define-terminal Command

The **define-terminal** BASIC program displays the available terminal (driver) definitions and allows changing existing definitions or adding new ones.

Each terminal definition is an item in the **devices** file with the terminal name used as the item-ID.

The definitions that are marked with an asterisk are also stored as items using the terminal type codes as the item-ID.

These item-IDs are used by the **term** command to set up terminal characteristics.

Menu Option	Description
1 and 2	Requests the terminal name. If the definition for the specified terminal is on file, the definition displays. If it is not on file, the option to create a new terminal definition is offered.
3	Displays a prompt requesting confirmation of the deletion.
4	Requests the terminal name. The corresponding item in the **devices** file is flagged as the selected terminal type and the menu redisplays.
5	Requests the terminal name. The flag marking the corresponding item is deleted and the menu redisplays.

Example(s)

These terminals are defined in the **devices** file.

```
define-terminal
System Cursor Definition Utility
Terminals marked with an asterisk are selected to be included in your
system cursor definition.
*A   ADDS580       G    GTC         M    MIME        *W   wy-50-link
*A   ADM-11        *G   IBM3161     *N   WY-100      *W   wy-50132
A    ADM3A         *G   IBM3164     *P   PERTEC      *W   wy-60
A    CIE-ANT       *G   mn3161      *Q   QVT102      *W   wy50-132
*B   AMPEX210      *G   mn3161-132  *R   R25         *W   WY60
B    BEEHIVE       H    HONEYWELL   S    ansi.c      *W   WY60.1
*B   IBM3151       I    AT.MON      *S   SOROC       x    DATAGRAPHIX
*B   ibm3151-132   I    IBM         T    TEC         *x   xterm
*B   mark          I    IBM3010     T    TV920       x    xterm-dim
*B   mn3151        *I   MM-MON      *T   TV950       Z    ANSI
C    DTC           *I   port-mon    *V   VP          Z    att605
*c   VT-52         *J   VT100       V    VP-60       Z    att705
*D   DATAMEDIA     J    vt100-dim   *V   VP-A2       Z    HFT
D    dg            J    vt320       W    link        Z    rt.console
E    EMULOG200     *K   prism       *W   link125     Z    RT.MON
E    EX-34         *L   LSI         *W   w           Z    Z
*F   TV910         *M   AMPEX-D80   *W   WY-50
*G   g             *M   AMPEX.D80   *W   WY-50-132

1)   Create Terminal Definition
```

```
2)   Modify Terminal Definition
3)   Delete Terminal Definition
4)   Add Terminal To Selected Definitions
5)   Delete Terminal From Selected Definitions
EX   Exit DEFINE-TERMINAL To TCL

Enter Selection (1-5) or EX :
```

To modify your terminal, enter **2**.

```
Modify Terminal
Enter Existing Terminal Name To Be Modified ?prism
Terminal: prism
1.    Term                             K
      Type.........................
2.    Terminal                         Prism-IV
      Description...............
3.    Backspace and Screen             8,80,24
      Size..........
4.    Cursor Address                                          A
      Code................
5.    @(X,Y)                           Cursor                 CHAR(11)
                                       Addressing..........   Y
                                                              CHAR(16)
                                                              X
6.    @(-1)                            Clear Screen &         FF SOH
                                       Home........
7.    @(-2)                            Cursor                 SOH
                                       Home................
8.    @(-3)                            Clear to End of        ESC "J"
                                       Page.......
9.    @(-4)                            Clear to End of        ESC "K"
                                       Line.......
10.   @(-5)                            Start                  ESC "B"
                                       Blink................
11.   @(-6)                            Stop                   ETX
                                       Blink.................  HEX(20)
12.   @(-7)                            Start Protected
                                       Field......
13.   @(-8)                            Stop Protected
                                       Field.......
14.   @(-9)                            Cursor                 NAK
                                       Back................
15.   @(-10)                           Cursor                 SUB
                                       Up..................
16.   @(-11)                           Enable Protect
                                       Mode........
17.   @(-12)                           Disable Protect
                                       Mode.......
18.   @(-13)                           Start Reverse          ETX "D"
                                       Video........
19.   @(-14)                           Stop Reverse           ETX " "
                                       Video.........
20.   @(-15)                           Start                  ETX "0"
```

```
                                                      Underline............
21.   @(-16)                                          Stop                     ETX ""
                                                      Underline............    ""
22.   @(-17)                                          Slave                    ETX "R"
                                                      On...................
23.   @(-18)                                          Slave                    ETX "T"
                                                      Off..................
24.   @(-19)                                          Cursor                   ACK
                                                      Forward............
25.   @(-20)                                          Cursor                   LF
                                                      Down...............
26.   @(-21)                                          Graphics Character Set
                                                      On..
27.   @(-22)                                          Graphics Character Set
                                                      Off.
28.   @(-23)                                          Keyboard
                                                      Lock.............
29.   @(-24)                                          Keyboard
                                                      Unlock...........
30.   @(-25)                                          Control Character
                                                      Enable...
31.   @(-26)                                          Control Character
                                                      Disable..
32.   @(-27)                                          Write Status             HEX(E3)
                                                      Line.........
33.   @(-28)                                          Erase Status             HEX(E5)
                                                      Line.........
34.   @(-29)                                          Initialize Terminal
                                                      Mode...
35.   @(-30)                                          Download Function
                                                      Keys.....
36.   @(-31)                                          Non-embedded Stand-out
                                                      On..
37.   @(-32)                                          Non-embedded Stand-out
                                                      Off.
38.   @(-99)                                          Embedded Visual
                                                      Attributes?
39.   @(-100)                                         Half
                                                      Intensity............
40.   @(-101)                                         Full
                                                      Intensity............
Is table for terminal prism correct? y
```

Terminal Type	A single letter which identifies the terminal to the system. The `Terminal Type` field in the terminal table corresponds to the single-character value typed in at the **`term`** command.
Screen Size	Size of the display screen (in columns by rows). It is entered as *columns,rows*. If a BASIC **`print @()`** command exceeds these values, the maximum allowable value is used.
Cursor Address Code	A series of characters used to denote cursor addressing, followed by the row and column of the position. Sometimes the row and column are in binary,

	sometimes not. Sometimes it requires column and row. Others use an ASCII code in place of binary row/column addresses. Select the method your terminal manufacturer uses: **NOTE—** X is the X-axis, Y is the Y-axis.	

	`a{+}`	Adds type addressing:
		`row = char(y+64)`
		`col = char((int(x/10)*6)+x)`

	`l{+}`	Lear-Seigler addressing:
		`row = char(y+32)`
		`col = char(x+32)`

	`t{+}`	TEC addressing:
		`row = char(-(1+y))`
		`col = char(1(1+x))`

	`h{+}`	Hazeltine addressing:
		`row = char(x)`
		`col = char(y)`

`d{cr}{+}`	Decimal addressing. It outputs a 1-3-digit binary code (each) for row and column. • **`c`** specifies the number of digits to output for the column (1-3); • **`r`** specifies the number of digits to output for the row (1-3). If used, the code forces leading zeroes. A **`d`** alone generates *floating* decimal numbers from 1 to 3 digits. • **`row = char(x)`** as 1-3 digits • **`col = char(y)`** as 1-3 digits All cursor addressing codes may be followed by a **`+`**, which adds one the row and column addresses before generating the address codes, to take care of the terminals whose addresses starts at 0,0 instead of 1,1.	

**`@(`*`x,y`*`)`
`Cursor Addressing`**
	String of control and escape codes need to tell the terminal that a row and column address is coming up next. For example:
	`esc "=" y x`
	This is a cursor-positioning code for a Lear-Siegler terminal. The **`y`** and **`x`** is replaced by the proper ASCII codes (and is determined by the cursor address code above).
	NOTE—`@(-1)...@(-300)` are explained in the topic, @() Function.	

Entering Values for Codes

- ASCII codes may be entered in one of several ways:
- ASCII strings.
- ASCII characters are enclosed in double quotation marks, for example, "T02".
- ASCII control codes.
- ASCII control codes may be entered using the two or three-letter word, representing the unprintable ASCII control code:

```
NUL = 00   BS = 08   DLE = 16   CAN = 24   SP = 32
SOH = 01   HT = 09   DC1 = 17   EM = 25
STX = 02   LF = 10   DC2 = 18   SUB = 26   DEL = 127
ETX = 03   VT = 11   DC3 = 19   ESC = 27
EOT = 04   FF = 12   DC4 = 20   FS = 28
ENQ = 05   CR = 13   NAK = 21   GS = 29
ACK = 06   SO = 14   SYN = 22   RS = 30
BEL = 07   SI = 15   ETB = 23   US = 31
```

- ASCII character value in decimal, which can be entered by typing the word **char** and in parenthesis (the decimal ASCII value (1-255)). For example, **char(27)**.
- ASCII character value in hexadecimal, which can be entered by typing the word **hex** and in parenthesis (the 1-byte hexadecimal value (00-ff)). For example, **hex(1b)**.
- String function—strings can be entered as a BASIC **str** function. For example, **str("*",5)** or **str(char(12),x)**.
- Cursor Address variable—there are three cursor address variables available: ***x***, ***y***, and ***z***.

They cause the specified address (byte or decimal string) to be inserted into the control string at the specified position.

x	Contains the column.
y	Contains the row.
z	Contains the previous row referenced in an **@(x,y)** code, or 0 if the last reference was an **@(-1)** or **@(-2)**.

Any combination of the above may be used on a line. Different codes are separated with a space (not a comma), for example:

```
hex(1b) soh "1ts04"
```

or

```
char(27) soh "1ts04"
```

or

```
esc soh "1ts04"
```

There are other codes, which have been defined for use, but do not display in the maintenance screens. The **dm,devices,** device item-ID item has to be edited directly to insert these codes if they are needed.

There may be other features available only on specific terminals, and there are no existing `@`function codes for them, such as uppercase on and off. To take advantage of these features:

- ASCII sequences must be hard-coded into every program (forcing the use of that terminal forever).
- A separate file of control codes is created (such as VT100 terminals), and referenced instead of the **dm,devices,** file.
- Custom codes are added to the dm,devices, file at, such as `@`**`(-1001)`** and beyond.