programs. While you still have your DOS diskette available, use the DOS DISKCOPY command and make a backup copy of your program diskette. Also, if you intend to use the Download function, you may need to format some blank diskettes which you can do now too with the DOS FORMAT command.

A>

The following shows what you see on the screen when you put the DOS programs on your Communications Program diskette:

Enter today's date (m-d-y): 7-6-82
The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981

A>b:update

A>a:sys b:
System transferred

A>copy a:command.com b:
1 File(s) copies

A>copy a:date.com
1 File(s) copies

A>copy a:basic.com b:
1 File(s) copied

A>type b: message

Your program diskette contains the needed DOS programs. While you still have your DOS diskette available, use the DOS DISKCOPY command and make a backup copy of your program diskette. Also, if you intend to use the Download function, you may need to format some blank diskettes which you can do now too with the DOS FORMAT command.

A>

CHAPTER 2. GETTING STARTED

Before going on, if you have not put the DOS programs on the Communications Program diskette, then go to Chapter 1 and perform this task because the Communications Program will not run without the DOS programs.

Loading the Communications Program

To load the Communications Program, do this:

1. Insert the Communications Program diskette in the diskette drive (Drive A for a two drive system).
2. Turn on the computer.
3. When the screen displays:

```
A>date  
Current date is 00-00-80  
Enter new date:
```

Type in the date in the form mm-dd-yy, where: mm is the month, dd is the day and yy is the last two digits of the year (for example, 81 for 1981).

4. After you enter the date, the Communications Program loads and begins execution. It immediately loads the Base Program of the package. The Base Program is file RS232INT.BAS on the program diskette.
5. You then see the message:

```
What screen width (40 or 80 characters) do you wish?
```

Answer this request by entering either **40** or **80**. You should use a width of 80 characters unless the monitor you are using cannot display 80 characters legibly.

6. The first of the selection menus is now displayed.

7. If you wish to use a separate diskette for saving or loading terminal specifications or for transferring files, you may now remove the Communications Program diskette from the drive and insert your new diskette. When specifying filenames, remember that the default diskette drive is Drive A.

Selecting a Terminal

When you start up the Communications Program, you are in the *Terminal Selection Phase*. You are given a series of menus used for selecting the desired terminal that the IBM Personal Computer will simulate and the detailed features of that terminal. Although the items in the menus are, for the most part, self-explanatory, this section discusses all of them in detail.

Using Menus during Terminal Selection

There are three levels of menus you see during the Terminal Selection Phase. The first-level menu lists the different terminal types that can be selected. When one of these possible terminals is selected, a second-level menu lists the terminal options that can be specified for the selected terminal. When one of those options is selected, a third-level menu lists the possible choices for that option.

Half or Full Duplex Terminal Protocols

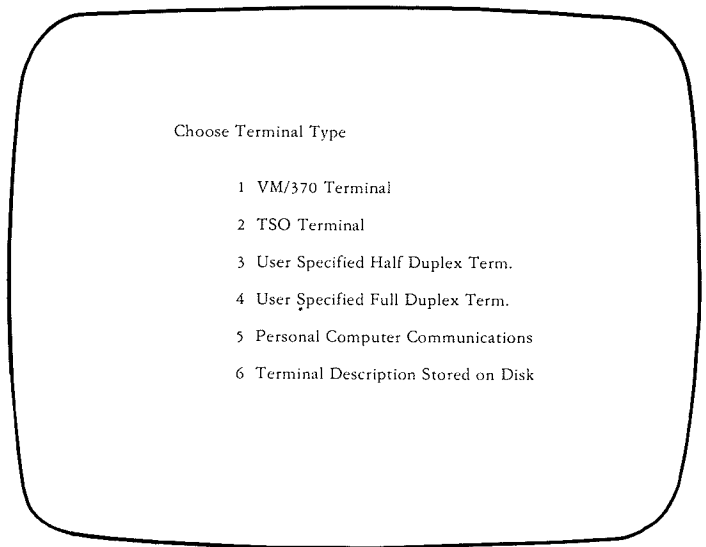
Before discussing the terminals you can select, let's describe the difference between half and full duplex terminal operation. There are two protocols used by most computer systems for asynchronous communication. A protocol is simply the set of rules set up for passing information back and forth between computers. A half duplex protocol assumes there is one circuit wire between the two computers that must be used for both sending and receiving. Though there may actually be separate physical circuits for sending and receiving, a half duplex protocol only requires one.

A full duplex protocol assumes there are two circuits available, one for transmission in each direction. With full duplex transmission, each computer can be sending information simultaneously. With half duplex transmission, rules must be set up so that transmission takes place in only one direction at a time.

The terminal execution portion of the Communications Program has a separate routine for each of these protocols. See Chapter 5 for more details on these two protocols.

Terminal Selection Menu

The Terminal Selection Menu looks like this:



VM/370 Terminal

A terminal that operates with the IBM VM/370 System Control Program running on an IBM computer. For a discussion of the protocol used by the VM/370 terminal see "Chapter 5. How Terminal Communications Works."

TSO Terminal

A terminal that operates with the IBM MVS TSO System Control Program running on an IBM computer. Again, this terminal is discussed in more detail in Chapter 5.

User Specified Half Duplex Term

Selecting this menu item lets you specify all of the pertinent parameters for a half duplex protocol terminal.

User Specified Full Duplex Term

Selecting this menu item lets you specify all of the pertinent parameters for a full duplex protocol terminal.

Personal Computer Communications

This menu item selects a full duplex terminal protocol so two IBM Personal Computers with appropriate features can talk to each other. Obviously, both machines should use the Asynchronous Communications Support and each should select this terminal type. This terminal protocol can also be used for talking to other terminals or non-IBM personal computers.

Terminal Description Stored on Disk

This choice recalls a terminal specification stored on a diskette file. When selected, you are asked for the name of the file where the specification is stored. The file is accessed and the parameters of the specified terminal are loaded. You are then given a menu that lets you change those parameters or lets you start up the specified terminal.

The program attempts to load the terminal description from the file *name.TER*. Here *name* is the specified file name. If you wish to specify the diskette drive on which the file resides, prefix the name with A: or B:. If the prefix is omitted, the program looks for the file on Drive A. See “Save Current Terminal Specification” later in this chapter for how terminal specifications are stored.

Terminal Parameter Selection Menu — Half Duplex

Once you select a particular terminal, you are given a menu listing the parameters that you can specify for that terminal. The number of items presented in this menu will vary depending on the terminal.

If you request **User Specified Half Duplex Term.** from the Terminal Selection Menu, the following menu is displayed:

Choose a Terminal Feature

- 1 Line Bit Rate
- 2 Type of Parity Checking
- 3 Number of Stop Bits
- 4 Line Turnaround Char. Sent to Host
- 5 Line Turnaround Char. Sent by Host
- 6 First Character to be Deleted
- 7 Second Character to be Deleted
- 8 Third Character to be Deleted
- 9 Line End Character Sent by Host
- 10 Start up Selected Terminal
- 11 Return to Terminal Selection Menu
- 12 Save Current Terminal Specification

Note: You need to understand some rather technical concepts to determine what the settings should be for some of the items in this menu. If you are unfamiliar with these concepts, someone at the computer center where your host computer is located should be able to help you. Another good way of determining settings is to check the equivalent setting on an ASCII terminal that you *know* works with the host computer you are planning to use.

Line Bit Rate

The line bit rate describes the rate (speed) at which characters are sent on the transmission line. The higher this rate, the faster the transmission will be. Generally, this rate is determined by the bit rate that the transmission equipment will handle and/or the bit rates available at the input ports for the computer being accessed by a terminal.

When you select Line Bit Rate, you are given the following menu:

Choose a Line Rate (Bits/Sec)	
1	75
2	110
3	150
** 4	300
5	600
6	1200
7	1800
8	2400
9	4800
10	9600

** Indicates current default

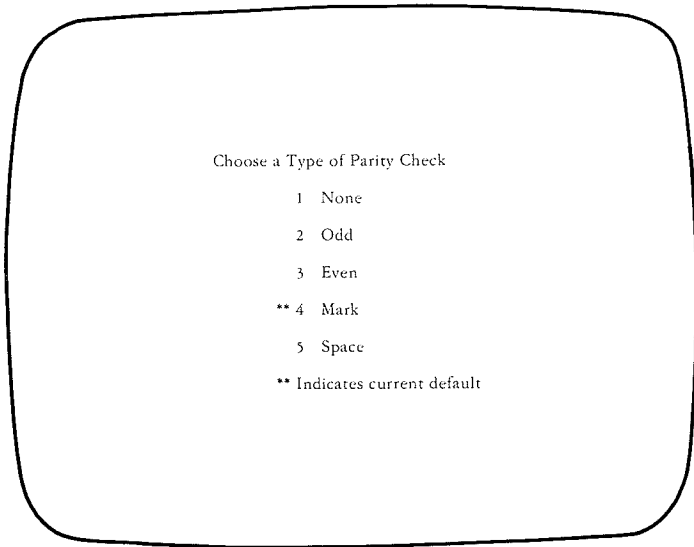
Notes: 1. Often bit rates are referred to as *baud* rates. Thus a *300 baud acoustic coupler* would typically transmit at 300 bits/second.

2. While higher bit rates can be selected, operation without a loss of data (except when listing a file at the terminal) is supported only for rates of 2400 bits/second and lower. Listing a file at the terminal is supported only for rates of 1200 bits/second and lower.

Type of Parity Checking

Characters transmitted over an asynchronous communications line are sent serially as sequences of ones and zeroes (represented by two different voltage levels) representing each character. A parity bit is a bit that is added onto each character transmitted so the character may be checked for accuracy at the receiving end. Usually you set the parity to match that expected by the host computer. The Communications Program lets you set this parity bit in a number of different ways.

If you select the **Type of Parity Checking** menu item, you are given the following menu:



None. No parity bit is added to the character transmitted. If None is specified, 8 bits of data are transmitted for each character. For all other parity specifications (as described below) the 7 low-order bits of the character are transmitted together with the parity bit. (In general, for the 128 character ASCII character set, 7 bits are sufficient for each character transmitted.)

Odd. The parity bit is set so the sum of all of the bits (including parity) of the character transmitted is odd.

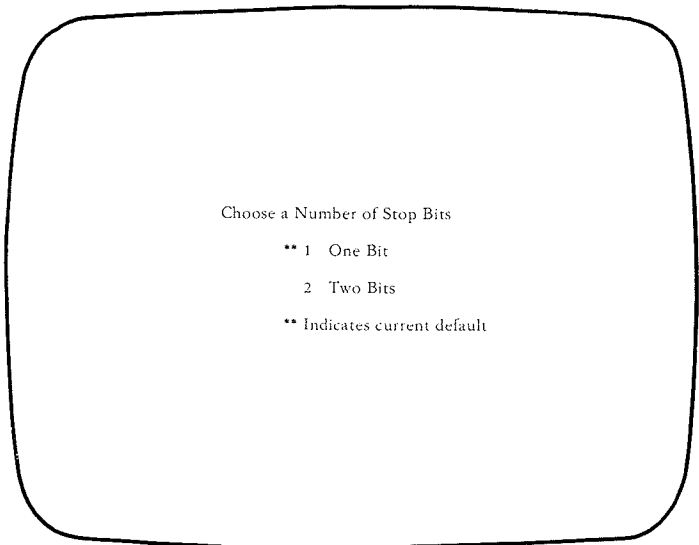
Even. The parity bit is set so the sum of all of the bits (including parity) of the character transmitted is even.

Mark. The parity bit is always set to 1.

Space. The parity bit is always set to 0.

Number of Stop Bits

In asynchronous transmission, one or two extra bits are always placed on the end of each character transmitted. These bits let the receiver detect the beginning of the next transmitted character. Usually, only one stop bit is required, but the Communications Program lets you select either 1 or 2 stop bits with the following menu:



Choose a Number of Stop Bits

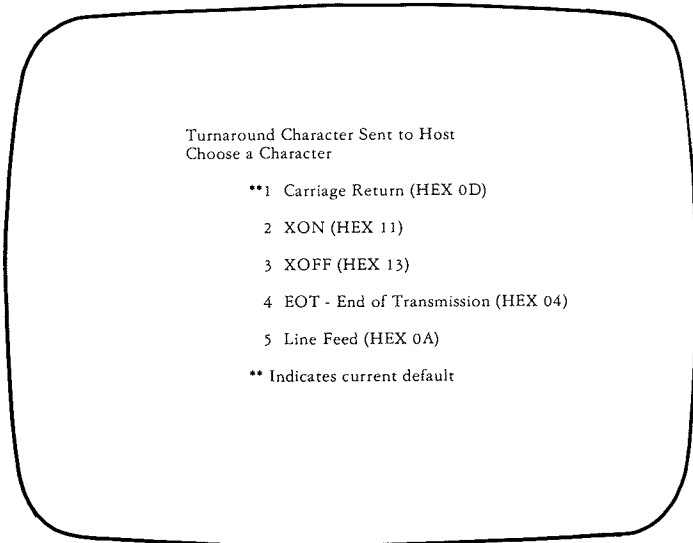
- ** 1 One Bit
- 2 Two Bits

** Indicates current default

Note: If 110 bits/sec. line bit rate is selected, the number of stop bits is always 2.

Line Turnaround Char. Sent to Host

You tell the host computer that you have completed the typing a line of text at the terminal, by pressing the Enter key. When you press the Enter key, the program translates the character produced by this keystroke to the character selected from the following menu:



For a half duplex protocol terminal, this turnaround character signifies to the host computer the end of transmission from the terminal. The host computer assumes it can go into *Sending* mode (transmitting its reply to the line just sent) and the terminal will be in *Receiving* mode.

For a full duplex protocol terminal, the turnaround character indicates the end of a line of input sent to the host computer. Typically the host computer takes action on that line and sends back a response. However, for many host computer systems, you may continue to transmit input from your terminal while the host computer is processing the previous line. Thus, one or more lines (each ending with a turnaround character) might be stacked in an input buffer of the host computer awaiting processing.

Note: The XON and XOFF characters that are used as turnaround characters are being used simply as control characters to signal a specific event to the host computer (or to your IBM Personal Computer). This use should not be confused with the use of these characters in the XON/XOFF support described in a section that follows. Specifically, VM/370 and TSO do not use the XON/XOFF support as described below. They may, however, use the XON or XOFF control characters for their own control purposes.

Line Turnaround Char. Sent by Host

With half duplex protocol, the host computer transmits a character to indicate it is ending its transmission and the terminal can begin its transmission. The same menu described above in "Line Turnaround Character Sent to Host" is also available for this turnaround character.

The line turnaround character from the host is only used in the half duplex protocol. It signifies that the host computer has completed its response and will not transmit any information until it receives a line of input from the terminal ending as described in "Line Turnaround Characters Sent to Host." The

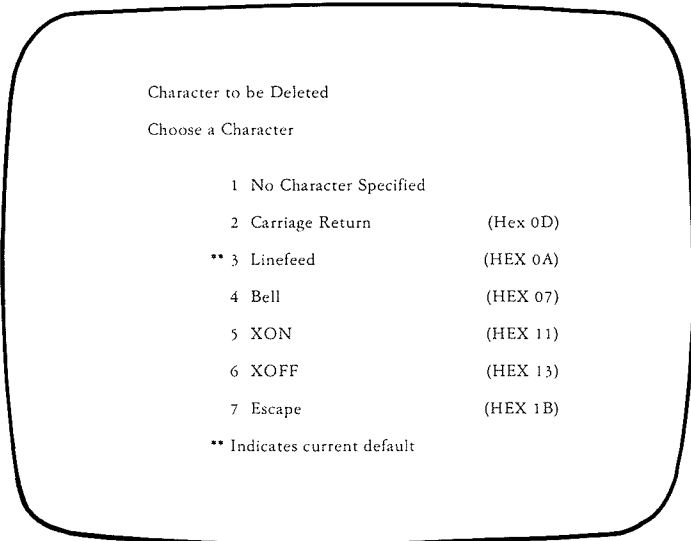
output from the host may, however, contain more than one line of text; these lines are separated by line end characters. (See "Line End Characters Sent by Host" later in this section.)

First, Second, and Third Characters to be Deleted

In communicating with a host computer, the host may transmit characters to a terminal that you do not want displayed on the screen. The Communications Program always deletes any Rubout characters (Hex 7F) detected in the text stream.

There are three menu items on the Feature Selection menu: **First Character to be Deleted**, **Second Character to be Deleted** and **Third Character to be Deleted**. If you select any one of these items, the menu of possible characters for deletion given below is displayed. For each of the **Character to be Deleted** menu items, you may choose one character to be deleted. Thus, you can delete a maximum of three different characters in addition to Rubout.

This is the menu displayed when you select any of the **Characters to be Deleted** items:



Note: If **No Character Specified** is selected, no character is deleted for that menu item.

For example, when specifying a terminal to operate with a VM/370 host system, you might wish to delete Rubouts, Linefeeds, and XOFF characters. Rubouts are always automatically deleted. From the **First Character to be Deleted** menu you would select **Linefeed**. From the **Second Character to be Deleted** you would select **XOFF**. The **Third Character to be Deleted** Menu would be left at the default setting of **No Character Specified** because only two characters need be deleted.

Line End Characters Sent by Host

The character selected from this menu specifies the end of line character sent from the host computer. This character indicates that a new line should be started on the terminal display.

Many host systems transmit both a carriage return and a linefeed character at the end of each line. In general, one of these characters should be deleted (see “**First, Second, and Third Characters to Be Deleted**”, above) and the other specified as the line end character sent by host.

The characters that can be selected for line end characters are the same as those for “**Line Turnaround Characters Sent to Host**” discussed above.

Start up of Selected Terminal

Selecting this item ends the Terminal Selection Phase of operation and starts up a terminal. The terminal started is the one with the currently specified parameters, (that is, either the one you may have just specified parameters for or the one that was just loaded from diskette). The section “**Starting Up as a Terminal**” covers this operation in detail.

Return to Terminal Selection Menu

Selecting this option returns you to the Terminal Selection Menu.

IMPORTANT: All terminal parameters that were changed are lost unless a terminal specification was stored as a diskette file.

Save Current Terminal Specification

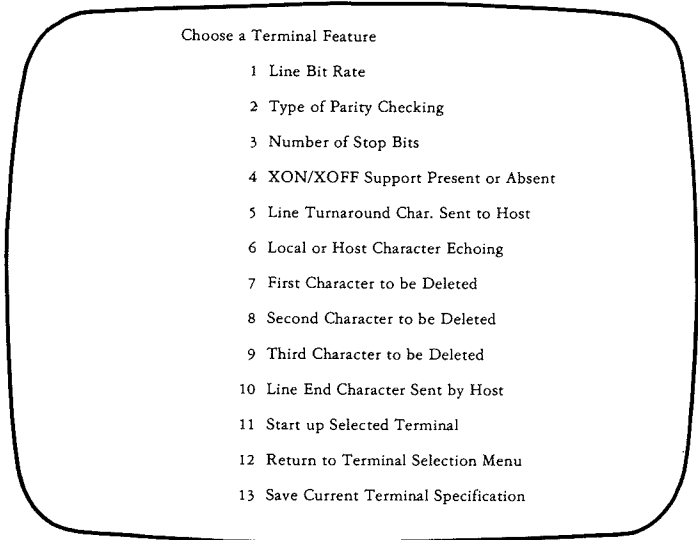
Choose this item to save the current terminal specification on a diskette file. You are asked the name of the file on which the specification is to be saved. When you enter the filename, the specification is stored on the file *name. TER* on the diskette in Drive A. Here *name* is the specified filename.

To place a file on a diskette in a specific drive, you can specify the drive letter as a prefix followed by a colon. Thus, giving the name *b:myterm* would save the terminal specification on file MYTERM. TER on the diskette in Drive B.

A saved specification can be loaded (see the previous section “Terminal Selection Menu”), modified, and stored back on the diskette under the same or a different name.

Terminal Parameter Selection Menu — Full Duplex

If you request **User Specified Full Duplex Terminal** from the Terminal Selection Menu, the following menu of parameters is given:



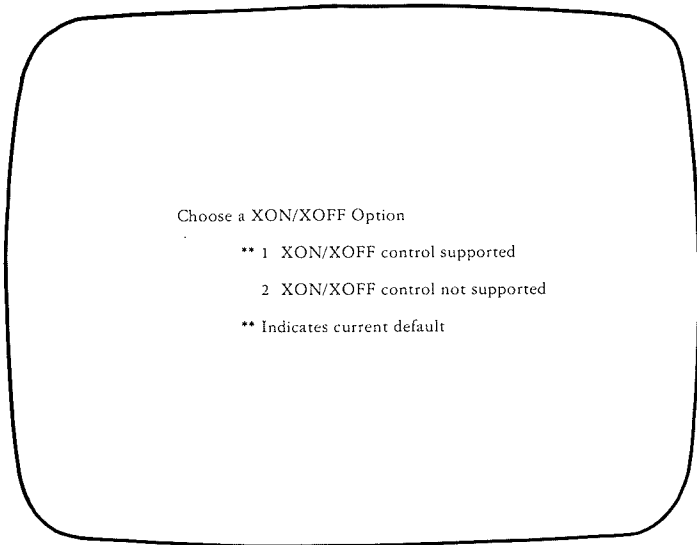
All of the items in this menu were already described in this chapter except for **XON/XOFF Support Present or Absent** and **Local or Host Character Echoing**.

XON/XOFF Support Present or Absent

The XON/XOFF support is part of a full duplex protocol. It is commonly used by many computer systems in talking with terminals (or other computer systems.) It lets a computer or a terminal signal another remote computer or terminal to stop transmitting data or to restart transmission.

The rules for this protocol are simple. When the IBM Personal Computer wishes a remote machine to stop transmitting, it sends XOFF (a Hex 13). When it wants the remote machine to start sending again, it sends XON (a Hex 11). The IBM Personal Computer also follows these rules during data transmission. If it is transmitting and receives an XOFF from the remote machine it stops until it receives an XON.

If you choose the **XON/XOFF Support Present or Absent** menu item, the following menu is displayed:



If you select **XON/XOFF control supported**, the terminal protocol follows the rules just discussed. An XOFF is sent by the IBM Personal Computer when there is danger of the receive buffer in the Base Program overflowing. When there is again a safe amount of space left in this buffer, an XON is sent.

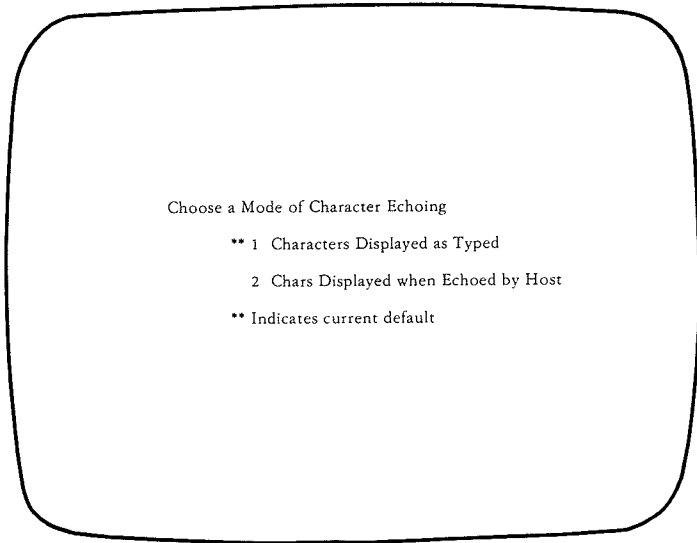
If you select **XON/XOFF control not supported**, this feature is turned off. Any system that does not support XON/XOFF must either transmit at a bit rate which the Communications Program can handle, or use some other protocol to guarantee that transmitted data will not be lost.

Local or Host Character Echoing

One commonly used full duplex protocol specifies that each character typed on a terminal keyboard and then transmitted will be echoed (sent back when it is received) by the host computer. The terminal then displays only characters received from the communications line. This mode of operation is *host echoing of characters*. If the characters are not being echoed by the host, they should be displayed on the terminal as they are typed. This mode of operation is *local echoing of characters*.

Host echoing of characters has the advantage that when you see a character displayed on your screen, you know it was received correctly by the host system. An additional advantage is that the host computer has full control over what is displayed on your local terminal. As a simple example, the host computer could suppress the display of a logon password when it is entered. A disadvantage to host echoing of characters becomes apparent when the host computer is located at such a distance from the terminal that the delays in transmission of characters become noticeable. In this case, the delay between the time when a key is struck and the time the character appears on the screen can be annoying.

If you select **Local or Host Character Echoing**, the following menu will be displayed:



If the host system you are using echoes characters, select **Chars Displayed when Echoed by Host**.

If the host system does not echo the characters typed at the terminal, then choose **Characters Displayed as Typed**.

Note: Some terminals and modems have a switch that can be set to either Half Duplex or Full Duplex. This switch actually selects whether there will be local character echoing (Half Duplex setting) or host character echoing (Full Duplex setting). Thus, the selection of a mode of character echoing with the Communications Program is equivalent to setting the Half Duplex/Full Duplex switch on some terminals.

Format of Parameters Stored for Terminals — A Summary

The parameters that describe a terminal are:

1. Terminal Bit Rate (in bits/second)

Values are:

- 1 75 bps
- 2 110 bps
- 3 150 bps
- 4 300 bps
- 5 600 bps
- 6 1200 bps
- 7 1800 bps
- 8 2400 bps
- 9 4800 bps
- 10 9600 bps

2. Type of Parity Checking

Values are:

- 1 None
- 2 Odd
- 3 Even
- 4 Mark
- 5 Space

3. Number of Stop Bits

Values are:

- 1 One Bit
- 2 Two Bits

4. Reserved

5. Half or Full Duplex Terminal Protocol (not displayed)

Values are:

- 1 Half Duplex
- 2 Full Duplex

6. XON/XOFF Support in Full Duplex

Values are:

- 1 XON/XOFF Supported
- 2 XON/XOFF Not Supported

7. Line Turnaround Characters Sent to Host

Values are:

- 1 Carriage Return (Hex 0D)
- 2 XON (Hex 11)
- 3 XOFF (Hex 13)
- 4 EOT End of Transmission (Hex 04)
- 5 Line Feed (Hex 0A)

8. Line Turnaround Character Sent by Host

- 1 Carriage Return (Hex 0D)
- 2 XON (Hex 11)
- 3 XOFF (Hex 13)
- 4 EOT End of Transmission (Hex 04)
- 5 Line Feed (Hex 0A)

9. Local or Host Character Echoing

Values are:

- 1 Characters displayed locally as they are typed
- 2 Only characters received from Host are displayed

10. First Character to be Deleted

Values are:

- 1 No Character Specified
- 2 Carriage Return (Hex 0D)
- 3 Linefeed (Hex 0A)
- 4 Bell (Hex 07)
- 5 XON (Hex 11)
- 6 XOFF (Hex 13)
- 7 Escape (Hex 1B)

11. Second Character to be Deleted

Values are:

- 1 No Character Specified
- 2 Carriage Return (Hex 0D)
- 3 Linefeed (Hex 0A)
- 4 Bell (Hex 07)
- 5 XON (Hex 11)
- 6 XOFF (Hex 13)
- 7 Escape (Hex 1B)

12. Third Character to be Deleted

- 1 No Character Specified
- 2 Carriage Return (Hex 0D)
- 3 Linefeed (Hex 0A)
- 4 Bell (Hex 07)
- 5 XON (Hex 11)
- 6 XOFF (Hex 13)
- 7 Escape (Hex 1B)

13. Reserved

14. Type of Terminal (for Upload/Download use)

Values are:

- 0 Upload/Download not supported
- 1 VM/370 Terminal
- 2 TSO Terminal
- 3 Two Personal Computers Talking

15. Line End Characters Sent by Host

Values are:

- 1 Carriage Return (Hex 0D)
- 2 XON (Hex 11)
- 3 XOFF (Hex 13)
- 4 EOT End of Transmission (Hex 04)
- 5 Line Feed (Hex 0A)

Parameters 16 through 18 are reserved. Parameters 19 and 20 are available for a user specified terminal.

Default Parameters for Various Terminals

The chart that follows gives the default parameters for the various terminals in the Terminal Selection Menu. The following notation is used.

- *(Asterisk)** Indicates that this parameter can be changed by the user by selection of a menu item during the terminal selection phase of operation.

- N/A** Not Applicable. Some parameters are only used by one terminal protocol. (For example, Character Echoing applies only to a full duplex terminal.) N/A parameters are ones not used by the protocol of a particular terminal.

- Not Used** These parameters are reserved.

TERMINALS PARAMETERS	VM/370	TSO	User Specified Half Duplex	User Specified Full Duplex	Personal Computer Communication
1. Line Bit Rate	*300 BPS	*300 BPS	*300 BPS	*300 BPS	*300 BPS
2. Parity Checking	* Mark	*Mark	*Mark	*Mark	Even
3. Number Of Stop Bits	One	One	*One	*One	One
4. Reserved	Not Used	Not Used	Not Used	Not Used	Not Used
5. Terminal Protocol	Half Duplex	Full Duplex	Half Duplex	Full Duplex	Full Duplex
6. XON/OFF Support	N/A	Not Supported	N/A	*Supported	Supported
7. Turnaround Char. To Host	*Carriage Return	*Carriage Return	*Carriage Return	*Carriage Return	Carriage Return
8. Turnaround Char. From Host	XON	N/A	*XON	N/A	N/A

TERMINALS PARAMETERS	VM/370	TSO	User Specified Half Duplex	User Specified Full Duplex	Personal Computer Communication
9. Local or Host Echo	N/A	Local	N/A	*Local	Local
10. First Deleted Char.	Linefeed	Carriage Return	*Linefeed	*None	None
11. Second Deleted Char.	XOFF	XON	*XOFF	*None	None
12. Third Deleted Char.	None	XOFF	*None	*None	None
13. Reserved	Not Used	Not Used	Not Used	Not Used	Not Used
14. Type Terminal	VM/370	TSO	Not Any	Not Any	Personal Computer
15. Line End Char. From Host	Carriage Return	Linefeed	*Carriage Return	*Carriage Return	Carriage Return
16-18 Reserved	Not Used	Not Used	Not Used	Not Used	Not Used
19 And 20 User Specified Terminal	N/A	N/A	N/A	N/A	N/A

Starting Up as a Terminal

Once you select the parameters of operation for a terminal, it is necessary to establish a hardware connection with a host computer system and then enter into the *Terminal Operation* phase of operation. This section tells you how to make that connection and how to operate the IBM Personal Computer as a terminal.

Note: For connecting an IBM Personal Computer to a VM/370 or TSO host system, this manual assumes you are familiar with the operation of those systems and in particular how to specify terminal characteristics on those systems. Whether you are a novice or an experienced user of VM/370 or TSO you should carefully read the sections below that discuss your host system.

Cable Connections

For the Communications Program to operate correctly, certain signals must be present on the cable connecting the Asynchronous Communications Adapter to the host computer or modem. If a cable is used that connects at least the pins described below (connecting other pins should have no adverse effect), the Communications Program should work correctly.

The cable should connect the following pins from the connector on one end to the connector on the other:

1, 2, 3, 4, 5, 6, 7, 8, 20

If the Communications Program fails to operate properly with a cable with these pin connections, see the section “I/O Signals used by the Communications Program” in Chapter 4.

See the section on the Asynchronous Communications Adapter in the *IBM Personal Computer Technical Reference Manual* for more details on the use of these pins.

Connecting to a Host Computer

There are two ways an IBM Personal Computer running the Communications Program can be connected to a host computer: by a direct cable connection or through a modem connected to a common carrier line (with another modem connecting the common carrier line to the host computer). When using the common carrier connection, the host computer may be located thousands of miles from the IBM Personal Computer. Either switched or dedicated (leased) common carrier lines can be used.

IMPORTANT: The modem used for such a connection must operate in a full duplex mode, even if the half duplex terminal protocol is being used.

The output of the Asynchronous Communications Adapter conforms to the EIA RS232-C Standard for communications between computers and external equipment. For more details on this interface, see the section on the Asynchronous Communications Adapter in the *IBM Personal Computer Technical Reference Manual*.

To connect to a host computer, proceed as follows:

- If you have a direct cable connection to the computer, make sure it is plugged into the Asynchronous Communications Adapter connector at the rear of your IBM Personal Computer.

- If using an acoustic coupled modem connecting to a common carrier line, then:
 - Connect power to the modem and turn on the modem.
 - If switches are provided on the modem, set them as follows:

Originate/Answer set to **Originate**

Full Duplex/Half Duplex set to **Full Duplex**

- Connect the modem to the Asynchronous Communications Adapter connector at the rear of your IBM Personal Computer.
-
- If using a direct-coupled modem connected to a common carrier line, then:
 - Connect the modem to the common carrier cable. If a telephone set is provided with the modem, connect it to the modem. Make the power connection to the modem and turn it on.
 - Set the modem so it is in Originate and Full Duplex modes.
 - Connect the modem to the Asynchronous Communications Adapter connector at the rear of your IBM Personal Computer.

- Now specify the number of the menu item **Start Up Selected Terminal** and press the Enter key.

- The message “Computer Connection NOT Established” may appear. This message is given if the Data Set Ready signal is not present on the communications cable, an indication that the host computer (or the modem) is not “ready to do business” with the terminal.

Note: If the “Computer Connection NOT Established” message does not appear, it does not necessarily indicate that the computer connection is established. If, for example, the cable from the modem to the Asynchronous Communications Adapter is not attached, the line controlling Data Set Ready signal could be “floating” and the Data Set Ready could read either present or absent.

- The message “You are Starting up as a Terminal. Check Computer or Modem Connection” is given. This message is a warning. If you have not established communications with a remote computer, you should check the connection to that computer or the local modem. See Chapter 4 for troubleshooting details.
- If your IBM Personal Computer is cable connected to the host computer, a message indicating that the host computer is ready for input should soon appear on the screen. The message “Line Connected” may also appear on the bottom line of the screen. This message indicates that the Data Set Ready signal has come on.
- If you are using an acoustic coupled modem, dial the number of the host computer and wait until you hear the tone indicating computer access. Then place the telephone receiver in the acoustic coupler cups provided on the modem.

Note: Make sure the mouthpiece and earpiece go in the correct cups.

If there is a Ready light on the modem, it should come on and stay on. In addition, the message “Line Connected” should appear on the bottom line of the screen. A message indicating the computer is ready for input may also appear.

- If you are using a direct coupled modem, set the switches on the telephone set so that a dial tone is heard on the receiver. Dial the number of the host computer. When you hear the tone indicating computer access, the message “Line Connected” should appear on the bottom line of the screen of your IBM Personal Computer. You are now connected to your host computer.

Note: Replacing the telephone receiver on the telephone set may break your connection with the host computer. See the instructions for your modem for more details.

Function Keys Used in Terminal Operation

If you are not familiar with the use of the function keys when running as a terminal, review the section “Using Function Keys When Running as a Terminal” in Chapter 1.

Escaping from and Returning to Terminal Operation

To exit from Terminal Operation, press the F2 key. You will be given the Function Selection Menu. The first three items on this menu are:

- 1 Return to Terminal Operation
- 2 Return to Terminal Selection
- 3 Return to BASIC

These functions operate as follows:

Return to Terminal Operation

Selecting this item puts you back in Terminal Operation, communicating with the host computer.

Return to Terminal Selection

Selecting this item returns you to the beginning of the Terminal Selection Phase of the program. You can then modify terminal parameters and return to Terminal Operation. You will not be disconnected from the host computer during this process.

Return to BASIC

Selecting this item returns you to BASIC.

During operation as a terminal, you can return to BASIC and then go back to Terminal Operation without interrupting connection to the host system.

In BASIC, the following BASIC commands return you to the Communications Support Program:

- **CONT** — This command returns you to the Function Selection Menu. (This command can be executed only if you have not modified any statements in the BASIC program.)

For example, to list the files on the local system, you could return to BASIC, give the BASIC command **FILES**, get a list of the available files, and then type **CONT** to return to the Function Selection Menu.

- **RUN** — This command restarts the entire Communications Program (including a reload of the Base Program). It may disconnect you from the host computer and cause a forced logout of the host system. **RUN** should be used after any modification (or reload) of the Communications Program.

In addition you can return to the IBM Personal Computer DOS with the BASIC command SYSTEM. This command will erase the Communications Program.

Using VM/370

This section describes some special details of operating as a terminal connected to a VM/370 system.

If you select the item **VM/370 Terminal** on the Terminal Selection Menu, you have the options of selecting Line Bit Rate, Type of Parity Checking, and Line Turnaround Character Sent to Host. For some VM/370 systems it may be necessary to change other parameters. If you wish that option, request **Terminal Description Stored on Disk** and specify A:VMMOD as the name of the file where the terminal description is stored. You are given a VM/370 terminal description in which any of the pertinent parameters can be modified. Set these parameters to what you wish and then save the new terminal description under a different filename for future use.

Note: If you use the **User Specified Half Duplex Term** menu item to create a terminal description for use with VM/370, you will not be able to upload or download files using that terminal.

Once you have established a connection to the computer running VM/370 the word RECEIVING should be displayed in the lower right-hand corner of the screen. You should then see the message

VM/370 ONLINE

There may be a short delay before this message appears so wait at least 30 seconds for it.

It is now necessary to signal the host computer that you are ready to begin logging in. A **BREAK** signal is used for this purpose. It is produced by pressing the F1 key.

VM/370 should respond by displaying a blank line followed by a single period (.) at the beginning of the next line. This period indicates VM/370 is ready for your terminal input. The message **SENDING** in the lower right-hand corner of the screen also indicates you can type input on the keyboard.

Note: If this period and the **SENDING** message fail to appear when VM/370 has completed output, it is an indication that **AUTOREAD** may be set **OFF** in CP (the Control Program of VM/370). In this case, Communication Attention, or **BREAK**, signals (F1 key) are required after every output from VM/370. The CMS command (after you logon and IPL CMS) **SET AUTOREAD ON** will restore the display of the period. This is the mode in which you should run VM/370.

You may now type the logon command. VM/370 will respond by requesting a password. If you enter the correct password, logon is completed.

VM/370 now operates as it would from any ASCII terminal. The following points concerning special features should be noted:

- The Terminal Attention (which produces a **BREAK** signal on the communications line) is produced by the F1 key as described above. This signal is generally used when VM/370 is producing output and you wish to stop that output.

- When you press the Backspace ← key, the cursor on the screen moves back one space. The backspace character (Hex 08) is also transmitted to the VM/370 system. By setting the Character Delete character in VM/370 equal to this character, backspacing over characters on an input line deletes the corresponding characters that were sent to VM/370. You can set the Character Delete character in VM/370 by issuing a CP TERMINAL command. This is done directly from the terminal by entering:

```
TERMINAL CHARDEL ←
```

However, if you plan to use VM/370 frequently with the IBM Personal Computer as a terminal, you can set the Character Delete character automatically by including the TERMINAL command as a line in your PROFILE EXEC. Since an ASCII Hex 08 is translated to an EBCDIC Hex 16 by VM/370, this latter value should be used in your PROFILE EXEC. To do this, when editing your PROFILE EXEC, enter the line:

```
CP TERMINAL CHARDEL X
```

Then return to edit mode and enter the subcommand:

```
Alter X 16
```

- When logging off VM/370 and using a modem, use the CP LOGOFF HOLD command to maintain the dial-up VM370 connection after logoff.

- When operating the IBM Personal Computer as a terminal, the Communications Program handles any length line (up to 255 characters) received from a host system. Thus you should set the terminal linesize in VM/370 to the maximum of 255 characters with the command:

```
TERMINAL LINESIZE 255
```

Note: If you DISCONNECT from VM/370 and then log back onto VM/370, you must reset the LINESIZE.

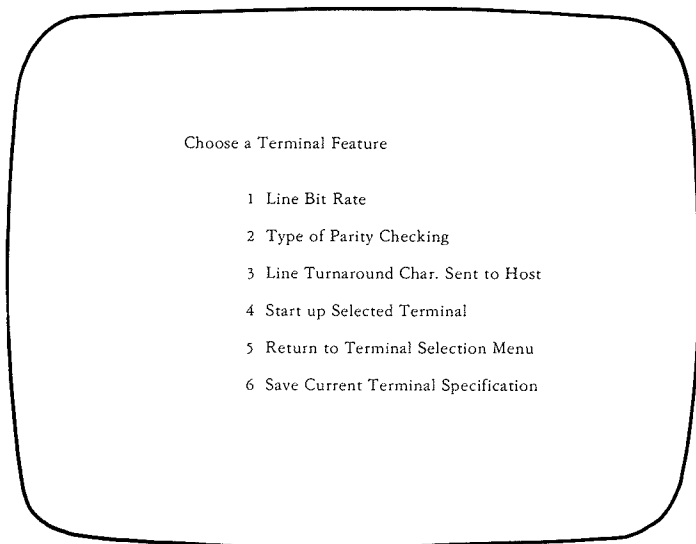
Logging onto VM/370 — An Example

Suppose you have access to VM/370 at your computer center through a 1200 bits/second cable connection. You wish to log onto VM/370. Load the Communications Program in your IBM Personal Computer and specify a line width. The following menu is displayed, giving the possible terminals you can select:

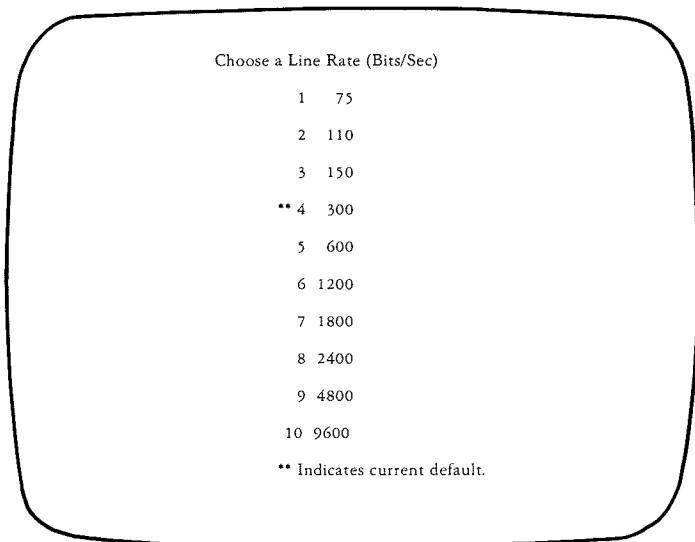
Choose a Terminal Type

- 1 VM/370 Terminal
- 2 TSO Terminal
- 3 User Specified Half Duplex Term.
- 4 User Specified Full Duplex Term.
- 5 Personal Computer Communications
- 6 Terminal Description Stored on Disk

As you wish to access VM/370, you enter a 1. The following menu is displayed:

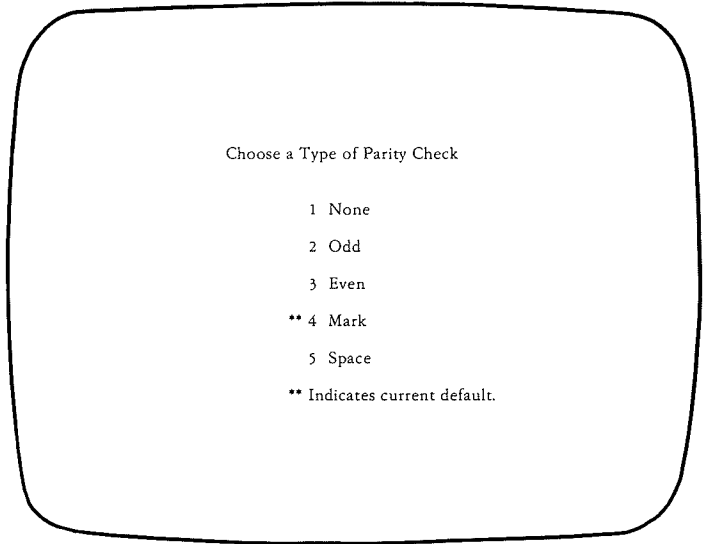


To change the Line Bit Rate, you enter a 1 and see the following menu:



The ** next to the 300 indicates that the VM/370 terminal will start up with a communications line speed of 300 bits/second (often called 300 baud) unless you change it. This is known as the *default* value.

As you are connecting to a 1200 bits/second computer port, you enter the number 6. This entry sets the Line Bit Rate to 1200 bits/second. The menu changes back to the Terminal Feature Menu. You now wish to check on the type of parity checking to be performed on the characters sent between your IBM Personal Computer and VM/370. You select item 2 in the Terminal Feature Menu and are given the following menu:



You know that VM/370 uses mark parity, so you don't change the specification. Pressing the Enter key returns you to the Terminal Feature Menu.

You can specify one other parameter for a VM/370 terminal, the line turnaround character sent to host. However, you know that the default character for this parameter (a carriage return) is the correct one for your system. Thus, you are ready to start up as a terminal.

You enter a 4 for **Start Up Selected Terminal** on the Terminal Feature Menu. You see the following messages on your screen:

Computer Connection NOT Established.

You are Starting up as a Terminal.

Check Computer or Modem Connection.

As the message "Computer Connection NOT Established" can mean that the cable connecting you to the computer center may not be physically connected at either end, you check your end of the cable and discover it is plugged into the Asynchronous Communications Adapter connector at the rear of your IBM Personal Computer. You also know that it may take a little while for VM/370 to recognize that you are ready to logon. You therefore wait. After about 10 seconds your IBM Personal Computer beeps and displays the message:

Line Connected

on the bottom of the screen. At the same time, the message

VM/370 ONLINE

displays higher up on the screen. This message indicates that VM/370 is ready to talk to you.

You now press the F1 (Communications Attention) key to signal to VM/370 that you are ready to logon. VM/370 responds with the following message:

!

The period indicates that you can now enter data. (The word SENDING in the lower right-hand corner of the screen also indicates you can enter data.)

You enter:

LOGON LOVELACE

VM/370 responds with the following:

ENTER PASSWORD:

HHHHHHHHH

SSSSSSSS

The lines of text below the password request obliterate the password on a printing terminal. You enter your password:

COUNTESS

VM/370 responds with the following:

LOGMSG - 08:41:02 EDT MONDAY 09/28/81

THERE WILL BE NO MEETING TODAY 9/28.

LOGON AT 09:06:14 EDT MONDAY 09/28/81

CMS 5.08 1/26/79

Note that your connection to VM/370 is established so that you automatically IPL CMS at logon. You now press Enter to complete startup of CMS. VM/370 displays the following:

Y (19E) R/O

R; T=0.03/0.08 09:07:07

You wish to use the Backspace ← key on your keyboard for correcting errors in entry to VM/370. You therefore enter the CP command:

TERMINAL CHARDEL ←

From now on you can correct errors with the Backspace key.

You wish to make sure that data you have VM/370 display on your IBM Personal Computer screen is correctly formatted. You therefore enter the CP command:

TERMINAL LINESIZE 255

You are now ready to use VM/370.

Using TSO

This section describes some special details of operating your IBM Personal Computer as a terminal attached to a TSO Host system.

- If you select the item **TSO Terminal** on the Terminal Selection Menu, you have the options of selecting Line Bit Rate, Type of Parity Checking, and Line Turnaround Character Sent to Host. For some TSO systems it may be necessary to change other parameters. If you

wish that option, request **Terminal Description Stored on Disk** from the Terminal Selection Menu and specify A:TSOMOD as the name of the file where the terminal description is stored. You are given a TSO terminal description in which any of the applicable parameters can be modified. Set these to what you wish and then save the new terminal description under a different filename for future use.

Note: If you use the **User Specified Full Duplex Term** menu item to create a terminal description for use with TSO, you will not be able to upload or download files using that terminal.

- The terminal selected with the TSO Terminal menu operates with full duplex protocol. Anything you type on the keyboard is immediately transmitted to TSO whether or not TSO is transmitting data to your IBM Personal Computer. If TSO is still processing a line of input when a second line of input is typed, TSO may, or may not, stack the second (and subsequent) lines for later processing. It is simple enough to determine which is the case for the TSO system you are using. Just type two commands in rapid succession and see if they both get processed.
- The Terminal Attention (which produces a BREAK signal on the communications line) is produced by pressing the F1 key, as described in Chapter 1. It is used by TSO to return you back one level of operation. For example, suppose you are trying to enter the TSO Editor and have failed to give a filetype. The TSO Editor prompts you for a filetype. If you wish to escape without answering that prompt, use the Terminal Attention (press F1).

- When operating the IBM Personal Computer as a terminal, the Communications Program handles any length of line (up to 255 characters) received from a host system. Thus, the terminal line size specified in TSO should be set to the maximum of 255 characters with the command:

```
TERMINAL LINESIZE (255)
```

- The Backspace ← key moves the cursor on the screen back one space. The backspace character (Hex 08) is also transmitted to TSO. By setting the Character Delete character in TSO equal to "backspace", backspacing over characters on an input line deletes the corresponding characters that you sent to TSO. You can set the Character Delete character in TSO by issuing the command:

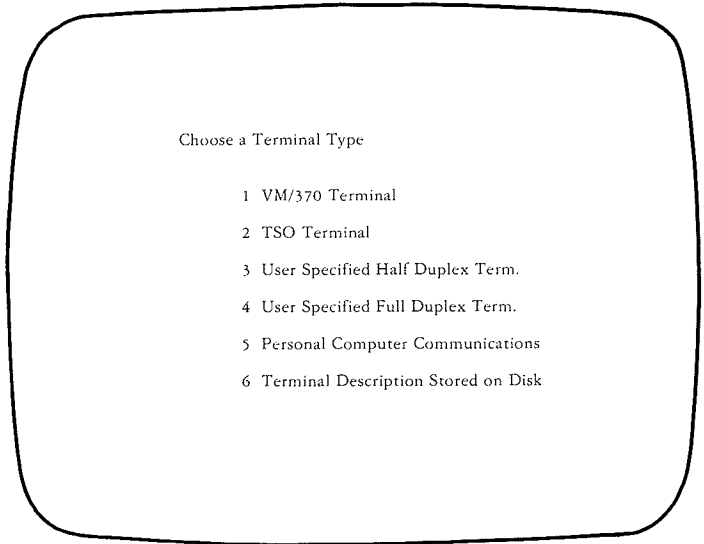
```
PROFILE CHAR(BS)
```

Note: On some TSO systems it may not be possible to set the delete character to backspace. If this command fails to have the desired effect, check with your TSO support group.

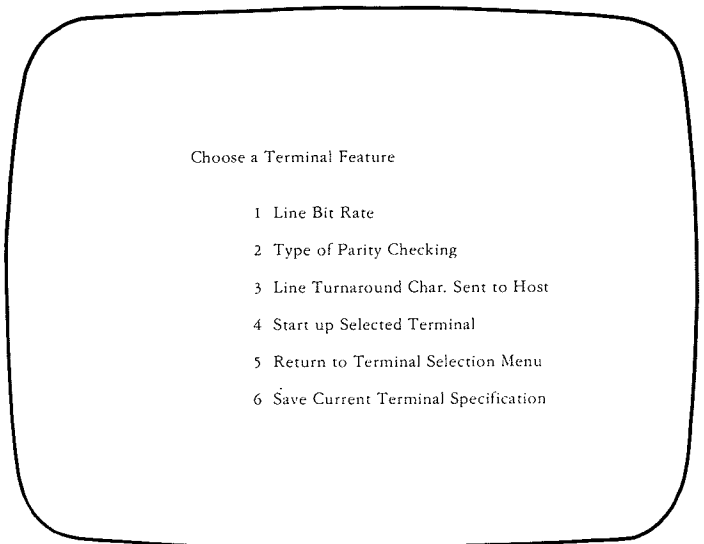
Logging onto TSO — An Example

Suppose you wish to use your IBM Personal Computer as a terminal connected to a remote computer running TSO. You plan to make this connection using an acoustic coupler that operates at 300 bits/second.

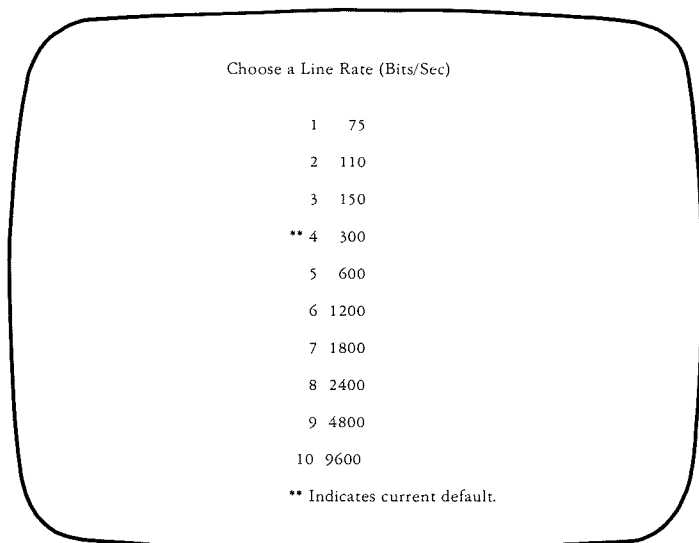
After you load the Communications Program and specify the line width, the following menu is displayed, giving the possible terminals you can select:



As you wish to access TSO, you enter a 2. The following menu is displayed:

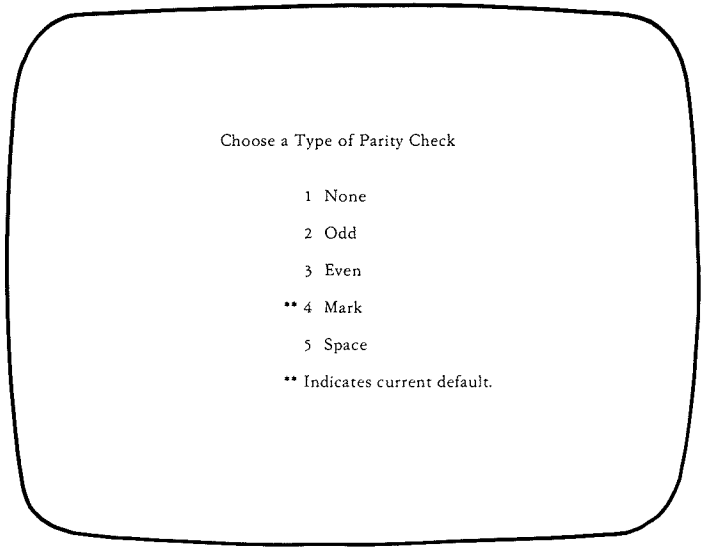


To check on the Line Bit Rate you enter a 1 and see the following menu:



The ** next to the 300 indicates the TSO terminal will start up with a communications line speed of 300 bits/second unless you change it.

As you are going to use a 300 bits/second modem for connecting to your computer center, you simply press the Enter key, and the default of 300 bits/second remains as the selected value. The menu changes back to the Terminal Feature Menu. You now wish to check on the type of parity checking to be performed on the characters sent between your IBM Personal Computer and TSO. You select item 2 in the Terminal Feature menu and are given the following menu:



You know that the port you are using to access TSO is set to use Odd Parity. Therefore, enter a 2 to select Odd. Once again you are returned to the Terminal Feature Menu.

You can specify one other parameter for a TSO terminal, the line turnaround character sent to host. However, you know the default turnaround character (a carriage return) is the correct one for your system. You are ready to start up as a terminal and connect to TSO.

You enter a 4 for **Start up Selected Terminal** on the Terminal Feature Menu.

The following message appears on your screen:

Computer Connection NOT Established.

You Are Starting up as a Terminal.

Check Computer or Modem Connection.

You now check the cable that connects your acoustic coupler with the Asynchronous Communications Adapter connector at the rear of your IBM Personal Computer. You make sure your acoustic coupler is On and set to Full Duplex and Originate. You call the operator at the computer center that is running TSO and request a number you can call to get a 300 bits/second ASCII port that can access TOS. You dial the number the operator gives you and when you hear the tone indicating the computer is ready to connect, place the telephone receiver in the cups on your acoustic coupler.

Almost immediately your IBM Personal Computer beeps and you see the following message on the bottom line of the screen:

LINE CONNECTED

Your TSO system gives you no message to indicate that it is ready for you to logon. You type your logon message as follows:

LOGON SMITH

TSO responds:

ENTER CURRENT PASSWORD FOR SMITH

You respond with your password:

SECRET

TSO replies as follows:

ICH700011 SMITH LAST ACCESS AT 06:12:46 ON MONDAY, SEPTEMBER 28, 1981

SMITH LOGON IN PROGRESS AT 06:14:05 ON SEPTEMBER 30, 1981

NO BROADCAST MESSAGES

READY

The **READY** indicates you have successfully logged onto TSO.

To use the Backspace ← key on your keyboard for correcting errors in TSO, you enter the TSO command:

PROFILE CHAR(BS)

To make sure that data you have TSO display on your IBM Personal Computer screen is correctly formatted, you enter the TSO command:

TERMINAL LINESIZE(255)

You are now ready to start your terminal session.

Operating as a Full Duplex Terminal

The **User Specified Full Duplex Term** option on the Terminal Selection Menu permits your IBM Personal Computer to be configured as a terminal that can access a wide variety of computer systems that provide ASCII support with 8 bit character codes (either 8 bits without a parity bit or 8 bits including a parity bit). To access such a system you should determine the line bit rate of the input port being accessed, the type parity checking used, the number of stop bits used, whether characters are echoed by the host system, and whether the XON/XOFF protocol is used. It is sometimes possible to determine these parameters by trial and error, but it is best to obtain them from a knowledgeable person before trying to access a particular host system.

In general, you can specify the carriage return character for both the line turnaround character sent to host and the line end character sent by host. If the system is initially run with no characters deleted,

extraneous characters transmitted by the host system may be detected. The Hex Listing feature (discussed in “Using Function Keys When Running as a Terminal” in Chapter 1) is particularly useful in determining what are the hexadecimal codes of characters you may wish to delete.

In particular, if you get extra spaces between lines of the display, you should try setting line feed as one of your deleted characters.

Communicating Between IBM Personal Computers

To communicate between two IBM Personal Computers use the item **Personal Computer Communications** on the Terminal Selection Menu.

Establishing Communication

To communicate between two IBM Personal Computers, the operators of both computers should do this:

- Start up the IBM Personal Computer and load the Communications Program, described previously.
- Specify the item **Personal Computer Communications** on the Terminal Selection Menu.
- The only parameter you may need to modify is the Line Bit Rate (the default setting is 300 bits/second). The bit rate should be the same for both IBM Personal Computers and should be set to the maximum that the communications line and modems will allow.
- Set up the modems as follows:
 - Connect power to the modems and turn them on.
 - Full duplex modems are required. If there is a switch with a Full Duplex/Half Duplex setting, set it to **Full Duplex**.

- Set the modem attached to one IBM Personal Computer to **Originate**. Set the modem attached to the other to **Answer**. If there is no switch setting on the modems, one must be an Originate modem and the other an Answer modem.
- Connect the modems to the Asynchronous Communications Adapter connectors at the rear of each IBM Personal Computer.

- Select the menu item **Start up Selected Terminal**.
- The message “Computer Connection NOT Established” usually appears followed by the message “Check Computer or Modem Connection.” See the previous section in this chapter “Connecting to a Host Computer” for more details on these messages.
- One user should now call the other. For acoustic coupled modems, once both users are talking, they need only place their respective telephone receivers in the cups on the acoustic couplers more or less simultaneously.
- Both computers should then indicate a successful connection by displaying “Line Connected” on the bottom lines of their screens.

Full duplex communications should now be established. Anything typed on either IBM Personal Computer will appear on the screens of both computers. If you press the Enter key, it will start a new line on both computer screens.

If both you and the user at the other Personal Computer type at the same time, things could be a bit confusing, so the two of you should work out some simple communications protocol. If you use a

backspace to correct an error in typing, it will delete the character on your screen, but will transmit as a delete character to the remote Personal Computer and display as a rectangular box character on its screen.

Note: If using a direct cable connection between two IBM Personal Computers, a special cable that reverses the send and receive lines must be used. When starting up the computers, connection may be established even though the “Line Connected” message does not appear.

Changing Parameters Other than Line Bit Rate

If you select the item **Personal Computer Communications** in the Terminal Selection Menu, the only parameter you can modify is the Line Bit Rate. The rest of the parameters are set as described in the section “Default Parameters for Various Terminals” in this chapter. If you wish to change any of these parameters, request **Terminal Description Stored on Disk** on the Terminal Selection Menu. When the program requests the name of the file where the terminal description is stored, reply with:

A:PCMOD

You will be given a terminal description in which any of the pertinent parameters can be modified. Set these parameters to what you wish and then save this new terminal description under a different filename for future use.

Note: If you use the **User Specified Full Duplex Term** menu item to create a terminal description for communication between IBM Personal Computers, you will not be able to use the file transfer function.

CHAPTER 3.

TRANSFERRING FILES

The Upload, Download, and Compare functions in the Communications Program permit you to transfer files between a host computer and an IBM Personal Computer. The transmit and receive IBM Personal Computer file functions let you send a file from one IBM Personal Computer to another.

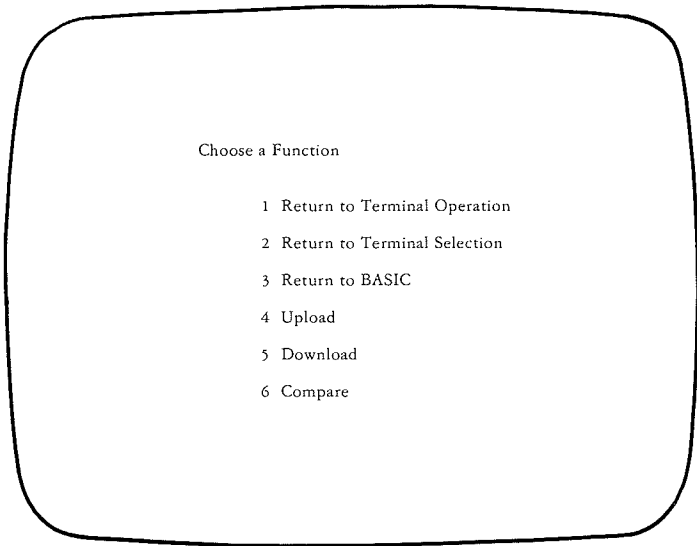
To perform these functions, you first establish communication between an IBM Personal Computer and the host computer (or between two IBM Personal Computers). You then begin the desired file transfer function. Files may include program source files, documents, or any other matter in textual form of fixed or variable record length. Program object files cannot be transferred unless converted into a text format (that is, with each byte represented by a two character hexadecimal code).

As communications must first be established between your IBM Personal Computer and a host computer (or another IBM Personal Computer) before file transfers can take place, you should be familiar with the contents of the appropriate sections of Chapter 2. You should also be familiar with the operation of the host system you are using.

IMPORTANT: Before attempting to transfer files, be sure to *read* this chapter.

Transferring Files Between an IBM Personal Computer and a Host Computer

For a VM/370 host system, with your Personal Computer in the SENDING state, press function key F2. For a TSO host system, when the READY message is displayed, press function key F2. The Function Selection Menu is then displayed:



The first three of these menu items have been previously described in Chapter 2.

The other items on this menu perform the following functions:

Upload. Selecting this item starts the operation of the Upload program. It transfers a file from the IBM Personal Computer to a host system. (See “Uploading a File” in this chapter.)

Download. Selecting this item starts the operation of the Download program. It transfers a file from the host system to the IBM Personal Computer. (See “Downloading a File” in this chapter.)

Compare. Selecting this item starts the operation of the Compare program. It compares a file on the host system with a file on the IBM Personal Computer. You can use it following either an Upload or a Download to check on the validity of the file transmitted. (See “Comparing Files” in this chapter.)

General Rules for Operation

- Only files that can be displayed as text on your terminal (with the TYPE command in DOS, for example) can be uploaded and downloaded. To upload a BASIC file, you must store it on the diskette with the A (ASCII) option of the SAVE command. Text may be in either upper or lower case. For downloading, files on VM/370 or TSO can be either fixed or variable length. Uploaded files are always placed on variable length files.

Note: The maximum length line you can upload is 130 characters. Uploading will be terminated if a line longer than 130 characters is encountered.

- You can specify filenames using either upper or lower case letters.

- In the discussion that follows, we refer to *lines* in a file. A file line is a string of characters written on a file with a line end delimiter at the end. It is essentially the same as a file record, if you are familiar with that term.
- Some modifications of files can take place during transmission. Because a single line in a host computer file may contain multiple carriage return characters, and because a carriage return character always terminates a file line on the IBM Personal Computer, downloading may translate a host file line into two or more IBM Personal Computer file records. Uploading the file back to the host computer then produces a host computer file that is not identical to the original host computer file. However, the content of the files (from the point of view of, say, printing them) would be the same. Any of these files should also compare using the Communications Program's Compare function.

IMPORTANT: If a null file line is detected in uploading, downloading, or comparing it is always converted to a line containing a single blank. This conversion is necessary to properly upload such lines (a null line signals the end of uploading to the host computer). This conversion should have no serious effect on operations except in the rare case where null lines are used to signal specific operations.

- Upload, Download, and Compare programs use the editors on VM/370 and TSO to accomplish these functions. See Chapter 5, "Operation of the Upload, Download, and Compare Programs" for more details.

Transferring Files with VM/370

The Upload, Download and Compare functions invoke the CMS Editor. Thus, the CMS Editor must be available on the VM/370 system being accessed and should be invoked when the EDIT command is given in CMS. Specifically, if the VM/370 SP is invoked, the command EDIT may invoke the XEDIT editor (through a system EXEC). The command SET IMPEX OFF can force the EDIT command to invoke the CMS Editor. The VM/370 EDIT EXEC, discussed later in this chapter, can also be used.

- If files being uploaded to VM/370 contain line end, line delete, character delete, or escape characters, then the specified actions for these characters will be performed in the process of uploading files. For example, if the escape character was set to the double quote mark (”), then double quote characters will be deleted from uploaded text. The following statement will turn off the line end, line delete, and escape characters in VM/370:

```
TERMINAL ESCAPE OFF LINEND OFF LINEDEL OFF
```

If the character delete character was changed to backspace (as described in the section “Using VM/370” in Chapter 2), then the possibility of a character deleting the previous character is unlikely.

- You should also turn off the CMS BLIP character when transferring files to VM/370. You can turn off the BLIP with the CMS command:

```
SET BLIP OFF
```

- Only files with a line length of 160 characters or less can be downloaded.

- For downloading files, the terminal linesize should be set in VM/370 so that it is larger than the longest line of any file to be downloaded. If, as discussed in “Using VM/370” in Chapter 2, the LINESIZE has been set to 255, no problems will occur.
- When downloading files, if messages are displayed on your terminal, they may appear as lines in the downloaded file. Thus, after you have downloaded a BASIC program you may find in the middle of your program a statement like:

“Hi Charlie, Please send me your version of Zorphwar. Joe”

You can suppress such messages with the VM/370 command:

SET MSG OFF WNG OFF.

Note: Setting WNG OFF cuts off warning messages from the operator.

- If Upload abnormally ends prior to saving on VM/370, you have an option of returning to the CMS Editor. At that point, any action possible in the Editor may be taken. For example, if the uploaded file was not saved due to lack of space, you can ERASE old files to make space and then save the file with a FILE command.
- In downloading a file from VM/370, cents sign characters (¢) in the file will be lost in transmission. There is no corresponding character in the ASCII character set so VM/370 deletes these characters during EBCDIC-to-ASCII translation.

VM/370 EDIT EXEC

One way you can make sure that all of the parameters required by VM/370 are properly set for file transfer operations is to use the VM/370 EXEC given below. This EXEC should be given the name EDIT EXEC. When a file transfer operation (for example, Upload) invokes the CMS Editor, it calls up this EXEC. The correct CP and CMS operational parameters are then set and the CMS Editor is invoked. Note that this EXEC does not reset the CP and CMS operational parameters so they remain as set in the EXEC after exit from the EXEC:

```
EDIT EXEC
```

```
&CONTROL OFF
```

```
CP TERMINAL ESCAPE OFF LINEND OFF LINEDEL OFF
```

```
CP SET MSG OFF WNG OFF
```

```
SET BLIP OFF
```

```
CP TERMINAL LINESIZE 255
```

```
EDIT &1 &2 &3 &4 &5 &6 &7
```

Transferring Files with TSO

When Operating with TSO, correct operation of the Upload, Download and Compare functions is dependent upon a number of parameters being set correctly. *If these parameters are not set as specified,* failure of the file transfer operations is likely.

TSO Profile Settings

The following parameters must be set as indicated using the TSO PROFILE command:

NOPROMPT	Required for correct recovery from erroneous input.
NOINTERCOM	Required to prevent spurious messages from interfering with file transfers.
NOPAUSE	Required in some situations so TSO responses can be recognized by the transfer program.
NOMSGID	Required so the TSO Editor responses can be recognized by the transfer programs during initialization.
NOMODE	Required so transfer programs can properly synchronize with responses from the TSO Editor.
NOWTPMSG	Required to prevent spurious messages from interfering with file transfers.

In addition, for Download to operate properly in all cases, the **LINESIZE** parameter should be set with the **TERMINAL** command to 255. Enter the following command:

```
TERMINAL LINESIZE (255)
```

If you do not wish to operate with the above parameters as your defaults, it is suggested you write a TSO command procedure that can be used to set these values prior to starting file transfer operations.

The following are restrictions on file transfers when using TSO:

- A file can only be uploaded to a file of filetype TEXT. This filetype must be specified as part of the filename using the dot (.) notation. Thus, a valid name for uploading would be MYFILE.TEXT. If you wish to upload a file of a type other than TEXT, upload to a TEXT file and then copy the uploaded file to a file of the desired filetype. A TSO command procedure can be written to perform such a copy.
- You cannot use quotes (") around a filename.
- Blanks are not permitted in specifying filenames for uploading, downloading, or comparing.
- Any valid filetype may be specified for downloading. It should be placed at the end of the filename using the dot (.) notation. Thus a valid name for a file to be downloaded would be ZORPH.VSBASIC.
- Filenames may be entered in either lower or upper case.

Uploading a File

To upload a file from the IBM Personal Computer to a host system proceed as follows:

- Start up the IBM Personal Computer as either a TSO or VM/370 terminal and logon to the appropriate host systems. Get to a point in the system where an entry of a command will produce only the command response. (Thus, in VM/370, after an IPL CMS, you should press the Enter key again to complete the CMS startup.)
- Press the F2 key to obtain the Function Selection Menu.

- Select the **Upload** item on this menu.
- When it is requested, give the name of the file on the diskette you wish to upload. You should give the full name of the file (including any extension). You can specify the diskette drive where the file is by preceding the filename with a prefix of either A: or B:. Thus, to upload a BASIC program, you might give the filename as B:MYPROG.BAS. If the prefix is omitted, the file is assumed to be on the diskette in Drive A. If the specified file is not on the specified diskette, you receive a message to that effect. You can then specify another filename or return to the Function Selection Menu.
- When requested, give the name of the file on the host system where you wish the file saved. You should specify this name using the format for the system to which you are connected. Thus, you should specify a VM/370 file as *filename filetype* (and optionally) *filemode*. If the file already exists on the host system, an error message is given. You may specify another filename or return to the terminal mode of operation and erase the file before uploading.

Note: For TSO, only the filetype TEXT can be specified for uploading. The dot notation should be used. For example, MYFILE.TEXT is a valid file name for uploading to TSO.

- If the filename is accepted by the host system, the message "Host Ready for Input" is given and transfer of lines of text from the file on the IBM Personal Computer begins. The number of the line currently being transferred appears in the lower right-hand corner of the screen.

When all lines in the file are transferred, you see the message:

End of Local File.

- To terminate uploading at any point, press the F1 key (Communications Attention key). Uploading is terminated with a message indicating how many lines of text were transferred. If termination fails to occur after several seconds, press the F1 key a second time.
- If uploading is abnormally ended or if the host system is unable to save the file that was created, you can cancel the upload (the file on the host system is lost). Or you can enter terminal operation in the host system editor and attempt to recover the uploaded file.
- If the number indicating the current file line being transferred stops changing or if a “Line Disconnected” message appears on the bottom of the screen, you have probably been disconnected from the remote computer.
- If the uploading process stops in midstream (due to a disconnect from the host computer, for example), you can get back to the Function Selection Menu by repetitively pressing the F1 key. See the Section “Recovering From a Disconnect during File Transfer” for more details.

Downloading a File

To download a file from a host system to an IBM Personal Computer proceed as follows:

- Start up your IBM Personal Computer as either a TSO or VM/370 terminal and logon the appropriate host computer. Get to a point in the system where an entry of a command will produce only the command response. (Thus, in VM/370, after an IPL CMS, you should press Enter again to complete the CMS startup and its associated messages.)
- Press F2 to obtain the Function Selection Menu.
- Select the **Download** item on the menu.
- When it is requested, type the name of the file on your IBM Personal Computer where you want the downloaded file saved. If this file exists already, it will be rewritten by the Download program. You should give the full name of the file (including any extension). You can specify the diskette drive (A: or B:) where the file is to be saved as a prefix. If a prefix is omitted, the file will be saved on the diskette in Drive A. If your IBM Personal Computer is unable to open the named file, a message is given. You may then specify another filename or return to the Function Selection Menu.

IMPORTANT: If you give the name of an existing file, it will be erased, even if no downloading takes place.

- When requested, give the name of the file on the host system which you wish to download. You should specify the name using the filename format for the system to which you are connected. If the file does not exist on the host

system, you will receive an error message. Again, you may specify another filename or return to the Function Selection Menu.

- If the file exists and can be accessed, the message “Host File Accessed” appears and the transfer of lines begins. The number of the file line currently being downloaded appears in the lower right-hand corner of the screen. On completion of downloading, you see the message

Download Completed.

Pressing Enter returns you to the Function Selection Menu.

- To end Download at any time, press the Communications Attention key (F1). Downloading ends with a message indicating how many lines of text were downloaded. If termination does not occur after several seconds, press F1 again.
- Other errors (such as lack of local file space) can end downloading prematurely. The error causing the termination and the point at which it occurred are given. The portion of the file already transferred will be saved on your IBM Personal Computer system.
- If the number indicating the current file line being transferred stops changing or if a “Line Disconnected” message appears on the bottom of the screen, you have probably been disconnected from the remote computer.
- If the downloading process stops in midstream (due to a disconnect from the host computer, for example), you can get back to the Function Selection Menu by repetitively pressing the F1 Key. See the section “Recovering from a Disconnect during File Transfer” for more details.

Comparing Files

You may use the Compare function to check the validity of a file just uploaded or downloaded or to check for differences in files on the IBM Personal Computer and the host system. The Compare function operates like the Download function, except that lines brought down from the host computer are compared with the lines in the specified IBM Personal Computer file rather than being written on it. To compare two files, proceed as follows:

- When running as a terminal on either a TSO or a VM/370 system, press F2. The Function Selection Menu should appear.
- Select the **Compare** item on this menu.
- When requested, give the name of the IBM Personal Computer file you wish to compare. You can prefix the filename with the diskette drive (either A: or B:) where the file resides. If no prefix is given, it assumes the file is on the diskette in Drive A. If this file does not exist, you will get an error message. You can specify another filename or return to the Function Selection Menu.
- When requested, give the name of the file on the host system you wish to compare. You should give the filename using the conventions for downloading for the host system you are using. If this file does not exist, you will be given an error message. You can specify another filename or return to the Function Selection Menu.
- If the specified file exists, comparison of lines begins. During comparison, the number of the host system line being compared is displayed in

the lower right-hand corner of the screen. If the files compare, you see a message to that effect. If two lines do not compare, comparison stops and the two unequal lines are displayed. Errors in differing file lengths (that is, the host file runs out of lines before the local file) are also indicated.

- After either a successful comparison or a detected error, control returns to the Function Selection Menu.
- To end comparison at any time, press the F1 key (Communications Attention key). Comparison terminates with a message indicating where comparison stopped in each of the files.
- If the number indicating the current file line being transferred stops changing or if a "Line Disconnected" message appears on the bottom of the screen, you have probably been disconnected from the remote computer.
- If the comparing process stops in midstream (due to a disconnect from the host computer, for example), you can get back to the Function Selection Menu by repetitively pressing the F1 Key.

Uploading and Downloading BASIC Programs

One of the common uses you may make of the Upload and Download facilities of the Communications Program is the transfer of BASIC programs between your IBM Personal Computer and a host system. This section is a short summary of rules and suggestions you should follow when writing and transferring such programs. Many of these rules are discussed elsewhere in this manual as well.

When writing BASIC programs on your IBM Personal Computer for uploading to run on a host system, make sure you understand any limitations of the BASIC on that host system. Such limitations include not only language features but the physical format of statements as well. In particular, the length of a line that will be accepted by the host BASIC must be considered (in addition to the limit of 130 characters per line built into the Upload function).

- When uploading BASIC files, make sure they were saved as ASCII files. Thus, if you have just written a new version of your program, CHESS, on your IBM Personal Computer and wish to run it on your host system, then before uploading it save it with the command:

```
SAVE "CHESS",A
```

- When downloading a BASIC program, save the file on the IBM Personal Computer with an extension of .BAS. As an example, suppose you are downloading a program ZORPH from a host system. When you are asked for the name of the IBM Personal Computer file where it is to be saved, type the name:

```
B:ZORPH.BAS
```

The downloaded file will be saved on the file ZORPH.BAS on the diskette in Drive B. When you wish to load the program to run it on the IBM Personal Computer, you would give the BASIC command:

```
LOAD "B:ZORPH
```

- When you attempt to LOAD (or RUN) a BASIC program after downloading, you may receive the BASIC error message:

Direct Statement in File

This message means that a statement without a line number occurs in the file. Such statements tend to creep into BASIC files either as titles that someone (or some computer) has put on the front of the file, as lines that are continuations of lines with line numbers, or as messages that were transmitted to you while you were downloading.

To determine where the direct statement is in your BASIC program, LIST the program. The stowaway direct statement is usually the next line beyond the last statement listed. To delete direct statements from a BASIC program file, use EDLIN, the editor, provided with the Disk Operating System.

- If, after downloading a host program file, you discover that lines in the program were broken into shorter lines (with the second portion of these lines appearing as direct statements), you may not have set the `TERMINAL LINESIZE` large enough on your host system. See the prior section in this chapter “Transferring Files” for your host system for more details on setting `TERMINAL LINESIZE`.

Transferring Files between IBM Personal Computers

Files transferred between IBM Personal Computers must be in textual form and have no lines longer than 254 characters (for example, a BASIC program saved as an ASCII file).

To transfer a file between two IBM Personal Computers, first establish communications between the computers as described in the section “Communicating between Personal Computers” above. Once this communications is established, each user should obtain the Function Selection Menu by pressing F2.

The user who is transmitting the file should now select the menu item **Transmit a Personal Computer File**. The user who is to receive the file should select the item **Receive a Personal Computer File**. The order in which these actions are taken does not matter. No matter which of the functions (transmitting or receiving a file) is ready to go first, it synchronizes its operation with the other. In fact, your IBM Personal Computer can still be in Terminal mode with the other ready to send or receive. In this case, you see an appropriate message (Ready to Send or Ready to Receive) from the other IBM Personal Computer.

Note: The protocol used to transfer files between two IBM Personal Computers can also be used to transfer files between a non-IBM personal computer and an IBM Personal Computer. The section “Personal Computer File Transfer Protocol” in this chapter discusses how to make such transfers.

The actions of each of the IBM Personal Computers is now described separately.

Transmitting an IBM Personal Computer File

When the menu item **Transmit a Personal Computer File** is selected, the following actions take place:

- You are asked for the name of the file you wish to transmit. In response to the request, the full name of the file (including any extension) should be given. You may designate the diskette drive where the file is by preceding the filename with the prefix **A:** or **B:**. If the prefix is omitted, the file is assumed to be on the diskette in Drive A. If the specified file does not exist, or if there is some other reason that the file cannot be accessed, you will see an appropriate message. You can then specify another filename or return to the Function Selection Menu.
- If the file exists and can be accessed, the message "Ready to Start Sending File" is sent to the remote IBM Personal Computer. Your IBM Personal Computer then waits for a message from the remote computer that indicates it is ready to receive a file. You can interrupt this waiting and return to the Function Selection Menu by pressing F1, the Communications Attention key.
- When the "Ready to Receive File" message from the remote IBM Personal Computer is received, it is displayed. A message is then transmitted to the remote computer indicating that transmission is about to begin. The file is then transmitted a line at a time. The number of the line currently being transmitted appears in the lower right-hand corner of the screen.

- When all of the lines in the file are transmitted, the message "Transmission Completed" is displayed. A message indicating the end of the file was reached is sent to the remote computer. Control is then returned to the Function Selection Menu.
- Operation of file transfer can be stopped at any time by pressing F1, the Communications Attention key. If problems occur in reading the diskette file (such as someone removing the diskette from the drive), transmission is also stopped. In either case, a message indicating that transmission has ended (together with the cause) is sent to the remote computer.
- File transfer can also be stopped at the request of the receiving IBM Personal Computer. When this occurs, the message "Transmission Ended at Request of Receiver" is displayed with a message describing the cause.
- If the number indicating the current file line being transferred stops changing or if a Line Disconnected message appears at the bottom of the screen, then you have probably been disconnected from the remote computer. At this point you can probably get back to the Function Selection Menu by repetitively pressing the F1 key. After reestablishing your connection with the remote IBM Personal Computer, make sure you can communicate as a terminal before attempting another file transfer. See the section "Recovering From a Disconnect during File Transfer" for further details.

Receiving a File on Your IBM Personal Computer

When the menu item **Receive A Personal Computer File** is selected, the following actions take place:

- A request is given for the name of the file where the transmitted file is to be saved. In response to this request, enter the full name of the file (including any extension). The diskette drive where the file will be placed can be designated by preceding the filename with either A: or B:. If the prefix is omitted, the file will be put on the diskette in Drive A. If there is some reason that the file cannot be written, an appropriate error message is given. You may then specify another filename or return to the Function Selection Menu.

IMPORTANT: If you give the name of an existing file, it is erased, even if no file transfer takes place.

- If the file can be written, the message "Ready to Receive File" is displayed. This message is also transmitted to the remote IBM Personal Computer that will be sending the file. A control code is appended to this message indicating the local computer is ready to receive file lines.
- The local computer waits for a message from the remote computer indicating that file lines are about to be transmitted. Periodically the message "Ready to Receive File" and an associated code are sent to the remote computer. You can interrupt this waiting and return to the Function Selection Menu by pressing the F1 key, the Communications Attention key. When waiting, any messages from the remote computer are displayed on your screen.

- When the remote computer is about to send a file, the message "Starting File Transmission" appears on your screen. As lines are received, the number of the line currently being received is displayed in the lower right-hand corner of your screen.
- When all of the lines of the file are transmitted, the message "Transmission Completed" is displayed. You may then return to the Function Selection Menu.
- You can interrupt transfer of a file at any time by pressing the F1 key, Communications Attention key. Pressing this key stops storing the received lines on your file and sends a message to the remote computer that should stop the transmission of file lines. Problems with the writing of your local file (such as running out of diskette space) can also stop the transfer operation. In that case you will see an appropriate message and a message will be transmitted to the remote computer to cause it to stop transmission.
- If the number indicating the current file line being transferred stops changing or if a Line Disconnected message appears at the bottom of the screen, you were probably disconnected from the remote computer. At this point you can probably get back to the Function Selection Menu by repetitively pressing the F1 key. After reestablishing your connection with the remote IBM Personal Computer make sure you can communicate as a terminal before attempting another file transfer. See the section "Recovering From a Disconnect during File Transfer" for further details.

Personal Computer File Transfer Protocol

To transfer files between an IBM Personal Computer and a personal computer built by another manufacturer, the program described in previous sections for transferring between two IBM Personal Computers can be used on the IBM Personal Computer. The non-IBM Personal Computer must be equipped with some type of communications interface and associated software. It is also necessary to write a program for the non-IBM personal computer that handles file transmission to and/or from the IBM Personal Computer. This program must match the protocol built into the file transfer program in the IBM Personal Computer. This section describes what the transmission and reception portions of such a program must do. On many personal computers such a program can be written in BASIC. The portion of the IBM Personal Computer Communications Program that handles Personal Computer file transfers can be used as a model for such a program. This portion of the Communications Program starts at line number 16200.

The following rules apply to both transmitting files and receiving them:

- All data is transmitted in lines and each line is terminated with a carriage return (CR\$) character.
- The following control codes are used in the descriptions below:

CR\$ Carriage Return (Hex 0D)

XON\$ XON Character (Hex 11)

XOFF\$ XOFF Character (Hex 13)

IBG\$ Begin Transmission Code (Hex 1C)

ITM\$ Terminate Transmission Code (Hex 17)

- Transmission operates with a full duplex terminal protocol in which the program checks for received information between transmission of lines of data.
- Depending on the non-IBM personal computer, carriage returns may or may not be present at the end of input lines from a communications interface. Also, they may or may not have to be placed at the end of lines sent as output to the communications interface. You should verify how line ends are handled by the non-IBM personal computer.

Transmitting a File from a Non-IBM Personal Computer

The program written for a non-IBM personal computer to transmit a file to an IBM Personal computer should operate as follows:

1. The program opens for input the file to be transmitted.
2. The program loops, reading the communications line and waiting for reception of a text line ending with the control characters IBG\$ CR\$.
3. When such a line is received, the program sends a text line ending with IBG\$ CR\$. (This line can contain an informative message as well, such as, Starting File Transmission.)

4. The program transmits the file. Each line (record) in the file should be sent as a line ending in a carriage return (CR\$).
5. While transmission is taking place the program should monitor the input from the communications line and take the following actions:
 - a. If an XOFF\$ CR\$ is seen, stop transmission of lines. When an XON\$ CR\$ is seen, resume transmission.
 - b. If a line ending in ITM\$ CR\$ is seen, stop all transmission. This line will contain as text the reason the receiving IBM Personal Computer has requested termination.
 - c. When all lines in the file have been sent, the program should send a line ending in ITM\$ CR\$. This line can contain an appropriate message such as, File Transmission Completed.

Receiving a File Transmitted by an IBM Personal Computer

The program that runs in a non-IBM personal computer and receives a file from an IBM Personal Computer should operate as follows:

1. A file is opened for output, ready for writing the received file.
2. The program loops, sending out a message ending in IBG\$ CR\$ every 15 to 20 seconds. This message may also contain text, such as, Ready to Receive.

3. During the loop in Step 2, the communication line is continually monitored for messages from the IBM Personal Computer. When a line ending in IBG\$ CR\$ is received, the program moves on to Step 4.
4. Each line received (after the one ending in IBG\$ CR\$) is stored as a file record. As these lines end with carriage returns (CR\$), the program might delete the CR\$ before storing a line. Before storing a line, the program checks it to see if it ends in ITM\$ CR\$. If it does, the program does not store that line, but closes the file and stops operation.
5. The program can stop transmission by the IBM Personal Computer by sending a line ending with an ITM\$ CR\$. This line may also contain a message giving the reason for the termination.
6. If the program is receiving lines faster than they can be stored, it can suspend transmission by sending a line consisting of an XOFF\$ CR\$ to the IBM Personal Computer. When it has caught up with the input, it can start up transmission by sending a line consisting of an XON\$ CR\$ to the IBM Personal Computer.

Recovering from a Disconnect during File Transfer

If you are transferring files between an IBM Personal Computer and a host computer or between two IBM Personal Computers and the connection between computers is broken, then it is possible to recover the portion of the file that was transferred.

First return from the file transfer program to the Function Selection Menu, usually, by repetitively pressing the F1 key. You will receive messages indicating the next actions to take. When recovering from an upload, it may be necessary to press the F1 key again after replying YES to the message asking if you wish to return to the Function Selection Menu. If this approach fails, then use the Ctrl-Break key combination to return to BASIC and then use the RUN command to restart the Communications Program.

If you experience continued problems during file transfers where the lines disconnect so frequently that it is difficult to transfer files, we suggest that you break a large file into smaller files before transmitting them.

Recovering a File after Disconnecting During an Upload

If the host system you are using permits you to restart at the point of operation where the line disconnect occurred, you should be able to log back onto the host system and be at the point in the host system editor where the disconnect occurred. You may then save the portion of the file that was

uploaded. Then return to the local operation of your IBM Personal Computer and make a smaller file for that portion of the file that has yet to be uploaded. After logging back onto the host system, you may upload the smaller file to the host using a different filename. The two files on the host system can then be merged into one file.

Because this process can be time consuming, we suggest that it only be used in situations where the length of the file involved is such that it would take significantly longer to upload the entire file again.

Recovering a File after Disconnecting during a Download

When you return to the Function Selection Menu after disconnecting during a download, the portion of the file downloaded will be preserved as a file on your IBM Personal Computer diskette. You can thus log back onto the host system, copy the portion of the host file that was not downloaded into a new file and download the new file. The two files on the IBM Personal Computer can then be merged into one file.

As mentioned above we suggest this process only be used for large files.

Recovering a File after Disconnect between Two IBM Personal Computers

If a disconnect occurred while transferring a file between two IBM Personal Computers, the return to the Function Selection Menu on the receiving computer causes that portion of the file transferred

to be saved on the diskette. It is then necessary for the sending user to make a smaller file for that portion of the file not transferred and send it as a separate file. The user of the receiving computer should save this portion of the transferred file as a separate file and then merge it with the first part of the file.

If the file is a BASIC file, the two files can be merged by using `LOAD` to load one part and `MERGE` to merge the other.

As before we suggest this process only be used for large files.

CHAPTER 4. TROUBLESHOOTING PROBLEMS AND UNDERSTANDING ERROR MESSAGES

Getting Connected to a Host System

If you are using an acoustic (or direct-wired) coupler on a common carrier (for example, telephone) line, the line quality may be poor and prevent good access to the computer. This problem is generally indicated by the connection “dropping” spontaneously shortly after the computer has been dialed up and the connection established. If there is a READY light on the modem, it will go out when the connection drops. If this happens, it is usually not a problem caused by the IBM Personal Computer. If it occurs frequently, make arrangements for a higher quality communications link.

Data Set Ready (DSR) is an input signal on one of the wires in the cable connecting to the Communications Adapter. It comes from the computer (in a cable connected system) or from the modem (in a remote connected system). This signal indicates the presence or absence of a connection to the computer. DSR is monitored by the terminal program and when its

status changes, a message is displayed on the bottom line of the screen. (See “How Messages Are Displayed” below.)

Message	Meaning
Line Disconnected	The DSR signal was turned off.
Line Connected	The DSR signal was turned on.
Line Disconnected/ Reconnected	The DSR signal was turned off and then turned on again.

Troubleshooting a Cable Connected System

When an IBM Personal Computer is cable connected to a host computer, the DSR signal may be off when the IBM Personal Computer starts up as a terminal for the first time. In that case, the message “Computer Connection NOT Established” will appear. It may be necessary to wait until (up to 30 seconds) the host computer recognized the presence of the IBM Personal Computer and activates the DSR signal.

If the “Line Connected” message (indicating the presence of a DSR signal) fails to appear in that time, check the connections between your IBM Personal Computer and the host computer, and the status of the host computer.

Troubleshooting a Modem Connected system

In a system connected through a modem (for example, an acoustic coupler), the DSR signal should appear as soon as the computer connection is established. It may still be necessary to wait a few seconds before a message appears from the remote computer indicating it is ready for input. If the modem has successfully made connection with the remote computer and the "Line Connected" message has not appeared on the IBM Personal Computer screen, check the cable connection between the IBM Personal Computer and the modem. If the DSR signal is present, and it is still not possible to communicate with the remote computer, check the switch settings on the modem. (See "Connecting to a Host Computer" in Chapter 2.)

If you do establish communications with the remote computer and get two characters on the screen for each character you type, check the Full Duplex/Half Duplex switch on the modem. It should be set to Full Duplex. There are systems where the host computer echoes back every character received from a terminal. See the section "Local or Host Character Echoing" in Chapter 2 for how to set a terminal specification to handle this situation. The modem should always be set to Full Duplex, no matter what type of character echoing you specified.

I/O Signals Used by the Communications Program

Signals on the lines (described below) are used by the Communications Program. The pin numbers referenced are those of the Asynchronous Communications Adapter connector at the rear of the IBM Personal Computer.

The following lines are used for output data and control signals from the IBM Personal Computer:

Transmit Data (Pin 2): The data transmitted by the IBM Personal Computer is sent on this line.

Request to Send (RTS) (Pin 4): This line is turned OFF when the Communications Program is first loaded. It is turned ON at the time the message "You Are Starting Up as a Terminal" appears during terminal startup. It remains on until the Communications Program is restarted (for example, with a BASIC RUN command).

Data Terminal Ready (DTR) (Pin 20): This line is turned OFF and ON at the same times the RTS line is turned OFF and ON.

The following lines are used for input data and control signals by the Communications Program:

Receive Data (Pin 3): The data received by the IBM Personal Computer is taken from this line.

Clear to Send (CTS) (Pin 5): This signal must be ON (together with DSR and DCD) in order for characters to be transmitted by the Communications Program.

Data Carrier Detect (DCD) (Pin 8): This signal is sometimes called Receive Line Signal Detect. It indicates the presence of a valid carrier signal detected by the nearest modem. It is used to control the transmission and reception of characters as described below.

Data Set Ready (DSR) (Pin 6): This signal is used by the Communications Program as an indication that the connection to the host computer or modem has been established. See the beginning of this chapter for a discussion of how this signal affects the messages displayed by the Communications Program. See below for a discussion of how this signal controls transmission and reception of data.

The other input control signal that is sensed by the Asynchronous Communications Adapter is the Ring Indicator (Pin 22). This signal is ignored by the Communications Program. However, it is possible to access this signal using the Base Program. See Chapter 7 for details.

For characters to be transmitted on the communications line by the Communications Program, the Clear to Send, the Data Carrier Detect, and the Data Set Ready signals must all be ON. If one or more of these signals are OFF, characters typed on the keyboard during terminal operation are stacked in a transmit buffer until all of these signals are ON. If sufficient characters are typed with one of these signals OFF, the transmit buffer will overflow.

For characters to be received by the Communications Program and placed in the receive buffer, the Data Carrier Detect and the Data Set Ready signals must both be ON. If either or both of these signals are OFF, received characters are ignored and lost.

Providing Control Signals for Program Operation

From the description in the previous section, you can see that for transmission and reception of data by the Communications Program, the Clear to Send, Data Set Ready, and Data Carrier Detect lines must all be ON. An ON condition requires that the line in question be held at greater than +3 volts. If any of these lines are left floating (unconnected), their state is indeterminate (either ON or OFF). Thus, the cable connection used must guarantee that Pins 4, 6, and 8 are turned either ON or OFF by a connection to a voltage that goes above +3 volts (ON) or below -3 volts (OFF).

A typical acoustic coupler handles control of these lines by tying all of them together and turning them OFF when no connection to a remote computer is present and turning them ON when that connection is established.

For a direct cable connection to a host computer, these three lines may also be tied together and turned ON and OFF to indicate whether the host system is ready or not ready. Another way a direct cable may be wired is with the Request to Send line (Pin 4) connected to the Clear to Send line (Pin 5). With this connection, when your IBM Personal Computer turns ON the Request to Send line, it automatically turns on its Clear to Send input line.

If you have problems using the Communications Program you suspect may be due to erroneous signals on the control lines, check with someone knowledgeable about the signals produced by the modem you are using or the cable connection to your host computer.

How Messages Are Displayed

The Communications Program displays a large number of information and error messages to assist you in specifying a terminal and running your IBM Personal Computer as a terminal. The error messages and their meanings are given in Appendix A. This section is a description of the different types of messages and when they appear.

Dynamic Messages, at the Bottom of the Screen

Dynamic Messages occur when your IBM Personal Computer is operating as a terminal or is transferring files. These messages appear on the bottom line of the screen. If the bottom screen line is empty when a message is generated, it is displayed there and a tone is sounded. If the bottom line of the screen is filled with a message, any other message just generated is “stacked” on a list, the tone is sounded, an asterisk (*) is displayed to the left of the bottom screen line. Up to 9 such messages are stacked.

To erase the current message and display the next one on the list (the earliest message stacked on the list is always the next one displayed), press the F3 key. The next message is then displayed. The asterisk (*) continues to appear to the left of the message line, until the last message on the list is displayed.

If you are familiar with “stacks” used by computers, the messages are placed in a FIFO (First In First Out) stack. Displaying the messages (or stacking them on the list) does not interrupt the operation of the IBM Personal Computer when running as a terminal or during file transfers.

Typical dynamic messages are the Line Connected and Line Disconnected messages described in the section “Getting Connected to a Host System” above.

If the Communications Program is waiting for the answer to a query (for example, the name of a file for file transfer), then new messages are not displayed and the F3 key does not function.

Enabling and Disabling Receive Errors

As characters are received from the communications line, a check is made to determine if parity, framing, or overrun errors have occurred. (See “SETSENSE” in Chapter 7 for a discussion of the meaning of these errors). In addition, a check is made to determine if a BREAK signal was received from the host system.

You have the option of having a dynamic message displayed each time one of these errors is detected, or suppressing these messages. The Enable/Disable Receive Error Checking key (function key F4) controls this function. Its operation is described in Chapter 1 in the section “Using Function Keys When Running as a Terminal.”

The display of these messages is normally *disabled* (turned off) when terminal operation is started. Thus, you will *not* see parity errors that may occur when connecting to a modem. While the error display function is disabled, errors that occur are still cleared. When you *enable* (turn on) the Receive Error Checking, you will see only the errors that occur *after* the function is enabled. Once you establish communication with a host system, it is suggested that you enable this function so that detected errors will be displayed.

If you appear to be getting parity errors on every character received, you should check to see what type of parity checking the host system is using. These are cases where it may not be possible to avoid parity error checks. In these cases, it may be necessary to run with Receive Error Checking disabled.

Static Messages

Other instructional or error messages may appear while you are specifying a terminal or during uploading and downloading of files. These messages are displayed above the bottom line of the screen and are generally self-explanatory. Often operation of the computer is halted until you specify what action should be taken next. You are told what options you have at any such time.

Some of the error messages you will encounter have to do with the accessing of files on your IBM Personal Computer. For the more common problems with files you will be given a specific message indicating the problem. For example, if you try to create a new file when you already have the maximum number of files (64) on your disk, you are given the message "Too Many Files".

However, for the less common file problems, you are given an error number to look up in the *IBM Personal Computer BASIC* manual. The numbers for different errors are given in Appendix A of that manual. For example, if you had no diskette in a drive when you specified a file for downloading, you would get the message "File Access Failed. BASIC Error 71." Looking up error number 71 in Appendix A of the

BASIC manual, you would find the following explanation:

71 Disk Not Ready

The diskette drive door is open or a diskette is not in the drive. Place the correct diskette in the drive and continue the program.

Messages during Upload, Download, and Compare

Upload, Download, and Compare use the TSO and VM/370 Editors to access files on their respective systems to perform the file transfer functions requested. Using this process, the text messages produced by these Editors (which normally would be displayed at a user terminal) are checked to make sure the requested function is proceeding correctly. In general, if the expected message is not correct, operation is ended, and an error message is displayed followed by the text received from the host system.

For example, if you are attempting to download a file from VM/370 and specify `SYSFILE.BASIC` as a filename for the host file, the Download program would return with the message:

Errors in Host Access. Message was:

INCOMPLETE FILEID SPECIFIED

The second line of this message comes directly from VM/370 and indicates the problem.

CHAPTER 5.

HOW TERMINAL COMMUNICATION WORKS

This section provides an overview of the Terminal Operation Phase of the Communications Program. It gives you background information if you are specifying terminal parameters or enhancing the BASIC program code.

Note: If you are reading the BASIC code, the names used in calls have been shortened from the names given in Chapter 7. Names used are as follows:

SS for SETSENSE

BR for BREAK

BF for BUFSIZE

RC for RECEIVE

SN for SEND

C% for CHAN%

D% for DAT%

E% for ERFLG%

T% for TYPE%

Initialization Phase

When the menu item **Start Up Selected Terminal** is specified, initialization code (common to both half and full duplex protocols) is executed. This code first sets a number of program constants and then makes a series of calls to the Base Program to set parameters as they were specified in the menus. In the setting of the Characters to be Deleted, the control characters are all cleared (so that none will be deleted) and then the specified characters are set to be deleted by calls to the SS (SETSENSE) entry (see Chapter 7 for details) with T% (TYPE%) set to 9.

Calls are also made to the SS entry to set the line end character sent by host and (if present) the line turnaround character sent by host so that these characters always appear as the last character of a string returned by the RECEIVE CALL.

Having specified control characters deleted in the Base Program and having the line end and turnaround characters always at the end of returned strings means that the BASIC program does not have to search strings for specific characters, permitting the BASIC terminal program to handle bit rates of up to 2400 bits/second.

The program then tests the Data Set Ready (DSR) modem to see if that signal is present. If the signal is not present, the message "Computer Connection NOT Established" is displayed.

At this point the program branches to one of two separate routines, the half duplex terminal protocol or the full duplex protocol.

Half Duplex Terminal Protocol

The half duplex terminal protocol is primarily used with VM/370. It assumes that the host computer and the IBM Personal Computer are in either a *receiving* or *sending* state. If one is in the sending state, the other is in receiving state.

Assume that initially the IBM Personal Computer is in the sending state and VM/370 is in the receiving state. Characters typed by the user are transmitted to VM/370. After typing a line of characters, the user presses the Enter key. Pressing this key generates a carriage return internally to the IBM Personal Computer. The terminal program translates this character into the character selected as the line turnaround character sent to host and transmits it. This character (which usually is a carriage return) lets VM/370 know that it should switch from receiving to sending mode. After sending the turnaround character, the terminal program switches to receiving mode.

VM/370 may send a number of lines in response to the line received from the IBM Personal Computer. Each of these lines is terminated by a line end character (typically a carriage return). When VM/370 has finished sending output, it terminates its transmission with a line turnaround character sent from host (usually an XON.). VM/370 then goes into the receiving state. The terminal program in the IBM Personal Computer goes into the sending state when it sees the turnaround character from VM/370.

There are two basic loops in the half duplex protocol terminal portion of the program. In the *Receiving State Loop*, the input buffer is continually read using the RC entry point. The output from this buffer is

formatted and displayed on the screen. The keyboard is also sampled to determine if any function keys were pressed. Thus, while it is not possible to type any alphanumeric characters when in receiving state, it is possible for example, to generate a BREAK signal on the communications line with the F1 key. The Receiving State Loop begins with a statement at line number 2480.

When a turnaround character is detected, the program changes to the *Sending State Loop*. In this loop the keyboard is repetitively sampled and the characters typed are sent out on the communications line using the SN (SEND) entry point. The Sending State Loop begins with a statement at line number 2602.

Full Duplex Terminal Protocol

In the full duplex terminal protocol, no specific sending or receiving states are set for the IBM Personal Computer. The Full Duplex Program operates in a single loop that reads and displays characters arriving on the communications line, again using an RC (RECEIVE) CALL, and also reads and transmits characters typed on the IBM Personal Computer keyboard. The Full Duplex Program begins at statement 5000.

The variable ILE in the program is used to specify whether or not local echoing of characters will take place. If local echoing is to be used, ILE is set to TRUE and any characters typed on the keyboard are immediately displayed on the screen. If ILE is FALSE, it indicates host echoing of characters and characters typed on the keyboard are not displayed except for those echoed back by the host system.

The full duplex protocol also supports the XON/XOFF protocol for controlling transmission from a remote computer. For a discussion of the operation of this protocol see the "XON/XOFF Support Present or Absent" described in Chapter 2. In the terminal program, the variable JO is used to indicate whether the XON/XOFF protocol is being used. If the protocol is in use (JO is TRUE), then the Base Program is set so that XON and XOFF characters always appear as the last character of a string returned by an RC (RECEIVE) CALL. Each line received is checked for these characters. If an XOFF is encountered, transmission of characters is suspended until an XON is received.

The program can stop transmission from the remote computer if it cannot keep up with the rate of data input. Each time through the program loop the receive buffer size is checked using the BF (BUFSIZE) CALL. If the number of characters in the buffer becomes larger than a specified amount, an XOFF is transmitted stopping transmission from the host system. The terminal program continues to read characters from the buffer. When the size of the buffer drops below another given number, an XON is transmitted and transmission from the host system begins again.

It should be noted that although the terminal and the host do not formally keep track of sending and receiving states with a full duplex protocol, the host computer in many systems is either waiting for input from the terminal or processing that input and producing output. The user at the terminal is either entering input (while the computer waits) or waiting for output from the host system. On many systems that work with a full duplex protocol, a user can type in several commands in a row (if one can work ahead of the current command) and the host system will stack up the inputs and process them in sequence.

Some systems take full advantage of the full duplex protocol by processing each character as it is typed and making immediate responses. For example, some editing systems will not wait for the user to type the full name of a command. As soon as such a system recognizes the command unambiguously from the first few characters typed, it is executed.

Operation of the Upload, Download and Compare Programs

The Upload, Download, and Compare programs use the CMS and TSO Editors for accessing files on their respective systems. The programs simulate a user sitting at a terminal accessing such an editor. Thus, for uploading, the system simulates a user calling up the editor on a new file, going into input mode, and then typing lines of input. The lines of input come from the local file that is being uploaded from the IBM Personal Computer. We talk about *lines* of input or output. For the purposes of this discussion, a *line* placed on or taken from a file is the same as *record* on that file.

Downloading

The system first requests a local filename and a host filename from the user. The file with the local filename is opened for output on the IBM Personal Computer. The program then attempts to invoke the system editor using the host filename given by the user. If this file does not exist on the host system, or cannot be accessed for some other reason, this fact is reported back to the user for the opportunity of entering another filename (or returning to the Function Selection Menu).

The process of invoking the editor is different for each system on the host. A separate program module (GOSUB subroutine) performs this function for each host system supported.

If invoking the editor starts it for an existing file, the program is now ready to begin downloading lines to the local file. At this point the program goes into a loop. A subroutine is called that obtains lines from the host system by having the editor type each successive line of the file (using the NEXT command in the VM/370 Editor, for example). Again, a separate access subroutine performs this function for each host editor supported. This subroutine returns one line at a time. If a line from the file on the host system contains more than one line end character (it might contain multiple carriage returns), it is broken up by this subroutine into multiple lines each representing a piece of text ending with a carriage return.

Carriage returns that appear in a line of a file with another carriage return or nothing in front of them are translated to a single space. Thus, a *null* line is translated into a line consisting of a single blank character. When the subroutine returns a null line, it is an indication the actual end of the file on the host system was reached.

As each line is received from the host system, it is stored as a line on the local file. Because extra carriage returns in host file generate null lines, the number of lines written on the local file may be larger than the number read from the host file. Counters keep track of the number of lines on each of these files and these numbers are displayed in error messages.

When the last record from the host file is read, the program calls another subroutine. It gives commands to the host editor to exit the editor (for example, a QUIT command for the VM/370 Editor). Again, a separate subroutine is used for each of the host systems being accessed.

Comparing

The Compare function uses the same access of the host system as does Download. The commands that the host system sees are identical for Download and Compare. The Compare function opens the local file for input. As each line is received from the host system it is compared with the corresponding line on the local file. This operation continues until a line fails to compare or the program runs out of lines on the local and/or the host system. When one system runs out of lines ahead of the other, an appropriate message is given to the user.

Uploading

The Upload program also operates in a manner quite similar to Download. The names of the local and host files are requested. The local file is checked to make sure it exists; the editor is then invoked for the host filename given. Again a special purpose subroutine matched to the host editor being used is invoked. If the host file already exists, the program refuses to upload to it. If the host file can be opened as a new file, the editor is put into input mode. Control then returns to the Upload program. The program now goes into a loop, reading lines from the local file and transmitting them one at a time to the host system.

A null line on the local file is translated into a single blank. Thus, null lines are never transmitted, so the editor is never taken out of input mode. A separate subroutine for each host system is used to transmit the line to the host and test the host reply to see if it was properly received. A null character string reply from this subroutine indicates the uploading of the line was successful.

After all of the lines are read from the local file, another subroutine, again dependent on the host system being accessed, is called to save the file that was created during input to the editor. Thus, for VM/370 this subroutine would issue a FILE command to the editor. This subroutine then checks to see if the file was successfully saved. If file saving fails, the user is given the options of closing out the file (losing it on the host system) or returning to terminal operation while still in the editor. In the latter case, the user can attempt to rectify the cause of the failure to save the file. (For example, if the file was not saved due to a lack of space on the host system, the user could erase other files to make space and then manually save the file.)

Subroutine Error Indication Protocol

In general, the error indication protocol used in calling subroutines is as follows. The subroutine called returns a string (M\$). If the string is null, it indicates the requested action of the subroutine was successfully completed. If the string is not null it contains an error message with the reason for the failure. This error message can then be displayed, or even transmitted to a remote system.

Adding Access to Other Systems

To install Upload, Download, and Compare functions using another host system (or another editor on VM/370 or TSO) it should be necessary only to write new versions of the access subroutines described above. In many cases subroutines are already available to handle functions required by these subroutines. The subroutine selected in each case is determined by an ON GOSUB statement in the program. The branch that is taken on the GOSUB is determined by parameter 14 of the

parameters stored for defined terminals. Thus, by specifying the next number in sequence for this parameter and adding the appropriate branches to the GOSUB statements, it is possible to define a terminal type that will invoke these new subroutines and thus access a different editor and/or host system.

CHAPTER 6.

MODIFYING THE BASIC PROGRAM

This section discusses how to make modifications in the BASIC portion of the Communications Program. Where necessary, the operation of the program is briefly described. While emphasis here is on how to make specific changes, it is recommended that anyone attempting such changes should be familiar with BASIC and spend some time reading and understanding the program code.

Changing Terminal Selection Default Parameters

There are two sets of DATA statements used for specifying the default parameters for the terminals presented to the user and for determining if a user will have the option of changing a particular parameter when selecting a terminal type. These statements start at statements 600 and 740 in the program. The first set of DATA statements determine whether or not a particular feature will be user selectable. One DATA statement is used for each possible terminal type. There are 20 numbers in each DATA statement representing the 20 possible parameters. (See Chapter 2 for a description of these parameters.) A number may be either 1 (indicating that a parameter is user selectable) or 0 (indicating it is not). These parameters are read into the array ID for use by the program.

The second set of DATA statements contain the numbers of the default values for each of the parameters for each of the possible terminals. (See Chapter 2 for the meaning of these numbers.) These data statements are read into array IX for use by the program.

To let a user select a particular parameter for a selected terminal type, place a 1 in that parameter position for that terminal type. For example, if you wanted to let a user select the Number of Stop Bits to be used when starting up as a VM/370 terminal, it is only necessary to change the 0 to a 1 in the third number (parameter number 3) of the DATA Statement following the title VM/370 term. The user is then automatically given that item as a choice on the appropriate menu.

Similarly, the default parameters can be changed in the second set of DATA statements. Thus, if you generally access a full duplex system that do not support the XON/XOFF protocol, you could change the sixth number in the DATA statement following the comment Full Duplex User specified from a 1 to a 2. This change would give you the default value XON/XOFF Not Supported when you selected **User Specified Full Duplex Terminal**.

Adding Additional Terminals to the Selection Menu

To add an additional terminal to the selection menu, do the following:

- Change the statement that sets ICN (at statement number 235) so its value is increased by 1 (that is, change ICN=5 to ICN=6).
- Add one DATA statement containing 20 numbers to each of the two groups of DATA statements previously described. The first of these DATA statements would contain 1's or 0's in each of the parameter positions to designate which features the user would be able to specify. The DATA statement added to the second group would contain the numbers for the default parameters for all of the pertinent parameters. See Chapter 2 for an indication of which parameters are used by a half duplex protocol terminal and which are used by a full duplex protocol terminal.
- Add a title statement to the list of statements starting at statement 1000 (following the comment display terms). These statements are of the form C\$(n)="Menu Title". Add your statement to the end of the list taking the next consecutive number (which should be equal to the new ICN number you have set). For Menu Title in the above sample statement substitute the title you want to appear on the Terminal Selection Menu. To display on a 40 character width screen, keep the length of this title to 34 characters or less.

Adding Additional Parameters

If you decide to modify the existing terminal protocols or write your own, you may let the user select parameters not currently in the parameter list. There are spare parameters (19 and 20) at the end of the parameter lists that can be used for this purpose (see Chapter 2). To add an additional parameter do the following:

- Select a location for the start of the subroutine that will handle the setting of this parameter. Statements 8000 to 8999 are reserved for such modifications.
- Modify the ON IP GOSUB ... statement at statement 1520 so that the *n*th number following the GOSUB is changed from 9000 to the starting location of your subroutine. Here *n* is the number of the parameter you have selected.
- Write the subroutine to generate the menu for your parameter. This subroutine should:
 - Set TITLE\$ to the title for your menu.
 - Set a series of C\$ array variables to the different selection options for your menu.
 - Set IMAX to the number of options you have in your menu.
 - Call the menu display subroutine.
 - Return.

A typical menu generation subroutine that you can use as a sample is given following the comment Parity Check Menu (at statement 6200).

- Add a default value to the parameter in the DATA statement (or statements) that corresponds to the terminal type (or types) that use the new parameter. Put a 1 or 0 in the DATA statements that determine if the user will be able to modify the parameter in question.
- When you come to use the parameter in your program, you will find it set in MPARM(n) where n is the number of the parameter you have selected.

Modifying the Terminal Protocols

To add some new terminal function to the Communications Program in a general way, it should be added to both of the terminal protocols. There is some common use of subroutines by the two protocols that can be used to advantage. For example, a GOSUB 2512 calls a subroutine that takes the current set of characters read from the receive buffer (in the string B\$) and formats them for output on the screen. Examine these routines closely to see whether your function can be embedded in them.

In adding new function to the terminal protocols, it is important to consider the execution time of the code. In particular, try to avoid code that individually accesses *every character* that is received from the communications line. If it is necessary to look for a particular control character, the machine language program can be set so that character always occurs as the last one in the string returned from the RECEIVE CALL. The test for the control character can then be a single statement that tests the last character of the returned string rather than a time consuming loop that looks at each character in the string.

CHAPTER 7.

USING THE BASE PROGRAM

General Overview

The Base Program is an object module that cannot be modified. It interfaces directly to the Asynchronous Communications Adapter and provides a software interface to the INS8250 Asynchronous Communications Element (ACE) chip.

The Base program uses the system interrupts. It handles the input and output of characters asynchronously to the calling program. Separate buffers are provided to coordinate program requirements with communication line constraints. The *transmit buffer* is loaded by software calls to the Base Program with character strings to be transmitted over the communication line. It is emptied, one character at a time, by interrupts fielded by the Base Program as timing permits the next character to be sent. The transmit buffer holds 255 characters.

In reverse, the *receive buffer* is filled, one character at a time, by interrupt driven software as each character is received over the line. It is emptied into a caller's character string as CALLS are made to the Base Program. The receive buffer holds 2000 characters.

The system permits you to access the control or sense bits on the INS8250 except for those associated with interrupts, reading and writing of characters, or modem control.

Loading the Base Program

The Base Program can be accessed through the use of the BASIC CALL statement. The BASIC program that accesses the base program must first load it. Loading is done by defining a segment address and then executing a BLOAD statement:

```
150 DEF SEG segaddress
```

```
160 BLOAD "RS232INT", 0
```

segaddress is the desired segment address. It should be located in a location of memory not accessed by the BASIC program. The BASIC CLEAR statement defines an upper limit of BASIC memory and the segment address should be set above that limit. Remember that the CLEAR statement sets the memory space available starting at the top of the loaded BASIC system. The DEF SEG statement specifies an absolute address in memory. This value should exceed the sum of the CLEAR statement argument and the memory used by the BASIC interpreter.

Specifying CALL Name Offsets

The offsets into the Base Program are defined as follows:

Function Name	Offset (in Hex)
SETSENSE	0
BREAK	10
BUFSIZE	20
RECEIVE	30
SEND	40

Since you define these offsets, you may use any names you wish as long as the corresponding CALLs in the program have the same names.

Example:

Assume the DEF SEG and BLOAD statements given in the previous example have just been executed. Then the following sequence of BASIC statements causes a BREAK signal to be transmitted by the Asynchronous Communications Adapter:

```
210 CHAN%=1 'port address
220 BK=&H10 'BREAK offset in segment
230 CALL BK(CHAN%,ERFLG%)
```

Entry Points

BASIC CALL statements are used to access the Base Program entry points. In using CALL statements, parameters should always be variables of the correct type and should not be arrays. A parameter ending with a % is an integer variable. One ending with a \$ is a string variable. Constants should *not* be used in CALL statements. Make sure the correct number of parameters are provided in a CALL. Failure to have the correct number, type, and order of parameters generally results in an unrecoverable error.

Note: In all of the statements given below, CHAN% is the number of the communications port. Only Port 1 is supported.

SETSENSE (CHAN%, TYPE%, DAT%, ERFLG%)

CHAN%	Is the number of Communication Port. (Always 1.)
TYPE%	Is a code that specifies the type of set or sense function to be performed.
DAT%	Is the data to be set or returned (sensed) by the routine.
ERFLG%	Is an error flag. It is set to -1 (TRUE) if no error has occurred and to 0 (FALSE) if an invalid TYPE% has been specified or if the specified CHAN% is not available. Thus, if the Asynchronous Communications Adapter is not installed in a Personal Computer, an ERFLG% of 0 is returned. A positive number for ERFLG% indicates errors as described in the following descriptions for the calls.

The following values are valid for TYPE%:

TYPE% = 1: *Read Line Status Register*

This CALL tells you what types of line errors occurred since the last time it was issued. The INS8250 Line Status Register bits are continually being changed as characters are received. The Base Program maintains a 4-bit base line register used for determining if errors occurred. Whenever a BREAK interrupt, framing error, parity error, or overrun error is detected in the line status register, the corresponding bit in the base line register is set to 1. These bits are reset only by this CALL. The CALL places these bits along with the remaining line status register bits into the low 8 order bits of DAT%. The high 8 order DAT% bits are set to zero. Then the 4-bit base line register bits are reset to zero. If any of the error bits are set, ERFLG% is set to 1.

Error bit values are:

BREAK interrupt	Hex 10
Framing error	Hex 08
Parity error	Hex 04
Overrun error	Hex 02

Instantaneous line status register bits are

Transmitter shift register empty	Hex 40
Transmitter holding register empty	Hex 20
Data ready	Hex 01

The high order bit is always zero. Refer to the *IBM Personal Computer Technical Reference Manual* for a description of the INS8250 Line Status Register bits.

For example, suppose you wish to see if any parity errors have occurred while receiving data. The following code performs that test:

```
100 CHAN%=1:TYPE%=1
110 CALL SETSENSE (CHAN%,TYPE%,DAT%,ERFLG%)
120 IF DAT% AND &H04<>0 THEN PRINT "Parity Error"
```

The following statement is *not* correct for two reasons:

```
110 CALL SETSENSE (1,1,D,E)
```

First, constants should not be used as parameters of a CALL. Second, D and E (unless specified as integer variables) default to floating point variables.

TYPE% = 2: Read Modem Status Register

This CALL tells you what changes occurred on the four RS232 interface lines since the last time it was executed. The INS8250 Modem Status Register contains four bits for the interface lines and four bits (*delta bits*) that indicate when changes occurred on these lines. These bits are continually being updated. Whenever a delta bit is set, a corresponding bit in a 4-bit base modem register is set to 1. The bits in this base modem register are retained until this CALL is made, at which time they are read and reset. These bits along with the line status bits of the modem status register are read into the low 8 order bits of DAT%. The high 8 order bits are set to zero. If any of the delta bits are set, ERFLG% is set to 2.

Received Line Signal Detect (sometimes called Data Carrier Detect)	Hex 80
-----------------------------------------------------------------------	--------

Ring Indicator	Hex 40
----------------	--------

Data Set Ready	Hex 20
----------------	--------

Clear to Send	Hex 10
---------------	--------

Four delta bits are used for changes in value of the four interface lines:

Delta Received Line Signal Detect	Hex 08
Trailing Edge Ring Indicator	Hex 04
Delta Data Set Ready	Hex 02
Delta Clear to Send	Hex 01

The presence of a delta bit indicates that the corresponding modem status bit changed at least once since the last CALL was made.

Refer to the *IBM Personal Computer Technical Reference Manual* for a description of the INS8250 Modem Status Register bits.

TYPE% = 3 : *Set Line Bit Rate (Baud rate)*

The requested bit rate (bits/second) is specified by *DAT%*. Bit rates between 2 and 56000 are supported within the accuracy limitations of the adapter's oscillator.

If an unsupported bit rate is specified, the program returns with *ERFLG%* set to 3.

TYPE% = 4 : *Set Word Length*

A non-standard number of data bits within a character may be set using this function. A number between 5 and 8 is specified in *DAT%*, denoting the number of data bits to be included with parity, start and stop bits in each character. If any other number is specified, the program returns with *ERFLG%* set to 4.

Note: Set parity Bits ($TYPE\%=5$) sets the word length to match standard ASCII convention of 8 bits framed in each character; therefore this call should *never* be made *before* setting parity.

TYPE% = 5 : Set parity Bits

A number between 0 and 4 is specified in $DAT\%$. The parity bits for the specified channel are set as follows:

0 - No Parity Checking

1 - Odd Parity

2 - Even Parity

3 - Mark

4 - Space

The character word length is also set to the ASCII standard connection of a total of 8 bits framed per character. For non-standard word lengths see the note for $TYPE\%=4$.

If any other number is specified, the program returns with $ERFLG\%$ set to 5.

TYPE% = 6 : Set Stop Bits

A number, either 1 or 2, is specified in $DAT\%$. This number indicates the number of stop bits to be used for the specified channel. If any other number is specified, the program returns with $ERFLG\%$ set to 6.

TYPE% = 7 : Initialize, Start, and Stop

DAT% = 0 initializes the Base Program, clears the transmit and receive buffers of any pending characters, and clears any options previously set by SETSENSE calls of TYPE% = 9. This CALL should be the first CALL to the Base Program and should be made only once.

DAT% = 1 starts up the communications interface, brings up the DTR and RTS bits, and initializes the INS8250 ACE chip.

If any other number is specified, ERFLG% is set to 7 on return.

TYPE% = 9 : Set Receive Data Modifiers

The string ST\$ returned by the RECEIVE routine can be affected by ASCII control codes (X'00' through X'1F' and X'7F') in the incoming data.

Modifiers are specifically set (and reset) by identifying the control code of interest in the lower 8 bits of DAT%. Modifier bits affecting the rules are identified in the binary representation of the higher 8 bits, with the ability to combine rules by summing modifiers.

DAT% = (0 + 'ctl. char') resets all modifiers for the specified control character.

DAT% = (256 + 'ctl. char') causes all occurrences of the control character to be deleted from the ST\$ returned by the RECEIVE routine.

DAT% = (512 + 'ctl. char') causes a RECEIVE ST\$ to terminate with the control character, if it appears before the end of currently available data. The next sequential ST\$ starts with the character immediately following the control character.

If any invalid combination of bits is specified, the program returns with ERFLG% set to 9.

BREAK (CHAN%,ERFLG%)

A CALL to BREAK sends a BREAK signal on the specified channel. Because a BREAK signal continues for some time after being started, a CALL to this routine while a BREAK is in progress restarts the time period needed to produce a BREAK signal. ERFLG% returns a -1 unless the CHAN% specified is unavailable.

BUFSIZE (CHAN%,TYPE%,DAT%,ERFLG%)

A CALL to BUFSIZE returns in DAT% the number of characters currently in the specified buffer. TYPE% = 1 specified the transmit buffer. TYPE% = 2 specifies the receive buffer. The number of characters in the receive buffer may not equal the logical number of characters transferred by RECEIVE if options such as character delete are in effect.

If no error has occurred, ERFLG% is set to -1. It is set to 0 if an invalid TYPE% has been specified. It is set to 10 if the specified buffer is close to overflow (more than 192 characters for transmit or more than 1500 characters for receive). It is set to 11 if the specified buffer has actually overflowed.

RECEIVE (CHAN%,LN%,ST\$,ERFLG%)

Characters in the receive buffer for the specified channel are moved to the string ST\$. The number of characters placed in the string is determined by the lesser of a) the defined lengths of the string, or b) the number of characters in the receive buffer before a

terminating control character or end of buffer. The length of ST\$ is not altered, but the number of characters set into ST\$ is placed in LN%. To permit maximum data availability, the string should be pre-defined to maximum length (255) within the BASIC program. The simplest way of predefining such a string is:

```
ST$=STRING$(255," ")
```

ERFLG% is set to -1 if no errors have occurred. ERFLG% is set to 12 if a receive buffer overflow occurred or to 13 if a parity or framing error occurred on a received character, or for an overrun or BREAK condition.

If there are no characters in the receive buffer the program returns with LN% set to 0. See the section "Effect of Control Signals on Data Transmission and Reception" later in this chapter for how to prevent placing data in the receive buffer.

SEND (CHAN%,ST\$,ERFLG%)

The characters in ST\$ are appended to the transmit buffer and transmission of characters is initiated, if it is not already in progress. If no errors occur, ERFLG% is set to -1. If insufficient space is available for ST\$ in the transmit buffer, no characters are placed in the buffer and ERFLG% is set to 14. A null ST\$ (zero length) results in the setting of ERFLG%, but does not place data in the transmit buffer.

See the next section for a description of how characters can be held in the transmit buffer.

Effect of Control Signals on Data Transmission and Reception

For characters to be transmitted on the communications line by the Base Program, the Clear to Send, the Data Carrier Detect, and the Data Set Ready signals must all be ON. If any of these signals is OFF, characters passed to the Base Program by a CALL SEND are stacked in the transmit buffer until all of these signals are ON.

For characters to be received by the Base Program and placed in the input buffer, the Data Carrier Detect and the Data Set Ready signals must both be ON. If either of these signals is OFF, received characters are ignored and lost.

See the section “I/O Signals Used by the Communications Program” in Chapter 4 for more details about these signals.

Operation of Base Program

Initialization occurs in the Base Program when a SETSENSE TYPE% = 7 command is given with DAT% = 0. This CALL causes the communications interrupt vectors to be set and sets the default parameters into the INS8250 ACE chip. These default parameters are compatible with VM/370 logon and are as follows:

Bit Rate = 300

Parity = mark

Number of bits in a character = 7 + parity

Number of stop bits = 1

The user may then change these default parameters by calling SETSENSE with the appropriate TYPE% and DAT% values.

To start the communications interface, the user issues a SETSENSE TYPE% = 7 command with DAT% = 1.

APPENDIX A. MESSAGES

Abnormal Terminate by Receiver

Transmission of a file was terminated at the request of the receiving IBM Personal Computer.

Abnormal Terminate of Upload

The uploading of the file was either terminated before completion of transmission of the local file or did not complete exiting from the host editor correctly. Other messages should give more details concerning the problem.

All of File NOT Sent

Transmission of a file from an IBM Personal Computer was stopped by the transmitting IBM Personal Computer before the entire file was transmitted.

Answer either Y or N

At various places in the program you are given requests for Yes or No answers. These requests may be answered with either Y, YES, N, or NO. Any combination of upper and/or lower case letters may be used. Thus you could type Yes and the program would see it as a YES answer.

A Positive Number Should be Entered

In replying to the menu displayed, you must make specific selection by entering the number of one of the menu items.

Base Program Initialize Failed

The initialization of the Base Program has failed. It is likely that no Asynchronous Communications Adapter is installed in your IBM Personal Computer or an adapter set to be Channel 2 was installed.

BASIC Error Accessing RS232INT.BAS

An error occurred in trying to load the Base Program required for operation of the Communications Program. If this error occurs in searching the A diskette, then the B diskette will be tried. If it occurs on the B diskette, the program halts.

Break Sent

When you are operating as a terminal, pressing the F1 key will send a BREAK signal to the host computer. When this signal is sent, the message "Break Sent" is displayed.

Break Signal Received

This message indicates that a communication line BREAK was received from the host computer (or another IBM Personal Computer). Such a signal might be used by the host computer prior to sending a warning message to system users. The Communications Program takes no action when a BREAK signal is received other than to display this message. This signal often occurs when a modem is connected or disconnected from the common carrier line. In those situations it can be ignored.

Buffer Close to Overflow

The receive buffer is getting close to the point where it will become filled with data from the communications line and overflow. This problem indicates the Communications Program cannot keep up with the rate of data input. It is occasionally seen when running at 2400 bits/second and listing many short lines on the screen. (Typically, a LIST** command when running at 2400 bits/second in VM/370 can cause it to occur.)

You can avoid loss of data in this situation by pressing the F1 key, the Communications Attention key and waiting for the data in the buffer to display. You may have to wait for several seconds while the data in the buffer empties and you see the acknowledgement from the host computer that your break was received.

You may then press the Enter key and continue your terminal listing.

Buffer has Overflowed

The receive buffer was filled with characters from the communication line and overflowed. Some characters have probably been lost. When such an overflow occurs, characters that will not fit in the buffer are discarded. This message should normally be seen only when operating as a terminal at a high line bit rate (2400 bits/second or higher). Loss of data due to overflow can be avoided by stopping transmission of data when the buffer is close to overflow (see "Buffer Close to Overflow" message above).

If output from the host computer is stopped (with a Communications Attention), there will be significant delay before you see the message indicating that the

attention signal has been received by the host computer. The delay is due to the fact that the receive buffer contains 2000 characters when it overflows and these characters will all be subsequently displayed on the screen.

Check Computer or Modem Connection

This message is a reminder that connection to a computer (using local cable) or modem (for remote computer access) must be made before you can start operation as a terminal. If your connection is established at this time, you can ignore this message.

Comparison Ended Abnormally

The Compare function ended before all of the lines in the host system file were compared with the local file. The reason for this is given in an associated message.

Comparison Failed. Lines Still Left in Local File

During comparison the end of the host file was reached while there were still lines left in the local file. All of the lines compared up to that point compared successfully.

Comparison Failed. Local File ended. Lines still on Host File

During comparison the end of the local file was reached while there were still lines left to be compared on the host file. All of the lines compared up to that point compared successfully.

Comparison Failed at Local Line ## Host Line ##

The specified lines failed to compare. The lines in questions are listed following this message. Comparison is halted at this point.

Computer Connection NOT Established

In starting up for operation as a terminal, the Communications Program did not find the Data Set Ready line on. This signal indicates the absence of a connection to a host computer. See Chapter 4 for a discussion of problems in connecting to a host system.

Could not access Host Data Set. Message:

In attempting to upload a file using TSO, the TSO editor could not be properly invoked for input of data. The message that follows was generated by TSO.

Do not run using BASICA. Load BASIC.

You attempted to run the Communications Program after loading Advanced BASIC (BASICA). Due to memory limitation in a 64K IBM Personal Computer, the program only runs with Disk BASIC. Return to DOS, load Disk BASIC, and restart the program.

Downloading Ended Abnormally

Downloading of a file terminated before all of the lines in the host system file were moved to the local file. The reason for this is given in an associated message.

Do You Wish to Preserve Uploaded Data?

See the section "Uploading a File" in Chapter 3 for action to be taken in this situation.

Do You Wish to Try Another File Name?

The file you attempted to access could not be accessed for reasons given in previous messages. If you answer **Yes** to the request, you may enter another filename. Answering **No** returns you to the Terminal Selection Menu.

Error in Host Access. Message was:

The VM/370 EDIT editor was not properly invoked by the Upload, Download, or Compare functions. The message that follows explains the reason for this failure. If the message that follows contains a large number of cryptic lines, check to make sure that you are not invoking the XEDIT editor. See the section "Transferring Files with VM/370" in Chapter 3 for more details.

Error Writing Host File

During uploading of a file, the host system sent a return message after transmission of a line. The message sent by the host should follow.

Error Writing Local File. BASIC Error ##

An error has occurred in attempting to write a line on a local file. See Appendix A of the *IBM Personal Computer BASIC* manual for the meaning of the number given.

File Access Failed. BASIC Error ##

You failed to open a file for either input or output. The reason for this failure may be determined by looking up the given BASIC error number in Appendix A of the *IBM Personal Computer BASIC* manual.

File filename not Present

The file you have attempted to read does not exist on the diskette on the default (or specified) drive. Check to make sure you are accessing the correct diskette drive.

File Reading Error. BASIC Error ##

An error has occurred in attempting to read a line from a local diskette file. The reason for this failure is given under the number (##) in Appendix A of the *IBM Personal Computer BASIC* manual.

File Writing Error. BASIC Error ## File Probably Not Written.

An error has occurred when writing a local diskette file. The complete file was probably not written. The number of the error (##) can be checked in Appendix A of the *IBM Personal Computer BASIC* manual.

Host Data Set Already Exists

You have attempted to upload to an existing TSO data set. Either specify another data set name or return to terminal mode and erase the data set before attempting uploading.

Host File Already Exists

You have attempted to upload to a VM/370 file that already exists on the host system. Either specify another filename or return to terminal mode and erase the file before attempting uploading.

Host File not Present

In attempting a download using VM/370, no file was found to download.

Illegal Option Reached

This message should not appear unless changes have been made in the terminal selection portion of the BASIC program. You should check with persons who might have modified the program for the changes they have made.

INVALID FILEMODE LRECL

In a file transfer to VM/370 you may have failed to specify a VM/370 filetype.

Length=255. May not be ASCII File

You have probably attempted to transfer a non ASCII BASIC file or some other non-text file. If transferring a BASIC program, make sure it is saved using the ASCII option.

Line Connected

The connection to a modem or host computer was established. See Chapter 4 for more details.

Line Disconnected

The connection to a modem or host computer was broken. See Chapter 4 for more details.

Line Disconnected/Reconnected

The connection to a modem or host computer was momentarily broken and then reestablished. See Chapter 4 for more details.

Line Too Long to Upload

A line of length greater than 130 characters has been encountered during uploading of a file.

Messages Disabled

The F4 key is used to enable or disable certain error messages. This message appears on the bottom line of the screen when you disable (turn off) these messages. See the section “Using Function Keys When Running as a Terminal” in Chapter 1 for a more detailed discussion of the use of this key.

Messages Enabled

The F4 key is used to enable or disable certain error messages. This message appears on the bottom line of the screen when you enable (turn on) these messages. See the section “Using Function Keys When Running as a Terminal” in Chapter 1 for a more detailed discussion of the use of this key.

Name must be 1 to 8 Characters in Length

You have specified a filename that is too long or too short. See the *IBM Personal Computer DOS* manual for a discussion of filenames.

No Blanks Permitted in Name

For the Upload, Download or Compare functions with a TSO host system, no blanks can be specified as part of a file name. Filetypes should be specified using the dot (.) notation. Thus, a file with the name TRUTH and a filetype TEXT should be specified as:

TRUTH.TEXT

Not Used between Personal Computers

You attempted to access the Upload, Download, or Compare functions when operating as a terminal that is communicating with another IBM Personal Computer. These functions are used only with a host system such as VM/370 or TSO. See the section "Transferring Files between IBM Personal Computers" in Chapter 3 for further explanation.

Null File Found for filename

The file on your IBM Personal Computer that you are attempting to Upload or Compare is empty.

Number out of Range of Menu

Reply to this menu with one of the numbers given for the menu items. If no default value (** to left of menu item) is present, you must give a number.

Possible Problem on Host. Message was:

In ending an upload with VM/370, a proper exit from the editor to CMS did not occur.

READY not Found

In ending a file transfer with the TSO host system, a problem occurred in exiting from the TSO Editor. Return to terminal mode and make sure you have exited from the Editor.

Ready to Receive File

The receiving IBM Personal Computer is ready to receive a file. This message is transmitted roughly every 15 seconds until the transmitting IBM Personal computer sends the message "Starting File Transmission".

Ready to Start Sending File filename

The transmitting IBM Personal Computer is ready to send the specified file. This message should appear on the screens of both the transmitting and receiving IBM Personal Computers.

Received Char. Framing Error

The Asynchronous Communications Adapter has detected a framing error in a received character. Such an error indicates that the transitions between voltages that designate the separate bits in a character are not occurring at the proper times. If this error occurs repetitively, it indicates you may be operating at the wrong line bit rate. This error will sometimes occur when gibberish characters are received when a modem is connected or disconnected from the common carrier line. In this case it can be ignored.

Received Char. Overrun Error

A character was received by the Asynchronous Communications Adapter before the previous character was processed by the Communications Program. This error indicates that at least one character of data was lost. If you find situations where this error occurs repeatedly, record the details of that situation and report it as a possible program bug.

Received Char. Parity Error

A parity error has been detected on a character received from the communications line. If parity errors occur frequently during receipt of data, you may have selected the wrong parity checking option. See the section "Enabling and Disabling Receive Errors" in Chapter 4 for further details. If these errors occur together with the appearance of gibberish characters, it is likely that you are operating at the wrong line bit rate.

When a modem is connected to or disconnected from a common carrier line, garbage characters are often generated. Such characters can cause parity errors. An occasional parity error during operation as a terminal or during upload or download indicates that errors in transmission may be occurring. Check the quality of the communication line.

Redo from Start

You have probably entered alphabetic characters not recognized as numbers by the BASIC input routine. Type the requested number.

Return to EDIT Failed

In returning from TSO INPUT mode during termination of Upload, EDIT mode was not reached successfully. Return to Terminal Mode and check on the status of the file. Make sure you are out of EDIT before attempting another file transfer.

RS232INT.BAS Load Failed

The Communications Program has been unable to load the Base Program required for operation. Both Drives A and B were accessed for this program. If the previous messages indicate the program has not been found, mount a diskette with a copy of the RS232INT.BAS file. If other errors have occurred on accessing a diskette, check to make sure the diskettes are properly installed in the drives and restart the program. If an error occurred in reading the file, it may be necessary to use a different copy of the file.

RS232INT.BAS Not on Diskette B (or A)

The Base Program required for operation of the Communications Program was not found on the specified diskette. If the program is not found on Drive A, Drive B is searched for it. If the program is not found on Drive B, the program halts.

Saved Failed. Message:

On completing uploading to a TSO host system, the file could not be successfully saved with the TSO editor.

Single CR missing at beginning

An error occurred in accessing the VM/370 editor during a file transfer. Return to terminal operation mode and make sure that you are in CMS before attempting the file transfer.

Specified Channel Not Present

Communications Channel 1 is not present in the IBM Personal Computer. Either the Asynchronous Communication Adapter is missing from the machine or a card specifying a channel other than 1 has been installed.

Specify Either 40 or 80

In setting the screen width at program startup you have specified a width not equal to either 40 or 80. These are the only screen widths supported. A width of 80 should be specified unless your display will not legibly display 80 characters across the screen.

Starting File Transmission

The transmitting IBM Personal Computer is about to begin transmitting file lines. This message should appear on both the transmitting and receiving IBM Personal Computers.

Terminal Initialize Error

An error has occurred in the setup for operation as a terminal. This message should not occur unless your Communications Program has been modified. If you can reproduce this error on an unmodified version of the Communications Program, report this problem together with the exact terminal parameters that were set when it occurred, as a program bug.

Too Many Files

In opening a file for output you have exceeded the maximum number of filenames permitted in a diskette directory. The maximum number permitted is 64. You should erase one or more existing files or use a different diskette where filename space is available.

Transmission Completed

Transmission of a file between IBM Personal Computers completed with all file lines transmitted successfully.

Upload did NOT Complete Correctly

All of the local file has probably been transmitted correctly to the host system. However, the host system failed to properly exit the editor being used. Return to terminal mode and make sure the file has been properly saved.

What screen width (40 or 80 characters) do you wish?

You should reply to this message with either 40 or 80. This number will specify the number of characters per line that will be displayed on your screen. You should give a width of 80 unless your display will not display 80 characters per line legibly.

You Are Starting up as a Terminal

The program that communicates with the communications line has been initialized and you are ready to start operation as a terminal. See the section "Starting Up as a Terminal" in Chapter 2 for more details.

INDEX

A

- abnormal termination of file transfer 3-6, 3-11, 3-13, 3-22, 3-27
- acoustic coupled modem 2-30, 2-31, 2-48, 4-3, 4-6
- adding more file transfers 5-10
- adding terminal to menu 6-4
- Alt Ctrl Del 1-14
- ANSWER modem switch 2-30, 2-52
- asynchronous
 - communications adapter 1-6, 2-28, 2-29, 2-30, 4-4, 7-1
- attention key see F1 Key
- AUTOREAD in VM/370 2-35

B

- backing up diskettes 1-16, 1-19
- backspace
 - character (TSO) 2-44, 2-49
 - character (VM/370) 2-36, 2-42, 3-5
 - key 1-10
- base modem register 7-6

- base program 1-6, 2-2, 5-2, 7-1
 - data modifiers 7-9
 - entry 7-4
 - initialize 7-9
 - loading 7-2
 - operation 7-13
 - start 7-9
 - stop 7-9
 - use 7-1

BASIC

- base program CALL 7-2
- BLOAD 7-2
- CLEAR 7-2
- CONT 2-33
- DEF SEG 7-2
- errors 4-10
- file transfers 3-15
- files 3-3, 3-16, 3-29
- program changes 6-1
- program download 3-15
- program upload 3-15
- returning to 2-32
- RUN command 2-33
- SYSTEM command 2-34
- baud rate 2-9, 7-7
- begin transmission code 3-24
- bell character 1-11, 2-14, 2-23
- bit length of characters 2-10
- bit rate see line bit rate
- bit rates supported 2-9, 2-21

BLIP character 3-5
BLOAD 7-2
BREAK call 5-1, 7-3, 7-10
break interrupt 7-5, 7-11
break signal 1-11, 2-35, 2-41,
5-4, 7-3, 7-10
buffer close to overflow 7-10
buffer overflow 7-10, 7-11
buffer sizes 7-1, 7-10
BUFSIZE 5-1, 5-6, 7-3, 7-10

C

cable connections 2-28,
4-4
troubleshooting 4-2
CALLs to base program
7-2, 7-4
caps lock key 1-9
carriage return character
as line end 2-15, 3-23
as turnaround 2-12,
2-22, 5-3
deleting 2-14, 2-23
in downloading 3-4, 5-8
in file transfers 3-23
carriage return key 1-9
CHAN% 5-1, 7-4
changing parameters
personal computer
communication 2-53
terminal selection 6-1
character
bit length 2-10, 7-7
deletion 1-13, 2-14, 7-9
echoing 2-19, 2-23, 2-27
loss in downloading 3-6
width of screen 2-2
character delete character
in VM/370 2-36, 2-42,
3-5
in TSO 2-44, 2-49
checking parity see parity
checking
choosing terminal type 2-5
CLEAR 7-2
clear to send 4-5, 7-6, 7-12
CMS 2-35
BLIP character 3-5
editor 3-5, 3-6
common carrier 2-29, 2-30
communicating
between IBM personal
computers 2-6, 2-51
communications attention
1-11, 3-11, 3-13, 3-15, 3-20,
3-22
compare 3-3, 3-14
comparing files
error messages 4-11
errors in 3-15
filenames 3-14
how to operate 3-14
program internals 5-9
termination 3-15
connecting to host 2-28
CONT command 2-33
control bits 7-6
control characters
deletion 2-14, 2-23,
2-27, 7-9
on keyboard 1-11
control signals 4-4, 4-6,
7-6, 7-12
copying diskettes 1-16, 1-19
Ctrl Break 1-14, 3-27
Ctrl key 1-11
CTS 4-5, 7-6, 7-12

D

DAT% 5-1, 7-4, 7-6, 7-13
data carrier detect 4-5, 7-6, 7-13
data set ready 2-30, 4-1, 4-2, 4-5, 5-2, 7-6, 7-12
DATA statements 6-2, 6-4
data terminal ready 4-4, 7-9
date entry 2-2
DCD 4-5, 7-6, 7-13
DEF SEG 7-2
default drive 2-3
 comparing 3-14
 downloading 3-12
 file transfers 3-19, 3-21
 uploading 3-10
default parameters
 terminals 2-25
 base program 7-13
deleting characters 1-13, 2-14, 7-9
delta
 carrier detect 7-7
 clear to send 7-7
 data set ready 7-7
diagram of program 1-3, 1-4
direct cable connection 2-28
direct-coupled modem 2-30, 2-32
disabling
 error checking 1-12, 4-9
 hex listing 1-13
 receive errors 1-12, 4-9
disconnect in file transfer 3-20, 3-22, 3-27
DISKCOPY command 1-16, 1-19

diskettes
 copying 1-16, 1-19
 drive selection 3-10, 3-12, 3-14, 3-19, 3-21
displaying messages 1-12, 4-8
DOS programs 1-15, 2-1
download 3-3, 3-12
downloading
 BASIC programs 3-15
 broken lines 3-17
 disconnect recovery 3-28
 error messages 4-11
 how to operate 3-12
 loss of characters 3-6
 maximum line length 3-5
 messages in files 3-6
 modification of files 3-4
 program internals 5-7
 termination 3-13
 TSO files 3-9
DSR 2-30, 4-1, 4-2, 4-5, 5-2, 7-6, 7-12
DTR 4-4, 7-9
dynamic messages 4-8

E

echoing of characters 2-19, 2-23, 2-27
EDIT EXEC 3-7
editors
 CMS 3-5, 3-6, 3-7, 4-11, 5-7, 5-10
 on host 3-27
 TSO 3-4, 4-11, 5-7, 5-10
 VM/370 see CMS
enabling
 error checking 1-12, 4-9
 hex listing 1-13
 receive errors 1-12, 4-9

- ENTER key 1-9, 2-12, 5-3
- entry points to base program
 - 7-3
- EOT 2-12, 2-22
- ERFLG% 5-1, 7-4, 7-5 to 7-11
- error indication protocol 5-10
- error messages
 - dynamic 4-8
 - how displayed 4-8
 - list of A-1
 - during file access 4-11
 - static 4-10
- error recovery 1-14
- escape character (VM/370) 3-5
- escape control character 2-14, 2-23
- escape from terminal operation 1-12, 2-32
- even parity 2-11, 2-21, 7-8
- examples
 - TSO logon 2-44
 - VM/370 logon 2-37
- execution time 6-7
- exiting terminal operation 2-32
- extending the program 1-2, 6-1

F

- feature selection menu 2-8
- files
 - accessing errors 4-11
 - lines 3-4
 - modification in transfer 3-4
 - records 3-4

- recovery after terminate
 - 3-6, 3-27
 - you can transfer 3-1, 3-3, 3-9
- file transfers
 - disconnect recovery 3-27
 - general rules 3-3
 - installing for other systems 5-10
 - personal computer 3-18
 - restrictions 3-3, 3-9
 - TSO 3-7
 - VM/370 3-5
- filenames
 - compare 3-14
 - download 3-9, 3-12, 5-7
 - local 3-10, 3-12, 3-14, 3-19, 3-21
 - personal computer file transfers 3-18, 3-23
 - TSO 3-9, 3-10
 - upload 3-9, 3-10
 - VM/370 3-10
- first character deleted 1-13, 2-14, 2-23, 2-27
- fixed length files 3-3
- floating lines 2-31, 4-6
- force to sending state 1-13
- formatting diskettes 1-17, 1-19
- framing error 4-9, 7-5, 7-11
- full duplex terminal 1-9
 - protocol 2-4, 2-6, 2-13, 2-22, 2-26, 2-43
 - operation 2-49
 - program internals 5-5
 - specifying 2-17
- FULL DUPLEX modem
 - switch 2-30, 2-48, 2-51, 4-3

function keys 1-7, 1-11
function selection menu
 1-4, 1-12, 2-32, 3-2, 3-9,
 3-12, 3-14
F1 Key 1-3, 1-4, 1-11, 2-35,
 2-41, 3-11, 3-13, 3-15, 3-20,
 3-22, 3-27, 5-4
F2 key 1-3, 1-4, 1-12, 2-32,
 3-2, 3-9, 3-12, 3-14, 3-18
F3 key 1-3, 1-4, 1-12, 4-9
F4 key 1-12, 4-9
F5 key 1-13
F6 key 1-13

G

getting started 2-1
go to
 BASIC 2-32
 function selection 1-12,
 2-32, 3-2

H

HALF DUPLEX modem
 switch 2-30, 2-51, 4-3
half duplex terminal 1-9
 protocol 2-4, 2-6, 2-7,
 2-22, 2-26
 program internals 5-3
 specifying 2-7
hex listing 1-13, 2-50
host character echoing 2-19,
 2-23, 2-27, 5-5
host computer 1-2
host computer connection
 2-28
host filenames see filenames

I

I/O signals 4-4
IMPEX setting 3-5
initialize base program 7-9
initialize a diskette 1-15
installing other file transfers
 5-10
INS8250 7-1
 control bits 7-1, 7-12
 error bits 7-5
 line status register 7-5
 modem status register 7-6
 sense bits 7-6
internals of program 5-1
interrupting
 compare 3-15
 download 3-13
 execution 1-14
 file transfers 3-20, 3-22
 host program 1-11, 2-35
 upload 3-11

K

keyboard layout 1-8
keyboard use 1-7

L

line bit rate 2-8, 2-21, 2-26,
 2-34, 2-38, 2-42, 2-46, 2-49,
 2-51, 2-53, 7-7, 7-13
line break see break signal
LINE CONNECTED
 message 4-2
line delete character 3-5

LINE DISCONNECTED

- message 4-2
- line end character
 - from host 2-15, 2-24, 2-27, 2-49
 - VM/370 3-5
- line length
 - downloaded files 3-5
 - on screen 2-2
 - uploaded files 3-3
- line status register 7-5
- line turnaround character
 - sent by host 2-13, 2-22, 2-26, 5-3
 - sent to host 2-12, 2-22, 2-26, 2-34, 2-40, 2-42, 2-47, 5-3
- linefeed 2-12, 2-14, 2-15, 2-22, 2-23, 2-50
- lines in a file 3-4
- LINESIZE setting
 - TSO 2-44, 3-8
 - VM/370 2-37, 3-6
- listing in hex 1-13
- loading
 - base program 7-2
 - communications program 2-2
- local character echoing 2-19, 2-23, 2-27, 5-5
- local filename 3-10, 3-12, 3-14, 5-7
- logging onto TSO 2-42
- logging onto VM/370 2-34, 2-37
- LOGOFF HOLD 2-36

M

- mark parity 2-11, 2-21, 7-8

- maximum line length
 - downloading 3-5
 - uploading 3-3
- menu
 - half duplex terminal 2-7
 - terminal selection 2-5
 - use in terminal selection 2-4
- message display 4-8
- messages
 - bottom line 4-8
 - in compare 4-11
 - in download 3-6, 3-17, 4-11
 - in function selection 4-10
 - in upload 4-11
 - stacking 4-8
- modem connections 2-29
 - troubleshooting 4-3
- modem status register 7-6
- modems 2-29, 2-46, 2-51
- modifying BASIC program 6-1
- modifying terminal protocols 6-7

N

- no character specified 2-23
- no parity bit 2-10, 2-21
- no parity checking 2-10, 2-21, 7-8
- NOINTERCOM 3-8
- NOMODE 3-8
- NOMSGID 3-8
- non-IBM personal computer
 - receiving files 3-25
 - transmitting files 3-24
- NOPAUSE 3-8
- NOPROMPT 3-8

NOWTPMSG 3-8
null line 3-4, 5-8, 5-9
Num Lock key 1-13
number of stop bits 2-11,
2-21, 2-26, 7-8
numeric keypad 1-7, 1-13

O

odd parity 2-11, 2-21, 7-8
operation
 as full duplex terminal
 2-49
 of base routine 7-13
 of file transfers 5-7
ORIGINATE modem switch
 2-30, 2-52
overrun error 4-9, 7-5, 7-11

P

parameters for VM/370
 2-27, 2-34
parameters for TSO 2-27,
 2-42
parameters for terminals
 2-21
parity bits 2-10, 7-8, 7-13
parity checking 2-10, 2-21,
 2-26, 2-34, 2-39, 2-42, 2-47,
 2-49, 7-8
parity errors 4-9, 7-5, 7-11
PCMOD 1-6, 2-53
personal computer
 communication 2-6, 2-24,
 2-26, 2-51
 changing parameters 2-53
personal computer file
 transfer 3-18
 disconnect recovery 3-28
 protocol 3-23

print screen key 1-10
problem troubleshooting
 4-1
PROFILE TSO Command
 2-44, 2-49, 3-7
PROFILE EXEC 2-36
program execution time 6-7
program diskette 1-6, 1-15,
 2-1, 2-2
protocols
 file transfers 3-23
 terminal 2-4
PrtSc key 1-10

R

read line status 7-5
READY in TSO
READY light 2-31, 4-1
recalling terminal description
 2-7
RECEIVE CALL 5-1, 5-5,
 7-3, 7-10
receive buffer 7-1, 7-10
 overflow 1-10, 2-18, 7-10,
 7-11
 size 7-1
receive data line 4-4
receive data modifiers 7-9
receive errors
 checking 1-12, 4-9
 enable/disable 1-12, 4-9
received line signal detect
 4-5, 7-6, 7-7
receiving a personal computer
 file 3-21, 3-25
RECEIVING message 2-34
receiving state 2-12, 5-3, 5-4
records in a file 3-4

- recovering files
 - after download disconnect 3-28
 - after upload disconnect 3-6, 3-27
 - after file transfer disconnect 3-28
- request to send 4-4, 7-9
- required hardware and software 1-6
- returning to
 - function selection 2-33
 - terminal selection 2-16
 - BASIC 2-32
 - DOS 2-34
 - terminal operation 2-32
- ring indicator 4-5, 7-7
- RS232-C standard 2-29
- RS232INT 1-6, 2-2, 7-2
- RTS 4-4, 7-9
- rubout character 2-14
- RUN command 2-33, 3-27

S

- saving terminal descriptions 2-7, 2-16
- screen width 2-2
- second character deleted 1-13, 2-14, 2-23, 2-27
- segment address 7-2
- selecting a terminal 2-4
- SEND 5-1, 5-4, 7-3, 7-11
- SENDING message 2-35
- sending state 1-13, 2-12, 5-3
- sense bits 7-1, 7-6
- SET AUTOREAD 2-35
- SET BLIP 3-5, 3-7
- SET IMPEX 3-5
- set line bit rate 7-7

- SET MSG 3-6, 3-7
- set parity bits 7-8
- set receive data modifiers 7-9
- set screen width 2-2
- set word length 7-7
- SET WNG 3-6, 3-7
- SETSENSE 5-1, 5-2, 7-3, 7-4, 7-13
- setting terminal linesize 2-37, 2-42, 2-44, 2-49, 3-6, 3-17
- shift key 1-9, 1-11
- single drive system 1-15
- space parity 2-11, 2-21, 7-8
- special key combinations 1-14
- special symbols 1-9
- ST\$ 7-10, 7-11
- start base program 7-9
- starting up
 - communications program 2-2
 - for the first time 1-15
 - selected terminal 2-15, 2-28, 5-2
- static messages 4-10
- stop base program 7-9
- stop bits 2-11, 2-21, 2-26, 2-49
- storing terminal description 2-7, 2-16
- switched lines 2-29
- SYSTEM command 2-34
- system reset 1-14

T

- terminal
 - attention 2-35, 2-43
 - default parameters 2-25

- terminal (cont.)
 - linesize 2-37, 2-42, 2-44, 2-49, 3-6, 3-17
 - modifying protocols 6-7
 - operation phase 1-2, 1-4, 2-28, 5-1
 - parameter addition 6-5
 - parameter selection 2-7
 - parameter summary 2-21, 2-25
 - protocols 2-4, 2-26
 - specification files 1-4, 2-7, 2-16
 - startup 2-15, 2-18
 - types 2-5
- TERMINAL command
 - TSO 2-44, 2-49, 3-8
 - VM/370 2-36, 2-37, 2-42, 3-6, 3-7
- terminal descriptions
 - recalling from file 2-7
 - storing on a file 2-7, 2-16
- terminal selection 2-4
 - changing parameters 6-2
 - menu 2-5
 - menu additions 6-4
 - phase 1-2, 1-4, 2-4
- TERMINAL.BAS 1-6
- terminate transmission code 3-24
- text files 3-1
- TEXT filetype - TSO 3-9
- third character deleted 1-13, 2-23, 2-27
- trailing edge ring indicator 4-5, 7-7
- transferring files 3-1
 - between personal computers 3-18
 - with TSO 3-7
 - with VM/370 3-5
- transmit buffer 4-5, 7-1, 7-10
 - size 7-1
- transmit data line 4-4
- transmitter register 7-5
- transmitting files
 - from IBM personal computer 3-19
 - from non-IBM personal computer 3-24
- troubleshooting 4-1
 - cable connected system 4-2
 - host connection problems 4-1
 - modem connected system 4-3
- TSO
 - accessing 2-42
 - additional parameters 2-43
 - default parameters 2-26
 - downloading files 3-3
 - editor 3-4, 4-11, 5-7, 5-10
 - example 2-44
 - file transfer restrictions 3-9
 - file transfers 3-7
 - filenames 3-10
 - logon 2-44, 2-48
- PROFILE Command 2-44, 2-49, 3-7
- READY 2-48
- restrictions 3-9, 3-10
- starting up terminal 2-28
- terminal 2-6, 2-24, 2-42
- TERMINAL Command 2-44, 2-49, 3-8
- terminal parameters 2-26
- uploading files 3-10
 - using 2-42
- TSOMOD 1-6, 2-43

- turnaround character
 - sent by host 2-13, 2-22, 2-26, 2-49, 5-3
 - sent to host 2-12, 2-22, 2-26, 2-42, 2-47, 2-49, 5-3
- two diskette drive system 1-18, 2-2
- two personal computers 2-24
- type of terminal 2-24
- TYPE% 5-1, 7-4 to 7-9
- typewriter keyboard 1-9

U

- UPDATE 1-6, 1-16, 1-19
- uploading
 - BASIC programs 3-15
 - disconnect recovery 3-27
 - how to operate 3-9
 - error messages 4-11
 - maximum line length 3-3
 - program internals 5-9
 - termination 3-11
 - TSO files 3-10
- uppercase 1-9
- user specified
 - full duplex terminal 2-6, 2-17, 2-26, 2-43, 2-49, 2-59, 6-2
 - half duplex terminal 2-6, 2-26, 2-34
- using the base program 7-1
- using TSO 2-42
- using VM/370 2-34

V

- variable length files 3-3

VM/370

- accessing 2-28
- additional parameters 2-34
- backspace setting 2-36
- CMS commands 2-35, 3-5, 3-6
- Comparing files 3-14
- Changing defaults 6-2
- CP commands 2-36, 2-42, 3-5, 3-7
- default parameters 2-26
- disconnect 2-36
- downloading files 3-3, 3-12
- EDIT EXEC 3-7
- editor 3-5, 3-6, 4-11, 5-7, 5-8, 5-10
- error messages 4-11
- example 2-37
- file transfers 3-5
- filenames 3-10
- logoff 2-36
- logon 2-35, 2-41
- SET command 3-5, 3-6, 3-7
- starting up terminal 2-28, 2-34
- terminal 2-6, 2-24, 2-26, 2-34
- TERMINAL command 2-36, 2-37, 2-42, 3-5, 3-7
- terminal linesize 2-37, 2-42
- terminal operation 5-3
- terminal parameters 2-26
- transferring files 3-5
- uploading files 3-9, 3-10
 - using 2-34
- VMMOD 1-6, 2-34

W

word length 7-7

X

xedit editor - VM/370 3-5

xoff character 2-12, 2-14,
2-22, 3-24, 5-5

xon character 2-12, 2-14,
2-22, 3-23, 5-5

xon/xoff protocol 2-13,
2-17, 2-22, 2-26, 2-49, 5-5



The Personal Computer
Software Library

Product Comment Form

Asynchronous Communications Support

6025218

Your comments assist us in improving our products. IBM may use and distribute any of the information you supply in anyway it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Comments:

If you wish a reply, provide your name and address in this space.

Name _____

Address _____

City _____ State _____

Zip Code _____

Tape

Please do not staple

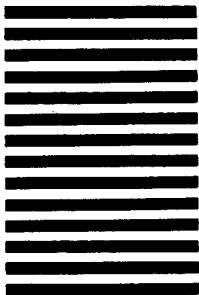
Tape

Fold here

BOCA RATON, FLORIDA 33432
P.O. BOX 1328-C
SALES & SERVICE
IBM PERSONAL COMPUTER

POSTAGE WILL BE PAID BY ADDRESSEE

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 123 BOCA RATON, FLORIDA 33432



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

