WANG

2200

BASIC-2 Multiuser Operating System Software Bulletin

Release 2.6

2200
BASIC-2
Multiuser Operating System
Software Bulletin
Release 2.6

2nd Edition — May 1985

715-0097A

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL. (617) 459-5000, TWX 710-343-6769, Telex 94-7421

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

The staff of Wang Laboratories, Inc., has taken due care in preparing this manual. However, nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreement by which the product was acquired, nor increases in any way Wang's liability to the customer. In no event shall Wang or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the product, the accompanying manual, or any related materials.

SOFTWARE NOTICE

All Wang Program Products (software) are licensed to customers in accordance with the terms and conditions of the Wang Standard Software License. No title or ownership of Wang software is transferred, and any use of the software beyond the terms of the aforesaid license, without the written authorization of Wang, is prohibited.

WARNING

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device, pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

PREFACE

The purpose of this Software Bulletin is to provide up-to-date documentation for software enhancements to the 2200 Series. Chapter 1 describes the changes and features of Release 2.6 of the 2200 BASIC-2 Multiuser Operating System. Chapter 2 summarizes changes in previous releases of the operating system (Releases 2.2 through 2.5). Chapter 3 describes the Initialize Date & Time utility.

This documentation is intended to be used in conjunction with the following manuals:

- BASIC-2 Utilities Reference Manual (700-6855)
- Wang BASIC-2 Language Reference Manual (700-4080)
- Wang BASIC-2 Disk Reference Manual (700-4081)

CONTENTS

FIGURES

TABLES

CHAPTER ONE
RELEASE 2.6

1.1 INTRODUCTION

Release 2.6 of the BASIC-2 Multiuser Operating System is being issued primarily to provide a new generalized printer driver. This release also includes the following revisions:

- The LIST DCT command now supports the use of wildcard characters.
- Enhancements have been made to the 2236MXE terminal controller microcode.
- The I94 error message has an additional meaning.

1.2 RELEASE INFORMATION

Release 2.6 of the BASIC-2 Multiuser Operating System is available on the following media:

Part Number	Size	Type	Comment
195-0049-3	8 inch	SSSD	2200 MVP, MVPC, LVP, LVPC (2270A Diskette Drive)
195-0049-5	8 inch	DSDD	2200 LVP, LVPC, SVP (DSDD Diskette Drive)
195-0049-9	5 1/4 inch	DSDD	All of the above (2275 Diskette Drive), all MVP-P packages

1.2.1 Hardware Requirements

It is important to observe the memory requirement for Release 2.6. Release 2.6 requires a minimum of 28K of control memory.

1.2.2 Contents of New System Diskettes

If you use a double-sided, double-density diskette, the system files for Release 2.6 are on one diskette. If you use a single-sided, single-density diskette, the system files for Release 2.6 are on on two diskettes. Diskette 1 contains the following files:

```
@@
@SYSMVPB
@DAVFU
@MXEO  (modified for Release 2.6)
@BACKUP
@INSTALL (modified for Release 2.6)
@BOOT
.STARTD
@PSTAT
@MOVEFIL  (modified for Release 2.6)
@RECOVER
@GENPART  (modified for Release 2.6)
@MOVE1
START
@MVP  (modified for Release 2.6)
@FORMAT
@DG
@CLOC
@MENU
@MODSYSF  (added for Release 2.6)
@SYSFILE  (modified for Release 2.6)
@DATE
@PM010V0  (added for Release 2.6)
@PM016V0  (added for Release 2.6)
```

Diskette 2 contains the following files:

```
@P
@A
@B
@C
@D
@\
@E
@MRTIAN
```

1.3 NEW GENERALIZED PRINTER DRIVER

The 2200 Generalized Printer Driver (GPD) is a table-driven driver that customizes print output to the attached printer. The GPD allows the 2200 system to use two Wang PC parallel printers as either terminal or system printers. The printers for which Release 2.6 provides translation tables are the PM010 and PM016. However, PC printers do not support the 2200 BASIC-2 $GIO statement. The GPD has no effect on the printers presently used on 2200 systems. The GPD also provides two transparent modes of operation, one permanent and one temporary.

The driver intercepts each byte of data from the application and determines whether the byte represents a printable character, a control code, or part of an escape sequence. Escape sequences and control codes are translated and then sent to the printer or used to set driver modes. Printable characters are translated, modes and attributes are checked, and then the modified character code is sent to the printer.

There are three cases in which the driver is not used. In transparent mode, in doing screen dumps to the terminal printer, and in any $GIO printer calls, all data bypasses the driver and goes directly to the printer. Thus, applications that use $GIO to print on printers will experience problems. For example, documents printed through 2200 WP on Wang PC printers will be printed without line feeds. If an application wishes to use one of the printers supported by the GPD, output should be done using PRINT statements and the escape sequences provided by the GPD.

The Generalized Printer Driver includes the following enhancements:

- a new @MODSYSF utility
- a modified @GENPART utility
- an enhanced $INIT statement
- modified X78 and X79 error messages
- an enhanced LIST DT command
- a new SELECT DRIVER statement

1.3.1 Installing the GPD

The following sections discuss installing and initializing the driver. All users should read "Using the @GENPART Utility" and "New @GENPART Error Messages" in this section. If you are updating from an earlier software release, you should read "@MODSYSF Utility". If you write your own @GENPART utility, you should read "Using the $INIT Statement".

@MODSYSF Utility

When updating from an earlier software release, you must execute @MODSYSF before installing Release 2.6. @MODSYSF works with the old version of @SYSFILE and is run only once.

The existing @SYSFILE needs to be expanded to receive the additional information necessary for the GPD. @MODSYSF loads @SYSFILE into memory, stores the information contained in @SYSFILE, and writes the information back to the disk in Release 2.6 format.

When you have run @MODSYSF, you are ready to reinitialize the system and use @GENPART to enter the new driver information. However, if necessary, @GENPART will run @MODSYSF.

Using the @GENPART Utility

In order to use the GPD, you must first tell the system what printer driver tables you want to use with what printers. You do this while configuring the system using the @GENPART utility.

Procedure

1. From the @GENPART menu, press SF7. The Printer Driver Screen (Figure 1-1) appears on the monitor.

```
***WANG 2200MVP PARTITION GENERATION PROGRAM***

                                                    LIST OF OPTIONS:
   Driver Table Name  Printer Address  Term #     SF'00 - clear partition
                                                 SF'01 - clear device table
 1. @PM016V0          215
 2. @PM010V0          004              1         SF'02 - divide mem. evenly
 3.
 4.                                              SF'04 - edit partitions
 5.                                              SF'05 - edit device table
 6.                                              SF'06 - edit $MSG
 7.
 8.                                              SF'07 - select prt. driver
 9.                                              SF'08 - load configuration
10.                                              SF'09 - save configuration
11.                                              SF'10 - delete config.
12.
13.
14.
15.                                              SF'15 - execute

Edit which driver table entry (default = 1)?_
```

Figure 1-1. Printer Driver Screen

2. Enter a number from 1 to 15 following the "Edit which driver table entry" prompt. Enter 1 if no printer driver entries have yet been created.

3. Enter the name of a file containing translation tables for the GPD in response to the prompt:

   Enter table driver name (enter 0 to delete from configuration):

   Table 1-1 contains a list of valid translation table names with their associated printers:

Table 1-1. Printer Driver Translation Tables

Tables	Printers
@PM010V0	Wang PC-PM010
@PM016V0	Wang PC-PM016

4. Enter the address of the printer for the table in response to the prompt:

   Enter associated printer address: _____

   The address consists of three digits. The first digit, either 0 or 2 depending on how the device was declared in @GENPART, represents the type of printer. With Type 0 devices, the operating system automatically supplies a line feed after each carriage return. Since the GPD also supplies line feeds, you must select Type 2 for printers. When the GPD is deselected, you must select Type 0 for printers.

   The second and third digits are the printer's address. For terminal printers, the address is 04.

5. If the address entered is 04, @GENPART prompts you for the number of the terminal.

   Enter terminal number (between 1 and 13): ___

6. Your responses to these prompts appear on the screen. An example of a valid set of responses follows.

	Driver Table Name	Printer Address	Term #
1.	@PM010V0	204	12
2.	@PM016V0	216	
3.	@PM016V0	217	
4.			
"			
"			

7. @GENPART then repeats the prompts listed in Steps 2 through 5, allowing you to associate tables with up to 15 printers.

   Since no more than seven different driver tables can be in memory at one time, you can repeat table names. However, a printer address other than 04 can be used only once.

8. To delete an entry from the GPD list, enter zero in response to the "Driver Table Name" prompt and press the RETURN key.

9. When you have specified drivers for all the printers, you can select any of the functions in the LIST OF OPTIONS.

New @GENPART Error Messages

When @GENPART is initializing the GPD, any of the following messages may appear on the screen:

Maximum Number of Drivers have already been chosen -- please enter 'RETURN'

This message appears when you specify more than seven different translation tables in @GENPART. To recover, change a table name.

Duplicate printer address -- enter another.

This message appears when the same printer address (other than 04) is used more than once in @GENPART. To recover, change or delete an address.

Table name indicated is not a driver table for the GPD. -- key 'RETURN' to check the list of driver table names.

This message appears when the table name given to the @GENPART utility is not the name of a file that is a translation table for the GPD. To recover, press the RETURN key. The table input display returns to the screen.

DRIVER Tables not on current platter -- mount correct platter and key 'RETURN'

This message appears when the platter currently mounted does not contain the tables specified by the @GENPART utility. To recover, mount the correct platter and press the RETURN key. The driver initialization then continues.

Driver Table error number ## was encountered initiating system -- correct the configuration or table file name

This message appears during initialization of the GPD when the system finds that a specified file is not a printer translation table. To recover, press the RETURN key. The processing returns to the table name prompt. Change the name of table entry ##.

Using the $INIT Statement

It is not recommended that you write your own @GENPART utility. However, if you do, use the $INIT statement to associate driver tables with printers. Refer to the 2200 BASIC-2 LANGUAGE REFERENCE MANUAL for a complete discussion of $INIT. To initialize the GPD, you assign the driver table file names and associated device addresses to an alpha variable string used as the seventh argument of the $INIT statement. Each entry in the alpha string must be 10 bytes long and the string itself must be 150 bytes long. Any unused bytes must be set to hex 20. The general form of the $INIT statement and the format for the entries in the seventh argument are as follows:

```
$INIT(alpha-1,alpha-2,alpha-3,alpha-4,alpha-5 [,[alpha-6][,alpha-7]])

    Where:

            alpha-1 = size of each partition
            alpha-2 = terminal number for each partition
            alpha-3 = mode of each partition
            alpha-4 = bootstrap program name for each partition
            alpha-5 = device table
            alpha-6 = optional reconfiguration password
            alpha-7 = optional printer driver associations
                      where:
                        bytes 0-7 = printer driver table file name
                        byte 8    = device address in hex
                        byte 9    = terminal number in hex, if
                                    byte 8 is 04; otherwise, 0
```

Two system error messages have been changed to support the enhancements to $INIT. If an X78 or X79 error occurs during execution of a $INIT statement, they have the following meanings. At all other times, their meanings are unchanged.

Error Code X78

This error code is returned when the system is asked to load more than seven printer driver tables or when not enough memory is available for the tables. To recover, reduce the number of tables specified. This code is also returned if you specify an invalid driver table name.

Error Code X79

This error code is returned when there is an attempt to associate more than 15 device addresses with printer driver tables or when an address associated with the printer driver tables is used more than once. To recover, change the incorrect address parameter. Note that since all terminals have a device address of 04, it is the terminal number that distinguishes between the terminals.

1.3.2 Using the GPD

The following sections explain GPD default settings, the new SELECT DRIVER statement and transparent mode, the GPD character set, control codes, and escape sequences, and an enhancement to the LIST DT command to support the GPD.

GPD Defaults

Unless you set them differently, the GPD uses the following default settings:

- 10 pitch
- 6 lines per inch
- no attributes enabled
- font #0 (if applicable)
- automatic line feed after carriage return enabled
- hex(0E) control code setting; emphasized print only, turn off when a hex(0F) is received

SELECT DRIVER Statement and Transparent Mode

In transparent mode, the GPD sends the codes it receives directly to the printer without interpreting them. You can select transparent mode permanently or temporarily. A new statement, SELECT DRIVER, allows you to establish transparent mode permanently. The general form of the SELECT DRIVER statement is as follows:

```
SELECT DRIVER device-address [OFF]

                                  /taa
    Where:      device-address =
                                  alpha-variable
```

ON is the default state for the driver. OFF turns the driver off for the specified address until the system is reinitialized or until you issue another SELECT DRIVER statement for the same address without the OFF. The GPD need not be turned on for any standard 2200 printer, for example, the 2245, 2263, or 2273.

If you want to use a feature available on a given printer but not supported by the GPD, you can do so by using Escape Sequence number 28, Enter Temporary Transparent Mode. This sequence allows you to pass hex codes directly to the printer without having them translated by the driver. Refer to the "Standard Escape Sequences" section below.

Generalized Printer Driver Character Set

Table 1-2 lists the standard character set for the GPD.

NOTE

All printers may not support all hex codes. If a printer does not support a code, the driver directs the printer to build the character, wherever possible. Where this is not possible, a space is printed. Close approximations of the hex 60 and hex E0 characters are supported for the PM010 and PM016 printers.

Table 1-2. Standard Character Set

HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER
hex(00)	(*)	hex(20)	space	hex(40)	@	hex(60)	°
hex(01)	(*)	hex(21)	!	hex(41)	A	hex(61)	a
hex(02)	(*)	hex(22)	"	hex(42)	B	hex(62)	b
hex(03)	(*)	hex(23)	#	hex(43)	C	hex(63)	c
hex(04)	(*)	hex(24)	$	hex(44)	D	hex(64)	d
hex(05)	(*)	hex(25)	%	hex(45)	E	hex(65)	e
hex(06)	(*)	hex(26)	&	hex(46)	F	hex(66)	f
hex(07)	(*)	hex(27)	'	hex(47)	G	hex(67)	g
hex(08)	(*)	hex(28)	(	hex(48)	H	hex(68)	h
hex(09)	(*)	hex(29)	)	hex(49)	I	hex(69)	i
hex(0A)	(*)	hex(2A)	*	hex(4A)	J	hex(6A)	j
hex(0B)	(*)	hex(2B)	+	hex(4B)	K	hex(6A)	k
hex(0C)	(*)	hex(2C)	,	hex(4C)	L	hex(6A)	l
hex(0D)	(*)	hex(2D)	-	hex(4D)	M	hex(6D)	m
hex(0E)	(*)	hex(2E)	.	hex(4E)	N	hex(6D)	n
hex(0F)	(*)	hex(2F)	/	hex(4F)	O	hex(6F)	o
hex(10)	â	hex(30)	0	hex(50)	P	hex(70)	p
hex(11)	ê	hex(31)	1	hex(51)	Q	hex(71)	q
hex(12)	î	hex(32)	2	hex(52)	R	hex(72)	r
hex(13)	ô	hex(33)	3	hex(53)	S	hex(73)	s
hex(14)	û	hex(34)	4	hex(54)	T	hex(74)	t
hex(15)	ä	hex(35)	5	hex(55)	U	hex(75)	u
hex(16)	ë	hex(36)	6	hex(56)	V	hex(76)	v
hex(17)	ï	hex(37)	7	hex(57)	W	hex(77)	w
hex(18)	ö	hex(38)	8	hex(58)	X	hex(78)	x
hex(19)	ü	hex(39)	9	hex(58)	Y	hex(79)	y
hex(1A)	à	hex(3A)	:	hex(5A)	Z	hex(7A)	z
hex(1B)	è	hex(3B)	;	hex(5B)	[	hex(7B)	§
hex(1C)	ù	hex(3C)	<	hex(5C)	\	hex(7C)	£
hex(1D)	Ä	hex(3D)	=	hex(5D)	]	hex(7D)	é
hex(1E)	Ö	hex(3E)	>	hex(5E)	↑	hex(7E)	ç
hex(1F)	Ü	hex(3F)	?	hex(5F)	_	hex(7F)	(*)

(*) -- see control sequences

Table 1-2. Standard Character Set (continued)

HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER	HEX CODE	PRINTER CHARACTER
hex(80)	space	hex(A0)	space	hex(C0)	@	hex(E0)	°
hex(81)	null	hex(A1)	!	hex(C1)	A	hex(E1)	a
hex(81)	null	hex(A1)	"	hex(C1)	B	hex(E1)	b
hex(83)	null	hex(A3)	#	hex(C3)	C	hex(E3)	c
hex(84)	null	hex(A4)	$	hex(C4)	D	hex(E4)	d
hex(85)	null	hex(A5)	%	hex(C5)	E	hex(E5)	e
hex(86)	null	hex(A6)	&	hex(C6)	F	hex(E6)	f
hex(87)	null	hex(A7)	'	hex(C7)	G	hex(E7)	g
hex(88)	null	hex(A8)	(	hex(C8)	H	hex(E8)	h
hex(89)	null	hex(A9)	)	hex(C9)	I	hex(E9)	i
hex(8A)	null	hex(AA)	*	hex(CA)	J	hex(EA)	j
hex(8B)	null	hex(AB)	+	hex(CB)	K	hex(EA)	k
hex(8C)	null	hex(AC)	,	hex(CC)	L	hex(EA)	l
hex(8D)	null	hex(AD)	-	hex(CD)	M	hex(ED)	m
hex(8E)	null	hex(AE)	.	hex(CE)	N	hex(ED)	n
hex(8F)	null	hex(AF)	/	hex(CF)	O	hex(EF)	o
hex(90)	â	hex(B0)	0	hex(D0)	P	hex(F0)	p
hex(91)	ê	hex(B1)	1	hex(D1)	Q	hex(F1)	q
hex(92)	î	hex(B2)	2	hex(D2)	R	hex(F2)	r
hex(93)	ô	hex(B3)	3	hex(D3)	S	hex(F3)	s
hex(94)	û	hex(B4)	4	hex(D4)	T	hex(F4)	t
hex(95)	ä	hex(B5)	5	hex(D5)	U	hex(F5)	u
hex(96)	ë	hex(B6)	6	hex(D6)	V	hex(F6)	v
hex(97)	ï	hex(B7)	7	hex(D7)	W	hex(F7)	w
hex(98)	ö	hex(B8)	8	hex(D8)	X	hex(F8)	x
hex(99)	ü	hex(B9)	9	hex(D8)	Y	hex(F9)	y
hex(9A)	à	hex(BA)	:	hex(DA)	Z	hex(FA)	(*)
hex(9B)	è	hex(BB)	;	hex(DB)	[	hex(FB)	§
hex(9C)	ù	hex(BC)	<	hex(DC)	\	hex(FC)	£
hex(9D)	Ä	hex(BD)	=	hex(DD)	]	hex(FD)	é
hex(9E)	Ö	hex(BE)	>	hex(DE)	↑	hex(FE)	ç
hex(9F)	Ü	hex(BF)	?	hex(DF)	_	hex(FF)	¢

(*) -- see control sequences

Standard Control Sequences

Table 1-3 lists the GPD Standard Control Sequences and identifies the actions associated with them. The next section, Implementation of Control Sequences, explains these actions in detail.

Table 1-3. Standard Control Sequences

Hex Code	Action
00 - 01	Null
02	Initiate escape code sequence
03 - 06	Null
07	Bell
08	Nondestructive backspace
09	Horizontal tab
0A	Line feed
0B	Vertical tab
0C	Form feed
0D	Carriage return
0E	Definable, see "Standard Escape Sequences"
0F	Definable, see "Standard Escape Sequences"
7F	Clear buffer
FA	Reverse line feed

Implementation of Control Sequences

The system implements the standard control codes listed in Table 1-3 as follows. All codes are in hex.

00 - 01

NULL. These codes are filtered out by the driver and are not sent to the printer.

02

INITIATE ESCAPE CODE SEQUENCE. This byte is used to start all escape code sequences.

03 - 06

NULL. These codes are filtered out by the driver and are not sent to the printer.

07

BELL. If applicable, this code generates an audio alarm. The specific alarm sounded is printer dependent.

08

NONDESCTRUCTIVE BACKSPACE. This code generates a nondestructive backspace in the printer's buffer, or on paper.

09

HORIZONTAL TAB. This code causes the printer to move to the next tab stop. Tab stops are located at every fifth column. A tab is not generated if it causes the carriage to move beyond the set line length.

0A

LINE FEED. This code advances the paper one line when the contents of the printer buffer have printed. The size of the line feed increment is the currently selected (by Escape Code number 19) size. The specific implementation of this code is printer dependent.

0B

VERTICAL TAB. Vertical tabs are located every 6 lines from the top of form. The current contents of the buffer will be printed before the tab is executed. If a vertical tab crosses the end of form, a form feed is generated instead.

0C

FORM FEED. This code advances the paper to the next top of form after printing the contents of the printer's buffer.

0D

CARRIAGE RETURN. This code causes the immediate printing of the contents of the printer's buffer. The carriage will return to the left margin without the paper being moved.

0E - 0F

The function of Control 0E's and 0F's are defined in detail in the next section, Standard Escape Sequences.

7F

CLEAR BUFFER. This code causes the contents of the printer's buffer to be cleared. This code is not supported for the PM010 and PM016 printers.

FA

REVERSE LINE FEED. This code causes the paper to move backwards one line feed when the contents of the buffer have printed. This code is not supported for the PM010 and PM016 printers.

Standard Escape Sequences

Table 1-4 lists the GPD Standard Escape Sequences and identifies the actions associated with them. The next section, Escape Sequence Support by Printer, identifies the sequences that are supported by each of the printers driven by the GPD. The section following that, Implementation of Escape Sequences, explains in detail the actions associated with the escape sequences.

In Table 1-4, the letters cc indicate a byte count of the following bytes, excluding the final hex(0E) or hex(0F). The letters dd, ee, ff, gg indicate bytes with user-specified values. All escape sequence bytes are hex values.

Table 1-4. Standard Escape Sequences

Escape Sequence (hex)	Action
1. 02 01 02 01 00 0F	Execute built in diagnostics
2. 02 02 dd 0F	Select character font 2a. dd = 00 - font zero (default) 2b. = 02 - alternate font # 1 2c. = 04 - alternate font # 2
3. 02 03 0F	Clear platen (Daisy printers only)

Table 1-4. Standard Escape Sequences (continued)

Escape Sequence (hex)	Action
4. 02 04 dd ee ff gg	Define the meaning of Control hex(0E) dd = 00 - overstrike disabled = 02 - overstrike enabled ee = 00 - underscore disabled = 04 - underscore enabled ff = 00 - emphasized print disabled = 02 - emphasized print enabled gg = 0E - turn on attributes until reset or a hex(0F) is received = 0F - turn on attributes upon receipt of a Control hex(0E); turn off on a Control hex(0F), carriage return, or reset
5. 02 05 01 dd 0F	Horizontal tab to column dd dd = column number
6. 02 05 02 dd ee 0F	Horizontal tab dd inches and ee quarter inches dd = number of inches ee = number of quarter inches
7. 02 06 dd 0F	Select paper tray (Daisy printers only) 7a. dd = 01 - rear tray 7b. = 02 - front tray
8. 02 07 0F	Deselect printer
9. 02 08 01 0E	Enable superscript
10. 02 08 01 0F	Disable superscript
11. 02 08 02 0E	Enable subscript
12. 02 08 02 0F	Disable subscript
13. 02 08 03 0E	Enable underscore
14. 02 08 03 0F	Disable underscore
15. 02 08 04 0E	Enable overstrike
16. 02 08 04 0F	Disable overstrike

Table 1-4. Standard Escape Sequences (continued)

Escape Sequence (hex)	Action
17. 02 09 01 02 dd ee 0F	Set pitch dd ee = value of the integral and fractional values of the desired pitch respectively Supported pitches: 5, 10, 12, 15, 16.5
18. 02 0A 01 01 dd 0E	Set line feed spacing and enable auto line feed after carriage return dd = number of lines per inch; must be greater than 00 and not greater than 0F
19. 02 0A 01 01 dd 0F	Set line feed spacing and disable auto line feed after carriage return dd = number of lines per inch; must be greater than 00 and not greater than 0F
20. 02 0A 0E	Enable auto line feed
21. 02 0A 0F	Disable auto line feed
22. 02 0A dd 0F	Partial line feed 22a. dd = 00 - 1/4 line feed 22b. 02 - 1/2 line feed 22c. 04 - 3/4 line feed
23. 02 0B 01 dd 0F	Set left margin dd = column number of left margin
24. 02 0B 02 0E	Enable proportional spacing
25. 02 0B 02 0F	Disable Proportional Spacing
26. 02 0C 01 02 00 dd 0F	Set form length to dd lines dd = number of lines per form
27. 02 0C 02 dd 0F	Set form length to dd inches dd = number of inches per form
28. 02 0D 01 cc dd...dd	Enter temporary transparent mode cc = number of bytes to pass transparently dd...dd = the bytes to be passed
29. 02 0D 0C 03 0F	Restore power on defaults

Escape Sequence Support By Printer

This section lists the GPD escape sequences and specifies whether or not they are supported by the PM010 and PM016 printers. Table 1-5 lists the sequences for the PM010. Table 1-6 lists them for the PM016.

Table 1-5. Escape Sequences for the PM010

Escape Sequence	Supported?
1. Execute built-in diagnostics	n
2a. Select character font (default)	y
2b. Select character font (alternate #1)	n
2c. Select character font (alternate #2)	n
3. Clear platen	n
4. Define the meaning of Control hex(OE)	y
5. Horizontal tab to column	y
6. Horizontal tab to inches/quarter inches	y
7a. Select paper tray #1 (rear)	n
7b. Select paper tray #2 (front)	n
8. Deselect printer	n
9. Enable superscript	y
10. Disable superscript	y
11. Enable subscript	y
12. Disable subscript	y
13. Enable underscore	y
14. Disable underscore	y
15. Enable overstrike	y
16. Disable overstrike	y
17. Set pitch	y
18. Set line feed spacing and enable auto line feed	y
19. Set line feed spacing and disable auto line feed	y
20. Enable auto line feed	y
21. Disable auto line feed	y
22a. Partial line feed (1/4)	n
22b. Partial line feed (1/2)	n
22c. Partial line feed (3/4)	n
23. Set left margin	n
24. Enable proportional spacing	n
25. Disable proportional spacing	n
26. Set form length in lines	y
27. Set form length in inches	y
28. Enter temporary transparent mode	y
29. Restore power-on defaults	y

Table 1-6. Escape Sequences for the PM016

Escape Sequence	Supported?
1. Execute built-in diagnostics	n
2a. Select character font (default)	y
2b. Select character font (alternate #1)	n
2c. Select character font (alternate #2)	n
3. Clear platen	n
4. Define the meaning of Control hex(OE)	y
5. Horizontal tab to column	y
6. Horizontal tab to inches/quarter inches	y
7a. Select paper tray #1 (Daisy only)	n
7b. Select paper tray #2 (Daisy only)	n
8. Deselect printer	y
9. Enable superscript	y
10. Disable superscript	y
11. Enable subscript	y
12. Disable subscript	y
13. Enable underscore	y
14. Disable underscore	y
15. Enable overstrike	y
16. Disable overstrike	y
17. Set pitch	y
18. Set line feed spacing and enable auto line feed	y
19. Set line feed spacing and disable auto line feed	y
20. Enable auto line feed	y
21. Disable auto line feed	y
22a. Partial line feed (1/4)	n
22b. Partial line feed (1/2)	n
22c. Partial line feed (3/4)	n
23. Set left margin	y
24. Enable proportional spacing	y
25. Disable proportional spacing	y
26. Set form length in lines	y
27. Set form length in inches	y
28. Enter temporary transparent mode	y
29. Restore power-on defaults	y

Implementation of Escape Sequences

This section explains in detail how the GPD implements the escape sequences. In general, a given device accepts and acts on those sequences that have meaning for it, and it ignores those sequences that have no meaning for it. All escape sequence bytes are hex values.

1. Execute built-in diagnostics (02 01 02 01 00 0F)

   This sequence is defined for all printers with built-in diagnostics escape sequences. Before the diagnostics start, any data in the buffer is printed, and a form feed generated.

2. Select character font (02 02 dd 0F)

2a. Default	dd = 00
2b. Alternate #1	02
2c. Alternate #2	04

The new font remains in effect until a new Select Font sequence is received, until a Reset Defaults sequence is received, or until the printer is powered off and then back on. The exact implementation is device dependent.

3. Clear platen (02 03 0F)

The purpose of this sequence is to move a single sheet of paper or an envelope from the platen to the receiving bin without feeding a new sheet or envelope into the platen. On devices other than Daisy printers, this sequence is ignored.

4. Define the meaning of Control hex(0E) (02 04 dd ee ff gg)

dd: 00 - overstrike disabled (default)
02 - overstrike enabled
ee: 00 - underscore disabled (default)
04 - underscore enabled
ff: 00 - emphasized print disabled (default)
02 - emphasized print enabled
gg: 0E - turn on attributes until reset or a hex(0F) is received, turn on attributes again when a Control hex(0E) is received
0F - turn on attributes upon receipt of a Control hex(0E); turn off on a Control hex(0F), carriage return or reset

When the GPD receives a Control hex(0E), it turns on the attributes associated with the 0E and continues processing as usual. The attributes remain in effect until a condition set by gg above is met. The set of conditions used is determined by the end of the Control hex(0E) definition. If the definition ends with a 0E, the attributes are turned on immediately; if it ends with a 0F, they are not turned on until a Control 0E is received. The default value of hex(0E) and hex(0F) is specified by the escape sequence:

02 04 00 00 02 0F

5. Horizontal tab to column dd (02 05 01 dd 0F)

dd = number of destination column

This sequence causes the printer to move to column dd of the current line. The printer moves to this column even if it has already passed it.

6. Horizontal tab dd inches and ee quarter inches (02 05 02 dd ee 0F)

dd = number of inches
ee = number of quarter inches

This sequence causes the carriage to move approximately dd inches and ee quarter inches from the current print position. If the carriage would move beyond the defined line length, the sequence will be ignored. If the number of characters per inch, or per quarter inch, is not a whole number, the number of characters is rounded to the nearest whole number. The number of quarter inches can be no greater than 3.

7. Select paper tray (02 06 dd 0F)

7a. Rear tray dd = 01
7b. Front tray 02

Upon receipt of this sequence, the current contents of the buffer will be printed. The correct tray will then be selected. This sequence is for Daisy printers only.

8. Deselect printer (02 07 0F)

The Deselect Printer sequence causes all data currently in the buffer to be printed, then disables the printer for receiving data. This sequence is not supported by all printers.

9. Enable superscript (02 08 01 0E)

This sequence causes the printer to enter its superscript mode, if superscript mode is available. If a given printer does not support superscripting, the platen moves by 1/2 of a reverse line feed. This mode remains enabled until a Restore Defaults sequence is received, a Disable Superscript sequence is received, or in some cases, until the printer is powered off and then back on.

10. Disable superscript (02 08 01 0F)

This sequence causes the printer to leave superscript mode.

11. Enable subscript (02 08 02 0E)

This sequence causes the printer to enter its subscript mode, if subscript mode is available. If a given printer does not support subscripting, the platen moves by 1/2 of a forward line feed. This mode remains enabled until a Restore Defaults sequence is received, a Disable Subscript sequence is received, or until the printer is powered off and then back on.

12. Disable subscript (02 08 02 0F)

    This sequence causes the printer to leave subscript mode.

13. Enable underscore (02 08 03 0E)

    This sequence causes all printable characters that follow it, including spaces, to be underscored. Underscoring remains in effect until a Restore Defaults sequence is received or a Disable Underscore Mode sequence is received.

14. Disable underscore (02 08 03 0F)

    This sequences causes the printer to leave the underscore mode.

15. Enable overstrike (02 08 04 0E)

    This sequence causes all printable characters that follow it, including spaces, to be overstruck. Overstriking remains in effect until a Restore Defaults sequence is received or a Disable Overstrike Mode sequence is received.

16. Disable overstrike (02 08 04 0F)

    This sequences causes the printer to leave the overstrike mode.

17. Set pitch (02 09 01 02 dd ee 0F)

    dd = value of the integral pitch
    ee = value of the fractional pitch

    pitches supported: 5, 10, 12, 15, 16.5

    This sequence selects the specified pitch. The selected pitch remains in effect until the occurrence of another Select Pitch sequence, the occurrence of the Reset Defaults sequence, or until the device is powered off and on. The default pitch is 10 characters to the inch. Any pitch setting not supported by the device will be ignored.

18. Set line feed spacing and enable auto line feed after carriage return (02 0A 01 01 dd 0E)

    dd = number of lines per inch; must be greater than 00 and not greater than 0F

    This sequence sets the line feed size and enables auto line feed after a carriage return.

19. Set line feed spacing and disable auto line feed after carriage return (02 0A 01 01 dd 0F)

dd = number of lines per inch; must be greater than 00 and not greater than 0F

This sequence sets the line feed size and disables auto line feed after a carriage return.

20. Enable auto line feed (02 0A 0E)

This sequence causes the GPD to send a line feed to the printer immediately after each carriage return. This is the default for the driver.

21. Disable auto line feed (02 0A 0F)

This sequence prevents a line feed from being sent to the printer after each carriage return.

22. Partial line feed (02 0A dd 0F)

22a. 1/4 feed dd = 00
22b. 1/2 feed 02
22c. 3/4 feed 04

This sequence generates partial line feeds in increments of 1/4, 1/2, and 3/4 of the currently selected, full line feed increment. Before execution of the partial line feed, this sequence suppresses auto line feed to avoid unwanted extra line feeds. A given printer may not support all these spacings.

23. Set left margin (02 0B 01 dd 0F)

This sequence sets the left margin to column dd. After execution of this sequence, the carriage location is at the left margin of the current line.

24. Enable proportional spacing (02 0B 02 0E)

This sequence enables the proportional spacing mode for those printers that support it. If a printer does not support proportional spacing, the sequence is ignored.

25. Disable proportional spacing (02 0B 02 0F)

This sequence disables the proportional spacing mode for those printers that support it. If a printer does not support it, the sequence is ignored.

26. Set form length in lines (02 0C 01 02 00 dd 0F)

dd = number of lines per form

This sequence sets the form length. You should not send this code unless it is directly preceded by powering on the printer, a Reset Defaults sequence, or a form feed.

27. Set form length in inches (02 0C 02 dd 0F)

dd = number of inches per form

This sequence sets the form length to dd inches, where dd is less than or equal to 16. Do not send this code unless it is directly preceded by powering on the printer, a Reset Defaults sequence, or a form feed.

28. Enter temporary transparent mode (02 0D 01 cc dd ... dd)

cc = number of bytes to pass to the printer
dd = the bytes to be passed

This sequence causes the driver to enter the transparent mode for the specified number of bytes. At most, 256 bytes may be passed to the printer in any one string. The bytes passed to the printer by this sequence are not modified in any way.

29. Restore power-on defaults (02 0D 0C 03 0F)

This sequence resets all of the power-on defaults of the printer to their original values. It also resets the driver to its default values. Any data in the buffer when this command is received is printed before the command is executed. If a partition is using the GPD with more than one printer, it should send this sequence to each printer before sending a print request, whenever it uses a different printer.

LIST DT Command Enhancement

The LIST DT command now shows you the driver tables and associated addresses available for the current partition and terminal. Driver information for other terminals is not available. The new information appears at the bottom of the LIST DT screen. LIST DT displays the information in the form ### NAME Y/N. LIST DT always displays the first digit as 0, regardless of how you specified the device type in @GENPART. The last two digits are the device address. NAME is the table file name. Y indicates that the table is enabled. N indicates that it is not enabled. The example in Figure 1-2 displays the table names for the five devices at addresses 15, 16, 04, 14, and 17.

```
0001 0001 0415 0000

000055000 00055000 02155000

0B30000000000000 0D70000000000000
0000000000000000 0000000000000000
0000000000000000 0000000000000000
0000000000000000 0000000000000000
0000000000000000 0000000000000000
0000000000000000 0000000000000000
0000000000000000 0000000000000000
000000000000000 0000000000000000

004      015-03  310-050 320-05X 330     07B     03A     03B     03C     03D
0FA      0FB     0FC     0FD     0FE     0FF     018     326     017

   015 @PM016V0  Y   016 @PM010V0  N  004 @PM010V0  Y  014 @PM016V0  Y
   017 @PM016V0  N
```

Figure 1-2. LIST DT Screen

1.4 LIST DCT COMMAND ENHANCEMENTS

The LIST DCT command now supports the use of wildcard characters, * and ?, that allow you to search a disk for specified groups of filenames.

The * represents any number of successive characters of any type. A LIST DCT command using * displays all filenames having any number of characters, or no characters, in the location of the *. For example, the following command displays the catalogue information for all files on disk 330 whose names end with the three letters PWP.

```
LIST DCT/330,"*PWP"
```

The ? represents one character of any type. A LIST DCT command using ? displays all filenames having any character at the location of the ?. For example, the following command displays the catalogue information for all filenames on disk 330 whose names are 6 characters in length and have TEST as the first 4 characters.

```
LIST DCT/330,"TEST??"
```

If you know the name of the file, you can enter it without any wildcard characters. The LIST DCT command will display the catalogue information for the specified file, if the file resides on the disk. For example, the following command displays the catalogue information for a file on disk D35 named MYPROG.

```
LIST DCT/D35,"MYPROG"
```

If the file is not found, the LIST DCT command returns nothing but the system prompt (:).

NOTE

Because the "*" and "?" are wildcards, you should avoid using these two characters in a filename.

1.5 ASYNCHRONOUS COMMUNICATIONS ENHANCEMENTS TO THE 2236MXE TERMINAL CONTROLLER

Release 2.6 provides the following asynchronous communications enhancements to the MXE controller:

- XON/XOFF flow control
- CTS/DCD, DTR flow control
- Monitorable RS-232 signals
- Direct user control over RTS and DTR signals

1.5.1 Communications Control Vector Extensions

Bytes 21 to 29 have been appended to the end of the communications control vector with no changes to the already existing byte definitions. If you want XON/XOFF or flow control via DTR, you should load all bytes up to Byte 29. If you do not want to use these functions, you need only load the first 20 bytes of the control vector. Therefore, no old applications are affected by this extension.

The new bytes are defined in Table 1-7.

Table 1-7. New Communications Control Vector Bytes

Byte	Bit	Meaning
21	01	Select CTS/DCD as flow control from connected device
	02	Select DTR as flow control from the MXE controller
	04	Select XON/XOFF as flow control from connected device
	08	Select XON/XOFF as flow control from the MXE controller
	10	Always zero
	20	Always zero
	40	Always zero
	80	Always zero
22		Escape code for two-byte escape sequences for XON/XOFF flow control from connected device. If this byte is zero, a one-byte flow control code is assumed. If this byte is not zero, the code specified must precede Byte 23 or 24 for the two-byte sequence to be recognized as a valid XON/XOFF sequence.
23		XON code from connected device.
24		XOFF code from connected device.
25		Escape code for two-byte escape sequences for XON/XOFF flow control from the MXE controller. If this byte is zero, a one-byte flow control code is assumed. If this byte is not zero, the code specified must precede Byte 26 or 27 for the two-byte sequence to be recognized as a valid XON/XOFF sequence.
26		XON code from the MXE controller.
27		XOFF code from the MXE controller.

Table 1-7. New Communications Control Vector Bytes (continued)

Byte	Bit	Meaning
28		XON threshold level. The MXE controller sends the XON byte(s) when the buffer level decreases to this number and there is a pending XOFF. When using DTR flow control, the MXE controller will enable DTR at this level.
29		XOFF threshold level. The MXE controller sends the XOFF byte(s) when the buffer level increases to this number and there is a pending XON. When using DTR flow control, the MXE controller will disable DTR at this level.

You should keep in mind the following points when using the new communications control vector bytes.

1. If the XON byte has the same definition as the XOFF byte, the MXE controller uses the defaults of hex 11 and hex 13, respectively.

2. Zero is a valid, though not recommended, flow control code.

3. If you select a two-byte flow control sequence, only valid XON/XOFF codes are deleted from the data stream. Two-byte sequences that are not valid flow controls are passed to the program.

4. When you are specifying the buffer empty levels, the values must be in the range of 1 to 254 (hex 1-FE), and the XOFF level must be greater than or equal to the XON level. If either value is out of range, or if the XON level is greater than the XOFF level, the MXE controller uses defaults of 32 and 200, respectively.

5. If you select XON/XOFF while half duplex is selected, the MXE controller forces full duplex mode during initialization.

6. When using XON/XOFF flow control, the MXE controller sends XON/XOFF controls even when the connected device has sent an XOFF code.

7. When using DTR flow control, the MXE controller accepts data from the connected device even when DTR is inactive. However, overrun may occur if data is received when DTR is low.

NOTE

If DCD is low and you have selected CTS/DCD flow control, the MXE controller will not recognize data on the line, including XON/XOFF bytes, as valid.

8. The MXE controller does not send XON/XOFF if CTS is low and CTS/DCD are selected as flow controls.

9. The MXE controller sends an XOFF byte on each of the following events:

   - the buffer level reaches the specified XOFF threshold
   - the buffer level reaches 254
   - a buffer overflow occurs

10. If reception is halted immediately after the MXE controller sends an XOFF character to the connected device, it is possible for a lockout condition to occur. To prevent this, the MXE controller sends an XON character when reception is enabled (CBS (08)), if the buffer level is less than or equal to the XON level. If DTR is selected, the signal is lowered or raised depending on the XON level.

1.5.2 Status Vector Extensions

An eighth byte has been added to the status vector that contains the current state of all control signals supported by the MXE controller. The bit positions are defined in Table 1-8.

Table 1-8. New Communications Status Vector Byte

Bit	Pin	Meaning
01	8	DCD (Received Line Signal Detector)
02	22	RI (Ring Indicator)
04	6	DSR (Data Set Ready)
08	5	CTS (Clear To Send)
10	20	DTR (Data Terminal Ready)
20	4	RTS (Request To Send)
40		XOFF byte received by the MXE controller from connected device
80		XOFF byte sent by the MXE controller to connected device

The last two bits (40 and 80) indicate the current receive/transmit state in the respective direction. When XON/XOFF is selected, the actual flow control bytes being received or transmitted determine the setting of the two bits. If CTS/DCD and DTR are selected without XON/XOFF, Bit 40 is set by the current state of CTS, and Bit 80 is set by the current state of DTR.

To allow compatibility with older applications, you need input only the first seven bytes of the status vector.

1.5.3 New MXE TC Command: Set Signals

A new TC command has been added to set or reset the state of RTS and DTR and optionally to lock these signals in a particular state.

Signal Sequences:

```
                              (0C)
    CBS(0F) OBS(xxyyzz) CBS
                              (00)
```

Where:

xx = a byte in which Bit 10 specifies the state of DTR and Bit 20 specifies the state of RTS

yy = a byte designating which bits in xx are valid

zz = a byte in which Bit 10 specifies the state of the DTR lock and Bit 20 specifies the state of the RTS lock

Byte xx designates the state of DTR and RTS. A bit value of one is the active state.

Byte yy is a mask that designates which bits in xx are valid. Bits 10 and 20 of yy correspond to bits 10 and 20 of xx.

Byte zz allows the option to lock these states so that no condition can change them except another Set Signals command, a Reset command (CBS (80)), or a Disconnect command. Bit 10 locks or unlocks the state of DTR, and Bit 20 locks or unlocks the state of RTS. Byte zz is not affected by the values of bytes xx and yy. It is therefore possible to set one or both signals and/or lock these signals in any combination.

This command sets DTR and RTS to the states desired no matter what flow control option is selected. Loading the communications control vector clears the lock states and sets DTR and RTS to their default values.

The signal states on the RS-232 connector are not changed until the CBS(0C) is received by the MXE controller. It is therefore possible to lock the current state of DTR and RTS without setting new signal states by writing three bytes to the MXE controller, with the third byte containing the lock state, but issuing a CBS(00) instead of a CBS(0C) to end the command.

The Disconnect command will work as it currently does by setting DTR to inactive for 3 to 5 seconds and then setting it to active. The current lock state of RTS and DTR does not affect the Disconnect command. All other commands that alter the state of RTS or DTR first check the signal's lock state before changing it. If the signal is locked to its current state, the signal is not modified until a new command to unlock it is sent.

You need not send all three bytes to the MXE controller each time you want to change a signal. The three bytes (xxyyzz) are saved by the MXE controller and are overwritten one-by-one each time the command is issued. Thus you can overwrite zero, one, two, or three bytes, as desired.

Examples:

These examples give signal sequences and sample BASIC-2 code, using the $GIO statement, for those sequences. For more information on the $GIO statement, refer to the Wang BASIC-2 Language Reference Manual.

The following example sets RTS, sets DTR, and locks neither:

Signal Sequence:

```
CBS(0F)  OBS(303000)  CBS(0C)
```

BASIC-2 Code:

```
A$ = Hex(303000)
$GIO#1(440F A000 440C)A$
```

The following example resets RTS, sets DTR, and locks neither:

Signal Sequence:

```
CBS(0F)  OBS(103000)  CBS(0C)
```

BASIC-2 Code:

```
A$ = Hex(103000)
$GIO#1(440F A000 440C)A$
```

The following example does not change RTS, sets DTR, and locks RTS:

Signal Sequence:

```
CBS(0F)  OBS(101020)  CBS(0C)
```

BASIC-2 Code:

```
A$ = Hex(101020)
$GIO#1(440F A000 440C)A$
```

The following example, by using CBS(00) instead of CBS(0C), changes neither signal but locks them both:

```
Signal Sequence:

CBS(0F)  OBS(303030)  CBS(00)

BASIC-2 Code:

A$ = Hex(303030)
$GIO#1(440F A000 440C)A$
```

1.6 ADDITIONAL MEANING FOR THE I94 ERROR MESSAGE

Communications to disk have been enhanced for Release 2.6. To support disk communications, an additional meaning has been assigned to the I94 error message. For those systems with a FORMAT key, I94 can still indicate that you attempted to access the disk while the format key was engaged. For all systems, I94 can also indicate that the system was unable to send a command to the selected disk due to transmission failure. For systems without the FORMAT key, I94 can only have the latter meaning.

CHAPTER 2
RELEASE 2.5, RELEASE 2.4, RELEASE 2.3, AND RELEASE 2.2

2.1 INTRODUCTION

This chapter summarizes enhancements made to 2200 BASIC in previous releases that are not yet incorporated into 2200 system manuals.

2.2 RELEASE 2.5

Release 2.5 of the BASIC-2 Multiuser Operating System was issued to provide support for the communications capabilities of the Model 2236MXE, provide a new, more efficient disk index structure and access method, and provide a facility for tracing disk activity

2.2.1 Changes to the System Platter

The system platter includes the MVP operating system and BASIC-2 language processor, the system diagnostics, and the system utilities. The following system files have been revised or deleted since Release 2.4:

@MVP	- MVP Operating System and BASIC-2 Language Processor (Rel. 2.5)
@MXE0	- MXE Microcode
@MRTIAN	- Martian Wars Game
@INSTALL	- System Install Utility
@MOVEFIL	- Move File Utility
@RECOVER	- Recover from Backup Utility
@SYSMVPB	- System Menu

- The MVP Operating System and BASIC-2 Language Processor file (@MVP) and the MXE Microcode file (@MXE0) have been updated to include the operating system enhancements.

- To accommodate the increased operating system code and the MXE asynchronous facility, the Martian Wars game (@MRTIAN) was deleted.

- The System File Installation utility and the System Menu file have been updated to reflect the deletion of the Martian Wars game.

- The Move File utility (@MOVEFIL) and the Recover from Backup utility (@RECOVER) have been updated to correct known problems with these utilities.

2.2.2 Corrected Anomalies

Previous releases of the Move File utility (@MOVEFIL) destroyed files under the following conditions:

- The operator specified the same address for both source and destination platters.

- The operator did not specify a destination file name different from the source file name.

This utility now displays an error message if you attempt to move a file to the disk on which it currently resides. Additionally, the @MOVEFIL utility now accepts lowercase characters for a disk address.

Previous releases of the Recover from Backup utility (@RECOVER) did not correctly recover 'all active files' if sectors in the catalogue index (i.e., @INDEX) contained duplicate file names. The utility has been modified to correct this problem; there has been no functional change to @RECOVER.

2.2.3 Command Mode of the Model 2236MXE

When a 2200 system is configured with a Model 2236MXE or an SVP Option-W Terminal Processor, it is possible to perform certain system procedures by evoking the Model 2236MXE Command mode and entering MXE commands. These MXE Command mode procedures include:

- Setting the transmission rate of a port on the terminal processor
- Locking and unlocking the transmission rate of the current port
- Displaying the status of all ports on the terminal processor
- Changing the primary user (i.e., Port 1)

Running 2236MXE Command Mode

You can run MXE commands at any time from any powered-on terminal currently connected to a port on the Model 2236MXE or an SVP Option-W. It is not necessary to configure the system before entering MXE Command mode. If the system is configured, the terminal from which you issue the MXE command does not have to be in the current configuration.

Before configuration, the system is under the control of the Bootstrap. The Set Primary Port command can be run only when the system is under Bootstrap control or when the system is running with the VP operating system.

When the system is running under Bootstrap control or running the VP operating system, there is a special MXE password needed to execute most MXE commands. On power-up, this password defaults to MXEPSW. You can change the password with the Change MXE Password command. When the system is running the MVP operating system, the MXE password is the same as the first six characters of the 2200 system configuration password.

Entering and Exiting 2236MXE Command Mode

Entering MXE Command mode alters the screen display. To enter MXE Command mode, press RESET, then press LOAD three times. After the third time, the prompt ENTER MXE COMMAND appears on the screen, followed by a percent character (%) on the next line. When the percent character appears, you can enter the MXE command. Every MXE command begins with a one-character MXE command code.

After processing the command, the terminal again displays the percent character and you can enter another MXE command. Pressing RETURN when the percent prompt appears, or pressing RESET exits MXE Command mode. Exiting MXE Command mode returns to the processing point from which MXE Command mode was entered.

NOTE

Although entering MXE Command mode halts processing on the current terminal, other terminals on the terminal processor operate normally. It is possible for more than one terminal to enter MXE Command mode simultaneously. Therefore, use caution when entering MXE commands that affect the performance of other terminals, such as Set Transmission Rate.

2.2.4 Status Command

General Form:

?

The Status command returns a chart indicating the status of all ports on the terminal processor to which the terminal issuing the command is connected. The first line of the chart uses the following format:

device R# (mode) option

where:

device = the model number of the terminal processor.

R# = the revision of terminal processor firmware. This number is the number of the general operating system release with the terminal processor release number appended to it. For example, Revision 2.51 or 2.53 of the terminal processor code runs with Operating System 2.5, while terminal processor Revision 2.92 runs with Operating System 2.9. Problems may occur if the number of the terminal processor firmware does not correspond to the operating system release number.

mode = the indication that the system has been configured with an MVP operating system, in which case the terminal displays MVP within the parentheses. If the system has not yet been configured or if the system has been configured with a VP operating system, the terminal displays BOOTSTRAP within the parentheses.

option = the indication of any additional options implemented by the MXE code.

The next four lines of the chart display the following information for each port on the terminal processor.

On/Off state -- If the terminal connected to the port is powered on (i.e., Data Set Ready on the RS-232 connection is active), the word ON appears on the screen. Otherwise, the word OFF appears.

Software selected transmission rate -- This is the transmission rate selected by MXE Command Code B, Set Transmission Rate. If MXE Command Code B has not been evoked, the transmission rate defaults to the rate set by the hardware switches.

Locked or unlocked state -- The Letter L indicates that MXE Command Code L, Lock, has been used to lock the transmission rate of the terminal. If the terminal is not locked, this field is left blank.

Hardware selected transmission rate -- This is the transmission rate set by the switches on the processor board.

Local/Remote state -- This field reflects the state of the Carrier Detect signal on the RS-232 line, indicating whether the terminal is a local terminal or is connected to the MXE through a modem. A remote terminal causes an active Carrier Detect signal.

Terminal type -- This field describes the format of the current communication protocol of the port. If the terminal on the indicated port is a standard Wang 2200 terminal or Wang 2200 terminal emulation (following Wang terminal protocol), the message WANG TERMINAL appears in this field. If the port is configured as a general asynchronous communications port, the message ASYNCH appears in this field.

An asterisk appears before the port to which the terminal issuing the status command is connected.

The following chart is an example of the chart the Status command returns:

```
2236 MXE R2.51  (BOOTSTRAP)
*PORT 1 ON  19200 L / (19200)  LOCAL   WANG TERMINAL
 PORT 2 OFF 19200   / (19200)  LOCAL   WANG TERMINAL
 PORT 3 OFF 19200   / (19200)  LOCAL   WANG TERMINAL
 PORT 4 OFF 19200   / (19200)  LOCAL   WANG TERMINAL
```

In this example, the terminal processor is a Model 2236MXE; the revision number is 2.51, indicating that this is the first bootstrap microcode revision that corresponds with Operating System Release 2.5; and the system has not yet been configured or has been configured with a VP operating system. The port issuing the status command is Port 1, which is the only port with a powered-on terminal. Port 1 on this 2236MXE terminal processor can be Port 1, Port 5, Port 9, or Port 13 on the system, depending on the number of terminal processors connected and the address of this MXE. The transmission rate at Port 1 is locked. All four ports are set to transmit at a rate of 19,200 bps and all terminals are local Wang terminals.

2.2.5 Change MXE Password Command

General Form:

C password1 password2

Where:

password1 = the current MXE password, consisting of six characters. The password does not appear on the screen; instead, the string "012345" appears.

password2 = the new MXE password.

Command Code C changes the MXE password used when the system is operating under control of the Bootstrap or with a VP operating system.

When the system is operating with an MVP operating system, the 2200 system reconfiguration password is required to execute MXE commands. The Change MXE Password command has no effect on the system reconfiguration password. The 2200 system password is eight bytes long. The MXE password is six bytes long. Therefore, upon configuration, the MXE password is set to the first six bytes of the system password.

After the system processes this command, the CRT displays the message OK.

Examples of valid syntax:

```
C OLPSWD NWPSWD
C secret Psword
```

2.2.6 Set Primary Port Command

General Form:

A password x

Where:

password = the current MXE password. The password does not appear on the screen; instead, the string "012345" appears.

x = the port designator, a number between 0 and 4 indicating which port to establish as the primary port of the 2236MXE. A port designator of 1, 2, 3, or 4 corresponds to the port number on the terminal processor. A port designator of 0 indicates the current terminal port.

Command Code A sets the primary port. The primary port supports the terminal that can perform master initialization. On a system with more than one terminal processor, you can issue this command only from a terminal connected to the first terminal processor.

Additionally, you can issue this command only when the system is under control of the Bootstrap. Unless a Change MXE Password command has been issued, the password is the default password, MXEPSW.

When this command is issued, Port 1 is logically swapped with the port indicated in the command. The MVP operating system will subsequently consider the swapped port to be Port 1 and Port 1 to be the swapped port. This command is used when Port 1 is not functioning or is not available.

After the system processes the Set Primary Port command, the CRT displays the message OK.

Examples of valid syntax:

```
A SYSTEM 3
A SECRET 0
A PSWORD 4
```

Note that the password itself does not appear on the screen.

2.2.7 Set Transmission Rate Command

General Form:

B password x rate

Where:

password = the current MXE password. The password does not appear on the screen; instead, the string "012345" appears.

x = the port designator, a number between 0 and 4 indicating the port to which the transmission rate is set. A port designator of 1, 2, 3, or 4 corresponds to the port number on the terminal processor. A port designator of 0 indicates the current terminal port.

rate = the transmission rate to set for the port indicated by the port designator. This command can set the transmission rate to 50, 75, 100, 110, 134.5, 150, 200, 300, 600, 1200, 2400, 4800, 9600, or 19200.

Command Code B sets the transmission rate of a particular port. Setting the transmission rate through software overrides any hardware switch settings. Master initializing or powering on the system resets the transmission rate to the hardware setting.

Command Code B can set the transmission rate of only those ports on the same terminal processor as the terminal issuing the command. The port must currently be a terminal port (i.e., the command will not work on a TC or other special type of port).

After you enter this command, the terminal displays the 2236MXE Status chart and prompts you to verify the entered information. You enter a Y to verify or an N to cancel.

Examples of valid syntax:

```
B SYSTEM 3 9600
B secret 0 300
B Psword 4 50
```

Note that the password itself does not appear on the screen.

2.2.8 Lock Command

General Form:

L

Command Code L changes the lock state of the port issuing the command. When a port is locked, no port can change its transmission rate, including the locked port itself. The default condition, set when the system is powered on, is unlocked.

When this statement is issued from a port that is unlocked, the port changes to a locked state and the CRT displays the statement: BAUDRATE LOCKED. When this statement is issued from a port that is locked, the port changes to an unlocked state and the CRT displays the statement: BAUDRATE UNLOCKED.

2.2.9 Communications Capabilities of the 2236MXE

Release 2.5 provides support for the asynchronous communications capabilities of the 2236MXE. With the new parameters of the SELECT statement, you can select any ports on the system (except Port 1) as telecommunications ports.

Asynchronous communications under the control of a user BASIC program or Wang ASYNC1 package is supported. 2227B asynchronous functionality is provided, with the following exceptions:

- Reverse channel is not supported.
- PRINT statements cannot be used for TC output; $GIO must be used.

A number of BASIC statements are used for controlling asynchronous communications. SELECT TC selects a port as a telecommunications port; SELECT TERMINAL converts a TC port to a terminal port. $OPEN and $CLOSE control access to a TC port. $GIO is used to issue TC commands, send output, and receive input. Refer to Section 2.6 for a description of each command.

2.2.10 New Disk Index

A new, more efficient disk index structure is available with Release 2.5. The new index method improves the performance of locating files within the index and is particularly effective on platters containing many files. The improved performance is reflected in the LOAD, SAVE, LIMITS, DATALOAD OPEN, and DATASAVE OPEN statements.

2200 Disk Index Structure

With Release 2.5, the 2200 operating system supports two distinct index structures. A platter has either the new index structure or the old index structure; however, both types can exist on the same system. The old index structure, created with the SCRATCH DISK statement, is fully supported with no change. A SCRATCH DISK ' statement creates the new index structure.

The first byte of sector 0 indicates which index structure exists on the platter. If Byte 1 of sector 0 equals hex 00, then the old index structure (i.e., SCRATCH DISK) exists on the platter. If Byte 1 of sector 0 equals hex 01, then the new index structure (i.e., SCRATCH DISK ') exists on the platter.

File entries in the new index structure are identical to those in the old index structure.

Access to the New Disk Index

The new index structure uses a more efficient hashing algorithm for locating files in the index, resulting in a more even distribution of file entries. Additionally, if an index sector is full, the system enters a new file into the next higher sector rather than the next lower, as is done with the old index. Entering the files into the next higher sector results in an improved performance on 2200 disks that provide a look-ahead read (e.g., the Winchester-style fixed drives of the 2200LVP and the Model 2280 Disk Drive).

The following BASIC-2 program simulates the new disk index hashing:

```
100 N$ = F$: REM F$ contains the name of the file to be entered
110 X$ = HEX(00)
120 FOR I = 1 TO 8
130  X1$ = STR(N$,I,1): REM X1$ contains the Ith byte of the filename
140  IF MOD(I,2) = 0 then 160
150  ROTATE (X1$,4):    REM Exchange upper/lower nibbles of odd bytes
160  X$ = X$ ADD X1$:    REM X$ contains the sum of all bytes
170 NEXT I
180 S = MOD(VAL(X$),C): REM C contains the number of index sectors
190 REM S contains the sector number into which the entry should go
```

If sector S is full, then S is set to MOD(S+1, C) and the entry goes in the new sector S. The system repeats this process until the entry is placed or until S equals its original value, thus indicating that the disk catalog is full.

WARNING

The new disk index structure is not appropriate for all users. Certain BASIC-2 programs make direct use of knowledge of the old index method, accessing the index with DATALOAD BA and DATASAVE BA. These programs will not operate properly on platters with the new disk index since the hashing algorithm has changed. Careful analysis should be done before converting to the new index since there is a risk that some software will not operate properly after the conversion. It is not a recommended practice to directly access the disk index with DATALOAD BA and DATASAVE BA. However, if such a program is written, it should support both the old and new index structures since it may not be possible or wise to convert all old platters to the new format.

Wang Laboratories, Inc. will update Integrated Support Systems (ISS) to support both the new and old index structures. However, Wang will not update several other packages (e.g., STAT). Therefore, such packages will only operate with the old index.

Conversion to the New Index

To convert an existing platter to the new index structure, perform the following:

- Establish a new disk index on a blank platter by executing a SCRATCH DISK ' statement.
- Move files from the existing platter to the new platter with a MOVE file statement.

2.2.11 New Statements

In summary, Release 2.5 includes the following new statements to the BASIC-2 language: $CLOSE, $GIO, $OPEN, SCRATCH DISK ', SELECT TC, SELECT TERMINAL, and TRACE DISK. Refer to Section 2.6 for descriptions of these statements.

2.3 RELEASE 2.4

Release 2.4 of the BASIC-2 Multiuser Operating System was issued primarily to provide support for the Model 2236MXE Terminal Processor and SVP Option-W. Additionally, Release 2.4 included a system date and time, a connect/disconnect detection facility, and a screen input facility.

2.3.1 MXE Support

Release 2.4 (or later) of the BASIC-2 Operating System is required for BASIC-2 systems with a Model 2236MXE Terminal Processor or SVP Option-W.

2.3.2 Disk Write

If a Data Error (Error 96) occurs during a disk write operation, the system will retry the operation up to three times. This is expected to reduce error rates for disk operations and should be particularly effective for multiplexed 2280 disk units.

2.3.3 DATE and TIME

The Model 2236MXE Terminal Processor, SVP Option-W, and the current version of the Model 22C32 Triple Controller include a time-of-day clock that is accessible through and maintained by BASIC-2.

Release 2.4 introduced the following BASIC-2 language enhancements to support these added features:

- The DATE statement sets or changes the system date; the DATE function returns the value of the current date.

- The TIME statement sets or changes the system time; the TIME function returns the value of the current date.

Refer to Section 2.6 for a description of the DATE and TIME functions and statements.

To set or adjust the time and date, Release 2.4 provided an Initialize Date & Time utility. This utility resides on the system platter; it is discussed in Chapter 3.

To support the DATE and TIME statements, Release 2.4 included the following new error message:

ERR X79 - Invalid Password

If an incorrect password is specified in either the DATE or TIME statement, the system returns an ERR X79. This error is recoverable (i.e., a programmer can intercept and respond to the error under program control). Refer to Chapter 9 of the BASIC-2 Language Reference Manual for more information on error control features.

2.3.4 Screen Input

Release 2.4, when used with a Model 2336DW or 2336DE Terminal and a Model 2236MXE Terminal Processor or SVP Option-W, allows the image displayed on the screen to be transmitted to a BASIC program for further processing. BASIC receives the screen data with an INPUT SCREEN statement.

The 2200 screen input facility enables a BASIC program to receive an image of a terminal screen display. This program would typically print the image on a system printer or save the image in a disk file for processing. A screen input can be requested by a BASIC program or by an operator. In the first case, a BASIC program executes an INPUT SCREEN statement that reads the screen of the terminal attached to the current partition. In the second case, the operator holds down SHIFT and EDIT, thereby attaching the terminal to a partition that has previously been made available for screen input. Refer to Section 2.6 for a description of the INPUT SCREEN statement and the procedure for operator-initiated screen inputs.

The screen input facility requires the following hardware and software:

- MVP BASIC-2 Release 2.4 (or later)
- 2200MVP, 2200LVP, or 2200SVP Central Processing Unit
- Model 2336DW or Model 2336DE Terminal
- Model 2236MXE Terminal Processor (for MVP and LVP systems)
- Option-W (for SVP systems)

2.3.5 Terminal Disconnect

The 2236MXE and SVP Option-W can alert the 2200 CPU when a local or remote terminal is connected or disconnected. The terminal connect/disconnect detection facility enables 2200 BASIC programs to be written to monitor system users. With this facility, the operating system can detect whether a terminal is connected or disconnected. Upon disconnection, the system can initiate bookkeeping procedures, force a terminal to disconnect automatically after a specified period of time, or initiate Logon and Logoff programs. The $DISCONNECT statement enables or disables disconnect detection. Refer to Section 2.6 for a description of the $DISCONNECT statement.

The following list summarizes the features of the terminal connect/disconnect detection facility:

- Connect detection for partition allocation and initiation of a user Logon program.

- Forced disconnection if you do not log on within the program specified time. Thus, you are prevented from tying up a port without completing the Logon procedure.

- Program control of the connect/disconnect facility through BASIC.

- Disconnect detection for initiation of user Logoff programs.

- BASIC TIME and DATE functions for logging system use.

The connect/disconnect facility requires the following hardware and software:

- MVP BASIC-2 Release 2.4 (or later)
- 2200MVP, 2200LVP, or 2200SVP Central Processing Unit
- Model 2236MXE Terminal Processor (for MVP and LVP systems)
- Option-W (for SVP systems)

NOTE

VP BASIC-2 will support the Model 2236MXE and the SVP Option-W. However, the DATE, TIME, INPUT SCREEN, and $DISCONNECT statements will generate syntax errors.

2.4 RELEASE 2.3

Release 2.3 of the BASIC-2 Multiuser Operating System was issued primarily to provide support for the additional memory banks available on the 2200MVPC and 2200LVPC. However, this operating system can also run on the 2200MVP, 2200LVP, and 2200SVP. Release 2.3 included the following features and enhancements:

- Revised operating system to support up to 512K bytes of user memory, provided by the 2200MVPC and 2200LVPC. (The maximum number of partitions remains 16; the maximum number of terminals remains 13.)

- Revised Partition Generator (@GENPART) and Partition Status (@PSTAT) System Utilities to support 512K bytes of user memory.

- Revised system diagnostics for testing control memory and user memory.

2.5 RELEASE 2.2

Release 2.2 of the BASIC-2 Multiuser Operating System was issued to provide enhancements to the System Utilities, as well as provide several new features. Refer to the BASIC-2 Utilities Reference Manual (700-6855) for further information about each utility.

Additionally, Release 2.2 introduced the #ID function and the $ALERT statement. The numeric function #ID returns the CPU identification number. With the $ALERT statement, partitions can interrupt each other's execution.

2.6 GENERAL FORMS

The general forms for the BASIC-2 functions and statements that accompanied the last four releases are presented on the following pages. Release 2.2 introduced the $ALERT statement and the #ID function. Release 2.4 introduced the DATE and TIME statements and functions, and the $DISCONNECT and INPUT SCREEN statements. Release 2.5 introduced the $CLOSE, $GIO, $OPEN, SCRATCH DISK ', SELECT TC, SELECT TERMINAL, AND TRACE DISK statements.

2.6.1 $ALERT

General Form:

```
$ALERT partition
```

Where:

partition = a numeric-expression specifying a partition number.

The $ALERT statement generates an interrupt to the specified partition. In order for the interrupt to have any effect, the alerted partition must execute a SELECT ON ALERT GOSUB statement. The SELECT ON ALERT GOSUB statement defines that alert interrupts are to be fielded and indicates a subroutine to execute when an alert interrupt occurs. (Refer to SELECT ON in Chapter 8 of the BASIC-2 Language Reference Manual.)

When an alert interrupt is acknowledged, the programmer knows that at least one $ALERT statement has been executed by some partition since the last occurrence of a $ALERT interrupt or a LOAD, CLEAR, or RUN command. The programmer does not know which partition executed the $ALERT, or whether or not several $ALERTs have been executed since the last $ALERT interrupt was acknowledged.

Example:

```
500 $ALERT 5: REM ALERT Partition 5
```

If Partition 5 has enabled alert interrupts, the alert interrupt is fielded as soon as Partition 5 comes to the end of processing a BASIC statement that is not in the interrupt handling subroutine.

Examples of valid syntax:

```
$ALERT 5
$ALERT T(N)
```

2.6.2 $CLOSE

General Form:

```
                              device-address          device-address
    $CLOSE [line-number,]     file-number      [,     file-number      ] ...
                              port-number             port-number
```

Where:

device-address	=	/taa, where t is the device-type and aa is the physical device-address.
file-number	=	#n, where n is integer or numeric variable with a value 0 to 15, inclusive.
port-number	=	/Add, where dd is decimal value from 02 to 16.

The $CLOSE statement releases the specified devices and TC ports that are currently restricted by the $OPEN statement. If no device-address or port-number is specified, the $CLOSE statement releases all devices and TC ports opened for the current partition.

The specified port must be selected as a TC port previously, otherwise the system generates an illegal device error. If the TC port has not been opened previously, the system does not generate an error.

Once open, the current partition maintains exclusive control of the port until one of the following conditions occurs:

- Execution of a $CLOSE statement
- Execution of a program END statement
- Execution of a CLEAR or LOAD RUN command
- Execution of a SELECT TERMINAL statement
- Pressing of the RESET key

Examples of valid syntax:

```
$CLOSE
$CLOSE /A15, #4
```

2.6.3 DATE

General Form (as a statement):

```
                alpha-variable                  alpha-variable
    DATE =                          PASSWORD
                literal-string                  literal-string
```

General Form (as a function):

```
    DATE
```

DATE Statement

The DATE statement sets or changes the system date. The programmer can specify the new ASCII date in an alpha-variable or a literal-string. The date is specified as a six character alphanumeric value in the form YYMMDD (year, month, day).

The alpha-variable or literal-string following the keyword PASSWORD represents the system password. If the given password is correct (i.e., matches @GENPART password), the system date is updated. Otherwise, the system returns a recoverable error (ERR X79 - Invalid Password).

The system date can be set even if there is no system clock. This capability allows a clockless system to maintain the system date manually.

Examples of valid syntax:

```
DATE = "820801" PASSWORD "SYSTEM"
DATE = D$ PASSWORD P$
```

DATE Function

DATE is an alphanumeric function that returns a six character ASCII string containing the current date in the form YYMMDD (year, month, day). DATE is used as an operand in alphanumeric expressions.

Examples of valid syntax:

```
T$ = DATE & TIME
A$ = DATE
```

2.6.4 $DISCONNECT

General Form:

```
                ON  [expression]
$DISCONNECT
                OFF
```

Where:

0 ≤ value of expression ≤ 65534

The $DISCONNECT statement enables or disables terminal disconnection detection. Once enabled, the operating system can detect when a terminal is disconnected. A local terminal is considered to be disconnected when the terminal is powered off. A remote terminal is disconnected when telephone communication is terminated. The operating system cannot distinguish between disconnections caused by timeout, modem disconnection, or terminal power off.

A $DISCONNECT ON statement enables disconnect detection. A $DISCONNECT OFF statement disables disconnect detection; this is the default state of the system. Issuing a subsequent $DISCONNECT statement always overrides the previous command and sets a new state.

The optional expression in the $DISCONNECT ON statement represents the time in seconds after which the operating system forces a disconnection. If specified, the operating system decreases the value of the expression by one each second and forces a disconnection of the terminal when the value of the expression equals zero. The counter remains at zero until another $DISCONNECT ON statement sets the time expression. Therefore, even after the terminal is disconnected, the next terminal at the same port is not disconnected until the execution of another $DISCONNECT ON statement.

$DISCONNECT is a command to the port attached to the partition issuing the statement and not to the partition. Therefore, changing partitions or terminals does not affect the way $DISCONNECT affects a specific port (i.e., $DISCONNECT remains in effect after the execution of a $RELEASE PART or $RELEASE TERMINAL statement).

Terminal Connect Detection

A local terminal is considered to be connected when it is powered on. A remote terminal is considered to be connected when the phone line link between the terminal and the terminal processor is established.

When the operating system detects a terminal connection or a RESET from a terminal that does not control any partitions, it automatically assigns an available partition to that terminal. All partitions assigned to the null terminal (i.e., terminal 0) and waiting for a terminal (i.e., having accessed a statement that performs I/O to the terminal, such as PRINT or LINPUT) are available partitions. If the terminal already controls one or more partitions, or if no available partitions exist, the operating system performs no action.

Once a partition is assigned to a terminal, this partition can then execute a Logon program that issues a $DISCONNECT ON statement and sets a time limit for you to complete the Logon sequence. (Refer to Example 1, "Sample Logon Program".) If you do not complete the logon sequence within the program specified time, the operating system automatically disconnects you. Since you can initiate but cannot complete the logon procedure, forced disconnection prevents you from tying up a terminal port. If, however, you properly complete the logon procedure, the forced disconnection can be overridden by executing a $DISCONNECT ON statement with no specified disconnect time.

Example 1: Sample Logon Program

```
0010 REM @CONNECT -- Sample Logon Program
0020 REM To log system users, run this program (or similar
   : REM program) in all released partitions.  When a terminal is
   : REM connected, the operating system assigns a partition to
   : REM the terminal and begins execution of this program.
0025 REM For protection, the partition should be nonprogrammable.
0030 REM%

Disconnect terminal if not logged on within 2 minutes

   : $DISCONNECT ON 120
0040 REM%

Title

   : PRINT HEX(03); AT(0,30); "LOG ON"
0050 REM%

Obtain user's password

   : PRINT AT(10,20,8); "Enter password: ";
   : FOR I = 1 TO 8
   : KEYIN STR(P$,I,1)
   : PRINT HEX(8B);
   : NEXT I
0060 REM%

Verify user's password and log on user

   : REM Perform any required logon procedures

0070 REM%

Turn off the disconnect timeout

   : $DISCONNECT ON
0080 REM%

Load menu

   : LOAD RUN T "START"
   : REM START should check that user has logged on
```

Terminal Disconnect Detection

The operating system is informed of disconnections only if disconnect is enabled. Upon detection of a disconnect, the operating system forces all partitions assigned to that terminal to run a user-written BASIC program called @DISCNCT. (Refer to Example 2, "Sample Logoff Program".) The process is functionally equivalent to typing RESET followed by LOAD RUN "@DISCNCT". However, running the program occurs independently of the terminal.

The @DISCNCT program can perform any required Logoff or accounting procedures concerning system use. Typically, this program makes the partition available to other users by executing a $RELEASE PART statement. Thereafter, the released partition can execute a Logon program that interacts with the user assigned to this partition when terminal connections are detected.

If disconnect detection is not enabled for the terminal port, the operating system does not perform any action following the disconnection.

Example 2: Sample Logoff Program

```
0010 REM%

@DISCNCT -- Sample Logoff Program

0020 REM If terminal disconnect detection is enabled and a
   : REM terminal disconnects, the operating system automatically
   : REM runs the @DISCNCT program in all partitions assigned to
   : REM that terminal
0030 REM%

Release the partition

   : $RELEASE PART
0040 REM%

Log off the user

   : REM Perform any required Logoff procedures
0050 REM%

Load a Logon program

   : LOAD T "@CONNECT"
```

Examples of valid syntax:

```
$DISCONNECT ON
$DISCONNECT ON 60
$DISCONNECT OFF
```

2.6.5 $GIO

General Form:

```
                    file-number,
   $GIO [comment]                  (arg-1 [,arg-2])[arg-3 [,arg-3 ...]]
                    port-number,
```

Where:

file-number	=	#n, where n is integer or numeric variable with a value 0 to 15, inclusive.
port-number	=	/Add, where dd is decimal value from 02 to 16.

The specified port must be selected as a TC port previously, otherwise the system generates an illegal device error. If the specified port is opened by another partition, the $GIO statement waits until the specified port is available.

During $GIO through TC port, the device address change commands, 71hh and 73r0, are not allowed.

For a description of the $GIO commands for asynchronous communications, refer to the Asynchronous Communications User Guide for Model 2236MXE Terminal Processor and Option-W Terminal Processor (700-8098).

2.6.6 #ID Function

General Form:

```
#ID
```

The #ID function returns the value of the CPU identification number. The value is a number from 1 to 65535. With the #ID function, a program can distinguish one CPU from another. This capability is useful in licensing software to specific installations.

2.6.7 INPUT SCREEN

General Form:

```
INPUT SCREEN alpha-variable
```

The INPUT SCREEN statement reads the screen of the terminal attached to the current partition and stores an image of the screen in the alpha-variable. Each character on the screen and its associated display attributes are represented in the screen image.

INPUT SCREEN can only be used with controllers (e.g., Model 2236MXE or SVP Option-W) and terminals (e.g., Model 2336DW or 2336DE) that support the screen input facility. Attempting to execute an INPUT SCREEN statement with other terminals or controllers results in an error.

During a screen input, the terminal sends a total of 4080 bytes (characters) to the alpha-variable. Therefore, the alpha-variable should be at least 4080 characters in length in order to receive a full screen. The screen image consists of the following items:

- Terminal self-identification message
- Current cursor position
- Characters currently displayed
- Display attributes for each character

The first 78 bytes contain the self-identification message. The message identifies the type of terminal and its character set. The next 2 bytes identify the cursor location. The first byte identifies the row position; the second byte identifies the column position. Rows are numbered 0 to 24; columns are numbered 0 to 79.

The following 2000 bytes (25 rows by 80 columns) represent the characters currently displayed. The terminal uses the character codes of the Alternate Character Set (refer to Figure 2-1). Codes are sent row by row, starting at the first character in the first row. The 25th row of characters is all zeroes.

Each character has an associated attribute byte describing how the character is displayed. The 2000 bytes representing the characters are followed by 2000 bytes representing the attributes. The 25th row of display attributes is only used for the box graphics under the characters of the 24th row. A display attribute byte has the following format:

```
bit 80 = 1 if character graphic
bit 40 = 1 if reverse video
bit 20 = 1 if blink
bit 10 = 1 if high intensity
bit 08 = 1 if underline
bit 04 = 1 if left horizontal box graphic segment
bit 02 = 1 if right horizontal box graphic segment
bit 01 = 1 if vertical box graphic segment
```

Only a foreground partition (i.e., a partition with a terminal attached) can execute an INPUT SCREEN statement. If a background partition with a terminal assigned to it issues INPUT SCREEN, execution is suspended until the terminal is attached to this partition.

Operator-Initiated Screen Input

Executing INPUT SCREEN in a partition with no terminal assigned to it (i.e., a $RELEASE PART statement is executed) notifies the operating system that this partition is available to receive screen input initiated by a terminal operator. Execution is suspended until a request for screen input is received.

To initiate a request for screen input, the operator must hold down SHIFT and EDIT for at least two seconds. The operating system then temporarily attaches the terminal to a partition waiting to receive screen input. If no such partition is available, the screen input request remains pending until a partition becomes available, or until the operator aborts the request. The operator can abort the request by pressing RESET, or SHIFT and RESET. Pressing RESET terminates the screen input request and leaves the CRT screen intact. Pressing SHIFT and RESET terminates the request and clears the screen.

To handle more than one screen input concurrently, more than one partition can be set up to receive screen input. The operating system selects one of the available partitions for each screen input. Once the terminal is attached to a partition, the INPUT SCREEN statement resumes execution.

When INPUT SCREEN is finished, the partition remains assigned to the terminal, but becomes a background partition. The programmer can identify the terminal that executed the screen input request by using the #TERM function. As soon as the screen image is processed, the partition can be made available for another screen input by reexecuting the $RELEASE PART and INPUT SCREEN statements.

Example:

The following example outlines how to structure a program to receive screen input:

```
10 DIM A$(51)80
20 REM Release partition from any terminal
 : $RELEASE PART
30 REM Wait for screen input
 : INPUT SCREEN A$( )
40 REM Process screen input
50 GOTO 20
```

Example of valid syntax:

```
INPUT SCREEN A$( )
```

High-order HEX Digit

Low-order HEX Digit

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	â	Space	0	@	P	°	p	.	•			□	◘	◙	■
1	ê	!	1	A	Q	a	q	♦	◊			◚	◛	◜	◝
2	î	"	2	B	R	b	r	▶	▲			◞	◟	◠	◡
3	ô	#	3	C	S	c	s	◀	▼			◢	◣	◤	◥
4	û	$	4	D	T	d	t	→	↓			◦	◧	◨	◩
5	ä	%	5	E	U	e	u	⌙	¬			◪	◫	◬	◭
6	ë	&	6	F	V	f	v	|	√			◮	◯	◰	◱
7	ï	'	7	G	W	g	w	··	○			◲	◳	◴	◵
8	ö	(	8	H	X	h	x	'	{			◶	◷	◸	◹
9	ü	)	9	I	Y	i	y	`	}			◺	◻	◼	◽
A	à	*	:	J	Z	j	z	^	Δ			◾	◿	▀	▁
B	ë	+	;	K	[	k	§	■	□			▂	▃	▄	▅
C	ù	,	<	L	\	l	£	!!				▆	▇	█	▉
D	Ä	-	=	M	]	m	é	↕				▊	▋	▌	▍
E	Ö	.	>	N	↑	n	ç	ß				▎	▏	▐	▒
F	Ü	/	?	O	←	o	¢	¶				▓	▔	▕	▖

Figure 2-1. Alternate Character Set

2.6.8 $OPEN

General Form:

```
                              device-address          device-address
$OPEN [line-number,]          file-number       [,    file-number       ] ...
                              port-number             port-number
```

Where:

device-address = /taa, where t is the device-type and aa is the physical device-address.

file-number = #n, where n is integer or numeric variable with a value 0 to 15, inclusive.

port-number = /Add, where dd is decimal value from 02 to 16.

The $OPEN statement requests exclusive use of a device (i.e., peripherals) or TC port for the current partition. To request exclusive use of a device, the programmer specifies either the device-address or the file-number in the $OPEN statement. To request exclusive use of a TC port, the programmer specifies the port-number in the $OPEN statement.

The system signals an ERR P48 - Illegal Device Specification if one of the following conditions occurs:

- The specified device is not in the Master Device Table.
- The specified device is already opened for exclusive use by another partition.
- The specified port was not selected as a TC port before executing the $OPEN statement.

Once open, the current partition maintains exclusive control of the device or port until one of the following conditions occurs:

- Execution of a $CLOSE statement
- Execution of a program END statement
- Execution of a CLEAR or LOAD RUN command
- Execution of a SELECT TERMINAL statement
- Pressing of the RESET key

If another partition has already opened the specified device or TC port, the $OPEN statement branches to the specified line-number. If no line-number is specified, the $OPEN statement waits until the specified device or TC port becomes available for this partition.

If the programmer specifies multiple devices (including TC ports) in a $OPEN statement and the system cannot open one of the devices, then $OPEN does not open any of the devices.

Examples of valid syntax:

```
$OPEN /A13, /215, /A07
$OPEN /A02, #4
```

2.6.9 SCRATCH DISK '

General Form:

```
                                file #
SCRATCH DISK ' platter                          [LS = exp-1,]  END = exp-2
                                disk-address,
```

Where:

'	=	an indication that the new disk index structure is used. The absence of the single quote indicates that the old disk index structure is used.
LS	=	a parameter specifying the number of sectors to be set aside for the Catalog Index.
exp-1	=	an integer or expression whose value is from 1 to 255. If the LS and expression-1 parameters are not included, the system automatically sets the size of the Catalog Index to 24 sectors.
END	=	a parameter specifying the last (highest) sector address in the Catalog Area.
exp-2	=	an expression whose value must be less than or equal to the last (highest) sector address on the disk.

The SCRATCH DISK ' statement creates a disk index with the new format. The new disk index provides faster access to files by providing a better distribution of file entries in the catalog than is provided with indexes created by the SCRATCH DISK statement.

Examples of valid syntax:

```
SCRATCH DISK ' T/D25, LS = 50, END = 52607
SCRATCH DISK ' T/D10, END = 3873
```

2.6.10 SELECT TC

General Form:

SELECT TC port-number

Where:

port-number	=	/Add, where dd is a decimal value from 02 to 16. <alpha-variable>, where alpha-variable contains the port-address Add.

The SELECT TC statement selects the specified port as a telecommunications (TC) port instead of a terminal port. No partition, including the partition issuing the SELECT TC statement, should be attached or assigned to the specified port when the SELECT TC statement is executed. If a partition is assigned or attached to the port, the system generates an illegal device error (P48). Therefore, a programmer should release all partitions assigned to the requested port by issuing $RELEASE PART statements before issuing a SELECT TC statement or configure the system with no partitions assigned to the TC port.

The device address /Add or <alpha-variable> indicates which port to select. Port 01 cannot be selected as a TC port since Port 01 serves as the terminal port for the system console. If the specified port is selected as a TC port already, the system does not generate an error.

To avoid losing control of a port while selecting it as a TC port, perform the following. This example assumes that A$ is set to the device-address.

```
$BREAK
SELECT #1 <A$>
$RELEASE PART
SELECT TC <A$>
$OPEN #1
```

The port designated by A$ should be attached to the partition initially.

Examples of valid syntax:

```
SELECT TC /A03
SELECT TC /A12
SELECT TC <A$>
```

2.6.11 SELECT TERMINAL

General Form:

```
SELECT TERMINAL port-number
```

Where:

port-number	=	/Add, where dd is a decimal value from 02 to 16. <alpha-variable>, where alpha-variable contains the port-address Add.

The SELECT TERMINAL statement selects a port as a terminal port or converts a TC port to a terminal port. The specified port should not be opened (by means of a $OPEN statement) as a TC port when executing a SELECT TERMINAL statement. If the specified port has been opened as a TC port by another partition, the system generates an illegal device error. When the system is powered on, the system selects all ports as terminal ports.

Examples of valid syntax:

```
SELECT TERMINAL /A04
SELECT TERMINAL <B$>
```

2.6.12 TIME

General Form (as a statement):

```
                alpha-variable                  alpha-variable
    TIME =                          PASSWORD
                literal-string                  literal-string
```

General Form (as a function):

```
    TIME
```

TIME Statement

The TIME statement sets or changes the system time. The programmer can specify the new ASCII time in an alpha-variable or a literal-string. The time is specified as a six character alphanumeric value in the form HHMMSS (hours, minutes, seconds).

The alpha-variable or literal-string following the keyword PASSWORD represents the system password. If the given password is correct (i.e., matches @GENPART password), the system time is updated. Otherwise, the system returns a recoverable error (ERR X79 - Invalid Password).

The system time can only be set if a system clock exists. If a clock does not exist and a user attempts to set the time, the system signals an error.

Examples of valid syntax:

```
TIME = "170001" PASSWORD "SYSTEM"
TIME = T$ PASSWORD P$
```

TIME Function

TIME is an alphanumeric function that returns an eight character ASCII string containing the current time in the form HHMMSSCC (hours, minutes, seconds, centiseconds). TIME is used as an operand in alphanumeric expressions.

The TIME function returns a value of 99999999 (i.e., invalid) if there is no system clock.

Examples of valid syntax:

```
A$ = TIME & " " & DATE
T$ = TIME
```

2.6.13 TRACE DISK

General Form:

TRACE DISK [OFF]

The TRACE DISK statement enables the disk tracing facility. The TRACE DISK OFF statement disables the disk tracing facility. If disk tracing is not enabled, the system ignores the TRACE DISK OFF statement.

TRACE DISK lists all disk activity performed by the system for the current partition. The system sends the trace output to the currently selected CO device (refer to the SELECT statement in the Wang BASIC-2 Language Reference Manual.) The trace remains in effect until a TRACE DISK OFF statement is executed or RESET is pressed.

The TRACE DISK statement is useful in analyzing the disk activity of application packages.

The trace disk output has one of the following forms:

/taa hh ddddd pp or /taa 20hh ddddd pp

where:

/taa = the address of the disk drive being accessed.

hh or 20hh = a disk command, where:

00 = Read sector
40 = Write sector
80 = Compare sector (i.e., read-after-write)
20hh = Special command to Disk Processing Unit (DPU)

Special commands to the DPU include service and diagnostics operations, as well as, the following operations:

2001 - Copy sectors
2002 - Format platter
2010 - Start of multisector write
2011 - End of multisector write
2012 - Verify sectors

Not all 2200 disk units use 20hh commands.

ddddd = the sector being accessed. The sector number is in decimal starting from 0.

pp = the partition issuing the disk command.

Example:

```
TRACE DISK

/D10 00   00000 01
/D10 00   00011 03
/D10 00   00012 03
/D12 40   00123 11
/D11 2004 01234 10
```

Examples of valid syntax:

```
TRACE DISK
TRACE DISK OFF
```

CHAPTER 3
INITIALIZE DATE & TIME UTILITY

3.1 INTRODUCTION

The operator can select the Initialize Date & Time utility from the System Utilities menu (refer to Figure 3-1). The Initialize Date & Time utility (@CLOC) allows the operator to edit the system date and, if a system clock exists, the system time. A system equipped with a clock automatically advances the date at midnight. If the system does not have a clock, the operator must reset the date each day.

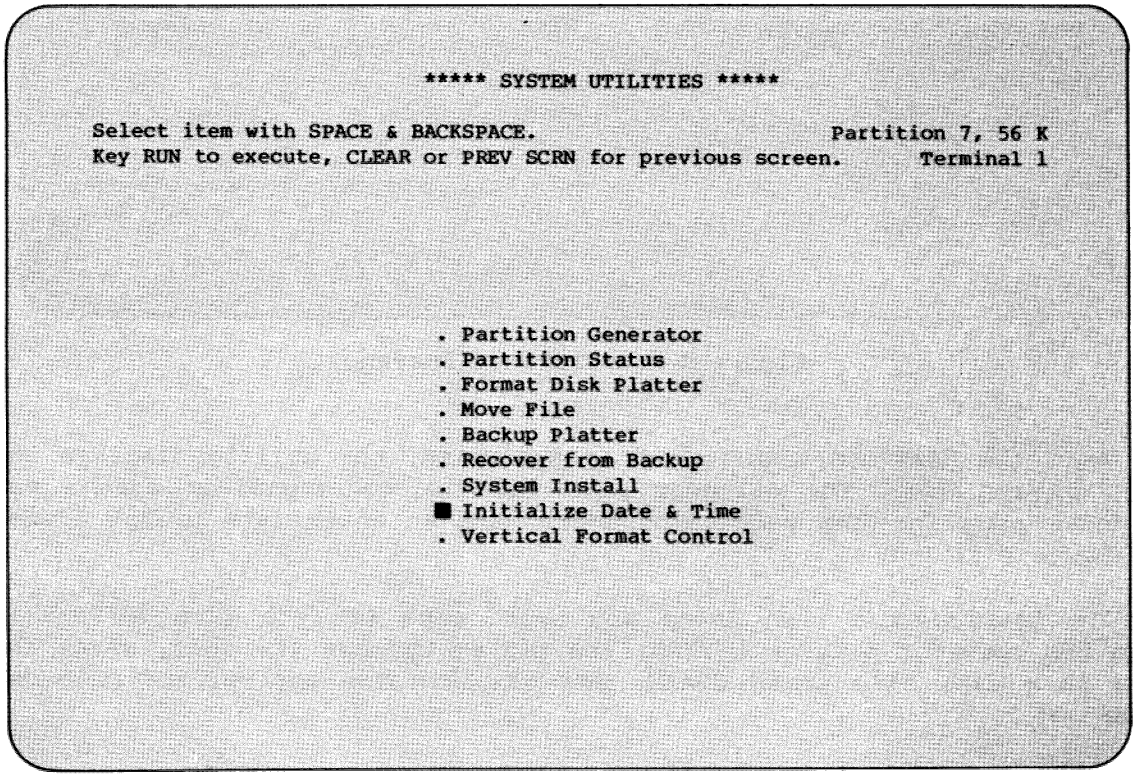

Figure 3-1. System Utilities Menu

3.2 OPERATION

Whenever the system password is not SYSTEM, the utility first requests the system password (refer to Figure 3-2). The operator enters the system password and presses RETURN to proceed editing the date and/or time.

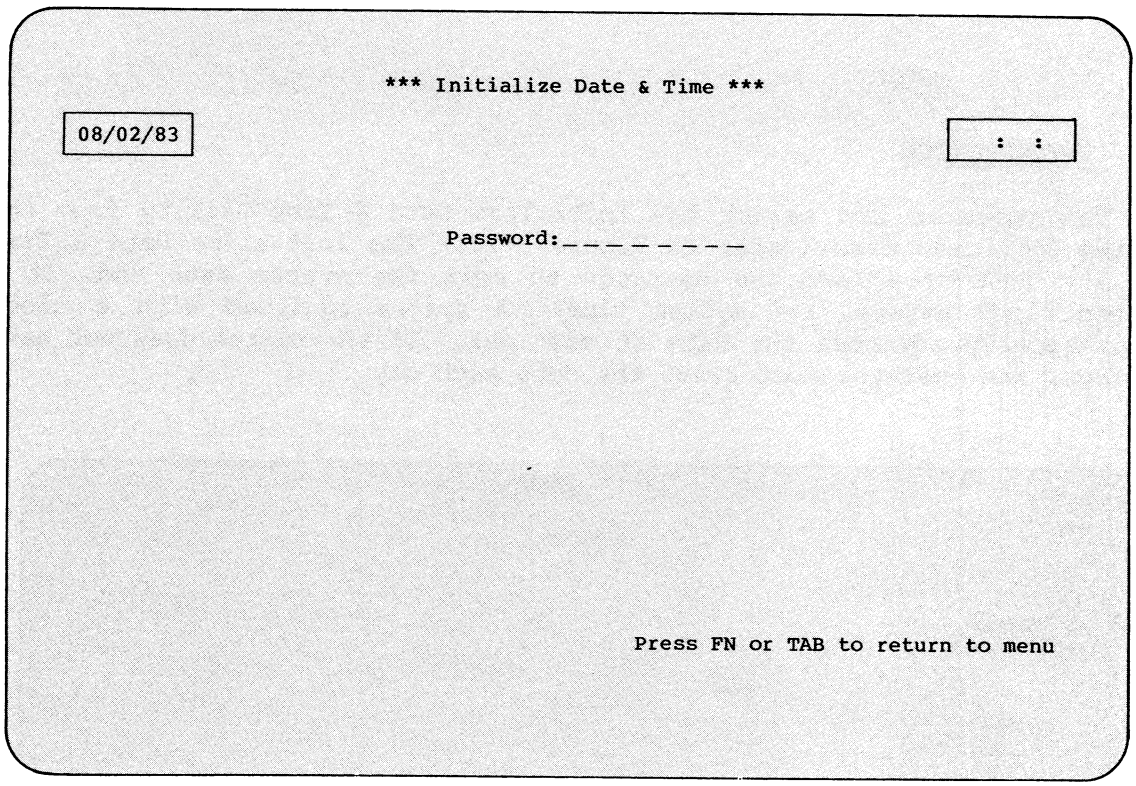

Figure 3-2. Password Screen

When the password requirement is satisfied, the utility displays the last entered date, the calendar page for the corresponding month, the calendar page for the succeeding month, and an enter date prompt (refer to Figure 3-3).

The operator can then enter any valid date (in the form MM/DD/YY). When the operator presses RETURN, the system updates the date, saves the date in a data file named @DATE, and displays the appropriate calendar pages.

If the system permits, the operator can enter any valid time (in the form HH:MM:SS). When the operator presses RETURN, the system updates the time.

At this point, the operator can re-edit the date and/or time. Pressing FN or TAB accepts the date and/or time and returns the System Utilities menu to the screen.

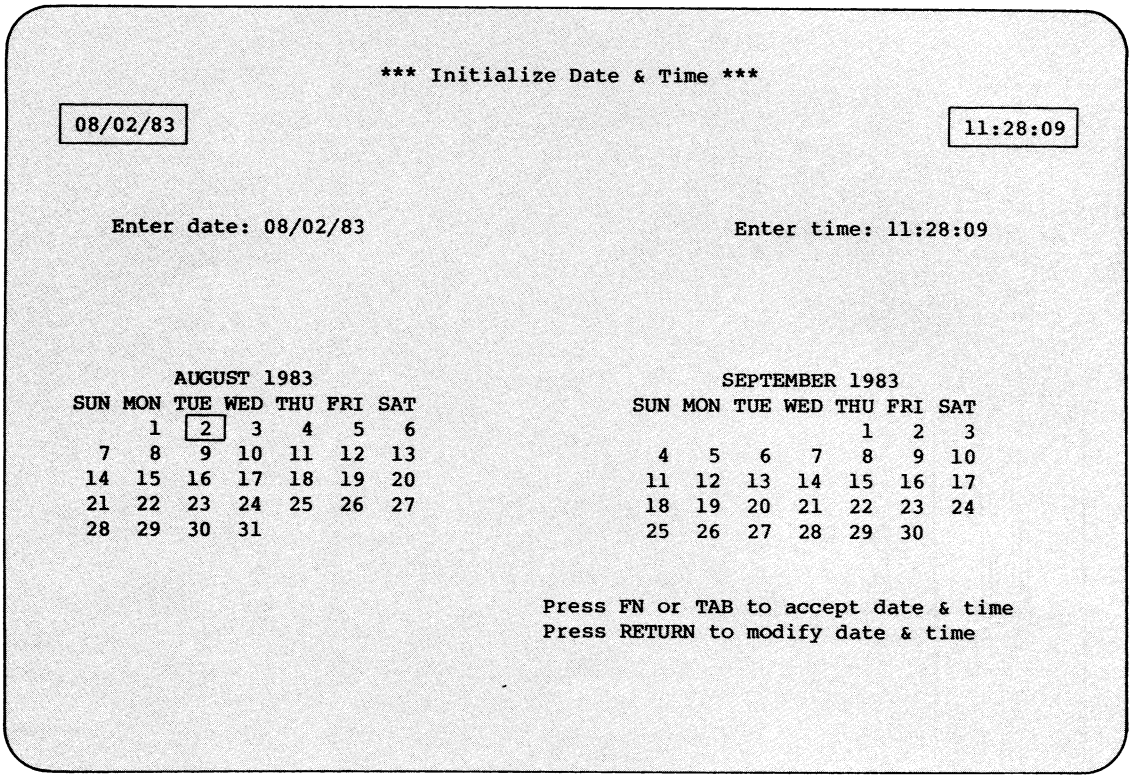

Figure 3-3. Edit Date & Time Screen

WANG

Customer Comment Form

Publication Number 715-0097A

Title 2200 BASIC-2 MULTIUSER OPERATING SYSTEM SOFTWARE BULLETIN RELEASE 2.6

Help Us Help You . . .

We've worked hard to make this document useful, readable, and technically accurate. Did we succeed? Only you can tell us! Your comments and suggestions will help us improve our technical communications. Please take a few minutes to let us know how you feel.

How did you receive this publication?

- ☐ Support or Sales Rep
- ☐ Wang Supplies Division
- ☐ From another user
- ☐ Enclosed with equipment
- ☐ Don't know
- ☐ Other _____

How did you use this Publication?

- ☐ Introduction to the subject
- ☐ Classroom text (student)
- ☐ Classroom text (teacher)
- ☐ Self-study text
- ☐ Aid to advanced knowledge
- ☐ Guide to operating instructions
- ☐ As a reference manual
- ☐ Other _____

Please rate the quality of this publication in each of the following areas.

	EXCELLENT	GOOD	FAIR	POOR	VERY POOR
Technical Accuracy — Does the system work the way the manual says it does?	☐	☐	☐	☐	☐
Readability — Is the manual easy to read and understand?	☐	☐	☐	☐	☐
Clarity — Are the instructions easy to follow?	☐	☐	☐	☐	☐
Examples — Were they helpful, realistic? Were there enough of them?	☐	☐	☐	☐	☐
Organization — Was it logical? Was it easy to find what you needed to know?	☐	☐	☐	☐	☐
Illustrations — Were they clear and useful?	☐	☐	☐	☐	☐
Physical Attractiveness — What did you think of the printing, binding, etc?	☐	☐	☐	☐	☐

Were there any terms or concepts that were not defined properly? ☐ Y ☐ N If so, what were they? _____

After reading this document do you feel that you will be able to operate the equipment/software? ☐ Yes ☐ No ☐ Yes, with practice

What errors or faults did you find in the manual? (Please include page numbers) _____

Do you have any other comments or suggestions? _____

Name _____ **Street** _____

Title _____ **City** _____

Dept/Mail Stop _____ **State/Country** _____

Company _____ **Zip Code** _____ **Telephone** _____

Thank you for your help.

Printed in U.S.A. 14-3140 7-83-5C

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 LOWELL, MA

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
PUBLICATIONS DEVELOPMENT
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Fold

Cut along dotted line.

WANG

To Order by Phone, Call:
(800)225-0234
From Mass., Hawaii, and Alaska
(617)256-1400
TELEX 951-743

Order Form for Wang Manuals and Documentation

① Customer Number (If Known)

② Bill To: Ship To:

③ Customer Contact: ( ) ( ) ④ Date Purchase Order Number
Phone Name

⑤ Taxable ⑥ Tax Exempt Number ⑦ Credit This Order to
Yes ☐ A Wang Salesperson
No ☐ Please Complete Salesperson's Name Employee No. RDB No.

⑧ Document Number	Description	Quantity	⑨ Unit Price	Total Price

⑩ _____ _____
Authorized Signature Date

☐ Check this box if you would like a free copy of the **Corporate Publications Literature Catalog** (700-5294)

Sub Total	
Less Any Applicable Discount	
Sub Total	
LocalStateTax	
Total Amount	

Ordering Instructions

1. If you have purchased supplies from Wang before, and know your Customer Number, please write it here.
2. Provide appropriate Billing Address and Shipping Address.
3. Please provide a phone number and name, should it be necessary for WANG to contact you about your order.
4. Your purchase order number and date.
5. Show whether order is taxable or not.
6. If tax exempt, please provide your exemption number.
7. If you wish credit for this order to be given to a WANG salesperson, please complete.
8. Show part numbers, description and quantity for each product ordered.
9. *Pricing extensions and totaling can be completed at your option; Wang will refigure these prices and add freight on your invoice.*
10. Signature of authorized buyer and date.

Wang Supplies Division Terms and Conditions

1. **TAXES** — Prices are exclusive of all sales, use, and like taxes.
2. **DELIVERY** — Delivery will be F.O.B. Wang's plant. Customer will be billed for freight charges; and unless customer specifies otherwise, all shipments will go best way surface as determined by Wang. Wang shall not assume any liability in connection with the shipment nor shall the carrier be construed to be an agent of Wang. If the customer requests that Wang arrange for insurance the customer will be billed for the insurance charges.
3. **PAYMENT** — Terms are net 30 days from date of invoice. Unless otherwise stated by customer, partial shipments will generate partial invoices.
4. **PRICES** — The prices shown are subject to change without notice. Individual document prices may be found in the Corporate Publications Literature Catalog (700-5294)
5. **LIMITATION OF LIABILITY** — In no event shall Wang be liable for loss of data or for special, incidental or consequential damages in connection with or arising out of the use of or information contained in any manuals or documentation furnished hereunder.

Printed in U.S.A. 14-3141 7-83-5C

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 NO. CHELSMFORD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
Supplies Division
c/o Order Entry Dept.
M/S 1711
800 Chelmsford Street
Lowell, MA 01851

Fold

WANG

ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851
TEL. (617) 459-5000
TWX 710-343-6769, TELEX 94-7421

Printed in U.S.A.
715-0097A
5-85

To: Ken Pappas

From: Dave Barrett

Cc: Scott Tagen

Date: August 27, 1985

Subject: MVP (Multi-User) Basic-2 Maintenance Pre-Release 2.6.3

Enclosed is a pre-release copy of the MVP (Multi-User) Basic-2 Operating System Interim Maintenance Release 2.6.3 for evaluation. All Probe Reports against the operating system itself that were sent to R&D before 8/27/85 are corrected by this release. In addition, new versions of @GENPART and @MOVEFIL are enclosed. These files contain fixes to correct Probes F008620 and F009320.

Other Probes such as F008852 and F009073 have not been fixed yet as they are problems with the MXE micro-code or 2229 tape utilites. These problems will be corrected at a later date.

The following Probe Reports are now closed and are fixed by this release.

PROBE NUMBER	DESCRIPTION
F008620	@MOVEFIL does not span files correctly
F008843	Restricting disks to single partitions
F008844	INPUT statement errors 'eating' user memory
F009320	@GENPART error when deleting 15th printer driver table entry
F009363	$INIT command errors with 6 printer driver tables entries specified
F009496	$RELEASE TERMINAL TO #,STOP does not halt destination partition
F400162	LIST DC wildcards not functioning properly

The following addition is to be made to the MVP 2.6. Software Release Bulletin page 1-24 under the heading "LIST DCT COMMAND ENHANCEMENTS".

LIST DCT masks now have the following restrictions placed on them.

1) Only one '*' wildcard is allowed per mask. The presence of more than one in a mask will prevent any matches from being made.
2) A '?' wildcard may not immediately follow a '*' wildcard. Doing so will prevent any matches from being made.
3) Trailing blanks in filename are ignored.