

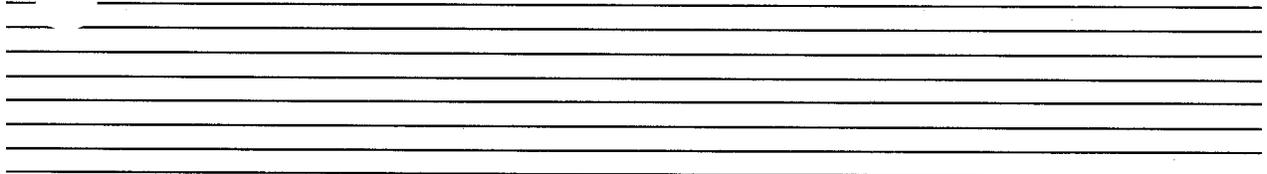
MOVE-IT™
INTERCOMPUTER
COMMUNICATION
SYSTEM

User Manual


DIGITAL
RESEARCH™

THE CP/M
LIBRARY™

MOVE-IT™
***INTERCOMPUTER
COMMUNICATION
SYSTEM***
User Manual



COPYRIGHT

Copyright © 1982 WOOLF SOFTWARE SYSTEMS. All rights reserved. Reprinted by Digital Research, Inc. P.O. Box 579, 160 Central Avenue, Pacific Grove, CA 93950 with the express permission of WOOLF SOFTWARE SYSTEMS.

No part of this publication may be reproduced, transmitted, or by any means transcribed without prior written permission of WOOLF SOFTWARE SYSTEMS, 6754 Eton Ave., Canoga Park, CA 91303.

WOOLF SOFTWARE SYSTEMS makes no representations or warranties, written or otherwise, with respect to the contents hereof. Further, WOOLF SOFTWARE SYSTEMS reserves the right to revise this publication without obligation of WOOLF SOFTWARE SYSTEMS to notify any person(s) of such revision.

CP/M and CP/M-86 are registered trademarks of Digital Research. MP/M and DDT are trademarks of Digital Research. MS-DOS is a trademark of MicroSoft. Altos is a registered trademark of Altos Computer Systems. Eagle is a registered trademark of AVL EAGLE Computer Investment Group, Inc. Pet is a trademark of Commodore Business Machines, Inc. Cromemco is a registered trademark of Cromemco Incorporated. TU-ART is a trademark of Cromemco Incorporated. Godbout is a registered trademark of Godbout Electronics Inc. Hayes MicroModem 100 is a trademark of Hayes Microcomputer Products, Inc. IMS is a registered trademark of IMS International. Imsai is a registered trademark of IMSAI International. Kaypro II is a trademark of Non-Linear Systems, Inc. Monroe is a registered trademark of Monroe Systems. Morrow and MicroDecision are trademarks of Morrow Designs. NorthStar and Horizon are trademarks of North Star Computers, Incorporated. Otrona is a registered trademark of Otrona Advanced Systems Corporation. Attache is a trademark of Otrona Advanced Systems Corporation. Radio Shack is a registered trademark of Tandy Corporation. TeleVideo is a registered trademark of TeleVideo Systems, Inc. Vector is a registered trademark of Vector Graphics Incorporated. BitStreamer and BitStreamer II are trademarks of Vector Graphics Incorporated. Bullet is a trademark of Wave Mate, Inc. Xerox is a registered trademark of Xerox Corporation.

NOTATION USED IN THIS MANUAL

In this manual, the following notation is used:

1. Characters typed by the user are in green type.
2. The symbol <cr> means to press the return key. On some systems, this key may be labeled "enter".
3. The symbol <esc> means to press the escape key. For computers without an escape key (usually labeled ESC), usually pressing the control key and the left square bracket key at the same time will accomplish the same thing.
4. The term "local computer" refers to the computer initiating the commands. The term "remote computer" refers to the computer at the other end of the communications line.
5. Move-It is designed to be compatible with a wide variety of computers and operating systems. Where a specific implementation differs from the norm, a note will be made in parenthesis. Filenames, where used, follow the conventions of CP/M®.

Table of Contents

I	Introduction	1
II	Commands	3
	ANSWER Command	4
	ASCII Command	5
	BINARY Command	6
	CALL Command	7
	Creating an Auto-dial Directory	8
	EXITING MOVE-IT Command	9
	GET Command	10
	GETTING HELP—THE ? Command	11
	HANGUP Command	12
	LDIR Command	13
	LUSER Command	14
	MESSAGE Command	15
	NOCONSOLE Command	16
	RDIR Command	17
	RUSER Command	18
	SEND Command	19
	TALK Command	21
	Getting Help in Talk Mode	21
	Sending Files in Talk Mode	21
	Talk Mode Switches	24
	Getting Files in Talk Mode	23
	Exiting Talk Mode	25
	TRIES Command	26
III	Move-It Messages	27
IV	Installation	31
	Writing a Custom MOV-BIOS*	35
	Installing Move-It on a Non-supported Computer*	36
	Troubleshooting the Installation	38
V	Operational Examples	41
VI	Using Move-It to Run a Program on the Remote Computer	43

Table of Contents (continued)

A	Sample Move-It BIOS (MOV-BIOS.ASM)	45
B	Configuration Information for Move-It	49
C	Glossary of Computer Terms	69
D	ASCII Code Chart	73

Section I

Introduction

Congratulations! You have purchased Move-It™, one of a growing line of quality software products from WOOLF SOFTWARE SYSTEMS. Move-It represents what we believe to be among the most sophisticated communications programs available to the microcomputer user.

The Move-It program allows the user to SEND files to or GET files from any other computer system running the Move-It program. Files can be transferred regardless of the particular hardware environment in use in either of the two systems. Thus, programs and data can be transferred between 8" and 5 1/4" systems, between hard disk and floppy systems, and even between systems running different operating systems, such as CP/M® and MS-DOS™. The transfer of files is accomplished using a packet format, including message numbers, data count, and 16-bit checksumming. This results in an extremely low error rate, insuring excellent file integrity even with poor phone lines or long cable runs. In addition, Move-It provides commands to retrieve both local and remote directories, control both local and remote user numbers, send messages to the remote computer, control auto-dial modems from a user-modifiable phone number file, control the number of data bits sent down the communications line, and to emulate a smart terminal to a remote timesharing system.

In the smart terminal (TALK) mode, the user can both send and receive files to timesharing systems, as well as control duplex, parity and other communications parameters.

The Move-It program is invoked by typing either:

MOVE-IT <cr> (Case 1)

-or-

MOVE-IT command line <cr> *** (Case 2)

In either case, the Move-It program is loaded into the system and executed. In case 1, command lines are read directly from the console and acted upon. After a command is executed, the Move-It program will prompt the user for another command by displaying a *. This prompting and executing will continue until the user enters a control-c at the console, or the Move-It program detects a message from the other computer system.

In case 2, the Move-It program executes the command line and immediately returns to the operating system.

Note: An exception case exists to the above rule. If the command on the command line is other than SEND or GET, Move-It does not return to the operating system but instead acts as if case 1 was used.

Section II Commands

In this section, several conventions will be used to describe the format of each of the commands. Lower case letters included in the command can be considered to be optional. Thus, the command:

Send FILENAME.TYP

can be abbreviated to:

S FILENAME.TYP

In addition, letters in parenthesis describe additional modifiers to the commands. These modifiers are generally used to route the file to a different drive or to give it a different name, or to provide options to the basic commands. These are more fully explained in the discussion on each command.

Commands and filenames can be entered in either upper- or lower-case and still be processed correctly by the program.

If a list of all the commands Move-It can execute is desired, simply press the question mark and then press return. Move-It will display the list of commands together with a short explanation of each.

The Move-It program recognizes the following commands as valid:

ANSWER

The ANSWER command has the following syntax:

ANswer

The ANSWER instructs Move-It to answer the phone connected to the modem specified during installation. This command generally answers the line at the next ring. There are no options available to this command.

ASCII

The ASCII command has the following syntax:

Ascii

The ASCII command conditions Move-It to transmit and receive messages in 7-bit format. This enables networks and computers with less than the normal 8 bit data path to transfer files. Under most conditions, you should not need to use this command.

Note: The ASCII and BINARY commands only change the version of Move-It running locally. To put both computers in ASCII mode, the ASCII command would have to be typed on both machines.

BINARY

The BINARY command has the following syntax:

Binary

The BINARY command conditions Move-It to transmit and receive messages in 8-bit format. This enables faster transfer of files for those computers that can support a full 8 bit data path. Move-It normally comes up in BINARY mode.

Note: The ASCII and BINARY commands only change the version of Move-It running locally. To put both computers in BINARY mode, the BINARY command would have to be typed on both machines.

CALL

The CALL command has the following syntaxes:

Call NUMBER

Call NAME

Call ?

The CALL command allows Move-It, when configured with appropriate auto-dial software (see the section on Installation) to call remote computers and information services.

The CALL command can be used in three ways. First, if a number follows the CALL command, Move-It will simply dial the number. The digits 0–9 are allowed, as well as the *, which means pause for 2 seconds. This can be used to cause a time delay to allow for a second dial tone before dialing the rest of the number. All other characters can be included for readability and will be ignored by the program.

For example, to dial 9, wait 2 seconds, then dial 555-1212, use the command:

```
CALL 9*555-1212
```

The CALL command can also be used in conjunction with a user-created file called PHONE.NUM to call locations by name.

For example, if the above number was stored in PHONE.NUM on the default drive under the name Directory, the call could have been initiated by typing:

```
CALL DIRECTORY
```

Move-It will look up the number in the file, then dial it and return to the prompt. If the name is not found in PHONE.NUM, an error message will be displayed.

Lastly, if CALL is followed by a question mark, Move-It will display the names contained in PHONE.NUM (see next section).

CREATING AN AUTO-DIAL DIRECTORY

To fully exploit the CALL command, the user should put all his frequently-called numbers in PHONE.NUM so that he can reference them by name. To do this, use any editor (like ED or Wordstar®) to create a file with the name PHONE.NUM according to the following format:

```
NAME1, NUMBER1, SWITCHES <carriage return, line feed>
NAME2, NUMBER2, SWITCHES <carriage return, line feed>
```

where NAME is the name you wish to refer to (like JIM or SOURCE), NUMBER is the phone number to dial (including asterisks, if needed), and SWITCHES controls the default communications parameters used in the TALK mode. The SWITCHES field can be any combination of the following, separated by commas:

- HALF Sets half duplex
- FULL Sets full duplex (default)
- EVEN Sets even parity
- ODD Sets odd parity
- NOPARITY Sets no parity (default)
- ALF=ON Sets auto-linefeed on
- ALF=OFF Sets auto-linefeed off (default)
- ECHO=ON Sets remote echo on (default)
- ECHO=OFF Sets remote echo off
- PRINT=ON Turns printer on in Talk mode
- PRINT=OFF Turns printer off (default)
- XON=<hex number> Re-defines XON char (default: 11)
- XOFF=<hex number> Re-defines XOFF char (default: 13)

These options and their effects are more fully described in the section covering TALK mode, elsewhere in this manual. Also, a sample PHONE.NUM file is contained on the distribution diskette.

EXITING MOVE-IT

The Exit command has the following syntax:

Exit

When this command is exited, Move-It will return to the operating system.

GET

The GET command has the following syntax:

Get DRIVE:FILENAME.TYP (As DRIVE:FILENAME.TYP)

This command allows the user to get a file (or files) from the remote computer (also running Move-It) and store them on local computer's disk.

As with the SEND command, the files from the remote computer are normally saved on the local computer under the same name. This can be changed by specifying the AS option. This option lets the user "rename" the file as it comes from the remote computer. For example: to get the file FAREND.TXT and place it on drive B: under the name NEAREND.DOC, enter the command:

```
GET FAREND.TXT AS B:NEAREND.DOC
```

In addition, the AS option can be used to specify the destination drive for a wildcard GET. To get all the files from the remote computer's drive C: and put them on the local drive D:, type:

```
GET C:*. * AS D:
```

After the command is entered, Move-It will print:

```
Receiving Filename.Typ . . .
```

As each block of data is being sent to the remote computer, Move-It will flash working on the console to let you know the transfer is in progress.

Upon completion of the transfer, Move-It will print complete. and either start the next transfer (if a wildcard was specified) or return to the prompt.

As with the Send command, if the local disk or directory becomes full during the transfer, Move-It will delete the partially-built file and display its full message. At this point the user can either press control-c to exit back to CP/M, or insert another disk in the local disk drive and press the return key. This feature allows the user to transfer files from a high-capacity disk to a lower-capacity disk, and have Move-It continue the transfer across disk boundaries.

GETTING HELP – THE ? COMMAND

Move-It contains 4 help files to assist the user. These help files are invoked by pressing the ? key. At the main * prompt, the ? command will display a list of the commands. In the Call command, a ? will display all the names in the PHONE.NUM file. In Talk Mode, <esc>-? will display a list of all the escape commands. Lastly, in Talk Mode's Send function, a ? will invoke a display of the possible protocols available to the user.

HANGUP

The HANGUP command has the following syntax:

Hangup

The HANGUP command instructs Move-It to hang up the auto-dial modem specified during installation. There are no options available to this command.

LDIR

The LDIR command has the following syntax:

```
LDIR DRIVE:FILENAME.TYP
```

The LDIR command commands Move-It to obtain a directory from the local disk denoted by DRIVE:. All of the combinations of drive names, file names, and types are supported.

For example, the command:

```
LDIR A:*.ASM
```

would display on the user's console all the files with the type of .ASM on the local drive A:.

LUSER

The LUSER command has the following syntax:

LUser <optional user number>

The LUSER command allows the user to control the user areas on the local computer under CP/M and other operating systems. The LUser command has two forms. First, if the command is specified without the optional user number, Move-It will report the current user number. Second, if the user number is specified, Move-It will set the computer to the user number specified, and then report it back to the user.

When Move-It is exited, the user number will be the same as when Move-It was entered.

MESSAGE

The MESSAGE command has the following syntax:

Message (message to be sent)

The message command allows the user to send a message to the remote computer's console. Using this command, the user could request the operator at the other computer to mount different disks, and perform other operations.

For example:

Message Please mount the payroll disk on B:

would result in the following message on the remote console:

Message from remote computer:

Please mount the payroll disk on B:

NOCONSOLE

The NOCONSOLE command has the following syntax:

```
NOConsole
```

This command prevents Move-It from echoing prompts and other messages to the user's console. This is useful in MP/M™ and other timesharing systems, where the communication line is used both as the console and the data line. This command also permits the user to start Move-It under MP/M and then detach from the job.

Suppose you wish to allow transfers to and from your MP/M system. To start Move-It, you would type

```
MOVE-IT NOCONSOLE
```

and then detach from the console. This would prevent MP/M from hanging, waiting for a non-existent console.

In addition, the NOConsole command is useful in systems where the console can be assigned to the modem port. If the user assigned the console in this way, and obtained an auto-answer modem, Move-It could be used to allow fully unattended remote access to the computer system. In this mode of operation, the NOConsole command would be used to prevent any confusion on the communication line when a remote computer, also running Move-It, called in to transfer files.

For example, if the user set up one system with an auto answer modem, a remote user could call in, supply any passwords or other information (using the TALK command, explained later in this manual), and then use the SEND, GET, RUSER and RDIR commands to perform the transfer.

Note: The user should NOT specify NOCONSOLE mode if, in fact, a console exists. This is because Move-It will not accept commands from the "nonexistent" console, and the computer will either have to be reset, or the remote caller will have to exit from Move-It by entering TALK mode and sending 5 ESCapes.

RDIR

The RDIR command has the following syntax:

```
RDir DRIVE:FILENAME.TYP
```

The RDIR command instructs Move-It to obtain a directory from the remote disk denoted by DRIVE:. All of the combinations of drive names, file names, and types are supported.

For example, the command:

```
RDIR B:*.ASM
```

would display on the user's console all the files from the remote system's drive B: with the type of .ASM.

After the RDIR command is entered, Move-It will display:

```
Receiving filenames, please be patient . . .
```

And then commence to display the filenames as they come in over the communications line.

RUSER

The RUSER command has the following syntax:

RUser (optional user number)

The RUSER command allows the user to control the user areas on the remote computer. Like the LUSER command, RUSER has two forms: First, if the command is specified without the user number, Move-It reports the user number of the remote computer. If the user number is specified in the RDIR command, Move-It will attempt to set the user number in the remote computer, and then report the user number back to the user. If the user number reported back to the user is not the same as the user number specified, the remote computer has the RUSER command locked out, and will not permit the user area to be changed.

SEND

The SEND command has the following syntax:

Send DRIVE:FILENAME.TYP (As DRIVE:FILENAME.TYP)

This command allows the user to send a file (or files) to another computer running Move-It. The transfer is performed with full error checking and correction. As Move-It observes all the normal file name rules, all of the following forms for DRIVE:FILENAME.TYP are valid:

```
A:TEXT.DOC
TEXT.DOC
*. *
program.com
*.DOC
TEX?.DOC
```

Normally, the file is saved on the remote computer with the same name as the file on the local computer.

For example, SEND MYFILE.COM would create the file MYFILE.COM on the remote computer. This can be modified by using the AS option. Say we wanted to send the file MYFILE.COM to the remote computer and have it reside as YOURFILE.COM on drive D:. The proper command for this would be:

```
S MYFILE.COM AS D:YOURFILE.COM
```

The same option can be used to change the destination drive of a wildcard transfer. For example:

```
S *.COM AS D:
```

would transfer all the .COM files from the logged-on drive to drive D: on the remote computer.

After the command is entered, Move-It will print:

```
Sending Filename.Type . . .
```

As each block of data is being sent to the remote computer, Move-It will flash working on the console to let you know the transfer is in progress.

Upon completion of the transfer, Move-It will print complete. and either start the next transfer (if a wildcard was specified) or return to the prompt.

SPECIAL NOTE:

If the remote disk or directory becomes full during the transfer, Move-It will delete the partially-built file and display its full message. At this point the user can either press control-c to exit back to the operating system, or insert another disk in the remote disk drive and press the return key on the local computer (the one that printed the message). This feature allows the user to transfer files from a high-capacity disk to a lower-capacity disk, and have Move-It continue the transfer across disk boundaries.

TALK

The talk command has the following syntax:

Talk (Half) (Full) (Odd) (Even) (Noparity)

The talk command allows the user to use Move-It as a "smart terminal". When entering this mode the user can specify half duplex, full duplex, odd parity, even parity, or no parity. These parameters are usually specified by the remote computer installation. All the settings in Talk Mode will stay in effect until explicitly changed. This means that the user can exit Talk mode and then re-enter without having to re-specify all the parameters. In the absence of any parameters, Move-It defaults to No Parity, Full Duplex.

When the Talk mode is entered, Move-It will display the duplex and parity in effect, and report to the user the number of characters in the trapping buffer (see GETTING FILES IN TALK MODE).

In general, any characters entered through the keyboard will be sent out the communications port, and any characters coming in the communications port will be displayed. Through the use of escape commands (explained below), the user can also send files to the remote computer, trap incoming data to a file, send data to the printer, and control other line parameters. This mode can be used to talk to databases like The Source, to log into a timesharing system, or to converse with another Move-It in the same mode.

GETTING HELP IN TALK MODE

To obtain a list of the Talk Mode escape commands, simply press <esc> and then the ? key. Move-It will display a list of the possible escape commands, together with a short explanation of each.

SENDING FILES IN TALK MODE

Move-It's Talk mode allows the user to send text, hex, and other Ascii files to practically any other computer system. To send a file to another computer, press <esc>-S. Move-It will display:

Filename to send to remote computer?

Enter the name of the file to send, or press the return key to abort and return to normal Talk Mode. After the filename is entered, Move-It will display:

Please enter the letter that describes the protocol you wish to use (N,E,C,T,X,?), or press Return to abort.

To see a list of the possible protocols, press ? and then the return key. The valid protocols are:

- N —This is really no protocol at all. Characters are sent out the port at full speed until the entire file is transmitted or control-c is pressed. This protocol should be used only if the user is sure that the remote computer can accept characters at full speed (some can't).
- E —The echo wait protocol can be used where the remote computer echos back characters to the local screen (full duplex). This protocol sends 1 character to the remote computer and waits for its return. Move-It then compares the two characters. If they are different, a message is displayed, and the user is asked if he wants to continue. If they are the same, the next character is sent. This sending, checking sequence continues until entire file is transmitted or the user aborts by pressing control-c. This is the only protocol available in Talk Mode that verifies the transmission. However, since Move-It has to wait for the return character, transmission speed is halved.
- C —In the character wait protocol, Move-It sends 1 line and waits for a user-specified character to be sent by the remote computer. This mode is useful if the remote computer displays a prompt when it expects a line of data. Most editors operate in this manner. Move-It will ask for the character to wait for when this mode is entered. As soon as the character is entered, Move-It will start sending the file. The character is entered as a single keystroke from the keyboard, with no return required. If desired, the transmission can be aborted by pressing control-c.
- T —The time wait protocol is similar to the character wait protocol described above in that Move-It transmits the file on line at a time. Time wait, however, does not wait for any particular character after the line is sent. Instead, time wait sends the line, then waits 1 second and transmits the next line and so on. This mode is used in systems where data is processed one line at a time, but no character is sent upon completion.

- X —Xon/Xoff protocol is the most common protocol in use. In this protocol, characters are sent at full speed until the remote computer sends the XOFF character, and resumes when the remote computer sends the XON character (See Modifying XON and XOFF characters). This protocol provides the fastest data transmission speed while still insuring that the remote computer received all the data. Normally, the XON character is a control-q and the XOFF character is a control-s, but these can be changed with the <esc>-X command, described elsewhere in this manual.

At least one of the protocols described above should work with the computer system you wish to send data to. If you are in doubt, talk to the system operator (sysop) or manager. Also, you can abort a talk mode send by pressing control-Z.

GETTING FILES IN TALK MODE

Move-It also allows the user to trap incoming data into a file, for later printing or display. This allows the user to call up another system, get the information he needs, hang up the phone, and then look over the information, resulting in much lower phone bills.

To Get a text file from the remote computer, press <esc>-G. Move-It will ask for the name of the file to save the data into. Move-It will open the file, and turn the trapping switch on. This will enable data to be saved into memory and then to disk. To toggle the trapping switch, refer to the <esc>-T command.

When Move-It's internal buffer fills up, Move-It sends the XOFF character to the remote computer, waits until the remote computer stops sending data (or inputs an additional 256 characters), and saves the data off to the disk. After the save is complete, Move-It sends the XON character and continues.

To close the trapping file, use the <esc>-E command (end talk mode).

TALK MODE SWITCHES

To allow the user more control over the session, Move-It allows various switches to be set at the user's discretion. These are described below:

Auto-linefeed

Setting this switch will cause Move-It to insert a linefeed after every carriage return typed by the user. This switch is set by pressing <esc>-A. Move-It will display the appropriate status message in square brackets.

Remote Echo

Normally, Move-It displays characters received from the remote computer on the user's console. Sometimes it is desirable to suspend the display. This happens when the data is coming in faster than the console can display. The remote echo switch is toggled by pressing <esc>-C. Move-It will display the appropriate status message in square brackets.

File Trapping

When the Talk Mode Get function is invoked, file trapping is turned on. This means that the incoming data is saved into memory and later to disk. With the File Trapping toggle, the user can selectively turn the trapping function on and off. The File Trapping toggle is invoked by pressing <esc>-T.

Printer

The printer can be toggled on and off in Talk Mode by pressing <esc>-P. The printer should be used with great caution, however, if no data is to be lost. In general, the printer speed should be at least twice the speed of the incoming data. Also, some printers will lock up when certain character groups are received. If you experience lost data, use the <esc>-G function to trap the data into a file, and then print it later.

XON and XOFF characters

Some Talk Mode functions use the XON and XOFF functions to control the sending and receiving of data. Normally, these characters are set to control-Q (11 hex) and control-S (13 hex), respectively. When this function is enabled by pressing <esc>-X, Move-It will display the current values of XON and XOFF in both hex and ASCII, and then ask for the new values. These values should be input in hex, or the return key should be pressed to keep the current values.

EXITING TALK MODE

To exit Talk Mode, simply press <esc>-E. If Get Mode has been invoked and there is data in the buffer, Move-It will ask if the data is to be saved. If so, the file will be closed. Move-It will then exit back to the command mode prompt (*).

TRIES

The TRIES command has the following syntax:

TRies

The tries command allows the user to display the number of retries Move-It has done in transmitting or receiving files since the last time the counter was reset. The counter is always reset when the program is started, and also whenever the user requests it via the TRIES command itself (see below).

When the command is invoked, the computer will print:

Number of retries: xxxxx

Where xxxxx is the decimal number of retries in the counter (0-65535).

The system will then print:

Do you want to clear the retry counter?

Respond with either yes or no (either upper- or lower-case) and then press the return key.

The tries command is generally used to gauge the quality of the phone line used to transfer files. If, by the end of the transfer, the tries command shows a large number of retries, you may want to hang up and re-dial the remote computer.

Section III

Move-It Messages

During the course of using Move-It, the following messages may appear:

File name already exists on remote end. Replace?

File name already exists on local end. Replace?

When Move-It discovers that a file it is about to transfer already exists on the destination system, it asks for permission to delete the file before continuing. Respond with either a Y for yes, or N for no, and then a carriage return.

Remote directory full. Insert new disk and press return to continue, or control-c to exit to Operating System.

Directory full. Insert new disk and press return to continue, or control-c to exit to Operating System.

These two messages signal that the destination directory is full. Move-It deletes the partially-built file and allows the user to insert another formatted disk in the destination drive and continue the transfer, or to exit to the operating system.

Remote disk full. Insert new disk and press return to continue, or control-c to exit to Operating System.

Disk full. Insert new disk and press return to continue, or control-c to exit to Operating System.

These two messages signal that the destination disk is full. Move-It deletes the partially-built file and allows the user to insert another formatted disk in the destination drive and continue the transfer, or to exit to the operating system.

[Auto Line Feed is on]

In Talk Mode, Move-It displays various messages in square brackets in response to escape commands. These are for information only, and have no effect on Move-It's operation.

There is no file open to accept the trapped data. Use `<esc>-G` to open a file.

If this message is displayed, the user tried to toggle the trapping function (`<esc>-T`) on, when no file had been opened by using the `<esc>-G` function. The command is ignored.

The remote computer echoed the wrong character. Press return to ignore and continue, or Control-C to abort.

If the character echo mode is used, and the echoed character is not the same as the transmitted character, this message is displayed. To continue, just press the return key.

Sorry, the remote computer does not support this command.

If the remote computer is running an older version of Move-It, all commands may not be supported.

An error exists in the switches for the name called.

If this message is displayed, the name called from the PHONE.NUM file has errors in the switches section. See "Creating an Auto-Dial Directory" for more information.

Name not found in PHONE.NUM.

The name the user specified in the Call command could not be found in PHONE.NUM. Perhaps a misspelling error?

There is data in the trapping buffer. Do you wish to save it?

The user used the `<esc>-G` command to open the trapping buffer and characters were received. If you wish to save the data and close the file, enter a Y. Otherwise, enter an N.

Not enough memory for function.

Move-It tries to protect the system by checking for memory overflow during commands. If an overflow is detected, this message is displayed. For most systems, however, this should never happen.

Warning – When an ambiguous name is transferred, only the drive name is permitted in the 'AS' field. File name ignored.

If either the file name or file type fields in a GET or SEND contain an asterisk and the AS field contains a file name, this message is printed, and the file name portion is ignored.

Communications Line Failure.

Either the comline is down or the remote computer is no longer running the utility.

Try again?

If Move-It fails to establish communications with the remote computer in 5 retries, it prints this message. If you wish to try again, type Y and then press the return key; otherwise, press N, return.

Section IV Installation

Installing Move-It on your computer system is done in three stages:

First, verify that the following files exist on the distribution disk:

MOVE-IT.UCF	(The unconfigured Move-It program)
MCONFIG.COM	(The Move-It configuration program)
MOV-BIOS.ASM	(A skeleton custom BIOS for Move-It)
PHONE.NUM	(A sample phone number file)

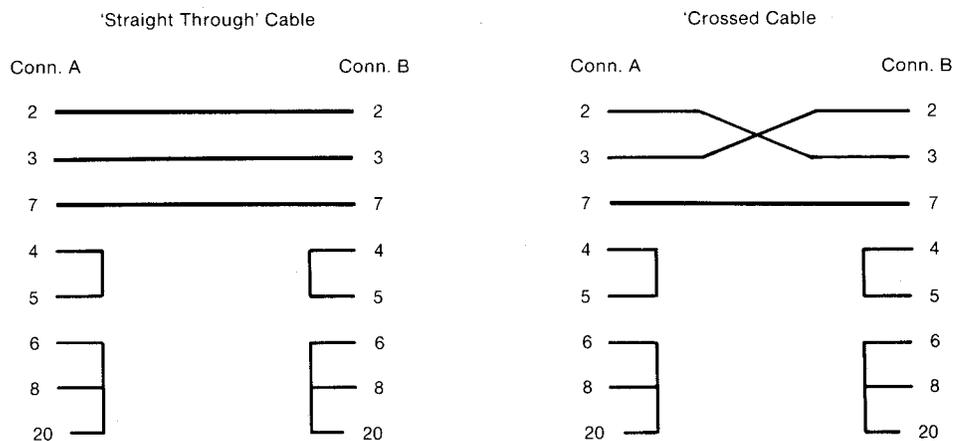
Also, the distribution disk should contain a group of files with an extension of .CNF as well as some number of files with an extension of .DIL.

If any of these files are not present, notify WOOLF SOFTWARE SYSTEMS immediately!

Next, if you are going to use Move-It to “talk” between computers locally, you will need to make a cable to connect the two computers together. To understand how to do this, a little history is in order. When the RS-232 specification was set up, the technology of the day was to use modems to talk to terminals for telegraph use. The specification consisted of DTE (data terminal equipment) and DCE (data communication equipment) devices. Since a terminal is a DTE device and a modem is a DCE device, a standard RS-232 cable would connect these devices and work every time. That’s fine for terminals and modems, but what about computers? When personal computers came about, the manufacturers of these machines had to decide whether to connect to terminals (DTE) or modems (DCE). About half decided one way and the other half decided the other.

Because of this, a cable to connect two computers together could be one of two types; a straight through cable (to connect DCE to DTE), or a crossed cable (to connect DTE to DTE or DCE to DCE). To tell which one you need, you can check the users manuals for the computers, or try both types (they are illustrated on the next page). If both computers normally use their RS-232 ports for printers, you’ll need a crossed cable.

The two types of cables can be made by obtaining two RS-232 connectors and wiring them together as follows:



Lastly, you'll need to tell Move-It how to communicate with your particular hardware. Follow this procedure for configuring the Move-It program:

1. **MAKE A COPY OF YOUR DISTRIBUTION DISK!** We cannot be held responsible if you clobber your distribution disk. However, if the unthinkable happens, call us and we'll arrange to send you another disk.
2. Log on to your copy of the distribution disk, type MCONFIG and press the return key. The MCONFIG program will sign on and ask for the filename of Move-It to configure. For now, just press the return key.
3. Next, MCONFIG will present a list of machines and I/O boards. If your system is one of these, simply type in the appropriate number, press return, and proceed to step 12.
4. If your machine was not one of the listed ones, you'll need to obtain the port and status information for your computer or I/O card. Check Appendix B of this manual for your machine. If it's there, proceed to step 5.

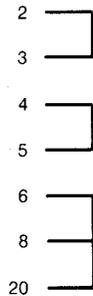
If not, the required information can usually be found in the user's manual for your computer. Generally, the computer "talks" to the serial device through two "ports". These ports allow the computer to: 1) Obtain the status of the serial

device (whether Move-It can send a character and whether a character is waiting for Move-It), and 2) Send and Receive data from the serial device. In the status information, there is usually 1 bit that is for Receive Buffer Full, RBF, (whether a character is waiting) and another bit for Transmit Buffer Empty, TBE, (whether Move-It can transmit a character).

Move-It also needs to know whether the status bit is true when it's a one or a zero. Lastly, since Move-It can initialize the serial port if required, the information on initialization should be obtained. The manufacturers of the various computer systems on the market sometimes provide this information in their manuals, or can provide it over the phone. If all else fails, you can call us at (213) 703-8112 during normal working hours, Monday through Friday. We can't guarantee anything, but we are constantly adding new information to Appendix B, and we may have added your system recently.

5. MCONFIG first needs the location of the data port. The current value for most questions is displayed in parenthesis. To keep this value, simply press return, or enter the new value from your notes or the Appendix.
6. Next, MCONFIG needs to know the location of the status port. Again, the default value is displayed in parenthesis. Enter the new value and press return.
7. Next, enter the transmit buffer empty mask. The default is displayed in parentheses.
8. Next, enter the receive buffer full mask.
9. MCONFIG next needs to know the polarity of the status bits. That is, whether the masks are true when they are a one or a zero. Practically all current machines use a polarity of one, so, unless yours is an exception, just press the return key.
10. Next, enter the initialization bytes. Most modern communications chips require initialization to work properly. Some times, this initialization is done in the system itself. If this is the case, simply press the escape key in response to this question. Otherwise, enter the bytes as pairs of numbers. The first number is the initialization data, and the second number is the port to send it to. This is typically the status port, but it may be different for your system. MCONFIG will allow you to enter up to 20 pairs of numbers. When you are finished, press the escape key.

11. MCONFIG next requests you to enter a descriptive line which will be added to the sign-on. This will allow you to later remember what port, baud rate, etc. you set up this version for. Terminate the line with a carriage return.
12. MCONFIG will next display a list of auto-dial modems. If you wish MCONFIG to include support for one of these, enter it's number. Otherwise, enter 99 and press the return key.
13. MCONFIG will then ask if you wish to allow the remote computer access to all the user areas on your system, or only the current user area. If you wish to allow access to all the user areas, answer yes to this question.
14. MCONFIG will next ask you if you wish to have it test the serial port information you have supplied. To properly test the port, the transmitter and receiver pins on the serial connector need to be connected together according to the following diagram:



MCONFIG will test the serial port and display either an error message, or an OK message. If you received an error message, proceed to the section "Troubleshooting the Installation".

15. If the serial port tested ok, or you didn't tell MCONFIG to test it, MCONFIG will ask for the name to save Move-It under. Normally, this would be Move-It.COM, but if you have multiple systems, you may want to save Move-It under a different name for each. Pressing the Return key only defaults to Move-It.COM.

Note: Move-It is licensed for single-site use only. This entitles the user to install Move-It on up to 5 computers at a given physical location. If you plan to install Move-It on more than one computer, give us a call to request more registration cards. You **MUST** fill out one of these cards for each machine you plan to use Move-It on. There is no charge for this, and, since we **WILL** go after people who violate our agreement (since software is our only business), it will save you (and us) a lot of headaches.

For most computers, that's all there is to it!

WRITING A CUSTOM MOV-BIOS *

On the distribution disk there is a file called MOV-BIOS.ASM. This is a skeleton custom MOV-BIOS, which should help you write your own. The information below describes the entry points in the MOV-BIOS (refer to Appendix A, MOV-BIOS listing):

JMP 0302H	This jumps into the Move-It program itself. It must not be changed.
JMP 4000H	This jump points to the lowest location Move-It will use to build filenames for transfers. The 4000H can be raised to 4800H to reserve the space from 4000H to 4800H for user code, if needed.
JMP INITI	This jump points to the initialization section needed to support the particular serial card used. In some cases, CP/M's BIOS will handle the initialization. If this is the case, simply have the JMP point to a return.
JMP FINISH	This jump points to the de-initialization section. This can contain code to reset the serial port for a printer or other device, if needed. Normally, this JMP points to a return.
JMP INSTA	This jump obtains the status of the input port, and returns the A reg. equal to 0FFH if a character is waiting, or 0 if not.
JMP INDAT	This jump gets the character from the input port and returns it in the A reg. Usually this is just an IN command followed by a RET.

* These op codes/locations refer to CP/M. Call for information on other operating systems.

- JMP OUTSTA** This jump obtains the status of the output port. Again the status is returned in the A reg., and is equal to 0FFH if the board is ready for character, or 0 is busy.
- JMP OUTDAT** This jump outputs the character in the A reg. Usually this is just an OUT instruction followed by a RET.
- JMP CONFIG** This jump points to a string of Ascii characters that contain the configuration line for Move-It. The string is terminated by a \$.
- JMP DIAL** This jump points to the modem support routine. Upon entry, the B register is a 0 to dial, a 1 to hang up the phone, and a 2 to answer the phone. If the B register is a 0, the HL register points to the string to be dialed. Normally, locations 200H to 2FFH are used for the modem support routine.

Once the MOV-BIOS is created and assembled, use DDT™ to combine it with Move-It, as follows:

```
A>DDT MOVE-IT.UCF<cr>
*INewBIOS.HEX<cr>
*R<cr>
*GO<cr>
```

```
A>SAVE 64 MOVE-IT.COM<cr>
```

INSTALLING MOVE-IT ON A NON-SUPPORTED COMPUTER *

Moving Move-It to a computer which cannot read one of our standard disks a little more complex. *A working knowledge of assembly language and CP/M is required.* Basically, PIP is modified on the destination computer to allow communication to the computer with Move-It on it. Next, a utility such as UNLOAD is used to convert Move-It.UCF and MCONFIG.COM to hex files, and then the Talk Mode Send command is used to transfer Move-It.HEX, MCONFIG.HEX, VERSIONS.DIL and VERSIONS.CNF to the target machine. Lastly, LOAD.COM is used to convert the hex files back to Move-It.UCF and MCONFIG.COM. Then the normal configuration instructions are followed.

* This section specifically applies to CP/M & MP/M, but the general idea can be applied to any operating system.

First, you'll have to use DDT to patch PIP to support the INP: device on the non-supported computer. This is a built-in device inside PIP. PIP expects to find a jump to the user's input section at location 103H. The user's input routine should put the byte received from the serial port in location 109H, after clearing the high bit (use an ANI 7FH).

PIP provides room from 110H-1FFH for the INP: routine.

The example below shows how the INP: code might be laid out.

(This routine assumes a data port of 21, a status port of 20, and a receive mask of 2. Be sure to modify these to fit your situation.)

Location	Code	Comments
103H	JMP 110H	Jump to input routine
110H	IN 20H	Get the status
112H	ANI 2	Mask it
114H	JZ 110H	Loop if not ready
117H	IN 21H	Get the data
119H	ANI 7FH	Clear the high bit
11BH	STA 109H	And save
11EH	RET	Go back to PIP

After PIP is patched, exit from DDT and do a save:

```
A>SAVE 29 PIPC.COM<cr>
```

Next, use UNLOAD (or whatever utility you have) to convert the Move-It .UCF and MCONFIG.COM files to hex:

```
A>REN MOVE-IT.COM=MOVE-IT.UCF<cr>
```

```
A>UNLOAD MOVE-IT 100<cr>
```

```
A>UNLOAD MCONFIG 100<cr>
```

Next, type the following command on the destination computer:

```
A>PIPC MOVE-IT.HEX=INP:[BEH]<cr>
```

And on the source computer, enter Talk Mode and type:

```
<esc>-S
```

Move-It will ask for the name of the file. Enter `MOVE-IT.HEX`, and for the protocol, enter `N`.

Move-It should transfer the file, and display its complete message. Now, press `<control> Z`. Pip on the receive side should exit back to CP/M. If nothing happens, try reversing pins 2 & 3 on one of the connectors, and try it again.

Next, use PIP to transfer `MCONFIG.HEX` and the other files in the same fashion.

Lastly, use `LOAD` on the destination computer to reload the files:

```
A>LOAD MOVE-IT<cr>
```

```
A>REN MOVE-IT.UCF=MOVE-IT.COM<cr>
```

```
A>LOAD MCONFIG<cr>
```

Now follow the normal configuration on the destination computer.

TROUBLESHOOTING THE INSTALLATION

If `MCONFIG` shows that the serial port tests OK, but you still can't transfer files, check the baud rate jumpers (if any) to verify that both computers are set to the same baud rate, 1 stop bit, 8 data bits and no parity. If you still have problems, install the Test plug (described in step 14 of the installation).

Next, run Move-It, enter Talk Mode, and type a few characters on the keyboard. These characters should be echoed on the screen. If they are, proceed to the next step. If not, re-check the configuration parameters and, if possible, check the computer's serial port. A surprising number are bad right from the factory.

If the above test checked out, try the same test on the other computer. If both machines check out, proceed to the next step.

Next, connect both machines together. Place both in Talk mode, and type on one of the keyboards. You should see the characters you typed on the OTHER machine's screen. Try this test on both keyboards. If you receive characters on the other screen, but the characters are not the same as the one's you typed, the two machines are set to different baud rates. If you get no response from either machine, try reversing pins two and three on ONE end of the cable only.

If the above test checks out, but Move-It still does not work, you probably have set one or both machines to even or odd parity. The computers MUST be set to NO parity for Move-It to function in binary mode. If you must leave parity enabled, set both machines to ASCII mode. All file types can still be handled, although the transfers will take longer.

If you have checked all the above items and Move-It still will not work, call us at (213) 999-3135, 10 AM to 5 PM Monday through Friday. We're here to help.

Section V

Operational Examples

1. Transfer local disk B to remote disk D:

```
SEND B:*.* AS D:<cr>
```

2. Transfers all .COM files from the logged-on disk to the remote disk A:

```
SEND *.COM AS A:<cr>
```

3. Transfer MYFILE.COM from disk A as YOURFILE.COM on disk B:

```
SEND A:MYFILE.COM AS B:YOURFILE.COM<cr>
```

4. Transfer all .COM files to the local logged-on disk from the remote disk C:

```
GET C:*.COM<cr>
```

5. Set NOCONSOLE mode:

```
NOCONSOLE<cr>
```

6. Set ASCII mode:

```
ASCII<cr>
```

7. Set BINARY mode:

```
BINARY<cr>
```

8. Get a local directory from drive C: for all .COM files:

```
LDIR C:*.COM<cr>
```

9. Get a remote directory from remote drive A:

```
RDIR A:<cr>
```

Section VI

Using Move-It To Run a Program On the Remote Computer*

Move-It can be used to execute a series of commands on the remote computer. To understand how this is done, a few words about CP/M's SUBMIT facility are in order. Whenever you type `SUBMIT XXX` at the console, SUBMIT looks for a file called `XXX.SUB`. If it finds one, it creates a file called `$$$SUB` and then returns to the CP/M prompt. If the `$$$SUB` file was created on any other drive but A:, nothing else happens, but if the file was created on drive A:, CP/M proceeds to execute all the commands in the file. This difference in how CP/M treats the `$$$SUB` file can be exploited to execute files on the remote computer.

To perform this small bit of magic, use this procedure:

1. On the local computer, use ED (or whatever editor you like) to create a file called `TEST.SUB`. In this file, enter the following lines:

```
DIR<cr>
STAT<cr>
MOVE-IT<cr>
```

2. Exit and save this file, log onto a drive other than A:, and type:

```
SUBMIT TEST<cr>
```

3. If all goes well, submit should run and create a file called `$$$SUB` on the logged-on drive. If, instead, the computer does the directory command, you're logged onto drive A:. Try logging onto another drive and try it again.

* This section applies to CP/M systems ONLY.

*(Thanks to K. Lombard for this suggestion)

4. If the \$\$\$\$.SUB file is in the directory, connect the remote computer type:

```
A>MOVE-IT<cr>  
*SEND $$$$.SUB<cr>
```

5. Enter Move-It's TALK mode and press the <ESCAPE> key five times. This causes the remote computer to exit Move-It and then execute the commands in the \$\$\$\$.SUB file. You should see the remote computer display the directory, report on free space on the disk, and then return to Move-It.
6. Lastly, press the <CTRL> key and the <D> key five times. This causes Move-It to exit TALK mode and re-enter command mode. You may now send files, look at directories, etc.

Any command the remote computer is capable of executing can be put into the .SUB file. In addition, through the XSUB command, you can even provide user input to other programs. See the CP/M manual for further information on these commands.

Appendix A

Sample Move-It BIOS (MOV-BIOS.ASM)

```

;
; CUSTOM MOV-BIOS SKELETON FOR MOVE-IT (10/1/82)
; FOR CP/M COMPATIBLE SYSTEMS
;
; WRITTEN BY: JEFF WOOLF
; WOLF SOFTWARE SYSTEMS
; 23842 ARCHWOOD ST.
; CANOGA PARK, CA 91307
;
ORG 100H

BIOS: EQU $ ;THIS IS THE USER'S BIOS!!!
START: JMP 0302H ;THE WAY TO THE PROGRAM
JMP 4800H ;END OF USER SECTION
JMP INITI ;COMLINE INIT.
JMP FINISH ;COMLINE DE-INIT.
JMP INSTA ;COMLINE STATUS
JMP INDAT ;COMLINE DATA INPUT
JMP OUTSTA ;COMLINE OUTPUT STATUS
JMP OUTDAT ;COMLINE OUTPUT DATA
JMP CNFGMS ;POINT TO THE CONFIGURATION MESSAGE
JMP DIAL ;AUTO-DIAL/ANSWER CODE

; THE CONFIG MESSAGE IS 80 CHARACTERS LONG, PLUS A CR AND TWO LFS
CNFGMS:
DB ' INSERT A MESSAGE THAT DESCRIBES '
DB 'YOUR MACHINE HERE '
DB 'ODH,OAH,OAH, '$'

INITI: ;INIT. COMLINE (NO ENTRY,EXIT PARAMS)

; — ANY CODE THE SYSTEM NEEDS TO INITIALIZE THE COMLINE —
; — SHOULD BE ADDED HERE —

RET

```

```
FINISH:                ;DE-INIT. COMLINE (NO ENTRY,EXIT PARAMS)
;  _____ ANY CODE THE SYSTEM NEEDS TO UN — INITIALIZE THE COMLINE _____
;  _____ SHOULD BE ADDED HERE _____
RET

INSTA:                 ;INPUT STATUS (A=FF IF CHAR READY,0 IF NOT)
;
;  ADD A ROUTINE TO READ THE STATUS OF THE INPUT PORT HERE
;  USE ANY REGISTERS YOU LIKE, AND UPON EXIT, SET THE A REG
;  TO OFFH IF A CHARACTER IS WAITING, OR TO 0 IF NOT
RET

INDAT:                 ;INPUT DATA (EXIT W/ CHARS. IN A REG)
;
;  ADD A ROUTINE TO GET THE DATA AT THE INPUT PORT HERE
;  RETURN THE CHARACTER IN THE A REGISTER (DO NOT CHANGE THE
;  CHARACTER IN ANY WAY!
RET

OUTSTA:                ;OUTPUT STATUS (A=FF IF RDY FOR CHAR, 0 IF NOT)
;
;  ADD A ROUTINE TO READ THE STATUS OF THE OUTPUT PORT HERE.
;  USE ANY REGISTERS YOU LIKE, AND, UPON EXIT, SET THE A REG
;  TO OFFH IF THE DATA PORT IS READY TO SEND DATA, OR TO 0 IF
;  NOT.
RET

OUTDAT:                ;OUTPUT DATA (CHAR IN A ON ENTRY)
;
;  ADD A ROUTINE TO OUTPUT THE CHARACTER IN THE A REGISTER TO
;  THE DATA PORT.
RET
```

```
DIAL:                ;SUPPORT AUTO-DIAL MODEMS

;   UPON ENTRY TO THIS ROUTINE, THE B REGISTER CONTAINS A 0
;   FOR DIALING, A 1 TO HANGUP, OR A 2 TO ANSWER THE PHONE.
;   IF THE B-REG. CONTAINS A 0, THE HL PAIR POINTS TO THE
;   PHONE NUMBER STRING

RET

END
```

Appendix B

Configuration Information for Move-It

Since the ease with which one configures Move-It depends largely on whether the information on the serial ports can be obtained, we have compiled the following list. This information comes from our users, dealers, manufacturer's literature, and our own in-house experience. If your machine is listed here, try the values suggested. Since, in a number of cases, the port numbers, etc. are dependent on jumpers set within the equipment, **Woolf Software Systems assumes no responsibility for errors or omissions in the list.** If you can't find your computer in the list, give us a call. We are adding new machines all the time, and we may have recently added yours.

The columns in the table refer to the various questions in the CONFIG program. These are:

- | | |
|--------------------|--|
| 1 - Data | This is the data port used by the I/O card. |
| 2 - Status | This is the status port used by the I/O card. |
| 3 - Xmit | This is the transmit buffer empty mask. |
| 4 - Recv | This is the receive buffer full mask. |
| 5 - Pol. | This is the polarity of the Xmit and Recv masks. A 1 means the ports are ready when the bits are a 1, and a 0 means the ports are ready when the bits are 0. |
| 6 - Init.
bytes | These are the initialization bytes required by the I/O board. Enter the bytes as pairs of numbers (AA,05 then press return, then 40,05 then press return, etc.). In the 16-bit versions of Move-It, the second number can be up to 4 digits. |

If you are able to successfully configure Move-It and we don't currently have information about your machine, let us know the particulars and we'll send you \$10.00 for your trouble.

Altair 88-2SIO

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	11	10	02	01	1	3,10 11,10

Altair SIO A,B,C (other than revision 0)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	01	00	80	01	0	None

Altair SIO A,B,C (revision 0)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	01	00	02	20	1	None

Altos® Series 5

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Printer Port	1E	1F	04	01	1	18,1F 04,1F 44,1F 03,1F C1,1F 05,1F EA,1F 00,1F

Altos Series 8

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Alternate Printer Port	28	29	04	01	1	18,29 04,29 44,29 03,29 C1,29 05,29 EA,29 00,29
Printer Port	1E	1F	04	01	1	18,1F 04,1F 44,1F 03,1F C1,1F 05,1F EA,1F 00,1F

AVL Eagle® I, II, III, IV, V

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Serial Port A	18	19	04	01	1	None
Serial Port A	1A	1B	04	01	1	None

Note: Use the ASSIGN utility provided with the Eagle to set up the baud rate.

California Computer Systems 2810 CPU

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	20	25	20	01	1	0F,24 83,23 80,20 01,21 03,23 00,21 01,25

Note: The init. bytes set up the port for 300 baud, 8 bits, no parity, and 1 stop bit.

California Computer Systems 2719

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	54	55	04	01	1	18,55 04,55 44,55 03,55 C1,55 05,55 EA,55 00,55 47,50 0C,50 **
Port B	56	57	04	01	1	18,57 04,57 44,57 03,57 C1,57 05,57 EA,57 00,57 47,51 0C,51 **

Note: The starred items in the last init. bytes line specify the baud rate for the port. The values specified (47,0C) will set the port to 9600 baud. To set the port to 300 baud, use 07 and 34.

Configuration Information**Move-It User's Manual****California Computer Systems 2820**

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	1C	1D	04	01	1	18,1D 04,1D 44,1D 03,1D C1,1D 05,1D EA,1D 00,1D
Port B	1E	1F	04	01	1	18,1F 04,1F 44,1F 03,1F C1,1F 05,1F EA,1F 00,1F

Commodore Pet™ (with Softbox)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Serial port	8	9	01	02	1	Use "Newsys" to set the port to 8 data bits, 1 stop bit and no parity.

Cromemco® TU-ART™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Device A	21	20	80	40	1	09,22 00,23 84,20 09,52 00,53 84,50
Device B	51	50	80	40	1	09,22 00,23 84,20 09,52 00,53 84,50

Note: The last two bytes in each line of the initialization group (84,XX) will set up the ports to 300 baud. For 9600 baud, use C0,20 and C0,50. Consult the manufacturer's documentation for other baud rates.

Delta Products CPU

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Serial Port 1	00	01	01	02	1	AA,01 40,01 4E,01 37,01
Serial Port 2	02	03	01	02	1	AA,03 40,03 4E,03 37,03

Note: The init. bytes specified will set the port to 8 bits, no parity, and one stop bit. Use the on-cpu dip switches to set the baud rate for the two ports.

Godbout® Interfacer I and Interfacer II

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	00	01	01	02	1	None
Port B	02	03	01	02	1	None

Note: Be sure to set the jumpers on the board to 8 bits, no parity, and 1 stop bit.

Godbout Interfacer III

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Rel. user 0	18	19	01	02	1	00,1F CE,1A 7E,1A 27,1B
Rel. user 1	18	19	01	02	1	01,1F CE,1A 7E,1A 27,1B
Rel. user 2	18	19	01	02	1	02,1F CE,1A 7E,1A 27,1B
Rel. user 3	18	19	01	02	1	03,1F CE,1A 7E,1A 27,1B
Rel. user 4	18	19	01	02	1	04,1F CE,1A 7E,1A 27,1B

Configuration Information**Move-It User's Manual**

Rel. user 5	18	19	01	02	1	05,1F CE,1A 7E,1A 27,1B
Rel. user 6	18	19	01	02	1	06,1F CE,1A 7E,1A 27,1B
Rel. user 7	18	19	01	02	1	07,1F CE,1A 7E,1A 27,1B

Note: The initialization will set the port(s) to 9600 baud, 8 data bits, no parity and one stop bit.

Godbout System Support I

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	5C	5D	01	02	1	4E,5E 75,5E 05,5F

Note: The init. bytes will set the port to 300 baud, 8 bits, no parity, and 1 stop bit. Also, be sure to set the jumpers at J2.

Gnat System 10

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	72	73	04	01	1	None

Hayes® MicroModem 100

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	80	81	02	01	1	None

Note: Use the on-board switches to set the rate and stop bits for this board.

Delta Products CPU

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Serial Port 1	00	01	01	02	1	AA,01 40,01 4E,01 37,01
Serial Port 2	02	03	01	02	1	AA,03 40,03 4E,03 37,03

Note: The init. bytes specified will set the port to 8 bits, no parity, and one stop bit. Use the on-cpu dip switches to set the baud rate for the two ports.

Godbout® Interfacer I and Interfacer II

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	00	01	01	02	1	None
Port B	02	03	01	02	1	None

Note: Be sure to set the jumpers on the board to 8 bits, no parity, and 1 stop bit.

Godbout Interfacer III

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Rel. user 0	18	19	01	02	1	00,1F CE,1A 7E,1A 27,1B
Rel. user 1	18	19	01	02	1	01,1F CE,1A 7E,1A 27,1B
Rel. user 2	18	19	01	02	1	02,1F CE,1A 7E,1A 27,1B
Rel. user 3	18	19	01	02	1	03,1F CE,1A 7E,1A 27,1B
Rel. user 4	18	19	01	02	1	04,1F CE,1A 7E,1A 27,1B

Configuration Information

Move-It User's Manual

Rel. user 5	18	19	01	02	1	05,1F CE,1A 7E,1A 27,1B
Rel. user 6	18	19	01	02	1	06,1F CE,1A 7E,1A 27,1B
Rel. user 7	18	19	01	02	1	07,1F CE,1A 7E,1A 27,1B

Note: The initialization will set the port(s) to 9600 baud, 8 data bits, no parity and one stop bit.

Godbout System Support I

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	5C	5D	01	02	1	4E,5E 75,5E 05,5F

Note: The init. bytes will set the port to 300 baud, 8 bits, no parity, and 1 stop bit. Also, be sure to set the jumpers at J2.

Gnat System 10

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	72	73	04	01	1	None

Hayes® MicroModem 100

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	80	81	02	01	1	None

Note: Use the on-board switches to set the rate and stop bits for this board.

Heath/Zenith Z89

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>	
DTE Serial Port	D8	DD	20	01	1	83,DB 03,DC 0C,D8 00,D9 03,DB 00,D9	**
DCE Serial Port (Printer port)	E0	E5	20	01	1	83,E3 03,E4 0C,E0 00,E1 03,E3 00,E1	**

** The sequences 0C,XX, 00,XX will set the bit rate to 9600 baud. To set the rate to 300 baud, use the sequence 80,XX, 01,XX. To set the rate to 1200 baud, use the sequence 60,XX, 00,XX.

IMS® 440 I/O Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>	
Device 0	11	10	02	01	1	36,17 0D,14 00,14 19,10	**
Device 1	13	12	02	01	1	76,17 0D,15 00,15 19,12	**

Note: The starred items in the 2nd init. line specify the baud rate for the port. The values specified (0D,1X 00,1X) will set the port to 9600 baud. To set the port to 300 baud, use A0,1X and 01,1X. For the new (1982) boards, the proper values are: 9600 - 08,1X 00,1X; 300 - 00,1X 01,1X.

Configuration Information**Move-It User's Manual****IMS 480 I/O Board**

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>	
Device 0	20	25	20	01	1	80,23 08,20 00,21 03,23	**
Device 1	28	2D	20	01	1	80,2B 08,28 00,29 03,2B	**
Device 2	30	35	20	01	1	80,33 08,30 00,31 03,33	**
Device 3	38	3D	20	01	1	80,3B 08,38 00,39 03,3B	**

Note: The starred items in the 2nd init. bytes line specify the baud rate for the port. The values specified (08,00) will set the port to 9600 baud. To set the port to 300 baud, use 00 and 01.

Imsai® MIO

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	42	43	01	02	1	00,43

Imsai SIO2

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	02	03	01	02	1	AA,03 40,03 CE,03 27,03

Kaypro II™ (Kaycomp II)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Communications Port	04	06	04	01	1	18,06 04,06 44,06 03,06 C1,06 05,06 EA,06 00,06

Note: Use the BAUD program to set the baud rate for the Kaypro's serial port.

Konan Octaplus Multi-port I/O Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port 1	00	01	01	02	1	AA,01 40,01 4E,01 37,01
Port 2	02	03	01	02	1	AA,03 40,03 4E,03 37,03
Port 3	04	05	01	02	1	AA,05 40,05 4E,05 37,05
Port 4	06	07	01	02	1	AA,07 40,07 4E,07 37,07
Port 5	08	09	01	02	1	AA,09 40,09 4E,09 37,09
Port 6	0A	0B	01	02	1	AA,0B 40,0B 4E,0B 37,0B
Port 7	0C	0D	01	02	1	AA,0D 40,0D 4E,0D 37,0D
Port 8	0E	0F	01	02	1	AA,0F 40,0F 4E,0F 37,0F

Note: The init. bytes specified will set up the port for 8 data bits, no parity, and one stop bit. Use the on-board jumpers to set the baud rate.

L & R Meteor

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Printer Port	02	03	04	01	1	None

Note: Use the switches on the processor board to set the baud rate.

Configuration Information**Move-It User's Manual****Monroe® OC 8820**

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Comm. Port	A4	A5	04	01	1	18,A5 04,A5 44,A5 03,A5 C1,A5 05,A5 EA,A5 00,A5 45,A8 0A,A8 **

**The last two pairs of numbers specify 9600 baud. For 1200 baud, use 45,A8 50,A8.

Morrow® Micro-Decision™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port 1	FC	FD	01	02	1	None
Port 2	FE	FF	01	02	1	None

Morrow Switchboard™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port 1	28	2A	08	04	1	None
Port 2	29	2A	80	40	1	None

NorthStar® Advantage

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Boardlet 1	50	51	01	02	1	AA,51 40,51 4E,51 37,51 7E,58

Note: The init. bytes specified will set up the port for 8 data bits, no parity, and one stop bit. The last two bytes (7E,58) set the baud-rate of 9600 baud. You can change this rate by using the setup program provided by Northstar.

Northstar Horizon

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port 0	02	03	01	02	1	None
Port 1	04	05	01	02	1	None

Northstar HSIO-4 Board (Base address = 10H)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	12	13	01	02	1	03,13 03,13 40,13 CE,13 27,13 06,10
Port B	16	17	01	02	1	03,17 03,17 40,17 CE,17 27,17 06,14
Port C	1A	1B	01	02	1	03,1B 03,1B 40,1B CE,1B 27,1B 06,18
Port D	1E	1F	01	02	1	03,1F 03,1F 40,1F CE,1F 27,1F 06,1C

Note: The last two bytes in the initialization sequence (06,XX) set the rate to 300 baud. For 9600 baud, use 01,XX.

Otrona® Attache™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	F0	F1	04	01	1	18,F1 04,F1 44,F1 03,F1 C1,F1 05,F1 EA,F1 00,F1

Problem Solver Systems CPU

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	13	12	02	01	1	0D,15 00,15

Processor Technology 3P+S Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	01	00	80	40	1	None

PMMI MM-103 Modem Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	C1	C0	01	02	1	None

Radio Shack® Model I and III

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	EB	EA	40	80	1	None

Note: Use the switches to set the rates, etc. on the Model I. On the Model III, use SETCOM. Set the interface to 8 data bits, one stop bit, and no parity.

Radio Shack Model II (with Pickles & Trout CP/M)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	F4	F6	04	01	1	None **
Port B	F5	F7	04	01	1	None **

**Use the SETUP utility to set the parameters for this computer. Be sure to specify 8 bits, no parity, and set the xmit on and xmit off characters to 0.

Seattle Computer Multi-port Card (Base port = 0)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Channel 0	0	1	01	02	1	AA,01 40,01 4E,01 37,01 **,08
Channel 1	2	3	01	02	1	AA,03 40,03 4E,03 37,03 **,09
Channel 2	4	5	01	02	1	AA,05 40,05 4E,05 37,05 **,0A
Channel 3	6	7	01	02	1	AA,07 40,07 4E,07 37,07 **,0B

Note: The ** in the second init. line for each port should be replaced by either 05 for 300 baud or 0E for 9600 baud. Consult you manual for other rates.

S D Systems SBC-200

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	7C	7D	01	02	1	AA,7D 40,7D 4E,7D 37,7D

Note: Since the SBC-200 has only one serial port, it is necessary to start Move-It in NOConsole mode and then plug in the communications cable instead of the terminal. The transfer is then controlled from the remote end.

SSM I/O 4 Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Serial Port 1	21	20	04	08	1	None
Serial Port 2	23	22	04	08	1	None

Note: The IO/4 Board can be set up for almost any I/O configuration. See the manufacturer's documentation for more information.

Configuration Information**Move-It User's Manual****Stratos**

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Modem port	6C	6D	01	02	1	AA,6D 40,6D 4E,6D 37,6D 55,65

Note: The last two bytes in the initialization (55,65) set the Modem port to 300 baud. For 1200 baud, use 77,65. For 9600 baud, use EE,65.

SuperBrain

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Main port	58	59	01	02	1	None
Aux port	40	41	01	02	1	None

Tarbell 3033 CPU-I/O Board

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	00	01	01	02	1	AA,01 40,01 4E,01 27,01
Port B	02	03	01	02	1	AA,03 40,03 4E,03 27,03

Televideo® 800, 802, 802H

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Modem Port	20	22	04	01	1	47,08 80,08
Printer Port	21	23	04	01	1	47,09 80,09

Note: The initialization specified will set the rate to 300 baud. For 9600 baud, use 47,0X 04,0X.

Televideo 801

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Right Serial Port	0C	0E	04	01	1	47,08 80,08

Note: The initialization specified will set the rate to 300 baud. For 9600 baud, use 47,08 04,08.

Televideo 806

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
RS-232 (OPT)	50	52	04	01	1	47,80 04,80
PRINTER	51	53	04	01	1	47,81 04,81

Note: Be sure to disable the service processor (MMOST). The init. bytes set the port to 9600 baud.

Vector® Bitstreamer™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	02	03	01	02	1	None

Vector BitStreamr II™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Port A	02	03	01	02	1	AA,03 40,03 4E,03 37,03
Port B	04	05	01	02	1	AA,03 40,05 4E,05 37,05
Port C	06	07	01	02	1	AA,03 40,07 4E,07 37,07

Vector VIP/2600

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	04	05	01	02	1	AA,05 40,05 4E,05 37,05

Note: The init. bytes specified will set up the port for 8 data bits, no parity, and one stop bit. Use the on-board dip switch to set the baud rate.

Vector Graphic ZCB

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	04	05	01	02	1	AA,05 40,05 4E,05 37,05

Wave Mate "Bullet"™

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
Channel A	00	01	04	01	1	None
Channel B	02	03	04	01	1	None

Note: Use the software supplied with the Bullet to set the baud rate, and to set the configuration to 8 data bits, 1 stop bit and no parity.

Xerox® 820

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>	
Communications Port	04	06	04	01	1	05,00 18,06 04,06 44,06 03,06 C1,06 05,06 EA,06 00,06	**
Printer Port	05	07	04	01	1	05,0C 18,07 04,07 44,07 03,07 C1,07 05,07 EA,07 00,07	**

** In each case, the first pair of numbers (05,XX) is the initialization for 300 baud. To set the port for 9600 baud, use 0E,XX.

I/O Chip Definitions

The following information pertains to those machines for which the I/O configuration information is not listed. For these machines, it is sometimes possible to obtain the port numbers and I/O chip types from the manufacturer. If these can be found, the rest of the information can be found in the following table. Since we have no idea which of the possible 256 I/O ports your machine uses, the designation DD refers to the "data port" number, and SS is the "status port" number. These can be obtained from the manufacturer.

8251 I/O Chip

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	DD	SS	01	02	1	AA,SS 40,SS 4E,SS 37,SS

Z80®SIO Chip (also Z80 DART)

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	DD	SS	04	01	1	18,SS 04,SS 44,SS 03,SS C1,SS 05,SS EA,SS 00,SS

Signetics 2651/2661

<i>I/O Port</i>	<i>Data</i>	<i>Status</i>	<i>Xmit</i>	<i>Recv</i>	<i>Pol.</i>	<i>Init. Bytes</i>
	DD	SS	01	02	1	4E,DD+2 3X,DD+2 27,DD+3

Note: In the init. bytes, DD+2 means "the data port plus 2". Therefore, if the data port was 4A, DD+2 would be 4C. The 3X in the next to last init. line depends on the baud rate chosen. For example, 300 baud would be 35 and 9600 baud would be 3E. Also, the 2661-1 is the same as the 2651, except the code for 300 baud is 36 instead of 35.

NOTES

NOTES



Appendix C

Glossary of Computer Terms

Alphanumeric Set of characters consisting of either the letters A through Z, the digits 0 through 9, special symbols, or a combination of all of these. For example, the set of characters, MISC. #2, is alphanumeric.

ASCII American Standard Code for Information Interchange. This is a code in which all the printable characters, as well as others, are represented by a 7 bit code. Also, in Move-It, Ascii mode is a seven-bit transmission mode.

Applications Program Program that performs a specific function, such as communications or word processing. Move-It is an example of an applications program.

Asynchronous Method of transmission in which the time interval between characters may be of unequal length. RS-232 is an asynchronous mode of transmission.

Autorepeat Automatic repetition of a character key. Usually done by either holding the key down, or by pressing the REPEAT key.

Back-up Stored data files which can be used if the original files are lost or incorrect.

BASIC BASIC is an acronym that stands for Beginners All-purpose Symbolic Instruction Code. BASIC is the computer programming language used with user-oriented computers.

Baud Measure of signaling speed, as in 300 baud.

Bit Smallest unit of information normally used by computers.

Byte Group of binary digits (bits) that the computer reads as a unit. A byte usually consists of eight or sixteen bits.

Carrier Tone sent by modems to allow communication between two computers.

Character One of a set of elements that may be arranged as groups to express information. Characters include the letters A through Z the digits 0 through 9, punctuation marks, and other special symbols.

CRT Stands for Cathode Ray Tube. This is the TV screen or video monitor which displays your computer information.

Cursor The small, sometimes flashing symbol that indicates the typing position on the video monitor.

Daisy-Wheel Printer Printer that uses a spinning element as a typeface. The results imitate a hand-typed letter, but these printers are slower than dot matrix printers.

Data Information in the form of letters, numbers, or symbols that the computer processes or produces.

Default Value Value assigned automatically by the computer when the operator does not enter any information before pressing the RETURN key.

Disk Cartridge Large volume data storage device that consists of a rigid platter with a magnetic coating. Sometimes one or more of these platters are enclosed in a plastic cartridge. A disk cartridge is also called a hard disk.

Diskette Data storage device that is coated with magnetic recording material and looks like a phonograph record sealed in a plastic envelope. The diskette and plastic envelope are inserted into the disk drive together. A diskette is also called a floppy disk or a flexible disk.

Disk Drive Electronic machine and housing used to store and retrieve information on a disk cartridge or diskette. When several disk units are used simultaneously, each one is assigned a number.

Field Single item of information. For example, a field could be a debtor name or number.

File Group or related records on the diskette. Each file has a name by which it is accessed.

FORMAT FORMAT is a system related program used to prepare new diskettes for use or an old diskette for re-use. Formatting DESTROYS all data on the diskette. For more information on formatting, consult your computer system manual.

Full-duplex Mode of transmission where both computers can communicate together, at the same time.

Half-duplex Mode of transmission where both computers can communicate together, but not at the same time. CB radio is a half-duplex scheme.

Hard Copy Computer output that is printed on paper.

Hard Disk Has a very large storage capacity and rapid access time. More expensive than floppy disks.

Hardware Actual equipment that makes up a computer system, such as the terminal, the video screen, the computer, the printer, etc.

Input Information that is entered into a computer from outside via the keyboard.

I/O Abbreviation of input/output.

Memory Capacity for storage of programs and data being used by the computer.

Menu Computer video display of options available to the operator.

Modem Electronic device that enables two computers, or a computer and a terminal, to communicate over phone lines.

Pixel Dot on the video screen. Different combinations of illuminated pixels form characters or graphics.

Processing Programmed computations and actions by which the computer uses input data to produce output data.

Program Set of sequential instructions that direct the computer in performing desired operations. Programs are collectively known as "software".

PROM Stands for Programmable Read-Only Memory.

Prompt Computer message on the video unit that indicates that the operator should type in a response. For example, A> is a prompt.

Protocol Formal set of rules that enable two computers to communicate. The protocol governs timing, message length and other parameters.

RAM Random Access Memory.

Record Collected information fields that make up a file.

ROM Read-Only Memory.

Software Software portion of a computer system includes programs, procedures and computer languages.

UART The abbreviation for Universal Asynchronous Receiver-Transmitter. The UART that enables the RS-232 port to work in most systems.

Appendix D ASCII Code Chart

- Table D-1. ASCII Code Chart

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
00	^@ (Null)	20	space	40	@	60	`
01	^A	21	!	41	A	61	a
02	^B	22	”	42	B	62	b
03	^C	23	#	43	C	63	c
04	^D	24	\$	44	D	64	d
05	^E	25	%	45	E	65	e
06	^F	26	&	46	F	66	f
07	^G (bell)	27	'	47	G	67	g
08	^H (bcksp)	28	(48	H	68	h
09	^I (tab)	29)	49	I	69	i
0A	^J (lnfd)	2A	*	4A	J	6A	j
0B	^K	2B	+	4B	K	6B	k
0C	^L	2C	,	4C	L	6C	l
0D	^M	2D	-	4D	M	6D	m
0E	^N	2E	.	4E	N	6E	n
0F	^O	2F	/	4F	O	6F	o
10	^P	30	0	50	P	70	p
11	^Q	31	1	51	Q	71	q
12	^R	32	2	52	R	72	r
13	^S	33	3	53	S	73	s
14	^T	34	4	54	T	74	t
15	^U	35	5	55	U	75	u
16	^V	36	6	56	V	76	v
17	^W	37	7	57	W	77	w
18	^X	38	8	58	X	78	x
19	^Y	39	9	59	Y	79	y
1A	^Z	3A	:	5A	Z	7A	z
1B	^[(esc)	3B	;	5B	[7B	[
1C	^\	3C	<	5C	\	7C	
1D]`	3D	=	5D]	7D]
1E	^_	3E	>	5E	^	7E	~
1F	^-	3F	?	5F	-	7F	delete

The ^ symbol indicates a control character. For example ^C is a Control-C, but hex 1E = ^^, or Control ^.

The standard ASCII character set does not define characters higher than 7F. Some terminals display such characters as special Graphic Characters, but there is no standard set of graphic characters.

Software Problem Report

At WOOLF SOFTWARE SYSTEMS, we are committed to providing the finest in computer programs for the business community. As a member of that community, you can help in our effort by reporting any problems or suggestions you may have on the form provided below. Please use only one form per problem/suggestion. Return form to: WOOLF SOFTWARE SYSTEMS, 23842 Archwood St. Canoga Park, CA 91307. Also, you can reach us by phone at (213) 703-8112, 10AM to 5PM Monday through Friday.

Date _____ Program _____

Name _____ Phone number _____

Organization _____

Address _____ City _____ St. _____ Zip _____

Serial No. _____ Please describe your hardware and operating system:

Version Number _____

In the space provided below, please provide a concise description of the problem and circumstances surrounding its occurrence. If possible, reduce the problem to a simple test case. Use extra sheets, if needed.
