TECHNICAL REFERENCE

REFLECTION™
SERIES
SOFTWARE

Reflection Technical Reference Manual, Summer 1987

Preface

Welcome to Reflection. The Reflection Series was created by Walker Richer & Quinn to allow you to make the most use of your IBM PC and workalikes. Now all the stand-alone features of your PC are complemented by the ability to communicate with your HP 1000, 3000, or 9000, DEC VAX/VMS, UNIX system, or other host.

File transfers are supported to HP 3000s, VAX/VMS, UNIX systems, and PCs with Walker Richer & Quinn's own file transfer protocol. XMODEM and KERMIT, public domain file transfer programs, are also supported by Reflection. HP 1000 users can use XMODEM or KERMIT, or purchase PCLINK-1000 from an outside vendor.

HP 2623A and 2627A graphics emulation documentation has been included in this manual. If you upgrade Reflection 1 to Reflection 3 or 7, you only need a software upgrade. These products let you take advantage of your PC's graphics capabilities, as well as your host computer's graphics programs. IBM PC, XT, AT, HP Vectras, and Compaq personal computers are currently supported with these graphics emulations, although most IBM-compatible PCs with the appropriate hardware should work. Reflection 7, color graphics, requires an enhanced graphics adapter card (with at least 128K of memory) and an enhanced color display.

Reflection Series

License Agreement

In return for payment of a one-time fee for this software product, Customer receives from Walker Richer & Quinn, Inc. a license to use the product subject to the following terms and conditions:

1. The product may be used on one computer system at a time; i.e., its use is not limited to a particular machine or user but to one machine and one user at a time.
2. The software product may not be duplicated or copied except for archive purposes, program error verification, or to replace defective media. All copies must bear copyright notices contained in the original copy. The host file transfer programs are not included in this restriction.
3. The software may not be installed on a network server for access by more than one personal computer without written permission from Walker Richer & Quinn, Inc.

Purchase of this license does not transfer any right, title, or interest in the software product to the Customer except as specifically set forth in the License Agreement, and Customer is on notice that the software product is protected under the copyright laws.

Limited Warranty Statement 90-Day Limited Warranty

Walker Richer & Quinn, Inc. warrants that this product will execute its programming instructions when properly installed on a properly configured personal computer for which it is intended. Walker Richer & Quinn, Inc. does not warrant that the operation of the software will be uninterrupted or error free. In the event that this software product fails to execute its programming instructions, Customer's exclusive remedy shall be to return the diskette to Walker Richer & Quinn, Inc. to obtain replacement. Should Walker Richer & Quinn, Inc. be unable to replace the diskette within a reasonable amount of time, Customer's alternate exclusive remedy shall be a refund of the purchase price upon the return of the product and all copies. Walker Richer & Quinn, Inc. warrants the diskette upon which this product is recorded to be free from defects in materials and workmanship under normal use for a period of 90 days from the date of purchase. During the warranty period Walker Richer & Quinn, Inc. will replace diskettes which prove to be defective. Customer's exclusive remedy for any diskette which proves to be defective shall be to return the diskette to Walker Richer & Quinn, Inc. for replacement.

ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE 90-DAY DURATION OF THIS WRITTEN WARRANTY. Some states or provinces do not allow limitations on how long an implied warranty lasts, so the above limitation or exclusion may not apply to you. IN NO EVENT SHALL WALKER RICHER & QUINN, INC. BE LIABLE FOR CONSEQUENTIAL DAMAGE. Some states or provinces do not allow the exclusion or limitation of incidental or consequential damages so the above limitation may not apply to you. This warranty gives you specific rights, and you may also have other rights which vary from state to state or province to province.

LIMITATION of WARRANTY: Walker Richer & Quinn, Inc. makes no other warranty expressed or implied with respect to this product. Walker Richer & Quinn, Inc. specifically disclaims the implied warranty of merchantability and fitness for a particular purpose.

EXCLUSIVE REMEDIES: The remedies provided herein are Customer's sole and exclusive remedies. In no event shall Walker Richer & Quinn, Inc. be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

P.O. Box 31876, Seattle, WA 98103-1876 U.S.A.

FAX #: 206 322 8151 TELEX #: 311 743 WRQ UR

Telephone: (206) 324-0350

Contents

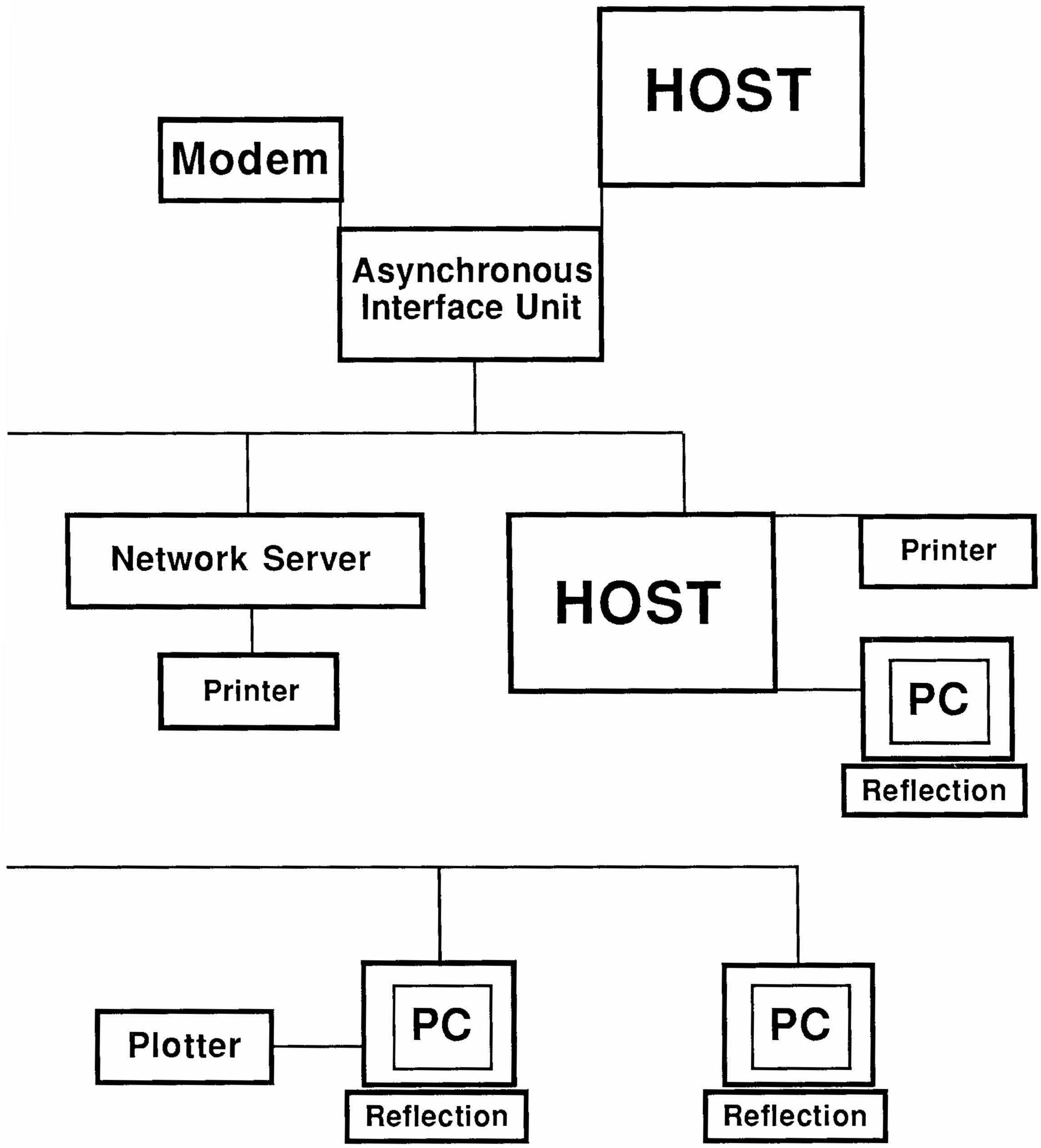
Figures

Tables

Section 1

Set-Up and Configuration

Contents



Connecting to the Host Computer

Connecting to the Host Computer

Once you've successfully connected the PC to the host, you are ready to use your PC as a terminal. Most configuration items can be set through the Basic Configuration menu. This section includes chapters on each of the configuration menus which allow you to make special settings.

NOTE: If you wish to maintain configuration files from an earlier version of Reflection, use NEWCFG.EXE. This program is included on the support files diskette. Use the following format at the DOS prompt:

C> NEWCFG <oldcfg> <newcfg>

A Quick Start Procedure

1. Find the serial communications port at the back of the PC. There may be four serial ports (referenced in this manual as COM1 to COM4), but only one at a time can be used for communicating with the host. Reflection's default data communications configuration is set for COM1, so use COM1 if you have a choice. If you have problems, consult page UM-198 in *Troubleshooting*.
2. If the host computer is close enough, connect it to the PC through the COM port by direct RS232C cable.

   Alternatively, connect COM1 to COM4 through the telephone system, using an external or internal modem. If you plan to connect by modem, you'll have to reconfigure Reflection's **Baud rate** field on the Basic Configuration menu.

   A LAN connection is also possible, see page TR-22.
3. Turn on the computer, load Reflection, set the correct baud rate, COM port, and parity on the Basic Configuration menu. Then save these values to R1.CFG (or R3.CFG/R7.CFG).

4. If you wish to emulate a VT102 terminal, press Alt-C and then select **global cfg**. Change the **Terminal class** field to *DEC* by pressing F1. When this value is activated, the current datacomm values are reset to the VT defaults. Once you have configured the DEC side, switching between terminal classes will maintain the configured datacomm values for whichever terminal you are emulating.
5. Press F4 to access the modes keys. The **REMOTE MODE** label should contain an asterisk (*). If it doesn't, press F4 to toggle remote mode on.
6. If you'll be communicating with an HP 3000 or a VAX/VMS, chances are the only fields you may have to reconfigure are the **Baud rate** and **Parity** fields.
   - HP mode defaults are set for an HP 3000 host computer, via direct connection at 9600 baud, with no parity, using the COM1 port.
   - The VT terminal emulation mode defaults are set for a VAX/VMS host computer via direct connection at 9600 baud, using COM1.
7. If the PC is connected directly to the host computer, press Return to bring up the host prompt.

   If the PC and the host are connected by modem, dial the host and establish a connection. For Hayes-compatible modems type **ATDT** followed immediately by the phone number. When *CONNECT* appears, press Return until you get the host prompt. Once a connection is established, you can log on to the host using procedures defined by the host computer.

If necessary, refer to the sections in this chapter which relate to special configuration settings required for the hosts discussed. You can also find information in the following chapters:

- *Data Communications Configuration*, page TR-27
- *Miscellaneous Host Configurations*, page TR-25
- *HP Terminal Configuration*, page TR-35
- *VT Terminal Configuration*, page TR-57

Locating the Serial Ports

To communicate with another computer, the PC needs either one serial communications port, which IBM calls an Asynchronous Communications Adapter, or a LAN connection (see page TR-22). A serial port is an option on the IBM Personal Computer, but it is part of the standard package on the IBM Personal Computer XT and many of the IBM-compatible computers.

The IBM PC's serial port is a male 25-pin D-shell connector at the back of the computer. It's called a D-shell connector because it's shaped like a long, narrow, capital letter D. (Don't confuse it with the parallel printer interface, which is a female 25-pin connector.) The IBM PC-AT uses a male 9-pin D-shell connector on its serial port.

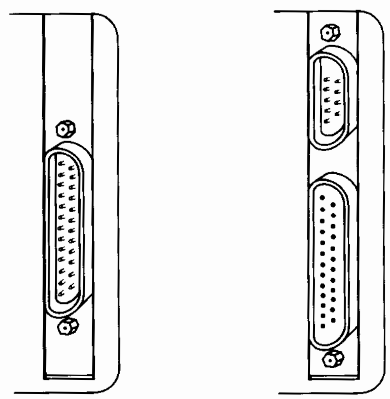


Figure 1.1 Closeup of IBM 25-Pin and 9-Pin Serial Ports

Selecting the Serial Port

Some PCs have more than one serial communications port; these are called COM1, COM2, COM3, and COM4. Reflection can use any one of these for communicating with a host computer, but only one of them can communicate at a time. The other serial ports can be used to drive a serial printer.

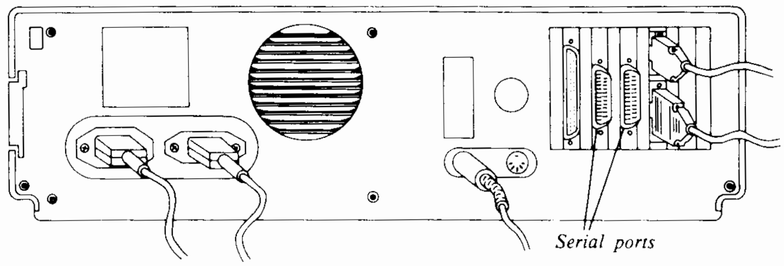


Figure 1.2 Back of the IBM PC - Serial Ports

Selecting the Connection Method

The remainder of this chapter is devoted to more detailed instructions for connecting the PC to the host computer and configuration. The PC can be connected to a host computer in three ways:

- Directly, using a standard RS232C cable. A direct connection is the most convenient method, but it's feasible only if the host computer is in the same general vicinity as the PC.
- Via modem, which allows the PC to transmit and receive data to and from the host over telephone lines. A modem connection is indicated if the host computer is far from the PC.
- Through a LAN, which can be physically connected in a variety of ways. Select the correct LAN option on the Datacomm Configuration menu in the **Datacomm port** field. See page TR-22.

Connecting Directly to the Host

For a direct connection, use a standard RS232C cable with a female connector at the cable end. If your computer is not an IBM PC, a male connector may be needed. Most terminals accept a male connector at the end of the cable; the IBM PC requires a female cable end as the serial ports are male. The parallel port is female.

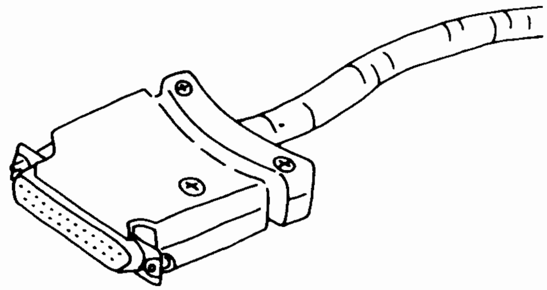


Figure 1.3 RS232C Serial Communications Interface

Only three pins must be present in the RS232C direct-connect cable:

Pin 2 - Transmit Data
Pin 3 - Receive Data
Pin 7 - Signal Ground

It's a good idea also to include pin 1, the protective ground line. Other pins may be present, but they are ignored by Reflection unless it is configured to use them.

If the host computer is an HP 3000, do not cross pins 2 and 3; they have already been crossed at the point of connection to the HP 3000.

When the connection to the host is made via direct-connect cable and Reflection is loaded, the PC is ready to use as a terminal.

To use the AT's serial port you will probably need a 9 to 25-pin adapter cable.

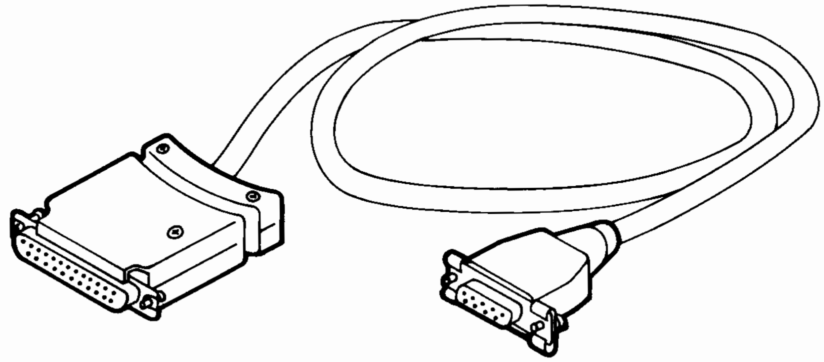


Figure 1.4 RS232C 9-Pin to 25-Pin Serial Interface

Table 1.1
RS232C AT Cable Wiring Scheme

9-Pin Pin #		*25-Pin Pin #*	*Signal Name*
8	-------------	*1*	*Protective Ground*
3	-------------	*2*	*Transmit Data*
2	-------------	*3*	*Receive Data*
20	------------	*4*	*Request to Send*
7	-------------	*5*	*Clear to Send (CTS)*
6	-------------	*6*	*Data Set Ready (DSR)*
4	-------------	*7*	*Signal Ground*
5	-------------	*8*	*Data Carrier Detect (DCD)*
22	------------	*9*	*Data Terminal Ready (DTR)*

Connecting via Modem

Two modem options are available for connecting the PC to a host computer:

- An external modem, either a direct-connect or an acoustic coupler style
- An internal modem, on an interface card inside the PC

Connect an external modem to the serial port of the PC with an RS232C modem cable. The following table shows the proper wiring scheme for this cable.

Table 1.2
RS232C Modem Cable Wiring Scheme

Computer Pin #	*Modem Pin #*	*Signal Name*
1	*1*	*Protective Ground*
2	*2*	*Transmit Data*
3	*3*	*Receive Data*
4	*4*	*Request to Send*
5	*5*	*Clear to Send (CTS)*
6	*6*	*Data Set Ready (DSR)*
7	*7*	*Signal Ground*
8	*8*	*Data Carrier Detect (DCD)*
20	*20*	*Data Terminal Ready (DTR)*

If the PC has an internal modem card, plug the telephone cord jack into the jack on the modem card. If there are two jacks on the modem card, one of them is for plugging in the telephone so it can be used for voice calls when the modem isn't in use.

Configuration Files

At start-up, Reflection looks for a configuration file on the command line, following the DOS prompt. If you enter only **R1** (R3 or R7) at the prompt, with no filename, Reflection looks on the default disk drive for R1.CFG. This file is used as the default configuration file.

Once Reflection has been reconfigured, the new configuration values can be saved to a disk file for future use. Save commonly used values to the default file. Any number of other configuration files can be saved for different circumstances, under different filenames.

To use a configuration file you've previously saved, follow the **R1** command at the DOS prompt with the name of the configuration file you want to use. For example, to use the configuration file, MINE.CFG, type **R1 MINE.CFG** at the DOS prompt and press $\boxed{\text{Return}}$.

The Contents of a Configuration File

The configuration file contains separate sets of parameters for the HP and VT emulation modes. When switching from one emulation mode to another while running Reflection, the configuration values are changed accordingly.

A single configuration file contains:

- All configuration menus
- The current definitions of the user keys (see page TR-103)
- The initial settings for remote mode, auto linefeed, block mode, and modify all
- Values set with a SET command (see page TR-214)

Every time one or more configuration parameters are changed and saved to a configuration file, the current values of all the other parameters listed above are re-saved. Type **VERIFY** at the command line to view current values.

Accessing the Configuration Menus

Any of the function keys that has a corresponding label on the config keys will bring up a configuration menu.

1. From the system keys ([F10]), press [F3], **basic config** and then press [F7], **config keys**. ([Alt]-[C] brings up the config keys from any screen).

datacomm config		terminal page 1	terminal page 2	1	1	printer config	enhancmt config	global config	system keys

Figure 1.5 Config Keys

Configuration Menus

The **basic config** key, [F3], from the system keys, only allows you to change a subset of configuration values. Reflection has several configuration menu screens that allow you to specify data communications parameters, terminal operating modes, printer characteristics, display enhancements, and file transfer parameters. From the config keys you have access to the following configuration menus:

- The **datacomm config** key brings up data communications parameters (see *Selecting the Serial Port* on page TR-9). VT and HP modes have identical data communications parameters, although the default settings vary somewhat and separate values are maintained for each in the configuration file. *Connecting to the Host Computer* on page TR-7, and *Data Communications Configuration* on page TR-27, contain detailed information about the datacomm menu.
- The **plotter config** key allows you to configure a serial plotter connection. The serial port can be COM1 to COM4, and is assumed to be a port not currently configured for datacomm or a printer. This menu is only available in graphics versions of Reflection.
- The **terminal page 1** and **terminal page 2** keys display menus which contain Reflection data operations, communications, and terminal operations parameters. There are two separate terminal configuration menu *pages* for both HP and VT mode.
- The **printer config** key displays printer control parameters. VT and HP modes have identical printer configuration menus.
- The **enhancmt config** key displays the menu which controls the appearance of images on the display screen. The enhancement menus differ slightly between VT and HP modes. Color graphics emulation has a separate enhancement menu.
- The **global config** key presents a menu which allows you to switch back and forth between terminal emulations as well as define language, keyboard, 7-bit operation, 132-column adapters, and Reflection memory allocations.
- The **video config** key is only available in R3 graphics emulations. It presents a menu for you to define your graphics hardware.
- The **config transfer** key is located on the File Transfer screen. It shows three menus on one screen. These menus contain parameters that control the operation of Reflection's special file transfer subsystem. The file transfer configuration menu screen is the only menu, besides Basic Configuration, not accessed through the config keys.

Configuration Parameter Values

A configuration menu field can have three values: the one currently displayed, the default value, and the active value.

- The currently displayed value is the one shown on the screen at the moment.
- The default value is the *as shipped* value. It is automatically set the first time the program runs and can be reset by loading Reflection with the **/D** switch.
- The active value is the one Reflection is currently using. The active value remains unchanged until a new set of values is activated or saved.

NOTE: To reset Reflection completely, load the program using **R1 /D**. Default values are reset for everything. Your old configuration values remain intact until you save a new configuration with the default filename (R1.CFG).

Most of the configuration fields allow you to choose from a predefined set of values simply by pressing [F1] for previous choice or [F2] for next choice. A few require keyboard entry.

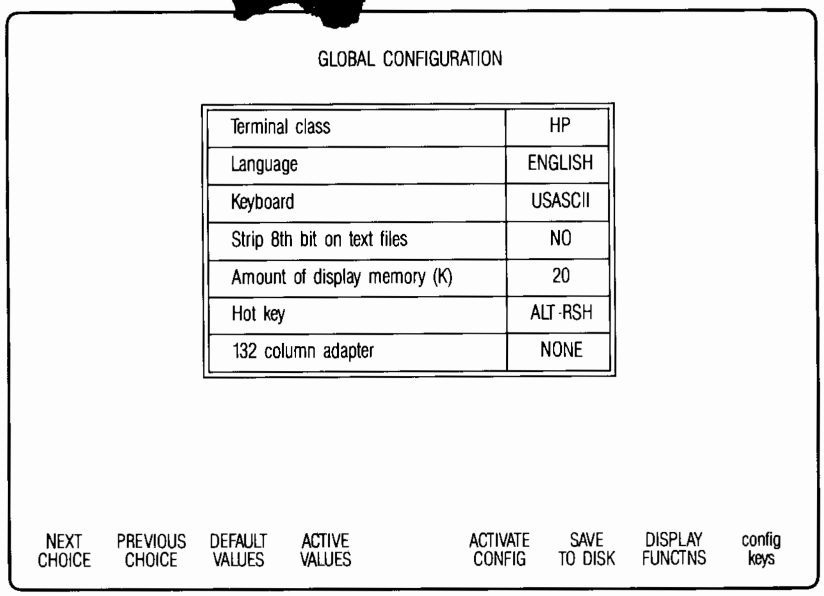


Figure 1.6 Global Configuration Menu

Selecting a Field

The current field is shown in inverse video. To move the cursor down the menu, press [Tab]; pressing [Shift] and [Tab] simultaneously moves the cursor back a field at a time. The [Up]/[Down] arrow keys may also be used.

When a field requires keyboard entry, the cursor blinks and row/column indicators are displayed at the bottom of the screen. The editing functions Insert char ([Ins]), Delete char ([Del]), and Clear field ([Alt]-[K]) allow you to edit your entries.

Using Function Keys in Configuration

The following table lists configuration menu screen labels and the actions performed by the corresponding function keys.

Table 1.3
Configuration Menu Screen Labels

Label	*Key*	*Action*
NEXT CHOICE	F1	Changes value of field to the next choice.
PREVIOUS CHOICE	F2	Changes value of field to the previous choice.
DEFAULT VALUES	F3	Changes all fields on the screen to their default values.
ACTIVE VALUES	F4	Changes all fields on screen to currently active values, i.e., the values that were most recently saved or activated.
ACTIVATE CONFIG	F5	Activates values currently shown on screen, changes labels to the config keys set, and sets to terminal mode.
SAVE TO DISK	F6	Asks for a filename for the configuration file. Saves all config values to that file.
DISPLAY FUNCTNS	F7	Toggles display functions mode on and off so that ASCII control codes can be entered into menu fields that require keyboard input.
config keys	F8	Clears the configuration menu from the screen, changes function key labels to the config keys set.

Saving a Configuration

Every configuration menu contains the label **SAVE TO DISK**. When this key is pressed, Reflection asks for a disk filename for the configuration file, as follows:

Enter filename: A:<filename>.CFG

Press [Return] if you want to save the configuration to the default file, R1.CFG. To supply a different configuration filename, key it into the inverse video area, directly over the default filename. Use [Backspace] to make corrections. Then press [Return].

If the default filename presented happens to be other than R1.CFG (R3 or R7 may be in place of R1), type in **R1.CFG** (using the correct number for the Reflection product you are using). Then press [Return]. This will make this setting automatic the next time you load Reflection.

If you're creating a separate configuration file, type in the filename you've selected for this file and press [Return]. The filename extension .CFG should be used for all configuration files to help you distinguish them easily from other files.

With the filename prompt, a **CANCEL** label also appears, which allows you to cancel the save operation and continue entering values in the current menu.

Multiple Configuration Files

You might want to save values for occasional use only. If so, type in a filename such as MODEM.CFG and press [Return]. These values can be loaded in from the command line whenever you use a modem. See the LOAD command in the *Command Language Summary* page TR-179.

See *Communicating with the Host* on page UM-47 for information on operations.

Configuring for Other than COM1

There is no way to determine which COM port a card has been set for except through the documentation for that card and looking at the jumper settings. Reflection's default configuration file uses the COM1 serial port. If you want to use another COM port as your host connection, you must reconfigure Reflection. Use the following steps:

1. Press [F3] from the system keys to bring up the Basic Configuration menu.

2. Press [F1] until the correct COM# appears in the **Datacomm port** field.
3. Press [F6] to save this value to a configuration file.
4. A prompt appears, asking for a filename. If the default filename shown on the prompt is **R1.CFG**, press [Return].

Reflection is now configured to communicate with the host computer through the correct COM port, and the configuration is saved to disk.

Configuring for an HP 3000

Reflection's default configuration values are correctly set for an HP 3000. Only a few fields must be checked:

Datacomm Configuration menu

The **Datacomm port** field must be configured for the serial port you've selected for data communications, COM1 to COM4, or for a LAN connection. The default value is **COM1**.

The **Baud rate** you specify must match the baud rate of your modem or of your direct connection.

The **Receive pacing** field can be set to *XON/XOFF* so that [ScrollLock] can be used as the STOP key.

Terminal Configuration menus

Typeahead (Page 1) can be set to *YES* to allow you to type ahead of the HP 3000. [Alt]-[T] may need to be used to obtain a host prompt after logging off or if handshaking must be reestablished.

Some models of the HP 3000, notably the Series 30 and 33, can't receive data blocks at 9600 baud (the default value) without danger of lost data. For this reason Reflection allows you to specify a time delay between character transmissions, to prevent overrunning the HP 3000. For a Series 30 or 33, specify a **Character xmit delay** (Page 2) of 3 milliseconds. The PC can then receive data at a full 9600 baud (960 characters per second), while cutting the effective transmission rate to the host to about 240 characters per second. Enter the new **Character xmit delay** parameter through the keyboard.

File Transfer Configuration menu

If you will be operating at 19200 baud, reduce the block size on the File Transfer Configuration menu to 116. This will prevent overrunning ATPs.

Configuring for a VAX/VMS

Reflection's default configuration values are correctly set for a VAX/VMS when VT emulation mode is selected. Only a few fields must be checked:

Basic Configuration menu

The **Datacomm port** field must be configured for the serial port or LAN you've selected for data communications. The default value is **COM1**.

The **Baud rate** field you specify must match the baud rate of your modem or of your direct connection.

Configuring for a UNIX System

Reflection's default configuration values are not set for a UNIX System. Besides the Basic Configuration menu settings, you must change the following fields:

Datacomm Configuration menu

Receive pacing and **Transmit pacing** must both be set to *XON/XOFF*.

Terminal Configuration items

Host prompt character (Page 2) should be set to *NONE* (use $\boxed{\text{Ctrl}}$-$\boxed{\text{Shift}}$-$\boxed{@}$).

Inhibit handshake and **Inhibit DC2** (Page 2) should both be set to *YES*.

Walker Richer & Quinn provides the host protocol program for UNIX systems. See *Transferring Files to UNIX-based Systems* on page UM-139 for how to configure for UNIX file transfers.

Configuring for an HP 1000

Cabling

HP 1000s usually do not have pins 2 and 3 crossed at the host. Use a Null Modem cable (where pins 2 and 3 are crossed) rather than that recommended for HP 3000s.

Basic Configuration menu

Check the baud rate and parity as discussed for HP 3000 and VAX hosts.

The **Parity field** is often *EVEN* on HP 1000s. Check to see what parity your system uses.

Reflection supports both XMODEM and KERMIT for file transfers to an HP 1000.*

Connecting via LAN

The configuration fields which affect LAN connections are **Datacomm port, Parity, Check parity**, and the **Enq/Ack pacing** fields. Select the LAN version you are using and read the related section below. If you are communicating with a host which uses Enq/Ack pacing, keep this field at *YES*. Either the server or the PC must have Enq/Ack pacing for communication with an HP 3000, setting both will not cause a problem.§

File transfers through a LAN may require that **;PARM=1** be added to the **Host startup sequence**. This setting will slow down file transfers somewhat. See page TR-89 for more information. Data communication between a PC and an HP 3000 over a LAN requires a hardware interface between the PC and the LAN, and a hardware interface between the HP 3000 and the LAN. In addition, there must be software available for Reflection to communicate with the PC hardware interface.

---

* A third party vendor, DO IT ALL Software, in Chicago, has a PCLINK/1000 (based on the Walker Richer & Quinn protocol) for the HP 1000.

§ Termtype=10 is assumed.

Using AdvanceNet (OfficeShare)

To run Reflection over OfficeShare, install the OfficeShare hardware and software as explained in the HP OfficeShare Network Installation and Configuration manual. Run the USRLOAD program as described in the HP OfficeShare Network User's Guide. After running Reflection, bring up the Basic Configuration menu and select *ADV.NET* in the **Datacomm port** field.

Walker Richer & Quinn's Command Interpreter prompt for AdvanceNet appears in display memory as the > character. At the prompt enter the command **CONNECT <nodename>**, where <nodename> is the name assigned by the HP 3000 network manager to the HP 3000 you wish to communicate with. The Command Interpreter returns a *Connected* message for a successful connection, and an error message for an unsuccessful connection. Once the connection is established, proceed as usual. There are two ways to break a connection. [Ctrl]-[F10] forces a disconnect from the PC side, and **:BYE** forces a disconnect from the HP 3000 side. When a connection is broken, Reflection displays *Disconnected*, and returns to the Command Interpreter.

File transfer needs no special configuration. A maximum block size is recommended.

Using the Ungermann-Bass Network

To run Reflection over an Ungermann-Bass network, the Ungermann-Bass hardware and software must be installed as explained in the Ungermann-Bass documentation. After Reflection is run, select **basic config**, and select *U.B.* in the **Datacomm port** field.

At this point you will be at the Ungermann-Bass Command Interpreter with its >> prompt. Enter **CONNECT <nodename>**, where <nodename> is the node name assigned by the network manager to the host you want to communicate with. The Command Interpreter returns *Successful* for a successful connection and an error message for an unsuccessful connection. Once the connection is established, proceed as usual. There are two ways to break a connection.

- [Ctrl]-[F10] forces a disconnect from the PC side.

- Ctrl-F8 puts the connection on hold. (Refer to the Ungermann-Bass documentation.)

After a disconnect or a hold, you are back at the Ungermann-Bass Command Interpreter prompt. Note the following configuration considerations:

- The Personal NIU for the PC should be configured with Break Sequence and Hold Sequence set to *NONE*.
- For an HP 3000, the NIU-130 or NIU-180 should be configured with Pace-NIU set to *NO*, Pace-Dev set to *ENQAK*, and Stop Count set to *80*. Receive Breaks should be set to *YES*.
- To optimize file transfer, set Binary Mode on the NIU-130 or NIU-180 to *YES*. To optimize terminal operation, set Binary Mode to *NO*, and set **;PARM=1** on the **Host startup sequence** in Reflection's File Transfer Configuration menu for file transfers.

Using the Bridge Network

To run Reflection over Bridge, you must have installed the hardware and software as explained in the Bridge documentation. Select *Bridge* from the Basic Configuration menu for the **Datacomm port** field.

Refer to the Bridge documentation for more information.

Using the NESTAR Network

To run Reflection over a NESTAR network, the NESTAR hardware and software must be installed as explained in your NESTAR documentation. After Reflection is run, select *NESTAR* in the **Datacomm port** field of the Basic Configuration menu. Activate or save this change to your configuration file.

At the network prompt enter **CONNECT <line identifier>**, where <line identifier> is the name assigned by the network manager to the line with which you want to communicate. Reflection returns *Successful* for a successful connection, and an error message for an unsuccessful one. Once the connection is established, proceed as usual. To break a connection, press Ctrl-F10; this forces a disconnect from the PC side.

Using INT-14

The INT-14 LAN connection provides a means for supporting various drivers or for writing a custom driver for another LAN. Select *INT-14* in the **Datacomm port** field of the Basic Configuration menu if your LAN is not listed as a choice. This only applies to LANs involving a mini or mainframe computer, not PC LANs.

Call Walker Richer & Quinn for further information.

Using LAT Protocol

To run Reflection over an ethernet network using the LAT protocol you must first install the ethernet hardware and SCH, DLL and LAT drivers as explained in your *DECnet-DOS Installation Guide*. After Reflection is run, select LAT in the **Datacomm port** field on the Basic Configuration menu. Activate or save this change to your configuration file.

At the network prompt (>>>) enter **CONNECT <LAT service>**, where <LAT service> is the name assigned by your network manager to the LAT service/node you wish to use. Reflection displays the message *connecting...* to indicate that it is attempting to establish a connection. If the connections succeeds, you will receive the host system login prompt; if the connection fails, you will receive the message *Disconnected* followed by the network prompt.

The LAT connection can be broken by either logging of the host system or pressing Ctrl-F10. The network prompt appears again after the connection is broken.

Miscellaneous Host Configurations

The table below gives configuration values for the following:

- Dow Jones Service or THE SOURCE
- VM/370
- MVS/TSO

NOTE: **Receive pacing** and **Transmit pacing** on the Datacomm Configuration menu should be set to NONE for all of these hosts.

Table 1.4
Configuration for Other Hosts

Host	*Value*	*Config Menu*	*Setting*
Dow Jones	*Parity*	*Datacomm*	*Even*
	Local echo	*Datacomm*	*NO*
VM/370	*Parity*	*Datacomm*	*1s*
	Local echo	*Datacomm*	*YES*
	Host Prompt	*Term. Page 2*	*DC1*
MVS/TSO	*Parity*	*Datacomm*	*1s*
	Local echo	*Datacomm*	*YES*

Host prompt character is the only field among the ones listed here that requires keyboard entry. The proper value for the VM/370 is the default HP emulation value, **DC1**, so no change should be necessary. If that field contains a value other than DC1, which is displayed as the symbol ◄, change the value to **DC1** by pressing [Ctrl]-[Q]. This value is sometimes referred to as *XON*.

To change the **Host prompt character** to *NONE* when configuring for Dow Jones, THE SOURCE, or MVS/TSO, enter the ASCII control character for NULL by pressing [Ctrl]-[@].

A value of 1's in the **Parity** field is sometimes referred to as *mark* parity.

Data Communications Configuration

This section covers in detail the configuration parameters for the data communications menu. The default values on this menu are set to allow Reflection to communicate directly with an HP 3000 (in HP mode) or VAX/VMS (in VT mode) from COM1.

The PC must have at least one serial port in order to work as a terminal, unless communication will be through a LAN. It may have four, called *COM1, COM2, COM3*, and *COM4*. Reflection can use any of these serial ports to communicate with a host computer. Other ports can be used to drive a serial printer or other serial device. Serial ports must be configured correctly; see page UM-198.

Using an Autodial Modem with Reflection on page UM-55 contains helpful information about configuring for data communications. Some parameters may need to be reconfigured before you can communicate successfully. Find additional tips for configuring for data communications in *Connecting to the Host Computer* on page TR-7.

Miscellaneous Host Configurations, on page TR-25, is helpful if you will be working with hosts other than the HP 3000 or VAX/VMS. Be sure to read this section and the previous one thoroughly before attempting to communicate with a host computer other than an HP 3000 or VAX/VMS. See *Pacing Data Transfer*, on page TR-31, and refer to pages TR-35 and TR-57 for terminal configuration fields.

Setting Datacomm Parameters

The following data communications configuration parameters apply to both VT and HP emulation modes. Selecting the values for these parameters is similar to setting other Reflection configuration parameters; see page TR-15 for details. Once you've selected the new values, you can activate them for current use, or you can save them to disk, which also activates them. When you save a value to disk, the current values of all other configuration menus are also saved.

DATACOMM CONFIGURATION

Datacomm port	COM1
Baud rate	9600
Parity	NONE
Check parity	NO
Receive pacing	NONE
Transmit pacing	NONE

Enq/Ack pacing	YES
Xmit indicator (*)	OFF
CTS required	NO
DSR required	NO
Stop bits	1
Session # (LAN)	0

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	ACTIVATE CONFIG	SAVE TO DISK	config keys

Figure 1.7 Datacomm Configuration Menu

Refer to pages TR-12 through TR-25 to determine whether the other default data communications configuration values are correct for the PC and host computer you'll be using. If the default data communications parameters are not appropriate to your equipment, change them before attempting to connect with the host.

Datacomm Configuration Menu Fields

Data communications configuration parameters for both HP and VT emulation follow. Default values are indicated by an asterisk.

Selecting the Connection Method

Reflection allows you to connect the PC to the host computer via serial port or through one of the supported LAN connections. Once properly connected (see page TR-7), you must select the corresponding value on the Data Communications menu.

The first field on this menu, **Datacomm port**, should match your hardware connection. Use [F1] to toggle through the choices.

Values:	**COM1***	
	COM2	
	COM3	
	COM4	
	ADV. NET	(HP's AdvanceNet or *OfficeShare*) See TR-23
	U.B.	(Ungerman-Bass's Network) See TR-23
	BRIDGE	(Bridge-3COM's Network) See TR-24
	NESTAR	(from Nestar Systems) See TR-24
	INT-14	(custom driver) See TR-25
	MNP-COM1/2	(Microcom's Networking Protocol) See TR-34
	LAT	(Digital's Network) See TR-25

Configuring for a Modem

Reflection must be properly configured and in remote mode to be able to communicate via modem. Specifically:

- The **Datacomm port** field must be configured for the correct communications port or the particular LAN you may be using.
- The value of the **Baud rate** field must match the modem baud rate.

Baud rate

Specifies the speed at which Reflection transmits and receives data through the serial port.

Values:	**110**	**1800**
	150	**2400**
	300	**4800**
	600	**9600***
	1200	**19200**

Selecting the Correct Parity

Parity is a method by which communications equipment can detect some kinds of errors in data transmission. When *odd* or *even* parity checking is used, one of the bits in each character is a parity bit. The transmitter sets this bit to *on* (1) or *off* (0) so that the entire character code, including the parity bit, contains an odd (odd parity) or even (even parity) number of 1 bits.

If **Check parity** (see below) is set to *YES*, the receiver tests each character for an odd or even number of 1 bits, and sets an error indicator whenever a character fails the test. Reflection always transmits, and expects to receive, a total of eight bits per character, not counting start and stop bits. The entire eight bits may be used to represent data, or one of the bits may be used as a parity bit, leaving seven bits for data. Values:

NONE*	Each character consists of eight data bits with no parity bit. In terminal mode, if the high-order bit is used, the character will be interpreted as an extended Roman 8 character.
0's	Each character consists of seven data bits, with the eighth bit always 0. This is often called *space* parity.
1's	Each character consists of seven data bits, with the eighth bit always 1. This is often called *mark* parity.
ODD	Odd parity.
EVEN	Even parity.

Check parity

When **Check parity** is *YES*, Reflection checks parity on all received characters. When an error is found, the character in error is replaced by an ASCII DEL character. When the field is *NO*, then parity on received characters is not checked. Parity is always generated for transmitted characters based on the **Parity** field.

Values: **NO***
YES

Pacing Data Transfer

Receive pacing

It is possible for a host computer to transmit data to Reflection faster than Reflection can process it. If this continued for too long, Reflection's *receive buffer* would overflow and data would be lost. One way to prevent this is to enable **Receive pacing**, provided the host computer recognizes the XON/XOFF handshake.

XON/XOFF receive pacing works as follows. When Reflection's receive buffer has a limited amount of space left, it sends an XOFF (ASCII DC3) character as a signal to stop transmitting. When Reflection has processed most of the backlog of characters in its receive buffer, it sends an XON (ASCII DC1) character to resume transmission from the host.

In VT mode when **Receive pacing** is set to *XON/XOFF* (the default), the hold screen key (ScrollLock) is activated.

There is an easy way to tell if the host computer recognizes the XON/XOFF handshake. In character/remote modes, enter a Ctrl-S (the DC3 character) while Reflection is receiving data from the host system. If the host system uses XON/XOFF, it will immediately stop transmitting, so that the display is *frozen*. Then enter Ctrl-Q (the DC1 character); the host system should start transmitting again.

Values: **NONE*** (HP mode)
XON/XOFF* (VT mode)

Transmit pacing

It is possible for Reflection to send data to the host computer faster than the host can process it. With this field, you can tell Reflection to stop transmitting whenever it receives an XOFF (DC3) character from the host system, and to resume transmitting only after receiving an XON (DC1) character.

This should not be used with most HP 3000s as DC3s will be echoed, hanging the terminal.

If *NONE* is selected for transmit pacing, problems may arise during file transfer for some systems. If Reflection were to transmit data as fast as it could, the host computer might not be able to keep pace. Set the **Character Transmit delay** field on the Terminal Configuration, Page 2 menu to 3. This will slightly reduce the speed at which information is sent to the host while allowing you to receive data at a higher baud rate. See page UM-210 for more detailed information.

Values: **NONE*** (HP mode)
XON/XOFF* (VT mode)

Enq/Ack pacing

The HP 3000 and HP 1000 computers use a form of handshaking called *Enq/Ack* (ENQuire/ACKnowledge) to prevent the terminal (in this case, Reflection) from falling too far behind the host system and losing data.

Enq/Ack pacing works as follows: whenever the host system has a block of more than 80 characters to send, it sends 80 characters followed by an ASCII ENQ character and stops transmitting. When the terminal has processed all of the characters preceding the ENQ, it sends an ASCII ACK character, which tells the host that the terminal has caught up and is ready for more data.

Values: **NO*** (don't use Enq/Ack pacing) (VT mode)
YES* (use Enq/Ack pacing) (HP mode)

Modem-Related Fields

Xmit indicator(*)

If you are using a modem connection to the host system, you may want to have Reflection display an indicator of the state of one of the RS232C lines. Be aware that some modems can be configured to force these lines to a *true* state, so that the indicator may have little meaning. For example, the factory settings of a Hayes Smartmodem cause it always to set the *Data Carrier Detect* signal to the terminal in a *true* state.

This field allows you to specify which modem status line, if any, you wish to have monitored. If the selected line (*pin*) is in a *true* state, an asterisk is displayed in the bottom row of the screen.

Values: **OFF*** (no indicator)
CTS (Clear To Send)
DSR (Data Set Ready)
DCD (Data Carrier Detect)

CTS required

When this field is set to *YES*, Reflection will not transmit unless the RS232C *Clear To Send* signal (pin 5) is active *(true)*; otherwise, the program ignores the signal.

Values: **YES**
NO*

DSR required

When this field is set to *YES*, Reflection will not transmit unless the RS232C *Data Set Ready* signal (pin 6) is active *(true)*; otherwise, the program ignores the signal.

Values: **YES**
NO*

Stop bits

Every character is followed by either one or two stop bits. You should specify one stop bit for all baud rates other than 110 baud, and two stop bits for 110 baud, unless you are positive that your host or serial device requires otherwise.

Values: **1***
2

Session # (LAN)

If you are using a LAN which supports multiple sessions (e.g., Ungermann-Bass), Reflection allows you to identify which session you would like as the current one. Enter the session number (0-255) and activate the configuration to select a session. If you are using normal serial ports, then the value of this item has no effect. See also *Using the Ungermann-Bass Network* on page 23.

Values: 0 - 255
Default: **0**

Microcom's Network Protocol

Walker Richer & Quinn has developed a serial port driver which is able to communicate with modems that use Microcom's Networking Protocol. MNP transmits the data in packets and performs CRC generation and checking. Any bad packets will be retransmitted. These functions ensure that all data which passes over a communications line is transferred error-free, regardless of line noise.

To take advantage of a modem which uses MNP, you must provide either a second MNP modem or use Reflection's MNP driver on the communications port of the PC with a non-MNP modem. Reflection's MNP driver is an option which must be purchased separately. It is significantly less expensive than purchasing a second MNP modem.

To install the MNP driver, use the following steps:

1. Make sure that your modem is set to transmit carrier upon connect and drop carrier upon disconnect. Hayes internal modems are factory set to perform this way, but external Hayes modems are not. A switch setting change is required to cause external Hayes modems to perform as required.
2. At the DOS prompt type **MNP1.COM** or **MNP2.COM** depending on which communications port you are using.
3. Load Reflection.
4. On the Basic Configuration Menu, select *MNP-COM1* or *MNP-COM2* in the **Datacomm port** according to the driver installed.

To drop out of MNP mode, press Ctrl-F10 (Disconnect). Use +++ to put a Hayes modem back into command mode. **ATH** will then cause the modem to terminate the call.

You can uninstall the MNP driver by typing **MNP<n>/U** at the DOS prompt.

HP Terminal Configuration

This chapter examines each HP terminal configuration field in the order it appears on the Terminal Configuration, Page 1 and Page 2 menus. The default configuration values are indicated by an asterisk (*).

To get to the terminal configuration menus press Alt-C. Then press F3, **terminal page 1.**

HP TERMINAL CONFIGURATION, PAGE 1

Type-ahead	NO
User label lines	2
Initial label set	SYSTEM
Bell	YES
Destructive bkspace	NO
Tab = spaces	NO
Local echo	NO
Caps lock	NO
Transmit functions	NO

SPOW (strap B)	NO
Inhibit EOL wrap	NO
Line/Page	LINE
Inhibit handshake	NO
Inhibit DC2	NO
Return definition	♪
Field separator	▼
Block terminator	▲
Start column	01

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	ACTIVATE CONFIG	SAVE TO DISK	DISPLAY FUNCTIONS	config keys

Figure 1.8 Terminal Configuration, Page 1 (HP)

As with other configuration screens in Reflection, some values are entered through the keyboard, while most require you to use the function keys to cycle through a predefined list of choices in order to select a value.

Once you've entered the new values, you can activate them for current use, or you can save them to disk (which also activates them). When you save a value to disk, the current values of all other configuration menus are saved in the same file.

- Change the value of the selected parameter by pressing [F1], **NEXT CHOICE**, or [F2], **PREVIOUS CHOICE**.
- Select the default values by pressing [F3], **DEFAULT VALUES**.
- Restore the currently active values to the screen by pressing [F4], **ACTIVE VALUES**.
- Temporarily activate new values by pressing [F5], **ACTIVATE CONFIG**.
- Save new configuration values by pressing [F6], **SAVE TO DISK**.

Terminal, Page 1 Menu Parameters

Typeahead

With an HP 3000 in character/remote mode, you must wait for a prompt from the host before entering each new line of data. When the host has finished processing a line, it transmits a prompt to the terminal, signaling that it's ready for another line. The host ignores characters it receives before it has issued its prompt.

The **Typeahead** feature allows you to type continuously. Each time the host sends a prompt character (see the **Host prompt character** field), Reflection sends the next line of keyboard input from its keyboard buffer.

Values: **YES**
NO*

User label lines

Specifies the number of screen rows to be used for user-key labels. The default value is *2*, which leaves only 23 lines of the PC's 25-line display available for text. If this field is set to a value of *1* or *NONE*, Reflection uses 24 lines for text whenever the user keys are called up.

Vectras and PCs with enhanced graphics adapters automatically provide 24 lines of text, two label lines, and a status line. EGAs require that the file EGA7X12.FT be present in order to load in the HP font which makes 27 lines possible. You can select the IBM font on either computer by using a /I switch. The 27 lines are reduced to 25 on EGAs. To force a complete clone of the IBM PC on the Vectra, use /C. This ensures 25 lines as well as the IBM font. Use this if interference with other programs occurs due to the 27-line mode, e.g., *Sidekick*.

When the user-key definition menu is enabled by an escape sequence, the value in this field determines the size of the labels. If you then change the number of lines by pressing [F9], that change remains in effect until another enabling escape sequence is received or until you press [Alt]-[U].

Values: **2***
1
NONE

NOTE: Only the user-key labels can be reduced in size; all other labels always use two lines.

Initial label set

Determines which set of screen labels will be displayed when Reflection is loaded.

Values: **SYSTEM***
USERS
MODES

Bell

When this field contains a value of *YES*, Reflection emits a beep sound under the following conditions:

1. When a character is typed from the keyboard into the column eight columns left of the right margin
2. In format mode, when the last character of an unprotected field is entered from the keyboard
3. In file transfer mode, when file transmission is complete or terminated in error

When *NO* is specified in this field, the bell sounds only when Reflection detects an input error, or when an ASCII BELL character (Ctrl-G) is received from the host or entered from the keyboard.

Values: **YES***
NO

Destructive bkspace

When this field contains a value of *YES*, Reflection erases the character to the left of the cursor each time an ASCII backspace character is entered. When the value is *NO*, Backspace simply moves the cursor to the left without erasing.

Values: **YES**
NO*

Tab = spaces

When this field contains a value of *YES*, pressing Tab generates the number of space characters needed to move the cursor forward to the next tab stop. In character/remote mode, the spaces are transmitted to the host computer. If there are no tab stops to the right of the cursor position when Tab is pressed, the bell sounds, the cursor does not move, and no spaces are generated.

The backtab key functions in a similar manner. When Shift and Tab are pressed, Reflection generates the number of ASCII backspace characters needed to move the cursor to the previous tab stop, or to the left margin if there are no tab stops between the cursor and the left margin. If the cursor is already at the left margin, Reflection sounds the bell, does not move the cursor, and does not generate any characters.

Values: **YES**
NO*

Local echo

Each time a character is typed at the keyboard of a terminal in remote mode, the terminal transmits the character to the host computer, and the host immediately sends the same character back to the terminal (i.e., it *echoes*). The character is not displayed on the screen until it is received back from the host. This mode is sometimes called *full duplex*.

If the host system does not echo, set this field to *YES* so that Reflection will immediately display all keyboard input.

When Reflection is in local mode or block mode, local echo is automatically enabled, even when the configuration value of **Local echo** is *NO*.

Values: **YES** (half duplex)
NO* (full duplex)

Caps lock

When this field contains a value of *YES*, Reflection limits the set of characters that can be produced from the keyboard to Teletype-compatible codes. Specifically:

1. No lower-case alphabetic characters are generated. All alphabetic keys are automatically upshifted.
2. [{], [}], and [|] generate the codes for [,], and \, respectively.
3. The ~ and ‘ codes are disabled.

Values: **YES**
NO*

Transmit functions

Most of the keys on the keyboard have an associated ASCII character. Several keys, however, perform functions for which there is no character defined in the ASCII standard, for example, [Home], [PgUp], [Alt]-[J] (clear display), and backtab. Certain host software programs, such as *HP Slate*, need to be informed when you press one of these non-ASCII keys. The **Transmit functions** feature provides a way for the terminal to inform the host system whenever you press one of these keys.

When **Transmit functions** is enabled and the terminal is operating in character/remote mode, each time you press one of these keys, the associated escape sequence is transmitted to the host. Most software that requires this feature sends the escape sequences to enable and disable the feature as required, so you probably will not need to enable it manually.

Values: **YES**
NO*

SPOW (strap B)

Ordinarily, spaces entered from the keyboard overwrite and erase existing characters. When **SPOW (strap B)** is configured to *YES*, spaces entered from the keyboard move the cursor over existing characters without replacing them; when configured to *NO*, the space bar functions normally. The SPOW switch is turned on by a carriage return. It is turned off by a linefeed, tab, or home up ([Ctrl]-[Home]).

Values: **YES**
NO*

Inhibit EOL wrap

When this field has a value of *NO*, Reflection automatically returns the cursor to the left margin in the next line when the cursor reaches the right margin or the right screen edge.

When **Inhibit EOL wrap** has a value of *YES*, the cursor is not automatically advanced when it reaches the right margin. As you type additional characters, each character overwrites the character at the right margin, until you explicitly move the cursor with a carriage return or other cursor movement key.

Values: **YES**
NO*

Line/Page

When Reflection is operating in block mode, a block of one or more characters is transmitted when you press [Enter] or when the host system requests a block transfer from terminal memory. The setting of this field determines how much data Reflection transmits on each block transfer.

Values: **LINE***— Transmit a line at a time, or a field at a time in format mode.
PAGE— Transmit an entire page or more.

Inhibit handshake, Inhibit DC2

The values in these two fields, along with some other factors, determine the type of handshaking that precedes each block transfer of data from Reflection to the host system.

When set to *YES*, the **Inhibit handshake** function inhibits the DC1 handshake for block transfers. The **Inhibit DC2** function inhibits the DC2 portion of the handshake.

Refer to page TR-259, *HP Block Transfers*, for a complete discussion of these fields.

Values: **YES**
NO* (for both fields)

[Return] definition

This field allows you to specify a string of one or two characters to be generated whenever [Return] is pressed. If the second character is a space, only the first character is generated.

NOTE: If you want to include a carriage return, tab , or backtab character in the definition, you must press the **DISPLAY FUNCTNS** key, [F7], before pressing any of those keys. Any other ASCII character can be entered without enabling **DISPLAY FUNCTNS**. Control characters (ASCII characters with values in the range 0-31) may be entered by holding down [Ctrl] while pressing the appropriate character key. See page TR-323 for a list of ASCII control characters and their associated keystrokes.

Values: Any two ASCII characters
Default: An ASCII carriage return character, followed by a space

Field separator

When Reflection is transmitting in block, page, and format modes, it sends a field separator character after each field of the formatted screen except the last one. This field allows you to specify which ASCII character is to be used as the field separator.

NOTE: If you want to include a carriage return, tab, or backtab character in the definition, see the note on page TR-41.

Values: Any ASCII character
Default: **US** (unit separator, decimal 31)

Block terminator

As explained in block mode transfers on page TR-259, under certain conditions Reflection transmits a block terminator character at the end of each block data transmission. This field allows you to specify an ASCII character to be the block terminator.

NOTE: If you want to include a carriage return, tab, or backtab character in the definition, see the note on page TR-41.

Values: Any ASCII character
Default: **RS** (record separator, decimal 30)

Start column

For every line in display memory, Reflection attempts to *remember* the leftmost column that was entered from the keyboard, as opposed to that received from datacomm. In this way, the program can distinguish the host prompt portion of each line from the user-entered portion. This information is then used when you enable **LINE MODIFY** or **MODIFY ALL** to determine the leftmost column that should be transmitted to the host when you press [Return] or [Enter]. Under some circumstances, it is impossible for Reflection to tell which column was the first user-keyed column; when that happens, it uses the value of **Start column** to determine the leftmost column to be transmitted.

Values: 1-80
Default: **1**

Terminal, Page 2 Menu Parameters

To access the Terminal Configuration, Page 2 menu, bring up the **config keys** with [Alt]-[C]. Then press [F4], **terminal page 2.** Reflection 3's Page 2 menu varies slightly; see below.

HP TERMINAL CONFIGURATION, PAGE 2

Left margin	1
Right margin	80
Columns per horizontal scroll	1
Host prompt character	◄
Character xmit delay (x .001 seconds)	0
Line xmit delay (x .1 seconds)	0
Color scrolling method/speed	SPEED 3
Terminal ID response	2392A
Display memory status response	4K
Forms buffer size (x 256)	0

NEXT CHOICE | PREVIOUS CHOICE | DEFAULT VALUES | ACTIVE VALUES | 3 | 59 | ACTIVATE CONFIG | SAVE TO DISK | DISPLAY FUNCTNS | config keys

Figure 1.9 Terminal Configuraton, Page 2 (HP)

Left margin

The left margin can be set by entering a number in this field. The value must be less than the right margin value.

Note that the left margin can also be set using the **margins/tabs/col** set of function keys, [F2] from the system keys.

Values: 1-80
Default: **1**

Right margin

The right margin can be set to any value from 0-10000, but it must not be less than the value of the left margin. When a value greater than 80 is entered, horizontal scrolling becomes possible, allowing for entry and viewing of lines longer than 80 characters without wraparound.

Note that the right margin can be set to a value of 80 or less using the **margins/tabs/col** set of function keys, [F2] from the system keys.

WARNING: If you configure a right margin greater than column 80, Reflection will not work properly with host computer software that is programmed to wrap around to the next line when column 80 is entered. Block mode fields are displayed incorrectly if more than 80 is set.

Values: 1-10000
Default: **80**

Columns per horizontal scroll

This field determines the number of columns, or character positions, Reflection will scroll horizontally with each right or left scrolling operation. The parameter has no effect if the right margin is set at less than 80.

Values: 1-80 (columns)
Default: **10**

Host prompt character

Some host computers, notably the HP 3000, send a prompt character to the terminal when ready to accept a line or block of characters. This field allows you to specify which ASCII character, if any, the host system uses as a prompt.

If the host system does not prompt in this manner, enter the ASCII NULL character in this field by pressing [Ctrl]-[@]. In this case, you may need to enter a non-zero value in the **Line xmit delay** field.

NOTE: When the **Typeahead** feature is enabled, Reflection waits for this character to be received from the host before it transmits the next line from the keyboard buffer.

Values: Any ASCII character
Default: **DC1** (decimal 17)

Character xmit delay (x .001 seconds)

When transmitting blocks of characters at high speeds, it may be necessary to *throttle down* the effective speed of the transmission to allow the host computer to keep pace. This frequently occurs when transmitting data blocks at 9600 baud to the HP 3000 Series 30 and 33 systems. It even occurs occasionally at 9600 baud on a heavily loaded Series 44s.

This field specifies the number of milliseconds of wait time that Reflection should insert after each character when transmitting blocks of characters to the host system.

When the baud rate is 9600, a value of 3 milliseconds works best with the HP 3000 Series 30 and Series 33 systems. This lowers the effective speed of data transmission to about 2400 bits per second.

Values: 0-99
Default: **0**

Line xmit delay (x .1 seconds)

If the host computer does not send a prompt character when it is ready for the next line of input (see **Host prompt character**), you may need to specify a non-zero value in this field. This field specifies the amount of time (in 1/10th seconds) that Reflection should wait after transmitting a carriage return character (the line delimiter) before it begins transmitting the next line.

Values: 0-99
Default: **0**

Color scrolling method/speed

The IBM color/graphics adapter card is not capable of scrolling fast enough to keep up with a high speed communications link (faster than 2400 bits/second) without losing display quality. If a program on the PC writes to the color/graphics display memory without waiting for a *retrace* interval, interference (*snow*) is produced on the monitor.

The display routines in the ROM BIOS, which are used by DOS, avoid display interference during scrolling by turning the display off, moving the data in the display memory, then turning the display back on. While this method produces the fastest possible scrolling without snow, it also causes the entire display to blink noticeably on each line scrolled.

This field allows you to choose the method of vertical scrolling that you want Reflection to use. The cleanest and slowest method is **SPEED 2**. The fastest method is the method used by DOS, called **BLINK**. In between are several speed selections, **SPEED 3** through **SPEED 6**. The higher numbers provide faster scrolling, but with more display interference. If you are communicating at 2400 baud or less, the default value, *SPEED 3*, should provide adequate performance.

Some color/graphics display adapter cards claim flicker-free operation. If you have one, try configuring this field to *MAXIMUM*. This tells Reflection to write to the display memory as fast as it can. Scrolling speed also affects monochrome monitors to provide a function similar to the terminal's smooth scroll feature.

NOTE: This field must be set to *MAXIMUM* when Reflection is run on an IBM 3270/PC or Zenith 170/171 (see page UM-12 concerning /M).

Values:		
	SPEED 2	**SPEED 6**
	SPEED 3*	**BLINK**
	SPEED 4	**MAXIMUM**
	SPEED 5	

Terminal ID response

This field specifies Reflection's response to a *terminal ID status request* received from the host computer. Normally, you should use the default, *2392A* in Reflection 1.

The HP 2624B terminal has either 16K or 32K of display memory, as compared to the HP 2622A's 4K. Some host software may use the **Terminal ID response** to determine the amount of display memory that it will use. You may want to specify *2624B* to cause some host programs to take advantage of some of Reflection's extensive display memory. Also, you should set this field to *2624B* if you want to use the forms cache feature with V/PLUS.

The 2626A terminal provides for lines up to 160 characters in length, using horizontal scrolling. You may want to set this field to *2626A* in order to use horizontal scrolling with some host software.

Values: **2392A***
2622A
2626A
2624B
2623A (Reflection 3 only; default)
2627A (Reflection 7 only; default)

WARNING: Reflection does not completely emulate an HP 2624B terminal. Specifying *2624B* in this field may cause some host programs to send escape sequences that Reflection cannot interpret, making the system operate incorrectly.

Display memory status response

This field specifies the amount of display memory to be reported to the host computer as part of a primary status response. Some host software may be able to use Reflection's extended display memory if you set this field to a value greater than 4K.

Values: **4K***
8K
12K
15K

Forms buffer size

This field specifies the amount of memory to be allocated to the forms cache buffer. Enter the desired number of 256-byte blocks. The value must be in the range 0 through 96. If you change this value and activate the configuration, the contents of display memory and the printer buffer are cleared. If there is not enough memory for the specified forms buffer, the size of the buffer is not changed.

Values: 0-96 blocks
Default: **0**

HP TERMINAL CONFIGURATION, PAGE 2

Left margin	1
Right margin	80
Cols / horizontal scroll	1
Host prompt character	◄
Char xmit delay	0
Line xmit delay	0

Color scrolling	SPEED 3
Terminal ID response	2623A
Initial graphics display	OFF
Tek 4010 compatibility	OFF
Display memory size	4K
Forms buffer size (x256)	0

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	13	27	ACTIVATE CONFIG	SAVE TO DISK	DISPLAY FUNCTNS	config keys

Figure 1.10 Graphics Terminal Configuration, Page 2 (HP)

Initial graphics display (Reflection 3 and 7)

The graphics display can be configured to be on or off when Reflection is loaded. A hard reset from the keyboard will reinitialize Reflection according to this setting. A hard reset from the host will *always* set the graphics display *ON*. Graphics reset ^{E}C*mR turns the graphics display on also.

Values: **OFF***
ON

Tek 4010 compat (Reflection 3 and 7)

This field allows you to set compatibility mode for Tektronix terminal emulation. Scaled mode divides the incoming X/Y coordinates by 2 so that the output of programs written for the 1024 x 780 Tektronix terminal can be displayed in full on the PC's smaller screen. There is some resolution loss. In unscaled mode the PC displays one-fourth of the image. Changing the origin changes the part of the image displayed.

Values: **OFF***
SCALED
UNSCALED

HP Display Enhancements Configuration

NOTE: If you select HP 2627A color terminal emulation, you will not be able to configure enhancements as described below. See page TR-54 for more information.

To bring up the Display Enhancements configuration screen:

1. [Alt]-[C] from any terminal mode screen brings up the config keys.
2. From the config keys, press [F6]. The Display Enhancements selection screen appears.

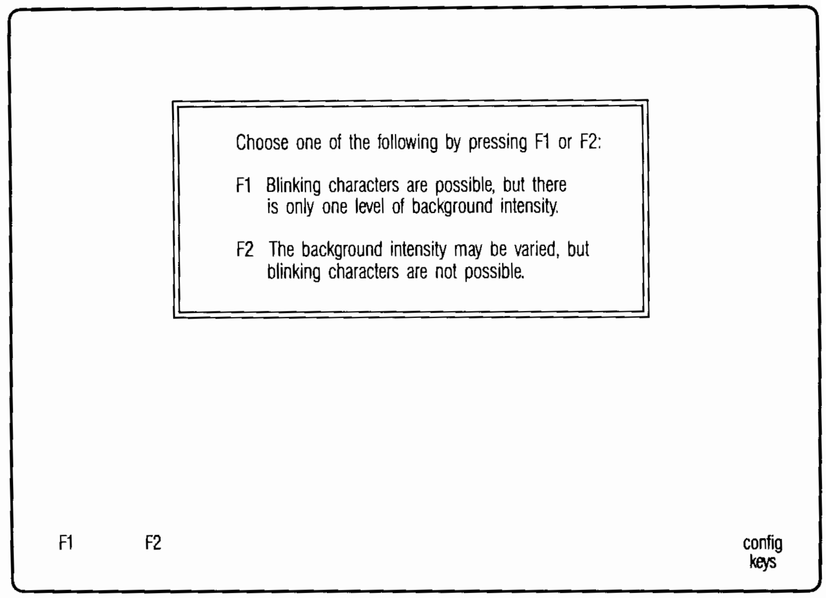


Figure 1.11 Display Enhancements Selection Screen

3. Select [F1] or [F2],depending on whether you want blinking characters or two levels of background intensity. The Display Enhancements configuration menu screen appears:

ENHANCEMENT CONFIGURATION

MONOCHROME	I	BG/FG	COLOR		HP ENHANCEMENT	B	BG	I	FG	
Black	0	000	Black	@	Normal Screen	0	000	1	111	SAMPLE
Underline	0	001	Blue	A	Normal,Blinking	1	000	1	111	SAMPLE
	0	010	Green	B	Normal,Inverse	0	111	0	000	SAMPLE
	0	011	Cyan	C	Normal,Inv,Blink	1	111	0	001	SAMPLE
	0	100	Red	D	Normal,Underline	0	000	1	000	SAMPLE
	0	101	Magenta	E	Norm,Undl,Blink	1	000	1	000	SAMPLE
	0	110	Brown	F	Norm,Inv,Undl	0	111	0	000	SAMPLE
White	0	111	White	G	Norm,Inv,Undl,Bl	1	111	0	000	SAMPLE
Black	1	000	Gray	H	Half Bright	0	000	0	111	SAMPLE
Underline	1	001	Light Blue	I	Half,Blinking	1	000	0	111	SAMPLE
	1	010	Light Green	J	Half,Inv (LABELS)	0	111	0	000	SAMPLE
	1	011	Light Cyan	K	Half,Inv,Blink	1	111	0	000	SAMPLE
	1	100	Light Red	L	Half,Underline	0	000	0	000	SAMPLE
	1	101	Light Magenta	M	Half,Undl,Blink	1	000	0	000	SAMPLE
	1	110	Yellow	N	Half,Undl,Inv	0	111	0	001	SAMPLE
Bright White	1	111	Bright White	O	Half,Undl,Inv,Bl	1	111	1	001	SAMPLE
					Border Color			0	000	

Legend: B — Blinking, BG — Background, I — Intensity, FG — Foreground

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES		ACTIVATE CONFIG	SAVE TO DISK		config keys

Figure 1.12 Display Enhancements Menu

4. A small arrow towards the upper right corner of the menu screen indicates the selected field. Move it by means of the cursor movement keys on the numeric keypad.

   The cursor appears as an inverse video block in the row of 0's and 1's in the selected field. Move it by means of the right and left cursor keys. When you press F1 or F2, you can toggle the bits on or off, which changes the colors displayed. The sample on the right shows what enhancements you are setting.

For example, if you want a light blue screen with black letters select the following for the row entitled *Normal Intensity*:

> Blinking attribute (or Bright) or B = 0
> Background attribute or BG = 001
> Intensity attribute or I = 0
> Foreground attribute or FG = 000

To set the function key labels to magenta with blue letters, set the following on the row entitled *Half Inverse*:

Blinking attribute (or Bright) or B = 0
Background attribute or BG = 101
Intensity attribute or I = 1
Foreground attribute or FG = 001

The following table lists the function-key labels for the display enhancements menu.

Table 1.5
HP Display Enhancement Labels

Label	Key	Action
NEXT CHOICE	F1	Changes value of field to next choice.
PREVIOUS CHOICE	F2	Changes value of field to previous choice.
DEFAULT VALUES	F3	Immediately restores all enhancements and the border color to their defaults, and redisplays the samples.
ACTIVE VALUES	F4	Restores all enhancements to their currently active values.
ACTIVATE VALUES	F5	Temporarily activates the values set and returns to terminal mode.
SAVE TO DISK	F6	Asks for a configuration filename, then saves a copy of the current configuration to the named file.
config keys	F8	Activates the enhancements as currently defined, clears the screen, switches to terminal mode, and displays the **config key** labels.

Configuring HP Dislpay Enhancements

The HP 2392A terminal supports 16 combinations of video display enhancements, including blinking, inverse video, underline, and half intensity. Any of the sixteen possible combinations can be selected by the escape sequence ^{E}c&d<x>, where <x> is one of the following:

@	No enhancement
A	Blinking
B	Inverse video
C	Blinking, inverse video
D	Underline
E	Blinking, underline
F	Inverse video, underline
G	Blinking, inverse video, underline
H	Half bright
I	Half bright, blinking
J	Half bright, inverse video
K	Half bright, blinking, inverse video
L	Half bright, underline
M	Half bright, blinking, underline
N	Half bright, inverse video, underline
O	Half bright, blinking, inverse video, underline

No single IBM PC adapter is capable of displaying all 16 enhancement combinations; therefore, Reflection lets you specify which of the 16 possible HP enhancements will be used on the PC. See page TR-281 for a table of combinations.

A monochrome adapter supports all four of the basic enhancements, but not all of the combinations. For example, it can't combine inverse video and underlining. Also, half intensity background and blinking foreground (i.e., characters) cannot be selected at the same time.

A color/graphics adapter doesn't support underlining. (The underline enhancement is blue on the color monitor.) If you are in graphics mode, inverse/underline is supported but blinking is not.

The current enhancements are shown in the **SAMPLES** column. Just to the left of the samples are rows of 0's and 1's, representing on/off toggles. The small arrow at the far right of the screen points to the current sample. To change the value, press $\boxed{\text{F1}}$, the **NEXT CHOICE** function key. Immediately, the sample on that line changes to show the effect, and the 0/1 toggle changes to reflect the current value of the enhancement.

For color display adapters, there is an additional field, **Border Color**, at the bottom of the screen. By changing these bit values, you can change the color of the display border.

Color Pairs in Terminal Enhancements

Configuring enhancements in Reflection 7 is done by selecting color pairs. Each pair consists of a foreground and a background color. User function key labels may also be configured for color; see page TR-111 for more information.

NOTE: Turning the background color to anything other than black will cause the graphics display to be completely blocked if the alpha display memory is on. Graphics are not visible when the alpha display is on and other than black. To create a background color for a graphics screen, select a background pen color with a graphics escape sequence.

To bring up the enhancement screen press F6 from the config keys.

ALPHA COLOR PAIR CONFIGURATION

COLOR PAIR #	FOREGROUND COLOR	BACKGROUND COLOR	SPECIAL USES
0	WHITE	BLACK	Normal text
1	RED	BLACK	
2	GREEN	BLACK	
3	YELLOW	BLACK	Half intensity
4	BLUE	BLACK	
5	MAGENTA	BLACK	
6	CYAN	BLACK	
7	BLACK	YELLOW	Function key labels

NEXT CHOICE | PREVIOUS CHOICE | DEFAULT VALUES | ACTIVE VALUES | ACTIVATE CONFIG | SAVE TO DISK | config keys

Figure 1.13 Alpha Color Pair Configuration

Only eight color pairs may be displayed at any one time. The default screen pair is 0, a black background with white letters. The default function key values are yellow labels with black letters. The default color pair options follow:

Table 1.6
Color Pair Defaults

Color Pair	*Foreground*	*Background*
0	*White*	*Black*
1	*Red*	*Black*
2	*Green*	*Black*
3	*Yellow*	*Black*
4	*Blue*	*Black*
5	*Magenta*	*Black*
6	*Cyan*	*Black*
7	*Black*	*Yellow*

VT Terminal Configuration

This reference section examines each VT terminal configuration field in the order it appears on the Terminal Configuration, Page 1 and Page 2 menu screens. An asterisk (*) indicates default values.

To access the configuration menus, press [Alt]-[C]. Then press [F3] again to bring up the Terminal Configuration, Page 1 menu.

VT TERMINAL CONFIGURATION, PAGE 1

Multi-page	YES
Initial label set	SYSTEM
User labels lines	2
Cursor style	LINE
Screen background	DARK
End of line wrap	NO
Caps lock	NO
Margin bell	YES
Auto Repeat	YES

UK character set	NO
Break key enabled	YES
Screen print extent	PAGE
Local echo	NO
Backspace key	DEL
Keypad mode	NORMAL
Cursor key mode	NORMAL
User features	UNLOCKED
Terminal type	VT102
DA response	VT102

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	ACTIVATE CONFIG	SAVE TO DISK	DISPLAY FUNCTNS	config keys

Figure 1.14 Terminal Configuration, Page 1 (VT)

As with other configuration screens in Reflection, some values are entered through the keyboard, while most require you to use the function keys to cycle through a predefined list of choices in order to select a value.

Once you've entered the new values, you can activate them for current use, or you can save them to disk (which also activates them). When you save a value to disk, the current values of all other configuration menus are saved in the same file. See page TR-18 for more information on changing configuration values.

Terminal, Page 1 Menu Parameters

Multi-page mode

Reflection allows your PC to emulate a multi-page terminal that dynamically stores data. Application programs designed for DEC terminals, however, require that the terminal have only a single page of memory (24 lines). If your application program allows only single-page mode, configure this field to *NO*.

When single-page mode is used, the following key functions are disabled: Next page, Prev page, Home up, Home down, Roll up, and Roll down.

Values: **YES***
NO

Initial label set

Initial label set determines which set of screen labels will be displayed when Reflection is loaded.

Values: **SYSTEM***
MODES
USERS
PF KEYS

User label lines

Specifies the number of screen rows to be used for user key labels. The default value is *2*, which leaves only 23 lines of the PC's 25-line display available for text. If this field is set to a value of *1* or *NONE*, Reflection uses 24 lines for text whenever the user keys are called up. (On 27-line displays 25 lines are used regardless of the labels presented.)

WARNING: Some software will not work properly with two label lines (e.g., *ALL-IN-1*).

Vectras and PCs with enhanced graphics adapters automatically provide 24 lines of text, two label lines, and a status line. EGAs require that the file EGA7X12.FT be present in order to load in the HP font which makes 27 lines possible. You can select the IBM font on either computer by using a /I switch. The 27 lines are reduced to 25 on EGAs. To force a complete clone of the IBM PC on the Vectra, use /C. This ensures 25 lines as well as the IBM font. Use this if interference with other programs occurs due to the 27-line mode.

Values: **2***
1
NONE

NOTE: Only the user-key labels can be reduced in size; all other labels always use two lines.

Cursor style

The cursor may appear as a blinking underline, a blinking box, or simply not be visible. This last option is often called a *text cursor*. If the cursor is selected as a blinking underline or box, pressing Ins or an escape sequence will toggle the cursor style between insert and replace mode.

Values: **LINE***
BLOCK
NONE

Screen background

Specifies a light or dark background for the screen. This overrides the Display Enhancements Configuration screen values.

Values: **DARK***
LIGHT

End of line wrap

When this field has a value of *YES*, Reflection automatically returns the cursor to the left margin in the next (lower) line when the cursor reaches the right margin or the right screen edge.

When **End of line wrap** has a value of *NO*, the cursor is not automatically advanced when it reaches the right margin. As you type additional characters, each character overwrites the character at the right margin, until you explicitly move the cursor with a carriage return or other cursor movement key.

Values: **YES**
NO*

Caps lock

When this field contains a value of *YES*, Reflection limits the set of characters that can be produced from the keyboard to Teletype-compatible codes.

1. No lower-case alphabetic characters are generated. All alphabetic keys are automatically upshifted.
2. [{], [}], and [|] generate the codes for [,], and \, respectively.
3. The ~ and ' codes are disabled.

Values: **YES**
NO*

Margin bell

When this field contains a value of *YES*, Reflection emits a beep under the following conditions:

1. When a character is typed from the keyboard into the column which is eight columns left of the right margin
2. In file transfer mode, when file transmission is complete or terminated in error

When *NO* is specified in this field, the bell sounds only when Reflection detects an input error, or when an ASCII BELL character ([Ctrl]-[G]) is received from the host or entered from the keyboard.

Values: **YES***
NO

Auto repeat

Setting this field to *YES* causes the keyboard to automatically repeat any key that is held down. When this field is set to *NO*, keys do not automatically repeat. When loaded, Reflection does not modify the repeat speed and initial delay in use by the keyboard processor.

Values: **YES***
NO

UK character set

Specifies whether the default character set (G0) for the terminal will be the normal ASCII or UKASCII. The only difference between the two character sets is the ASCII character #, which is replaced by the British pound symbol, £.

Values: **YES**
NO*

Break key enabled

Enables and disables the action of [Break].

Values: **YES***
NO

Screen print extent

When a screen print operation is performed with [Alt]-[PrtSc], the **Screen print extent** parameter determines how much of the screen is to be printed. A value of *PAGE* will cause Reflection to print the entire screen. A value of *REGION* will cause Reflection to print only the scrolling region, as defined by the top and bottom margins.

Values: **PAGE***
REGION

Local echo

When **Local echo** has a value of *YES*, each character typed at the keyboard is immediately displayed on the screen.

When remote mode is selected, each character typed at the keyboard is transmitted to the host computer. Most host systems (e.g., the HP 3000 and the VAX/VMS) immediately send the same character back to the terminal (i.e., *echo* the character). The character is not displayed on the screen until it is received back from the host. This is sometimes called *full duplex.*

When local echo and remote mode are both selected, each character is sent two places, directly to display memory (the screen) and to the host computer.

Therefore, if you are communicating with a host computer that echoes, and the Reflection **Local echo** parameter has a value of *YES*, each character you type will appear twice on the screen. For this reason, set this value to *YES* only when communicating with host systems that do not echo each typed character (for example, some public networks). When using such a system, you should set **Local echo** to *YES* so that you can see characters as you type them.

When Reflection is in local mode, **Local echo** is automatically enabled.

Values: **YES** (half duplex)
NO* (full duplex)

Backspace key

Selects the meaning of [Backspace]. On the standard PC keyboard, [Backspace] generates a backspace character (decimal 8, ASCII BS), and [Ctrl]-[Backspace] generates a delete character (decimal 127, ASCII DEL). By specifying *DEL* in the **Backspace key** option the assignment can be reversed so that [Backspace] will generate a DEL and [Ctrl]-[Backspace] will generate a BS character.

The BS character causes the cursor to back up one position on the current line, but does not erase the character at that position. The DEL character is used by some host systems to mean *backup and delete the previous character*. On many hosts, when a DEL character is encountered, the previous character in the input buffer is deleted and a three-character sequence of BS/space/BS is echoed back to the terminal. This effectively backs up the cursor and erases the previous character from the screen.

Values: **BKSP**
DEL*

Keypad mode

Selects the codes to be generated from the keys on the numeric keypad. Either normal numeric values or special application escape sequences are generated. This option is usually controlled by an application program running on the host computer. Changing it locally without informing the host computer may cause problems. See page TR-365.

Values: **NORMAL***
APPLCTN

Cursor key mode

Selects the codes to be generated from the cursor keys. Either the normal cursor control escape sequences or special application escape sequences are generated. This option is usually controlled by an application program running on the host computer. Changing it locally without informing the host computer may cause problems. See page TR-365.

Values: **NORMAL***
APPLCTN

User features

Locks the following items so that they cannot be changed by the host computer: **Screen background**, **Auto repeat**, **tab stops**, and **keyboard lock**.

Values: **LOCKED**
UNLOCKED*

Terminal type

Specifies which terminal should be emulated: VT 52, VT 102, or VT 220 with 7- or 8-bit controls. The VT 52 and VT 102 are full terminal emulations; however, some features of the VT 220 are not emulated. These choices affect the codes generated by the numeric keypad and the function keys, the interpretation of escape sequences, and the response to terminal identification requests.

Values: **VT 52**
VT 102*
VT 220-7
VT 220-8

DA response

Device attribute response. The value in this field determines how Reflection responds to a primary device attribute request. See the following table:

Table 1.7
Device Attributes Request Responses

Value	*Reply*	*Meaning*
VT102*	^{E}c[?6c	VT 102
VT100	^{E}c[?1;0c	VT 100
VT100av	^{E}c[?1;2c	VT 100 advanced video
VT100p	^{E}c[?1;11c	VT 100 w/printer
VT220	^{E}c[?62;1;2;6;7;8c§	VT 220 normal
VT220x	^{E}c[?62;1;2;6c	Reflection

* The default.

§ The VT 220 response means: VT 200 family terminal (62) with 132 columns (1), printer port (2), selective erase (6), downloadable character sets (7), and user-defined keys (8).

Terminal, Page 2 Menu Parameters

VT TERMINAL CONFIGURATION, PAGE 2

Left margin	1
Right margin	80
Columns per horizontal scroll	10
Character xmit delay (x .001 seconds)	0
Line xmit delay (x .1 seconds)	0
Color scrolling method/speed	SPEED 3
Automatic answerback	NO
Conceal answerback message	NO

Answerback message	♠

NEXT CHOICE | PREVIOUS CHOICE | DEFAULT VALUES | ACTIVE VALUES | ACTIVATE CONFIG | SAVE TO DISK | DISPLAY FUNCTIONS | config keys

Figure 1.15 Terminal Configuration, Page 2 (VT)

Left margin

The left margin can be set by entering a number in this field. The value must not be greater than the right margin.

Values: 1-80
Default: **1**

Right margin

The right margin can be set to any value from 2 through 10000, but it must be greater than the value of the left margin. When a value greater than 80 is entered, horizontal scrolling becomes possible, allowing for entry and viewing of lines longer than 80 characters without wraparound.

WARNING: If you configure a right margin greater than column 80, Reflection will not work properly with host computer software that is programmed to wrap at column 80.

Values: 2-10000
Default: **80**

Columns per horizontal scroll

This field determines the number of columns, or character positions, to be rolled right or left during one horizontal scrolling operation with [Ctrl]-[Left] or [Ctrl]-[Right] keystrokes. The parameter has no effect if the right margin is set for fewer than 80 columns.

Values: 1-80
Default: **10**

Character xmit delay (x .001 seconds)

When transmitting blocks of characters at high speeds, it may be necessary to *throttle down* the effective speed of the transmission to allow the host computer to keep pace.

This field specifies the number of milliseconds of wait time that Reflection should insert after each character when transmitting blocks of characters to the host system.

Values: 0-255
Default: **0**

Line xmit delay (x .1 seconds)

This field specifies the amount of time (in 1/10th seconds) that Reflection should wait after transmitting a carriage return character (the line delimiter) before it begins transmitting the next line.

Values: 0-255
Default: **0**

Color scrolling method/speed

The IBM color graphics adapter card is not capable of scrolling fast enough to keep up with a high speed communications link (faster than 2400 bits/second) without losing display quality. If a program on the PC writes to the color graphics display memory without waiting for a *retrace* interval, interference (*snow*) is produced on the monitor.

The display routines in the ROM BIOS, which are used by DOS, avoid display interference during scrolling by turning the display off, moving the data in display memory, and then turning the display back on. While this method produces the fastest possible scrolling without snow, it also causes the entire display to blink noticeably on each line scrolled.

This field allows you to choose the method of vertical scrolling that you want Reflection to use. The cleanest and slowest method is **SPEED 2**. The fastest method is the method used by DOS, called **BLINK**. In between are several speed selections, **SPEED 3** through **SPEED 6**. The higher numbers provide faster scrolling, but with more display interference. If you are communicating at 2400 baud or less, the default value, *SPEED 3*, should provide adequate performance.

Some color graphics display adapter cards made by add-on manufacturers claim flicker-free operation. If you have such an adapter, try configuring this field to a value of *MAXIMUM*. This tells Reflection to ignore retrace intervals and to write to the display memory as fast as it can.

Scrolling speed also affects monochrome monitors to provide a function similar to the terminal's smooth scroll feature.

NOTE: This field must be set to *MAXIMUM* when Reflection is run on an IBM 3270/PC or a Zenith 170/171. See page UM-12.

Values:		
	SPEED 2	**SPEED 6**
	SPEED 3*	**BLINK**
	SPEED 4	**MAXIMUM**
	SPEED 5	

Automatic answerback message

If set to *YES*, the answerback message is automatically transmitted to the host in response to an ENQ (decimal 5) character.

Values: **YES**
NO*

Conceal answerback

When configured to *YES*, the current answerback message is concealed. Changing to *NO* will not cause the answerback message to be redisplayed.

Values: **YES**
NO*

Answerback message

This 30-character message will be transmitted to the host when the answerback key (Ctrl-F2) is pressed or upon the receipt of an ENQ character (decimal 5) from the host if **Automatic answerback** is configured to *YES*.

Values: Any message up to 30 characters
Default: ACK character (Ctrl-F2) (decimal 6)

VT Display Enhancements Configuration

Access the Display Enhancements Configuration screen:

1. Alt-C displays the config keys.
2. From the **config keys** labels, press F6. The Display Enhancements Selection screen appears.

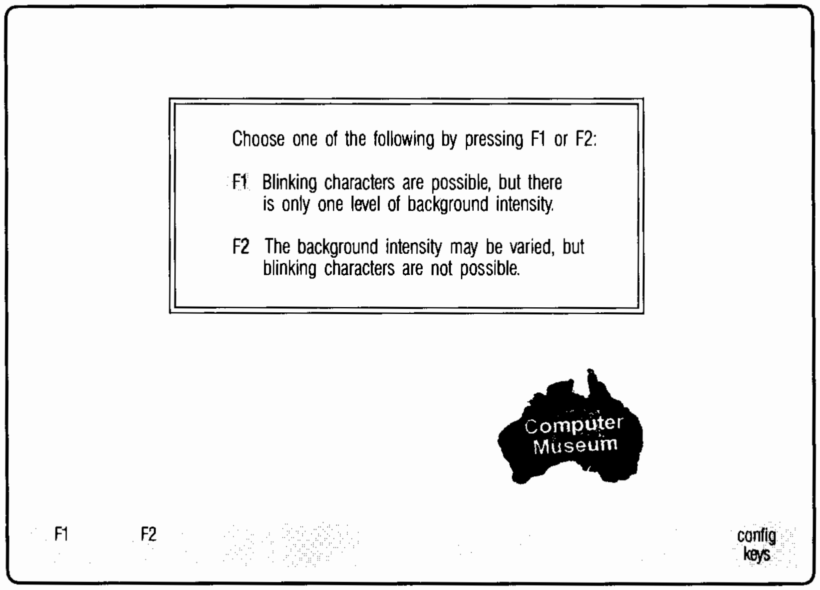


Figure 1.16 Display Enhancements Selection Screen

3. Select [F1] or [F2], depending on whether you want blinking characters or two levels of background intensity. The Display Enhancements Configuration menu appears.
4. A small arrow toward the upper right corner of the screen indicates the selected field. Move it by means of the [Up] and [Down] cursor keys on the numeric keypad.

The cursor appears as an inverse video block in the row of 0's and 1's in the selected field. Move it by means of the [Right] and [Left] cursor keys on the numeric keypad.

ENHANCEMENT CONFIGURATION

MONOCHROME	I	BG/FG	COLOR	DEC ENHANCEMENT	B	BG	I	FG	
Black	0	000	Black	Normal Screen	0	000	0	111	SAMPLE
Underline	0	001	Blue	Normal,Blinking	1	000	0	111	SAMPLE
	0	010	Green	Normal,Inv (LABELS)	0	111	1	000	SAMPLE
	0	011	Cyan	Normal,Inv,Blink	1	111	1	000	SAMPLE
	0	100	Red	Normal,Underline	0	110	1	001	SAMPLE
	0	101	Magenta	Norm,Undl,Blink	1	110	1	001	SAMPLE
	0	110	Brown	Norm,Inv,Undl	0	111	1	000	SAMPLE
White	0	111	White	Norm,Inv,Undl,Bl	1	111	1	000	SAMPLE
Black	1	000	Gray	Bold	0	000	1	111	SAMPLE
Underline	1	001	Light Blue	Bold,Blinking	1	000	1	111	SAMPLE
	1	010	Light Green	Bold,Inverse	0	111	0	000	SAMPLE
	1	011	Light Cyan	Bold,Inv,Blink	1	111	0	000	SAMPLE
	1	100	Light Red	Bold,Underline	0	110	0	001	SAMPLE
	1	101	Light Magenta	Bold,Undl,Blink	1	110	0	001	SAMPLE
	1	110	Yellow	Bold,Undl,Inv	0	111	0	000	SAMPLE
Bright White	1	111	Bright White	Bold,Undl,Inv,Bl	1	111	0	000	SAMPLE

Legend: B — Blinking, BG — Background, I — Intensity, FG — Foreground

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	ACTIVATE CONFIG	SAVE TO DISK	config keys

Figure 1.17 Display Enhancements Menu (VT)

For example, if you want a light blue screen with black letters select the following for the row entitled *Normal Intensity*:

> Blinking attribute (or Bright) or B = 0
> Background attribute (BG) = 001
> Intensity attribute or I = 0
> Foreground attribute (FG) = 000

The background color is the screen or label color; the foreground color is the color of the letters imposed on the background color.

To set the function key labels to magenta with blue letters, set the following on the row entitled *Normal Inverse*:

> Blinking attribute (or Bright) or B = 0
> Background attribute or BG = 101
> Intensity attribute or I = 1
> Foreground attribute or FG = 001

The following table lists the function key labels for the display enhancements menu.

Table 1.8
VT Display Enhancement Labels

Label	*Key*	*Action*
NEXT CHOICE	F1	Changes value of field to next choice.
PREVIOUS CHOICE	F2	Changes value of field to previous choice.
DEFAULT VALUES	F3	Immediately restores all enhancements and the border color to their defaults, and redisplays the samples.
ACTIVE VALUES	F4	Restores all enhancements to their currently active values.
ACTIVATE VALUES	F5	Temporarily activates the values set and returns to terminal mode.
SAVE TO DISK	F6	Asks for a configuration filename, then saves a copy of the current configuration to the named file.
config keys	F8	Activates the enhancements as currently defined, clears the screen, switches to terminal mode, and displays the config keys set of labels.

Configuring VT Display Enhancements

VT terminals support 16 combinations of display enhancements, including blinking, inverse video, underline, and bold intensity. Neither of the IBM PC monitors is capable of displaying all 16 display enhancement combinations, therefore Reflection lets you specify which of the 16 possible enhancements will be used on the PC.

A monochrome adapter supports all four of the basic enhancements, but not all of the combinations. For example, it can't combine inverse video and underlining. Also, the blinking enhancement for the foreground (i.e., characters) can have a background with only one level of intensity.

A color graphics adapter doesn't support underlining (the monochrome underline enhancement shows as blue on the color monitor).

The current attributes are shown in the **SAMPLES** column. Just to the left of the samples are rows of 0s and 1s, where 0 = off and 1 = on. A small arrow at the far right of the screen points to the current sample. To change the value, press F1, **NEXT CHOICE**, or F2, **PREV CHOICE**. Immediately, the sample on that line changes to show the effect, and the 0/1 toggle changes to reflect the current value of the enhancement.

For color display adapters, there is an additional field, **Border Color**, at the bottom of the screen. By changing these bit values, you can change the color of the border around the color display.

Using Display Enhancements

Reflection's command language allows you to send escape sequences to the terminal for a variety of purposes. You can send escape sequences to select enhancements when creating local screens, for example. By using the DISPLAY command, you can set an enhancememt and then display the text so that it is displayed with that enhancement value.

Any of these combinations may be selected by the escape sequence:

^{E}C[<n>;...<n>m

where **<n>** is one or more of the following:

Table 1.9
Values for Display Enhancements (<n>)

Attribute	*On*	*Off*
Bold	*1*	*22*
Underline	*4*	*24*
Blink	*5*	*25*
Inverse video	*7*	*27*
Clear all attributes		*0*

For example, entering ^{E}c[1;5m will cause all characters that follow to be displayed as bold blinking. In a command file use DISPLAY "^[[1;5m".

Printer and Plotter Configuration

Printer Configuration

Complete the following Printer Configuration menu even if you don't have a printer attached to the PC. Setting the **Printer interface** field to *NONE* indicates no printer is available and streamlines disk operations.

To access the Printer Configuration menu, use [Alt]-[C] to display the config keys; then press [F5], **printer config**.

PRINTER CONFIGURATION

Printer interface	PARALLEL
Device name	N/A

PRINTER FEATURES

Control codes	IBM/EPSN
Normal line width	80
Compressed line width	132
Expanded line width	40
Form Feed after Alt PrtSc	YES

DATACOMM SPECIFICATION

Buffer size (K bytes)	0
Baud rate (serial)	N/A
Parity (serial)	N/A
Flow control (serial)	N/A
Stop bits (serial)	N/A

NEXT CHOICE | PREVIOUS CHOICE | DEFAULT VALUES | ACTIVE VALUES | ACTIVATE CONFIG | SAVE TO DISK | config keys

Figure 1.18 Printer Configuration Menu

As with other configuration screens in Reflection, some values are entered through the keyboard, while others require you to use the function keys to cycle through a predefined list of choices in order to select the value. See page TR-18 for details on setting configuration menu values.

Once you've entered the new values, you can activate them for current use, or you can save them to disk (which also activates them). When you save a value to disk, the current values of all other configuration menus are saved in the same file.

Printer Configuration Parameters

Printer interface

This field specifies the type of printer interface, if any, through which the printer is attached to the PC. *PARALLEL* implies that the printer uses the parallel printer adapter on the IBM PC. Reflection allows you to define parallel ports 1, 2, 3, or 4 as the printer interface.

If you specify *SERIAL*, you must configure the proper baud rate and parity. Reflection allows you to define serial ports 1, 2, 3, or 4 as the printer interface.

If you specify *DEVICE*, you must enter the proper device name. You can default to whatever mode settings you already have defined by selecting a DOS device name: PRN, LPT1, or LPT2.

If you specify *NONE*, no further configuration is necessary.

Values:	**PARALLEL***	**SERIAL-1**
	PARALL-2	**SERIAL-2**
	PARALL-3	**SERIAL-3**
	PARALL-4	**SERIAL-4**
	NONE	**DEVICE**

Printer Features

Control codes

This field specifies the control codes, if any, that the printer uses to enable and disable expanded and compressed print. If the printer does not have these features, select the value *OTHER*.

Values:	**IBM/EPSON***	**HP**
	IBM PRO	**TOSHIBA**
	OKIDATA	**OTHER**
	ANSI	

Table 1.10
Printer Control Codes

Printer	*Expanded*	*Compressed*	*Normal*
IBM / Epson	Ec!SPC	Ec!EOT	Ec!NUL
Proprinter	DC2 EcWSOH	EcWNULSI	EcWNULDC2
Okidata	US	CAN	GS RS
Ansi (Tally)	Ec[0w	Ec[6w	Ec[4w
HP	Ec&k1S	Ec&k2S	Ec&k0S
Toshiba	Ec!	Ec[	Ec] Ec"
Other	(no codes)	(no codes)	(no codes)

CAN=24 DC2=18 EOT=4 Ec=27 GS=29 NUL=0
RS=30 SI=15 SOH=1 SPC=32 US=31

Normal, compressed, and expanded line width

These fields specify a maximum line width for each of the print sizes. Reflection limits the maximum number of characters that it will print on a line to these values.

Values: 1-9999
Defaults: Normal = **80**
Compressed = **132**
Expanded = **40**

Datacomm Fields

Buffer size (K bytes)

You can set aside a portion of memory to be used for printer buffering if your printer interface is other than *DEVICE*. Having a printer buffer frees the keyboard for use during printing.

When the printer buffer is initially set, it shares the memory set aside for display memory. Only as much memory as is currently available is used for the printer buffer. If saved to a configuration file, the full printer buffer size requested will be used the next time Reflection is loaded. Display memory will also be set according to its configured value, separately.

Values: 0-255 K bytes
Default: **0**

Baud rate (serial)
Specifies the speed at which Reflection transmits data to the printer through a serial port.

Values:		
	110	**1800**
	150	**2400**
	300	**4800**
	600	**9600***
	1200	**19200**

Parity (serial)
Check your printer documentation for the correct parity to use with your printer. See page TR-30 for more information on this field. Possible values:

NONE* Each character consists of eight data bits with no parity bit. If the high-order bit is used, and the printer has the Roman 8 character set available, the character will be interpreted as an extended Roman 8 character.

0's Each character consists of seven data bits, with the eighth bit always 0. This is often called *space* parity.

1's Each character consists of seven data bits, with the eighth bit always 1. This is often called *mark* parity.

ODD Odd parity.

EVEN Even parity.

Flow control (serial)
This field is used only if you specify *SERIAL* in the **Printer interface** field. It specifies how the flow of data from Reflection to the serial printer is to be controlled. Refer to the printer manufacturer's documentation to determine the correct value. Possible values:

NONE*	The serial printer has no way to stop and start the flow of data from the computer.
XON/XOFF	When the printer's buffer is nearly full, it sends an ASCII DC3 (XOFF) character to the computer. When the printer is ready for more data, it sends an ASCII DC1 (XON) to the computer.
ETX/ACK	After sending each 250-character block of data to the printer, the computer sends an ASCII ETX character. The printer sends an ASCII ACK character to the computer when it is ready for another block.
DSR	The computer sends characters to the printer as long as the RS232C Data Set Ready signal (pin 6) is active, *true*. The printer sets the line *false* when it is not ready for more data.
CTS	The computer sends characters to the printer as long as the RS232C Clear To Send signal (pin 5) is active, *true*. The printer sets the line *false* when it is not ready for more data.

Form Feed after PrtSc

When configured to *YES*, Reflection automatically advances paper to top of form after a print-screen operation (Alt-PrtSc).

Values: **YES***
NO

Stop bits (serial)

Every character is followed by either one or two stop bits. You should specify one stop bit for all baud rates other than 110 baud, and two stop bits for 110 baud, unless you are positive that your host system or serial device requires otherwise.

Values: **1***
2

Configuring for a Plotter

Plotters may be attached via eavesdrop cable or via direct serial connection. Although an eavesdrop connection requires only requires one serial port, it forces you to physically switch cables if you are sometimes plotting with PC applications and sometimes with host applications. Reflection requires that plotters connected directly to the PC be configured for the correct connection, parity, and baud rate. For either type of connection, pacing (XON/XOFF, etc.) is handled during plotter operations by the plotter itself, rather than by Reflection. This pacing is rarely of concern to the user in that it is normally set by an escape sequence from the host.

To bring up the Plotter Configuration menu, use [Alt]-[C] to display the config keys; then press [F2], **plotter config.**

PLOTTER CONFIGURATION

Connection	EAVESDRP
Parity	NONE
Baud rate	9600

NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	ACTIVE VALUES	ACTIVATE CONFIG	SAVE TO DISK	DISPLAY FUNCTNS	config keys

Figure 1.19 Plotter Configuration Menu

Plotter Configuration Parameters

Plotter connection

This field specifies the type of plotter interface, if any, through which the plotter is attached to the PC. *EAVESDROP* implies that the plotter is connected between the PC and the host via a special *eavesdrop* cable. still apply, so that changing those fields on this screen has no effect. If you do not have a plotter, the default setting of *EAVESDROP* is correct.

If you are directly connecting your plotter to a separate serial port, you must indicate whether it is through COM1, COM2, COM3, or COM4. You must also configure proper baud rate and parity on.

Values: **EAVESDROP***
COM1
COM2
COM3
COM4

Parity

Check your plotter documentation for the correct parity to use with your plotter. See page TR-30 for more information on this field. Possible values:

NONE* No parity indicates all 8 data bits sent to the plotter will be used.

0's Only seven data bits are used, with the eighth bit always 0. This is often called *space* parity.

1's Only seven data bits are used, with the eighth bit always 1. This is often called *mark* parity.

ODD Odd parity.

EVEN Even parity.

Baud rate

Specifies the speed at which Reflection transmits data to the plotter through a serial port.

Values:		
	110	**1800**
	150	**2400**
	300	**4800**
	600	**9600***
	1200	**19200**

File Transfer Set-Up and Configuration

Reflection supports five file transfer protocols:

- Reflection's 8-bit protocol to HP 3000s
- Reflection's 7-bit protocol to a VAX/VMS
- Reflection's 8-bit protocol to either a VAX/VMS or a UNIX system
- KERMIT public domain protocol for a variety of hosts
- XMODEM public domain protocol for a variety of hosts

You may also perform clear text transfers to any host. Host software is provided for MPE (HP 3000), VMS, UNIX, and some MS-DOS operating systems. Reflection's file transfer system has several advantages over KERMIT and XMODEM transfer techniques.

- Faster transfers because data compression is used.
- More control of file types and data translation is provided through configuration menus.
- The minimum number of keystrokes are necessary to complete a successful transfer.
- The Reflection PLUS backup and restore facility is available.

The defaults throughout the file transfer menu and configuration menus are for transfers to either an HP 3000 (HP terminal emulation) or a VAX/VMS system (VT terminal emulation). Specific information about each host and other protocols are in the chapters which follow.

Reflection file transfers can be optimized and controlled through the options described in the following chapter. Most KERMIT, XMODEM, and ASCII text transfers are not affected by these options.

Before you can use the Reflection file transfer, you must upload the host program to your host system. The process is automated with Reflection's command language.

Uploading the Host Program

Before you can transfer files between systems, a program must exist on the host to work with Reflection to error-check the files. (If you are merely transmitting data with the TRANSMIT command, an editor or other program must be running on the host to receive the data, and no error-checking is done).

Depending on the type of host you are transferring files to, the process for getting the host program on the host varies. See the page references below for each type of host.

Uploading PCLINK to an HP 3000	See page TR-84.
Uploading VAXLINK to a VAX/VMS	See page TR-86.
Uploading UNIXLINK to a UNIX System	See page TR-86.
HP 1000 Transfers using XMODEM	See page UM-147.
PC to PC Transfers using PC2PC	See page UM-149.
Transfers using KERMIT	See page UM-143.

Uploading PCLINK to an HP 3000

The Reflection diskette contains a program file called *PCLINK.PUB*, which you must copy to the HP 3000 host before performing any file transfers in HP mode. This process is called *uploading*.

You only need to do this once for each host computer. The next time you use Reflection, the uploaded file should still be on the host. Use the following steps to upload PCLINK.PUB:

1. If Reflection is configured for other than the default values, press **basic config** from the system keys to check the parity setting.
2. Tab to the menu field labeled **Parity**.
3. If the parity field contains a value other than *NONE* or *0's*, press [F1] until the field reads *NONE* or *0's*.

4. Press [F6], **SAVE TO DISK** and press [Return] to save the configuration to the default config file (or **ACTIVATE** these settings for use with the current session only). The menu screen disappears and the system keys reappear.
5. Log on to the host computer.
6. From the system keys, press [F6], **file transfer**.
7. Press [F7], the **upload xfer pgm** key. A screen entitled *UPLOAD HOST XFER PROGRAM* is displayed. Read the instructions on this screen.

```
                    UPLOAD HOST XFER PROGRAM

The following will help insure the success of the UPLOAD procedure:

1. Configure your communications port with parity set to 'NONE' or '0's (the default).
2. Make sure that you are signed on to a host HP3000 with term-type equal to '10'
   (normal value). You can force this to be true by signing on with 'HELLO
   <logon>;TERM=10'.
3. If UPLOAD is unsuccessful, then before re-trying insure that no host program is
   still executing. This can be done by typing:
                    <break> <esc> ':' 'ABORT' <cr>


Enter target program name: PCLINK.PUB.SYS






START    file                 18   30   DOS       config            system
UPLOAD   transfer                       command   transfer          keys
```

Figure 1.20 Upload Host Transfer Screen

8. At the prompt, either type the name to be given to the program on the host or press [Return] to upload under the default filename. PCLINK.PUB.SYS is the default name.
9. Press [F1], **START UPLOAD**. The message *Initiating upload* will appear. Then *Upload in progress* is displayed. A count of records transferred is displayed as the transfer proceeds.

10. If you want to abort the upload, press the **ABORT TRANSFER** key; if not, do nothing until the message *Successful Transfer* is displayed.

Uploading VAXLINK to a VAX/VMS

The Reflection diskette contains four files which upload VAXLINK for you. You only have to do this once for each DEC VAX/VMS computer. You must have VMS 4.0 or greater to use the VAXLINK host program.

To upload VAXLINK, do the following:

1. Make sure the following files are available in the current directory, or insert the support diskette in the current drive.
   - VAXLINK.HEX
   - WRQDEHEX.MAR
   - UPLOAD.VAX
   - WRQLOAD.COM
2. Start up Reflection and log on to the VAX.
3. Press [F6] from the system keys to display the file transfer keys.
4. Press [F7] **Upload xfer pgm.**

At this point the command file UPLOAD.VAX will transmit several files to the VAX. The process takes 12 - 18 minutes at 9600 baud. With a modem transfer you must have a clear connection, i.e., without a lot of noise on the line. After it is finished, you should see *Successful Transfer* displayed on the screen. You will then find a file called *VAXLINK.EXE* in your current directory on the VAX.

Uploading UNIXLINK to a UNIX System

The upload for UNIX systems is controlled by a command file called *UPLOAD.UNX*. This file describes the upload process, automatically sets the correct configuration values, and uploads the host file. The file is actually uploaded twice and then compared to ensure a successful transfer. Then the program is compiled (a C-compiler is required). Use the following steps to set up file transfers to a UNIX system:

1. Connect the PC to the host just as you would connect a terminal.
2. Make sure the files UNIXLINK.C and UPLOAD.UNX are available on the current drive of the PC. You must have a C compiler.
3. Load Reflection.
4. To execute the command file UPLOAD.UNX, simply use [Alt]-[Y] to bring up the command line and enter the filename UPLOAD.UNX. Correct configuration values will be set automatically.
5. A series of prompts will walk you through the upload process.

File Transfer Configuration Menu

Depending on your situation, you may configure file transfer only once or each time you transfer files. The menu is the same for HP and VT emulations, but the defaults differ slightly. To configure for file transfer, follow this procedure:

1. From the system keys ([F10]), press [F6], **file transfer.**

SEND TO HOST	RECEIVE FROMHOST	error recap	SHOW STATS	4	36	DOS command	config transfer	upload xfer pgm	system keys

Figure 1.21 File Transfer Keys

2. From the file transfer key labels, press [F6], **config transfer.**

FILE TRANSFER CONFIGURATION

Host startup sequence	RUN PCLINK.PUB.SYS; PARM = 0
Receive timeout (seconds)	15
Data block size (64-512)	512
Error retry limit	10

ASCII TRANSLATION TO HOST

CR/LF = Record separator	YES
Expand embedded tabs	YES
Use ctrl-Z as EOF	YES
Change Roman-8 to ISO7	NO

ASCII TRANSLATION FROM HOST

Record separator = CR/LF	YES
Change spaces to tabs	NO
Write ctrl-Z EOF	YES
Delete trailing spaces	YES
Change ISO7 to Roman-8	NO

NEXT CHOICE   PREVIOUS CHOICE   DEFAULT VALUES   ACTIVE VALUES   3   39   ACTIVATE CONFIG   SAVE TO DISK   DISPLAY FUNCTNS   exit config

Figure 1.22 File Transfer Configuration Menu

As with other configuration screens in Reflection, some values are entered through the keyboard, while others require you to use the function keys to cycle through a predefined list of choices in order to select the appropriate value. See page TR-18 for details about changing configuration values.

Once you've entered the new values, you can activate them for current use, or you can save them to disk, which also activates them. When you save a value to disk, the current values of all other configuration menus are saved in the same file.

If you will be using a variety of file transfer methods, you can save each configuration to a separate config file, as in UNIX.CFG, to ensure correct configuration values.

File Transfer Configuration Parameters

Host startup sequence

Each time you start a file transfer, Reflection transmits a host startup sequence, as entered in this field, to the host computer.

This is a command to the host telling it to run the host transfer program, using whatever name you assigned when you uploaded the program.

Value: A string of up to 31 characters.
Default: **RUN PCLINK.PUB.SYS;PARM=0** (HP mode)
RUN VAXLINK (VT mode)

Changing PCLINK Defaults

Some of the default values which govern file transfers to an HP 3000 can be altered by changing the **Host Startup sequence**. On the File Transfer Configuration menu, use the following format:

RUN PCLINK.PUB.SYS;PARM=<#>

The following table describes available PARM values.

(bit#)	*Compression on Binary File Xfer Disabled (#14)*	*7-bit Mode Enabled (#15)*
*PARM= 0**	*NO*	*NO*
1	*NO*	*YES*
2	*YES*	*NO*
3	*YES*	*YES*

PARM= can also be used to change the default timeout value for PCLINK (20 minutes). Bits 0-12 (left to right) are used to indicate the timeout interval in seconds. For example, to set a timeout of 30 seconds use **PARM=240** (240 is equal to 8 * 30). If the parm value of these bits is zero, the default timeout is used.

Changing VAXLINK Defaults

To change the defaults of the host startup sequence for VAX/VMS hosts you must install VAXLINK as a foreign command by adding the following to your LOGIN.COM file. <

$VAXLINK :== $[dev][directory]VAXLINK.EXE <

* The default settings.

§ NOTE: PCLINK uses HPPCPORT as the formal designator to determine the LDEV. It defaults to the session LDEV, but you may override this with a file equation. The first 20 words of PCLINK global data are unused. They are initialized to nulls.

If you install VAXLINK as a foreign command, then the following parameters may be used:

VAXLINK [e] [t <seconds>] [n <integer>] [r] [o]

e Specifies 8-bit file transfer. Using the **e** option increases the speed of transfers, but only works if there are no data-sensitive devices such as multiplexers or certain switches on your data line. See *VAXLINK 8-Bit File Transfers* at the end of this chapter.

t <n> Sets the number of seconds that will elapse between no response from the PC and a timeout. The default for 8-bit file transfer is 20 seconds; the 7-bit transfer mode default is no timeout. The VAXLINK timeout parameter should always be set to a value at least five seconds greater than the Reflection timeout parameter in the File Transfer Configuration menu (see below).

n <n> Sets the retry limit to <n> consecutive times; the default is 10.

r Puts file transfer into readsync mode.This setting should be used if file transfers fail with a *data overrun* error. Used only with 7-bit transfers.

o Allows you to perform file transfers to and from PCs using PC 2622 emulation software, an older Reflection product. Only used with 7-bit transfers.

Receive timeout (seconds)

This field tells the file transfer subsystem the maximum number of seconds to wait for a response from the host or PC. If nothing is received within the period specified, the program attempts to resynchronize. Two consecutive timeouts cause termination of the transfer. This setting also affects the PTRANSMIT command.

Values: 1-9999
Default: **15** (seconds)

Data block size (64-512)

Data block size tells the file transfer program how many bytes of data to transfer at a time.

The transfer program breaks a file into packets, which are transmitted one at a time until the entire file has been transmitted. You specify the number of data bytes each packet will contain (64-512). There need be no correlation between this field and the size of the records or blocks on the host computer.

The size of the block affects the speed of transfers. Too small a block results in long transmission times because of the overhead that occurs for each block. However, too large a block may increase the elapsed time because the chance of an error causing a retransmission increases with block size.

Additionally, the typeahead buffer may overflow if the block size is too large when transferring to a UNIX host. This will cause retransmission and possibly failure of the transfer. (You can combat this problem with the SET SUB-BLOCK-LENGTH command. A value of 50 will always work, however larger values may give a faster transfer.)

A large block size is more efficient when there are few transmission errors. Therefore, the value you assign to this field should depend on the likelihood of transmission errors. This is especially important if you're communicating to the host over a modem, which has a greater chance of error. Use 512 bytes per block in most situations with an HP 3000 host, and 64 bytes per block with a VAX host.

NOTE: If you are operating through ATPs at 19200 baud, lower your block size to 116. The length of the file transfers is a little better than at 9600 baud, and the small block size prevents lots of data errors or overrunning the port.

Values: 64-512 bytes
Default: **512** (bytes) (HP mode)
64 (bytes) (VT mode)

Error retry limit

The error retry limit tells Reflection how many consecutive times to attempt to correct an error before abandoning the transfer.

If an error condition occurs many times consecutively, you may have an insurmountable communications problem. With this parameter, you define the point at which an error state is declared unsolvable.

Values: 1-9999
Default: **10** (retries)

ASCII Translation to Host Fields

CR/LF = Record separator

Setting this field to *YES* tells the host program to generate a new record every time it receives the carriage return/linefeed sequence.

Reflection ignores this field when performing a binary file transfer. (Refer to page UM-128 for a discussion of binary and ASCII file transfer methods.)

Values: **YES***
NO

Expand embedded tabs

This field should be *YES* if you want the PC file's tab characters to be expanded to spaces in the host file. When this field has a value of *YES*, the system replaces each tab character with the number of spaces necessary to fill out to the next tab stop, where tab stops are assumed in column 9 and every eighth column thereafter.

Reflection ignores this field when performing a binary file transfer.

Values: **YES***
NO

Use Ctrl-Z as EOF

On the PC, ASCII text files normally end with a Ctrl-Z character. If you do not want this character to be copied into the host file, set this field to *NO*.

If you want to use the character count in the file directory to determine the file length, set this field to *NO*.

Reflection ignores this field when performing a binary file transfer.

Values: **YES***
NO

Change Roman 8 to ISO 7

If your host does not support the Roman 8 character set, set this field to *YES*. Roman 8 files will be translated to an ISO-compatible format during the transfer. The **Keyboard** setting on the Global Configuration menu determines the conversion. (See page TR-101.)

Values: **YES**
NO*

ASCII Translation from Host Fields

Record separator = CR/LF

If you want lines received from the host to be separated by a carriage return/linefeed on the PC, set this field to *YES*. Reflection ignores this field when performing a binary file transfer.

Values: **YES***
NO

Change spaces to tabs

If you want to generate tab characters to replace consecutive spaces, set this field to *YES*. Many software products on the PC understand how to use embedded tabs to indicate the standard 8-column tab stops. Configuring *YES* saves a lot of disk space on the PC.

NOTE: This field is also used outside the file transfer subsystem to control replacement of tabs with spaces when logging to disk or copying from display memory.

Reflection ignores this field when performing a binary file transfer.

Values: **YES**
NO*

Write Ctrl-Z EOF

When Reflection receives a file, it automatically adds an end-of-file marker at the end of the file if this field is set to *YES*. Many PC programs require the Ctrl-Z end-of-file marker.

Reflection ignores this field when performing a binary file transfer.

Values: **YES***
NO

Delete trailing spaces

Most PC text processing programs use delimited lines and paragraphs, and thus do not need blanks preceding a delimiter. You can save a great deal of disk space by setting this field to *YES*.

NOTE: This field is also used outside the file transfer subsystem when logging or copying to disk.

Reflection ignores this field when performing a binary file transfer.

Values: **YES***
NO

Change ISO 7 to Roman 8

If you want ISO files from the host to be converted to Roman 8, set this field to *YES*. ISO files will be translated to Roman 8. The **Keyboard** field on the Global Configuration menu determines the character conversion.

Values: **YES**
NO*

VAXLINK 8-Bit File Transfers

The VAXLINK file transfer program has an optional 8-bit protocol which is activated by appending the character **e** to the host startup sequence as described above. The 8-bit protocol improves transmission of ASCII files by 5-15% and of binary files by 25-50% over the default 7-bit protocol.

NOTE: Versions of Reflection prior to version 2.1 can also transmit files with this 8-bit protocol. Enter **SET HOST-PROMPT "^Q"** on the Reflection command line.

Transfer problems may occur when using 8-bit mode if there are communications hardware or software systems between the PC and the host computer. Typical examples of these intermediaries are public data networks like TYMNET and Telenet, or multiplexers. Some data networks can be configured to support 8-bit transfers; check with the service.

If it is necessary to abort an 8-bit transfer, press $\boxed{\text{F1}}$ **STOP**, at the File Transfer menu, and wait for an orderly termination. If this is unsuccessful, press $\boxed{\text{F5}}$ **ABORT**. If the file transfer program is still running on the VAX, STOP TYPING and wait for VAXLINK to timeout. This should take about 20 seconds unless you have changed the default timeout parameter. VAXLINK cannot be terminated by $\boxed{\text{Ctrl}}$-$\boxed{\text{C}}$ or $\boxed{\text{Ctrl}}$-$\boxed{\text{Y}}$ when you are using 8-bit mode.

Changing Global Characteristics

This chapter outlines *global* configuration. Global configuration allows you to alter basic characteristics of your terminal and keyboard. You can select a national language and 15 national keyboards. All menus and labels are displayed in the configured language. See page UM-97 for information about entering national characters.

Global Configuration Parameters

You can change the global configuration attributes temporarily or establish defaults which will be active any time you load Reflection. The selections include:

- Terminal class (or emulation mode)
- Operating language
- National keyboard
- 7-bit file operations
- Display memory size
- Hot-key definition
- 132-column adapter support

To bring up the Global Configuration menu, use the following steps:

1. Press $\boxed{\text{Alt}}$-$\boxed{\text{C}}$ to display the config keys.
2. Press $\boxed{\text{F7}}$ **global config.**

GLOBAL CONFIGURATION

Terminal class	HP
Language	ENGLISH
Keyboard	USASCII
Strip 8th bit on text files	NO
Amount of display memory (K)	20
Hot key	ALT-RSH
132 column adapter	NONE

NEXT CHOICE | PREVIOUS CHOICE | DEFAULT VALUES | ACTIVE VALUES | ACTIVATE CONFIG | SAVE TO DISK | DISPLAY FUNCTNS | config keys

Figure 1.23 Global Configuration Menu

If you decide to change any of these global settings after you are in Reflection, use the following steps:

1. At the Global Configuration menu, press [Tab] to get to the field to change ([Shift]-[Tab] moves backwards).
2. Press [F1] to display the next value.
3. Press [F5] to activate the new mode temporarily, or [F6] to save this choice to the configuration file as the default.

You are then returned to the system keys under the new configuration. See page TR-18 for more details on changing menu values in Reflection.

Terminal Class

The **Terminal class** field lets you select which class of terminals you wish to emulate, the Hewlett-Packard 2392A series or the Digital Equipment Corporation VT 100-200 series. The two emulations can be configured separately; a complete set of parameters can be stored in a configuration file for each emulation.

The **Terminal class** field lets you toggle between the two sets of configuration values. For example, if your PC has two serial ports you can simultaneously log on to a VAX and an HP 3000 by using different values in the **Datacomm port** field, and then switching between computers through the Global Configuration menu.

You may also use the SET TERMINAL-CLASS command to determine HP or DEC terminal class. If this command is used in a command file, the class will be switched, but the command file will abort. You cannot change this field once you have toggled Reflection into background mode. See page UM-23.

Values: **HP***
DEC

Language

The **Language** field lets you select the language that the labels, help screen, prompts, and error messages will use. Files with the extension .LOC (localization file) on the Reflection diskette contain the text for the various languages. English text is contained in the file R1U.LOC. In order to select a language, the appropriate LOC file must be available. When you want to return to English, the file R1U.LOC must be available to the program.

Values:	ENGLISH	(R1U.LOC)*
	FRENCH	(R1F.LOC)
	GERMAN	(R1G.LOC)

Keyboard

Some PCs have keyboards and keyboard drivers available for languages other than English. If you have a non-English keyboard, you should change the **Keyboard** field to match your keyboard.

In 8-bit operation the **Keyboard** field has no effect. In 7-bit operation it limits the characters that can be entered to those that are defined for the configured keyboard and determines the replacement characters that will be used during data communications. See *National Keyboards and Characters*, page UM-101 for a chart.

Values:	*USASCII**	*DEUTSCH*
	UK	*ITALIANA*
	NEDERL.	*NORSK*
	SUOMI	*ESP. LAT*
	CANADIEN	*ESPAÑOL*
	CANADIAN	*SVENSK*
	FRANCAIS	*SCHWEIZ*
	VLAAMS	*SUISSE*
	DANSK	

Strip 8th Bit on Text Files

The **Strip 8th bit on text files** field affects all disk input operations including ASCII file transfer, the TYPE, TRANSMIT, and PRINT commands, and all **READ DISK** operations. When this field is set to *YES*, the 8th bit of all data read in these operations will be set to zero.

If you are operating in English and do not need Roman 8 characters, set the **Strip 8th bit** field to *YES*. This ensures that any programs which use the 8th bit for special functions will translate properly. For example, *WordStar* files use the 8th bit for control sequences unique to *WordStar*. In order to view these files or transfer them to a mainframe system in a usable format, the 8th bit must be stripped.

If you are using the Roman 8 character set, the 8th bit must be kept. Set this field to *NO*.

Values: **NO** *
YES

Amount of Display Memory (K bytes)

Reflection allows you to configure the amount of memory to use for its own display memory. Both SHELL and the background option of Reflection allow you to execute DOS commands or programs from the command line. When you type SHELL without a command name, you exit Reflection temporarily to the DOS prompt. When you use a program name, you exit Reflection temporarily to the program prompt. Either **EXIT** typed at the DOS prompt or the program's exit command will return you to Reflection. (See *Command Language Keywords* on page TR-179.)

If you are using the background option, SHELL is not available to you. Instead you can use the hot-key to toggle out of Reflection to another program or to DOS. (See *Background Operations* on page UM-23.)

To make use of these options, you must set the **Amount of display memory (K bytes)** field to a sufficiently small value for running another program from within Reflection. 16K is recommended. Note that if printer and forms buffers are configured, that amount is subtracted from your available memory. Graphics work areas in Reflection 3 and 7 require a certain amount of memory which cannot be released without disabling their graphics capability.

Values: 0-700
Default: **16**

Hot-Key

The hot-key allows you to place Reflection in background mode in order to access DOS or another PC application. See *Background Operations* in the *User Manual* for more on this feature. The hot-key is initially set as [Alt]-[Shift] (right). When these keys are pressed, Reflection is immediately placed into background mode. Pressing these keys again brings Reflection back to the foreground.

A value of *NONE* prevents Reflection from being placed into background mode.

Values: **ALT-RSH** *
CTL-RSH
CTL-LSH
LSH-RSH
ALT-LSH
NONE

132-Column Adapters

This field allows you to indicate that you are using one of the following enhanced video adapters with a 132-column display: Tseng Lab's EVA or ULTRAPAK, or Chauffer HT from STB Systems. When set to *NONE*, Reflection uses horizontal scrolling if the right margin is greater than 80. An escape sequence for switching between 80 and 132 columns is on page TR-279.

Values: **NONE***
ULTRAPAK
EVA
Chauffer HT

Configuring User Keys

This chapter explains how to define the *user* set of function keys to make work with Reflection faster and easier in both HP and VT emulation modes. By redefining the meanings of function keys [F1] through [F8], the user can create one-stroke short cuts to perform repetitive, multi-stroke operations. For more technical information on escape sequences related to user keys, see *HP Escape Sequences* on page TR-274.

A user-key definition consists of the following:

- A 16-character label
- A string of up to 80 characters to be generated when the key is pressed
- An attribute that determines exactly how Reflection processes the string when the key is pressed

For technical information about the user keys, review *Defining the User Keys Programmatically* on page TR-109.

Displaying the User Keys

Press [F9] (or [Alt]-[U]) to display the currently-defined labels for the user keys and activate their current definitions. When you first load Reflection, the user keys are labeled as shown below, and are defined by escape sequences ^{E}cp through ^{E}cw .

Figure 1.24 User Keys

NOTE: When an HP 3000 program takes over the user keys, any definitions you have made may be erased. To restore the configured values for user keys, press [Alt]-[F]. Only saved user-key definitions are restored.

If the host program you are running does not define user keys, such as MCBA, any definitions you make may interfere with running it. In that case the host program expects the function keys to contain their default values. Restore these values by bringing up the definition menu and pressing F3. This temporarily removes Reflection definitions. See descriptions for STORE and RECALL in the *Command Language Keywords* appendix.

Changing the User-Key Labels

The height of the user-key labels may be set to two, one, or no screen rows. There are two ways to do this:

- Configure the number of user label lines in the Terminal Configuration, Page 1 menu. (See pages TR-35 and TR-57 for details). When the user keys are enabled, either from the keyboard or by the host, they occupy the configured number of rows.
- Display the user keys with F9. With each subsequent stroke of F9, the height of the label rows changes.

If the user keys are up, F9 toggles their height. Note that toggling the height with F9 is effective only for the current activation of the user keys. When the host system re-enables them, or when F9 (or Alt-U) is pressed, the number of rows reverts to the configured value.

Default User-Key Definitions

The default user-key definitions are as follows:

- The attributes are all **T** (transmit only).
- The labels are **f1** through **f8**.
- Each function key has a unique two-character escape sequence, as follows:

Table 1.11
User-Key Default Escape Sequences

Key	*Escape Sequence*
F1	^{E}cp
F2	^{E}cq
F3	^{E}cr
F4	^{E}cs
F5	^{E}ct
F6	^{E}cu
F7	^{E}cv
F8	^{E}cw

To use the default values of the user keys (when they are visible on the screen), press Shift with the appropriate function key. The defined function for that key will be ignored, and the default escape sequence will be sent to the host.

Defining the User Keys from the Keyboard

To create user-key definitions from the keyboard, press Ctrl-F9 to bring up the definition menu.

```
f1     T    LABEL          f1
←p
f2     T    LABEL          f2
←q
f3     T    LABEL          f3
←r
f4     T    LABEL          f4
←s
f5     T    LABEL          f5
←t
f6     T    LABEL          f6
←u
f7     T    LABEL          f7
←v
f8     T    LABEL          f8
←w

  NEXT    PREVIOUS  DEFAULT          1    4            SAVE     DISPLAY   system
 CHOICE    CHOICE   VALUES                            TO DISK   FUNCTNS    keys
```

Figure 1.25 User-Key Definition Menu

The screen displays a formatted user-key definition menu with two screen rows for each definition. The first row contains the function key label identification (f1, f2, etc.), a one-letter attribute field displayed in inverse video, the word *LABEL*, and two eight-character inverse video fields that make up the key's label. The second row contains space for the string of up to 80 characters that is to be generated when you press the key.

Move the cursor from field to field with Tab and Shift-Tab (Backtab) or Up/Down. The current field contains a blinking cursor.

One of three attributes must be assigned to each user-key definition. Change attribute fields by pressing F1 or F2, **NEXT CHOICE** and **PREVIOUS CHOICE**.

T Transmit-only. In remote mode, Reflection transmits the user-key string to the host computer, after completing a block transfer handshake, automatically transmitting a carriage return. In local mode, the key has no effect.

L Local-only. The user-key string is executed locally but is not transmitted to the host system.

N Normal keyboard input. The string is treated exactly as if it had been entered from the keyboard. A carriage return is *not* automatically transmitted. If Reflection is in local mode, the string is displayed on the screen, and embedded escape sequences are executed locally. In remote mode, with local echo off, the string is transmitted to the host computer, and is executed and displayed only if the host system echoes. In remote mode with local echo on, the string is executed and displayed locally, as well as being transmitted to the host. See *HP Terminal Configuration* on page TR-35 and *VT Terminal Configuration* on page TR-57 for information about the local echo feature.

In the two label fields, enter from the keyboard the user-key label exactly as it is to appear when the user keys are activated. The first field contains the upper half of the label, and the second field contains the lower half. For single-line user-key labels, make the first part of the label as descriptive as possible, since only the upper half will be displayed.

In the second line of each user-key definition, enter the exact character string the user key is to generate. Include escape sequences and ASCII control codes within the string by pressing [F7], the **DISPLAY FUNCTNS** key, before entering the control codes. Disable display functions mode immediately after entering the desired control code. (See page UM-63.)

When entering data into the label and definition fields, you can use all the editing keys ([Ins], [Del], clear line, etc.) as if format mode were active. The only exception is the clear display ([Alt]-[J]) function, which is disabled.

Reflection Commands in User Keys

User keys can be loaded with sequences which include command language keywords. Select *Local* and use the following sequence:

$^{E}{}_{C}$&oC<command>$^{C}{}_{R}$

Display functions must be on when you press [Esc] and [Return].

You could, for example, label a user key as DIR A: and include the following sequence:

$^{E}_{C}$&oCDIR A:$^{C}_{R}$

Pressing this user key would list the directory of the diskette in the A: drive.

User-Key Applications

Transmit user keys can be used to store commonly used commands such as modem dialing sequences or program run commands. Normal user keys can be used to store commands which have changing parameters. Local user keys are useful for Reflection commands and escape sequences. Some examples follow:

1. Store the sequence which invokes a host program you use frequently in a transmit function key. When you press the key while at the host prompt, the program is invoked.
2. Store a sequence where the parameters change occasionally in a normal function key. That way the command is not immediately executed when you press the user key. It waits for you to add any information, such as a filename, and then press $\boxed{\text{Return}}$.
3. The local user-key definition of $^{E}_{C}$&oCSHELL BASIC$^{C}_{R}$, would invoke Reflection's SHELL command and start the BASIC interpreter. When you left BASIC, you would return directly to Reflection.
4. Assume the file LOGON.CMD is a Reflection command file like the one on page TR-172 that dials your host computer and goes through the logon and password sequence. Then, by storing $^{E}_{C}$&oCLOGON.CMD$^{C}_{R}$ in a local user key, you could invoke this command file and, in one keystroke, log on to your system.
5. Using the above technique in conjunction with the user-key definition escape sequence discussed later in this chapter, you can build a tree of function labels. For example, assume that file USER2.CMD contained the following line:

   DISPLAY "^[&f1k0a8d16L EDTEDIT/EDT^M"

Then, if the definition for user-key 8 in your default configuration was ^{E}c&oCUSER2.CMD^{C}R, pressing user key 8 would change the definition of user key 1 to EDIT/EDT.

Saving User-Key Definitions

User-key definitions are automatically activated when you leave the definition menu. User-key definitions are saved along with configuration parameters in configuration files. These definitions can be saved in any configuration file you choose. After completing the user-key definition menu, press [F6], **SAVE TO DISK**, and enter the configuration filename to which you want to save them. Then press [Return].

Restoring Configured Values

Configured user-key values are sometimes not restored by host software. If a host application destroys your own user-key values, press [Alt]-[F]. (See below.) This matches the function of the RECALL command.

Storing and Recalling User-Key Definitions

Command language allows you to store Reflection's user-key labels for later recall. Because host application programs typically take over the user-key definitions, Reflection allows you to temporarily store values which can later be reset. The keywords are STORE and RECALL.

Defining the User Keys Programmatically

The host system can send an escape sequence to Reflection to define a user key. This escape sequence takes the form ^{E}c&f<attr><key><label length><string length><label><string>. Up to 160 characters can be loaded into a user key from the host.

The following table defines the parameters in the user-key definition escape sequence. The parameters **<attr>**, **<key>**, **<label length>**, and **<string length>** may appear in any order. However, the letter that identifies the last parameter must be in upper case, and all the preceding identifiers must be in lower case.

Table 1.12
User-Key Escape Sequence Parameters

Variable	*Parameters*	*Default*
<attr>	0a (Normal) 1a (Local only) 2a (Transmit only)	0a
<key>	key number, 1k-8k	1k
<label length>	0d-16d	0d
<string length>	0L-80L -1 erases the field definition	1L
<label>	a string of exactly <label length> characters	
<string>	a string of exactly <string length> characters	

If a parameter is omitted, the default value is used. If a value of zero is used for the **<label length>** or **<string length>**, then the current contents of the label or string are left unchanged.

The **<label>**, if its length is not zero, must precede the **<string>**.

The following example shows two equivalent escape sequences that assign the string **HELLO MGR.DEV** to [F1]. The attribute is normal (N), the label reads **LOG ON**. The hyphens represent spaces, and are required so that the function begins on the function definition line, not in the label.

Ec&f0a1k16d13L---LOG-----ON---HELLO MGR.DEV
Ec&f1k0a13l16D---LOG-----ON---HELLO MGR.DEV

Displaying the Menu

The host computer can display and remove the user-key definition menu by sending the following escape sequences:

Ecj   Displays menu.

Eck   Removes menu.

Reading the Menu

Once the user-key menu has been displayed, the host system can read it by sending the following escape sequences in succession:

$^{E}{}_{C}$&k1B	Turns on block mode.
$^{E}{}_{C}$&s1D	Turns on page mode.
$^{E}{}_{C}$H	Homes the cursor.
$^{E}{}_{C}$d	Requests terminal to send block.

Reflection responds by sending one $^{E}{}_{C}$&f... escape sequence for each of the eight user keys. This capability is used by some host programs to save the current user-key definitions before changing them, so that they can be reloaded with their original definitions when the program terminates.

User Keys in Color Graphics

Reflection 7 allows you to define color pairs and video enhancements for user keys. Each user-key label can have its own pair of colors as well as video enhancement combinations (blinking, underline, and inverse video). The rest of the parameters are identical to those discussed above in the *User-Key Escape Sequence Parameters* table.

NOTE: If you are only specifying a color pair and/or video enhancements, you must specify a 0 length parameter with 0L, as it normally defaults to one.

$^{E}{}_{C}$&f<color pair>c<video enh>v0L

User keys are displayed as color pair 7 under the default. If half-bright is selected as an enhancement, the color pair 3 is used. If a color is also selected, the half-bright enhancement is ignored.

Because the **<key>** parameter resets the color pairs and video enhancements to their defaults, you must place these values after **<key>**. The following sequence includes all possible parameters and a suggested order. Any combination of parts may be selected, but the final parameter must be capitalized to complete the sequence. The three parts shown here should appear on a single line.

^{E}C&f<attribute>a<key>k<color pair>c
<label length>d<string length>l
<video enh>V<label><string>

Table 1.13
Color Parameters for User Keys

Variable	Parameters	Default
<color pairs>	**0-7c**	**7**
<video enh>	**0-15v**	**0**

#	Value	#	Half-Bright and ...
0		*9*	*Blinking*
1	*Blinking*	*10*	*Inverse Video*
2	*Inverse Video*	*11*	*Blinking and Inverse Video*
3	*Blinking and Inverse Video*	*12*	*Underline*
4	*Underline*	*13*	*Blinking and Underline*
5	*Blinking and Underline*	*14*	*Inverse Video*
6	*Inverse Video and Underline*	*15*	*All Values*
7	*Blinking, Inverse, and Underline*		
8	*Half-Bright*		

Programming Graphics Functions

Contents

Reflection's HP Graphics Features

All graphics operations consist of specifying pixel locations on a screen to create specific images. These locations may be used to express the end points of a line, the corners of a rectangular area to be filled in by a pattern, or the location for the graphics cursor.

Graphics operations allow you (or a graphics program running on the host) to specify these points by means of predefined codes. Escape sequences are the codes that drive graphics operations. They may be entered from the PC keyboard, sent from the host computer, or embedded in a command file. Sequences may also be loaded into user keys. The following pages outline the escape sequences in detail.

Graphics Control Functions

Reflection 3 and 7's graphic control functions include the following:

- Display and memory control from the keyboard and by escape sequences issued from the host computer
- Graphics cursor control
- Graphics text control
- Several drawing modes
- User-definable and predefined line and area patterns
- Absolute and relocatable area fill
- ASCII and binary plotting control
- A *compatability mode,* which emulates a Tektronix 4010 graphics terminal
- A facility that allows you to print a *snapshot* of a graphics image on selected printers

- Rubberband lines which allow you to preview lines before making them permanent
- Polygon area fill
- Support for Microsoft's mouse (R7 only)

Reflection 7 also provides a variety of color support options; see page TR-147.

Graphics Escape Sequences

Graphics functions are controlled with escape sequences which begin with Ec*. The character following the asterisk, always lower case, specifies the type of sequence. For example, Ec*p begins a plotting sequence.

Table 2.1
Graphics Sequence Types

Ec*d	Display Control
Ec*l	Labeling
Ec*m	Drawing Mode
Ec*p	Line Plotting
Ec*s	Graphics Status
Ec*t	Compatibility Mode

Within escape sequences, control codes such as CR and LF are generally ignored. Ec cancels any previous sequence which was not terminated correctly and begins a new escape sequence.

Commands

Graphics commands are composed of both upper and lower case letters (A-Z and the ^ character). Upper and lower case letters execute the same function; an upper case letter terminates an escape sequence. To simplify input, you can give several commands in one sequence by using lower case letters until the ending command. Sequences can be any length and it is usually convenient to string several commands together, with their parameters. For example:

```
Ec*m4a7b100,100,100,200J
```

The a, b, and J are all commands under the Ec*m sequence (drawing mode). The J ends the sequence. Parameter values, all numbers here, come before the commands they modify. Omitted commands default to their currently active value. X and Y coordinates are separated by commas or spaces; 100,100 100,200.

The following sequence plots a line between the given coordinates by selecting a plotting sequence (^{E}c*p), lifting the pen and moving it to 100,300, and selecting ASCII absolute format to specify the coordinates. The capital Z ends the sequence.

^{E}c*paf100,300 300,300Z

NOTE: Any capital letter will terminate an escape sequence. Lower case characters which are not commands are ignored.

Parameters

Parameters come from columns 2 and 3 of the ASCII table (SPACE through ?). They are typically numbers which indicate coordinates or select a particular setting. Both ASCII and binary formats are possible; they are discussed later in this section. Note that in binary formats spaces are treated as data.

Parameters precede the command letter; for example:

^{E}c*m2a7b256,195J

- ^{E}c*m defines the type of sequence being used.
- 2a selects drawing mode 2.
- 7b selects the line type associated with number 7.
- 256,195J sets the relocatable origin and ends the sequence.

Each escape sequence type is discussed in detail below. Tables outline the possible commands for each sequence, as well as possible parameters which can be used to further define the sequence.

Restoring Graphics Defaults

^{E}c*mR restores all graphics defaults; see page TR-118. When programming escape sequences, you may want to ensure initial default conditions for all values. All of the defaults listed on page TR-118 can be reset at any time with the above sequence.

Restoring Selected Graphics Defaults

If a 1 is used as the <default flag> in the following sequence:

$^{E}{}_{C}$*m<default flag>R

a subset of default values is set. Otherwise all defaults are set. These are marked by a double asterisk below.

Table 2.2
Graphics Defaults

Parameter	*Default*
***Pen condition*	*Down*
***Line type*	*1 (solid)*
***Drawing mode*	*2 (SET)*
***User-defined line pattern*	*255,1 (solid line)*
***Area fill type*	*2 (user-defined pattern)*
***User-defined area fill*	*255, 255,. . ., (Solid)*
***User-defined dither pattern*	*255, 255,. . ., (Solid)*
***Current dither pattern*	*255, 255,. . ., (Solid)*
***Background pen*	*0 (black)*
***Primary pen*	*7 (white)*
***Secondary pen*	*0 (black)*
***Boundary pen*	*off*
***Graphics text*	*off*
***Text color*	*primary pen*
***Text direction*	*1*
***Text origin*	*1 (left,bottom)*
***Text size*	*1*
***Text slant*	*Off*
Relocatable origin	*0,0*
Alpha video	*On*
Alpha cursor	*On*
*Graphics video**	*On*
Graphics cursor	*Off*
Grapics cursor address	*0,0*
Rubberband line	*Off*
Compatibility mode	*Off*
Page full straps	*0 (out)*
GIN strap	*0 (CR only)*

* Reflection's default for the graphics display is *OFF*. This can be changed on the Terminal Configuration, Page 2 menu.

To change any of these defaults, use the graphics escape sequences which control the particular setting to be changed. To obtain information regarding the current status of various settings, use the graphics status request sequence outlined below.

Graphics Hard Reset

Initial values for Reflection match the defaults for the HP 2623A or 2627A terminals, except that the graphics keypad is initially off.

EC*wR performs a graphics hard reset. This changes all graphics parameters to their default values. It also clears raster memory buffer and places the primary drawing pen at 0,0. It is as if a hard reset were performed for the graphics only. This reset can only be performed with an escape sequence.

Graphics Status

To read graphics status, use EC*s<x>^ where x is one of the following:

Table 2.3
Graphics Status

x	*STATUS*
1	*Terminal ID*
2	*Pen position*
3	*Graphics cursor position*
4	*Read cursor position and wait for key*
5	*Display size*
6	*Graphics capabilities*
7	*Graphics text status*
8	*Read zoom status*
9	*Relocatable origin*
10	*Reset status*
11	*Area shading*
12	*Dynamics*

Graphics Display and Cursor Control

The graphics display on the IBM PC depends on the graphics adapter installed. The resolution of each adapter varies; see below for specific values.

The structure of the graphics display is different from that of the alpha display. The alpha display is composed of 80 column positions and 25 rows. The HP 2623A and the HP 2627A terminals' graphics display is composed of 512 columns and 390 rows of pixels.

In contrast, your graphics adapter probably uses a different number of pixel rows and columns. Reflection accommodates this difference by calculating the pixel on your adapter which is closest to that on the HP 2623A or 2627A. The PC you are using can accommodate 640 columns of pixels, but only 200 rows, assuming you have an IBM color graphics display adapter. The *Enhanced Graphics Adapter*, or EGA, may give the screen 640 x 350 pixels, depending on the kind of monitor you have. Hercules provides 720 x 348.

Each pixel can be represented by a set of two numeric values, which specify the row/column position of that pixel on the screen. Through the keyboard or a series of escape sequences issued by a host program, you can send the PC instructions to give a value of 1 (one) to specified pixels on the screen, making these pixels visible and forming the image you want.

It is not necessary to specify each individual pixel to create an image. Character keys and escape sequences do this for you, as you will see later in this chapter. The graphics display is initially *OFF* when Reflection is loaded.

Controlling the Display

The graphics display can be turned ON or OFF. Terminal Configuration, Page 2 allows you to configure whether the graphics display is initially on or off.

To turn the graphics display ON and OFF with escape sequences use the following:

Graphics display ON: ^{E}C*dC
Graphics display OFF: ^{E}C*dD

NOTE: PCs have two modes, graphics and alpha. Reflection initiates graphics mode whenever the graphics display is turned on. (If the **Initial graphics display** field is set to *ON*, graphics mode will be the initial state.) Graphics mode slows the scrolling speed of the alpha display.

The following escape sequence controls the graphics display or cursor:

Ec*d<z>

where <z> is replaced by one of the following:

Table 2.4
Display Control Functions

A	Clears graphics memory.
B	Sets graphics memory.
C	Turns on graphics display.
D	Turns off graphics display.
E	Turns on alphanumeric display.
F	Turns off alphanumeric display.
K	Turns on graphics cursor.
L	Turns off graphics cursor.
Q	Turns on alphanumeric cursor.
R	Turns off alphanumeric cursor.
S	Turns on graphic text mode.
T	Turns off graphic text mode.
Z	No operation is performed.
<X,Y>O	Moves graphics cursor to horizontal position **X** and vertical position **Y** (relative to the bottom left corner).
<X,Y>P	Moves graphics cursor to horizontal position **X** and vertical position **Y** (relative to its present location).

Graphics Keypad Control

The graphics keypad can be turned on or off by the following escape sequences:

Ec&k0O	Turn graphics keypad OFF.
Ec&k1O	Turn graphics keypad ON.

Graphics Cursor Control

A separate cursor is used to locate points in the graphics display. The graphics cursor is used to input position data or to interact with graphics application software. It is *OFF* when Reflection is first loaded. To turn the cursor on or off programmatically, use the following:

Graphics cursor on: Ec*dK
Graphics cursor off: Ec*dL (default)

The cursor is initially at position 0,0 after Reflection is loaded or after a full reset. The cursor can be positioned even while it is off. You can position the cursor either relative to its current position or with absolute coordinates. To position the cursor use the following sequences:

Absolute coordinates: Ec*d<X,Y>O
Relative coordinates: Ec*d<X,Y>P

For example, Ec*d10,10O places the cursor at the screen coordinates 10,10. The capital O ends the sequence. Ec*d10,10P places the cursor plus 10x and 10y in relation to its current position.

Mouse Control of the Graphics Cursor

Reflection 7 supports the Microsoft mouse for movement of the graphics cursor. Most mice and graphics tablets on the market have drivers that are compatible with the Microsoft mouse interface.

Normally, the left button acts as [Enter], and the right button acts as [Return]. In response to the Esc*s4^, *read graphics cursor position with wait* sequence, the left button returns a code of 128.

Cursor Status with Wait

The graphics keypad is automatically turned on when the Ec*s4^ or Ec*s33^ escape sequence is received. These escape sequences request the user to position the graphics cursor and then press a key or a mouse button, at which time the terminal transmits the cursor position. In the Tektronix emulation, the **GIN** mode acts in a similar fashion. At all other times, the graphics keypad must be manually activated before the keypad arrow keys will move the graphics cursor.

Graphics Memory Control

The graphics display can be turned on or off without affecting the data in the graphics memory. Set the display to all ones by setting graphics memory (a white screen) or all zeros by clearing memory (black screen).

Clear graphics memory: ^{E}c*dA
Set graphics memory: ^{E}c*dB

NOTE: You can print the graphics memory with ^{E}c&p7sF. Also see page UM-190.

Graphics Drawing Modes

Drawing commands with various parameters can specify the following:

- Whether data is stored in graphics memory as 1s or 0s
- Line or area patterns for drawing vectors
- The relocatable origin position
- Graphics text settings

Graphics drawing escape sequences begin with ^{E}C*m followed by one or more commands and parameters.

Selecting Modes

You can draw vectors by setting, clearing, or complementing the data in the graphics memory. Normally, vectors are drawn as white lines on a dark screen (bits turned off). To draw black lines on a white screen, set the graphics memory (turn bits on), select a clear or complement line type, and draw dark vectors.

NOTE: Reflection 7 provides a wider variety of drawing modes. See page TR-154.

^{E}C*m<x>A Selects the drawing mode.

Table 2.5
Drawing Modes - Reflection 3

Mode	*Name*	*Pattern*	*Effect*
0	*NOP*	*0,1*	*NOP*
1	*CLEAR*	*0*	*NOP*
		1	*Sets bit to 0*
2	*SET*	*0*	*NOP*
		1	*Sets bit to 1*
3	*COMP*	*0*	*NOP*
		1	*Complements bit*
4	*JAM*	*0*	*Sets bit to 0*
		1	*Sets bit to 1*

Clear mode

When clear mode is selected, area fill or solid lines are created by turning bits in the graphics display off. If the display is already off, the lines will be invisible. Unless the line pattern is solid, all bits in the pattern are turned off, and bits which are not part of the pattern are left as they were; i.e., dashed lines turn the bits which create the dashes off and leave the bits which normally create the spaces between dashes as they are, either on or off. No bits are actively turned on in clear mode.

Set mode

Set mode turns on all the bits in the selected line or area pattern. A dashed line turns on bits to create dashes and leaves bits between dashes as they are. If bits between dashes are already on, they are left on. No bits are actively turned off. Lines drawn in set mode are invisible when the graphics display is in inverse video (turned on).

Complement mode

Complement mode turns bits both on and off, depending on their current state. Only the bits which are on in the line pattern are affected, however. Dashed lines will turn bits on or off during the solid pattern, but the spaces between dashes are left unaffected, whether the bits there are on or off.

Jam mode

Jam mode affects both the on and off bits in the pattern. Bits are turned off or on depending on the pattern, regardless of their initial state. Unlike the other modes, all of the bits in the pattern, whether on or off, affect the display; i.e., if the display bits are initially off, the dashes are turned on and the space between dashes is turned off.

NOTE: Complement mode is useful for selective erasing of lines. To prevent gaps which occur when the erased line intersects with other lines, draw the line in complement mode initially. Then redraw the line in complement mode to erase it. This preserves the original graphics display. Use this technique to draw and erase temporary figures.

Plotting Lines

Lines can be plotted in relation to the graphics cursor position or a *pen* position. Depending on the escape sequence given, the first X,Y position is located at the cursor or the absolute coordinates of 0,0 (unless the *pen* has been moved). To plot a vector, issue the plot command ^{E}c*p followed by coordinates. For long sequences, when plotting several lines, use Z (no-op) to ensure synchronization.

Initially, the pen and cursor are both in the same position, 0,0. The pen is in the up position to begin with, so when it is moved to new coordinates no line is drawn. After the first move, the pen is in the down position unless otherwise directed. The new point becomes the current pen position. When new coordinates are given, a line is drawn from the initial position to the indicated point. You can specify an absolute position on the screen, or a relative one, by the parameters given in the plotting sequence.

You can position the pen in relation to absolute coordinates on the screen or to relative coordinates in relation to its current position. Coordinates can be given in either ASCII or binary.

Rubberband Line

Rubberband line mode allows you to temporarily establish a line and move it to various places in graphics memory before plotting it to a particular location.

1. The current pen position is used as the origin point. Turn on rubberband line mode using ^{E}c*dM.
2. Move the rubberband line by moving the graphics cursor, either directly or through escape sequences.
3. Plot the line by establishing the current cursor location as a point. The escape sequence ^{E}c*pC makes the line between the original point and the current cursor position permanent.

ASCII Formats

ASCII formats can be of three types: absolute, incremental, and relocatable. In ASCII formats the numbers 0 to 9 are used to specify X and Y coordinates. The first number given is considered the X coordinate and the second the Y coordinate. All values must be whole integers separated by commas or spaces. Any numbers following a decimal are ignored as well as any extra spaces or commas.

ASCII Absolute

This is the default plotting format. Values from -16384 through 16383 may be given, but only values within the graphics display range will be visible on the screen.

^{E}c*pa15,15 65,15 65,70 15,70 15,15Z draws a rectangle with the bottom left corner at 15,15 and the top right corner at 65,70. The **a** raises the pen and, since no format is defined, ASCII absolute is assumed. The pen is lowered after the first coordinate pair is received. The coordinates are determined from a 0,0 starting point.

```
--75
--70      |--------------------------------|
--65      |                                |
--60      |                                |
--55      |                                |
--50      |                                |
--45      |                                |
--40      |                                |
--35      |                                |
--30      |                                |
--25      |                                |
--20      |                                |
--15      |--------------------------------|
--10
--05
--00
Y    10 15 20 25 30 35 40 45 50 55 60 65 X
```

ASCII Incremental

All values are added to the current pen position. This allows you to use the same escape sequence to draw a figure anywhere on the screen merely by starting at a different pen position.

^{E}C*pg10,0 0,50 -50,0 0,-50

The first set of coordinates moves the pen 10 units along the X-axis and 0 units along the Y-axis. The second moves the pen 0 units further on the X-axis and 50 units on the Y-axis, etc. All vectors are relative to the current position rather than absolute positions on the screen.

ASCII Relocatable

When relocatable coordinates are being used, the X and Y coordinates of the *relocatable origin* are added to all input coordinates. This allows you to draw the same figure with the same escape sequences by just changing the relocatable origin.

The box defined by 25,50 50,50 50,100 25,100 25,50 can be repeated at 75,100 100,100 100,150 75,150 75,100 by setting a relocatable origin at 50,50, and sending the same drawing sequence again.

```
--150                 ___
--125                |   |
--100         ___    |___|
-- 75        |   |
-- 50        |___|
-- 25
-- 00
     25 50 75 100 125
```

Binary Formats

Binary formats can be of four types: absolute, short incremental, incremental or relocatable. In binary formats the bit patterns of ASCII characters are used to define the X and Y values. Depending on the format type, 4 to 6 bytes (characters) are required to provide each coordinate. Values can range from -16384 to 16383.

Binary Absolute

As in ASCII absolute, 0,0 is considered the initial origin. The given coordinates are plotted in relation to 0,0. Four bytes are required to define each point. Only points within the graphics display area are visible on the screen.

Table 2.6
Binary Absolute Format

BIT	*7*	*6*	*5*	*4*	*3*	*2*	*1*	
BYTE 1	*0*	*1*	*X9*	*X8*	*X7*	*X6*	*X5*	*HI X*
BYTE 2	*0*	*1*	*X4*	*X3*	*X2*	*X1*	*X0*	*LOW X*
BYTE 3	*0*	*1*	*X9*	*X8*	*X7*	*X6*	*X5*	*HI Y*
BYTE 4	*0*	*1*	*X4*	*X3*	*X2*	*X1*	*X0*	*LOW Y*

Programming Graphics

Table 2.7
Computation of Data Bytes

X = 0 = 00000	*00000*	*Y = 0 = 00000*	*00000*
HI X	*LOW X*	*HI Y*	*LOW Y*

BYTE 1 = 0100000 = SPACE HI X
BYTE 2 = 0100000 = SPACE LOW X
BYTE 3 = 0100000 = SPACE HI Y
BYTE 4 = 0100000 = SPACE LOW Y

X = 360 =01011	*01000*	*Y = 180 = 00101*	*10100*
HI X	*LOW X*	*HI Y*	*LOW Y*

BYTE 1 = 0101011 = + HI X
BYTE 2 = 0101000 = (LOW X
BYTE 3 = 0100101 = % HI Y
BYTE 4 = 0110100 = 4 LOW Y

Binary Short Incremental

This format allows you to alter the current pen position by incrementing the current X/Y coordinates. Under this format, only two bytes are required to give the change in position, but the range is only from -16 to +15 from the present coordinates. The five least significant bits are added to the current pen position to obtain the new end point. They are interpreted as a signed, two's complement number.

Table 2.8
Incremental (Short) Vector Bytes

0	1	2	3	4	5	6	7
	!	"	#	$	%	&	'
8	9	10	11	12	13	14	15
(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9
0	1	2	3	4	5	6	7
-8	-7	-6	-5	-4	-3	-2	-1
8	9	:	;	<	=	>	?

Binary Incremental

Using the incremental format allows you to specify a much larger range, but requires that you use six bytes of data. Delta X and delta Y can range from -16384 to +16383.

Table 2.9
Binary Incremental Format

BYTE1	*0*	*1*	*DX14*	*DX13*	*DX12*	*DX11*	*DX10*	*HI DELTA X*
BYTE2	*0*	*1*	*DX9*	*DX8*	*DX7*	*DX6*	*DX5*	*MID DELTA*
BYTE3	*0*	*1*	*DX4*	*DX3*	*DX2*	*DX1*	*DX0*	*LOW DELTA*
BYTE4	*0*	*1*	*DX14*	*DX13*	*DX12*	*DX11*	*DX10*	*HI DELTA Y*
BYTE5	*0*	*1*	*DX9*	*DX8*	*DX7*	*DX6*	*DX5*	*MID DELTA*
BYTE6	*0*	*1*	*DX4*	*DX3*	*DX2*	*DX1*	*DX0*	*LOW DELTA*

Table 2.10
Characters Used in Binary Data Formats

ASCII Char	*BIT Pattern*	*ASCII Char*	*BIT Pattern*
SP	*01 0 0000*	*0*	*01 1 0000*
!	*01 0 0001*	*1*	*01 1 0001*
"	*01 0 0010*	*2*	*01 1 0010*
#	*01 0 0011*	*3*	*01 1 0011*
$	*01 0 0100*	*4*	*01 1 0100*
%	*01 0 0101*	*5*	*01 1 0101*
&	*01 0 0110*	*6*	*01 1 0110*
'	*01 0 0111*	*7*	*01 1 0111*
(	*01 0 1000*	*8*	*01 1 1000*
)	*01 0 1001*	*9*	*01 1 1001*
***	*01 0 1010*	*:*	*01 1 1010*
+	*01 0 1011*	*;*	*01 1 1011*
,	*01 0 1100*	*<*	*01 1 1100*
-	*01 0 1101*	*=*	*01 1 1101*
.	*01 0 1110*	*>*	*01 1 1110*
/	*01 0 1111*	*?*	*01 1 1111*

Binary Relocatable

Under the relocatable format, the origin can be set to any position in the -16384 to +16383 range. The X and Y coordinates are then added to this origin. In this way the same figure can be drawn at different locations using the same data.

Table 2.11
Binary Relocatable Format

BIT	*7*	*6*	*5*	*4*	*3*	*2*	*1*	
BYTE 1	*0*	*1*	*X14*	*X13*	*X12*	*X11*	*X10*	*HI X*
BYTE 3	*0*	*1*	*X9*	*X8*	*X7*	*X6*	*X5*	*MID X*
BYTE 2	*0*	*1*	*X4*	*X3*	*X2*	*X1*	*X0*	*LOW X*
BYTE 4	*0*	*1*	*X14*	*X13*	*X12*	*X11*	*X10*	*HI Y*
BYTE 5	*0*	*1*	*X9*	*X8*	*X7*	*X6*	*X5*	*MID Y*
BYTE 6	*0*	*1*	*X4*	*X3*	*X2*	*X1*	*X0*	*LOW Y*

BYTE 1 = 01	*11111 = ?HI X*
BYTE 2 = 01	*01101 = -LOW X*
BYTE 3 = 01	*01000 = (HI Y*
BYTE 4 = 01	*00000 = SPHI X*
BYTE 5 = 01	*00110 = &LOW X*
BYTE 6 = 01	*01000 = (HI Y*

Drawing Patterns

Select the type of line to be used to draw vectors or to fill rectangular areas. Nine predefined line patterns can be selected as well as a user-defined line or area pattern.

Eleven line types are available. Once the pattern has been selected, all vectors are drawn with that type.

Table 2.12
Line Types

Ec*m<x>B   Selects a line pattern.

x	*Pattern*	
1	*Solid line (default)*	
2	*User-defined line pattern*	
3	*Current area pattern*	
4	*Line #1*	
5	*Line #2*	
6	*Line #3*	
7	*Line #4*	
8	*Line #5*	
9	*Line #6*	
10	*Line #7*	
11	*Point plot*	*. (point plot) .*

Point plot causes a single point to be plotted at the coordinates specified. This is useful for *scattergram* graphs.

If the area fill pattern is selected (3), the line patterns used are selected from the eight lines making up the pattern. The graphics display is divided into groups of eight rows and eight columns. Horizontal and vertical lines are drawn using the appropriate row or column from the area pattern. Diagonal lines are always drawn using a solid vector.

Defining Line Patterns

Besides the pre-defined line patterns, you can define a pattern using the bit pattern of any byte from 0 to 255. For example, 170 has the following pattern:

10101010 or ■_■_■_■_

Besides the basic pattern, you can define a scale which allows you to stretch the pattern. Each bit in the pattern is repeated X times, where X is the scaling factor. Scales range from 1 to 16. For example, 170, the example used above, could be given a scale of 2 to produce the following pattern:

To draw with the user-defined line type, select it as the line type before plotting any lines. Vectors will be drawn with dots according to the pattern. A series of horizontal or vertical lines using this line type can be used to create more complicated patterns than those possible using the area fill pattern.

EC*m<x><y>C

Defines an 8-bit segment of line pattern and a scale according to the following:

- **x** is a number from 0 to 255 which, when converted to its binary form, illustrates the segment of line pattern.
- **y** is a number from 1 to 16 which indicates the number of times each dot of the line pattern should be repeated, i.e., the scale.

Plotting Lines

Table 2.13
Plotting Functions

EC*p<x>

Performs the action indicated by **x**.

- **A** Lifts the pen.
- **B** Lowers the pen.
- **C** Uses the graphics cursor position as a new point.
- **D** Draws a point at the current pen position and lifts the pen.
- **E** Sets the relocatable origin to the current pen position.
- **F** Data is ASCII absolute.
- **G** Data is ASCII incremental.
- **H** Data is ASCII relocatable.
- **I** Data is binary absolute.
- **J** Data is binary short incremental.

K	Data is binary incremental.
L	Data is binary relocatable.
Z	No operation is performed.

Graphics Text Escape Sequences

Typically, text is sent to the alphanumeric memory where it can be easily scrolled or altered without affecting graphics memory. In order to send text to the graphics memory, you must turn graphics text mode on. Use the following sequences to turn graphics text mode on and off:

Begin graphics text mode: ^{E}c*dS
End graphics text mode: ^{E}c*dT

The graphics cursor marks where the next character will appear when text is entered in graphics memory. Other sequences can be sent to scale, rotate, or otherwise alter text as it is entered in memory.

Selecting Graphics Text Styles

Four sets of parameters determine the appearance of graphics text:

Size
Graphics text may be any of eight sizes. Graphics text size is determined by the escape sequence, ^{E}c*m<n>M, where <n> is a value, 1 through 8.

Slant
Graphics text may be upright or slanted to the right. The default is upright. To turn on slanted characters, type the escape sequence, ^{E}c*mO. To turn it off, type ^{E}c*mP.

Orientation
Graphics text can run from left to right, the normal orientation, or it can be upside down, running from right to left. It can also run from bottom to top or top to bottom. Orientation is set by the escape sequence, ^{E}c*m<n>N, where <n> is a value, 1 through 4.

Justification
Characters can be justified in relation to the current position of either the pen or the graphics cursor. The position is used

as a baseline. Left bottom justification (1) is the default. Justification is changed with Ec*m<origin>Q, where <origin> is replaced with a value between 0 and 9.

The following table summarizes the graphics text escape sequences.

Table 2.14
Graphics Text Escape Sequences

Parameter	*Sequence*	*Action*
Size:	Ec*m1M	Smallest characters
	Ec*m2M	
	Ec*m3M	
	Ec*m4M	
	Ec*m5M	
	Ec*m6M	
	Ec*m7M	
	Ec*m8M	The largest characters, about one and one-half times the height of the graphics cursor
Slant:	Ec*mO	Slanted characters
	Ec*mP	Upright characters
Orientation:	Ec*m1N	Normal orientation, default
	Ec*m2N	Rotated 90 degrees counter-clockwise, running from the bottom to the top of the screen
	Ec*m3N	Rotated 180 degrees counterclockwise, upside down
	Ec*m4N	Rotated 270 degrees counterclockwise, from top of the screen to the bottom
Justification:	Ec*m1Q	Text justified bottom left
	Ec*m2Q	Text justified middle left
	Ec*m3Q	Text justified top left
	Ec*m4Q	Text justified bottom center
	Ec*m5Q	Text justified middle center

EC*m6Q	Text justified top center
EC*m7Q	Text justified bottom right
EC*m8Q	Text justified middle right
EC*m9Q	Text justified top right
EC*m0Q	Text justified cursor position

Labels

A single line of graphics text can be sent to the graphics display memory with the following sequence. The current text size, angle, slant, justification and color are all used.

EC*l<text string>CRLF

For example, EC*lSales Data for 1986CRLF would send this label to the graphics memory at the current pen position.

Graphics Text Colors

In Reflection 7 use the following escape sequence to select a color for the graphics text. If no color is selected, the primary pen color is used:

EC*n<pen#>X

The color selected is used for all subsequent text and labels until it is changed by another escape sequence or the defaults are set.

Filling Rectangle and Polygon Areas

Both rectangle and polygon areas can be filled, although the sequences vary. See page TR-147 for more information on filling using color graphics.

Filling Rectangles

To fill a rectangular area with a pattern, give the lower left and upper right coordinates of the rectangle. Use either an absolute or relocatable format when giving coordinates.

Area Fill Absolute

Absolute area filling uses absolute coordinates for the area. Use the following sequence:

Ec*m<XLL,YLL> <XUR,YUR>E

where <XLL,YLL> and <XUR,YUR> are the absolute coordinates of the lower left and upper right corners.

Area Fill Relocatable

Relocatable area filling uses area coordinates in ASCII format. Use the following sequence:

Ec*m<XLL,YLL> <XUR,YUR>F

where <XLL,YLL> and <XUR,YUR> are the relocatable coordinates of the lower left and upper right corners.

Filling Polygonal Areas

Any polygon with up to 148 sides can be filled with the current area fill pattern. Polygon filling can be affected by any of the following if they are active:

- Current drawing mode
- Current area pattern
- Current boundary pen (Reflection 7 only)

- Current pen selections (Reflection 7 only)

Use the following steps:

1. Begin the polygon area fill with ^{E}c*ps.

   All subsequent coordinates are read as vertices of the polygon.

2. Specify the vertices of the polygon using the sequence ^{E}c*pas <X,Y X,Y X,Y ...>T

   If the letter **a** occurs within the string of coordinates, the current polygon is closed, the pen is lifted, and a new polygon is begun.

3. Any capital letter or the close polygon area fill command (t) closes off the polygon and fills the area.

See page TR-147 for more information about color and polygon filling, especially *dither patterns*.

Creating Area Patterns

Area patterns are created by defining an 8 x 8 pattern. The bit pattern of each parameter given in the definition is used as the pattern for one of the 8 rows.

The area pattern is typically used to fill rectangular areas. It can also be used as the line drawing pattern. If you are drawing horizontal or vertical lines, the corresponding row or column of the area fill pattern is used as the line pattern. Diagonal vectors are always drawn with a solid line. You can also create irregular shapes by selecting the area pattern as the line type and using successive lines.

Each of the 8 parameters in the area pattern can be different. A single row of the design is defined by each.

To define an area pattern, use the following sequence:

^{E}c*m<row 0> <row 1>...<row 7>D

Row 7 is the top of the area and row 0 is the bottom of the area. Enter a space or comma between each of the eight numbers. The following area pattern is created by the sequence:

EC*m170,181,170,181,170,181,170,181d

■_■■_■_■
■■_■_■■_
■_■■_■_■
■■_■_■■_
■_■■_■_■
■■_■_■■_
■_■■_■_■
■■_■_■■_

Relocatable Origins

Relocatable origins can be used for either of the following:

- Displaying the same figure at various locations
- Displaying different portions of a figure too large to fit on the PC's display

The figure data is left unaltered; the terminal adjusts where or what portion of the figure is displayed.

Relocatable origin absolute

The following escape sequence defines absolute coordinates as the new origin:

EC*m<X,Y>J

where X and Y are ASCII numbers between -16348 and 16383.

Relocatable origin at pen position

The following escape sequence defines the current pen position as the new origin:

EC*mK

Relocatable origin at graphics cursor

The following escape sequence defines the graphics cursor position as the new origin:

EC*mL

Setting Compatibility Mode

Reflection allows you to use graphics programs intended for Tektronix 4010 terminals. *Compatibility* mode presents either part (one-quarter) or all of a graphics image created for terminals with 1024 x 780 displays. To configure compatibility mode, set the **Tek compat** field of the Terminal Configuration, Page 2 menu to either *SCALED* or *UNSCALED*. The default is *OFF*.

Scaled mode scales the larger image down so that the entire image is displayed on the PC. Unscaled mode presents as much of the image as possible without altering the image size. By changing the origin in unscaled mode, you can view all of the image in four parts.

The following sequences set these modes programmatically:

^{E}C&s1p0Q - Scaled compatibility mode on
^{E}C&s0p1Q - Unscaled compatibility mode on
^{E}C&s0p0Q - Compatibility mode off (default)

P and Q settings in the above sequences determine the terminal's mode when initialized or after a hard reset.

Scaled Mode

In scaled mode the X and Y coordinates are divided by 2. The program maps the 1024 x 780 display as 512 by 390. Programs intended for the 4010 run unchanged and display the entire image with some loss in resolution.

In scaled mode text is sent to graphics memory unless redirected. The text size, origin, and direction are set to their default values, and cannot be changed. The initial origin is the bottom left portion of the display.

Unscaled Mode

In unscaled mode it is necessary to change the relocatable origin in order to view another part of the image. The relocatable origin is subtracted from the graphic data coordinates. The default is 0,0 which displays the lower left part of the image. Text in unscaled mode is sent to alphanumeric memory, unless graphics text is turned on.

Setting Terminal Straps

Along with compatibility in image size, Reflection allows you to set other straps available on the 4010 through escape sequences.

Graphic input terminator

The following straps allow you to configure what character will act as a terminator for cursor address information. You can select CR, CR and EOT, or select no terminator. The following escape sequences set this strap:

Ec*t0A - Carriage return terminator (default)
Ec*t1A - Carriage return and EOT terminator
Ec*t2A - No terminator

Page full break

This strap determines whether a 200ms break signal is sent to the host on a full page condition.

Ec*t0B - No break
Ec*t1B - Send break signal to host on full page

Page full busy

This determines whether the keyboard locks after a full page of text is received. The **Graphics Clear** function unlocks the keyboard. In unscaled mode this strap is ignored. The following escape sequences set this strap:

Ec*t0C - Normal, no keyboard locking
Ec*t1C - Keyboard locked on full page

The following escape sequences turn on scaled compatibility mode, define no terminator for input, and specify that the keyboard should be locked and a break sent after a full page:

Ec&s1p0QEc*t2a1c1B

Graphics Data in Compatibility Mode

Each coordinate requires two bytes in compatibility mode. A 10-bit coordinate from (0-1023) is determined by the lower five bits of the two bytes. When sending data to the terminal, the Y coordinate must be sent first: <Upper Y>, <Lower Y>, <Upper X>, <Lower X>. Data is returned to the computer with the X value first: <Upper X>, <Lower X>, <Upper Y>, <Lower Y>.

Bits 6 and 7 of each byte are used to indicate whether the coordinate is upper or lower.

BIT	*7*	*6*	*Value*
	0	*1*	*Upper X or Y*
	1	*0*	*Lower X*
	1	*1*	*Lower Y*

It is not always necessary to send 4 bytes for each X,Y coordinate pair. However, the following must always be sent:

- Lower X byte
- Any changed byte
- Lower Y if Upper X changes

Chapter 2.5

Color Graphics

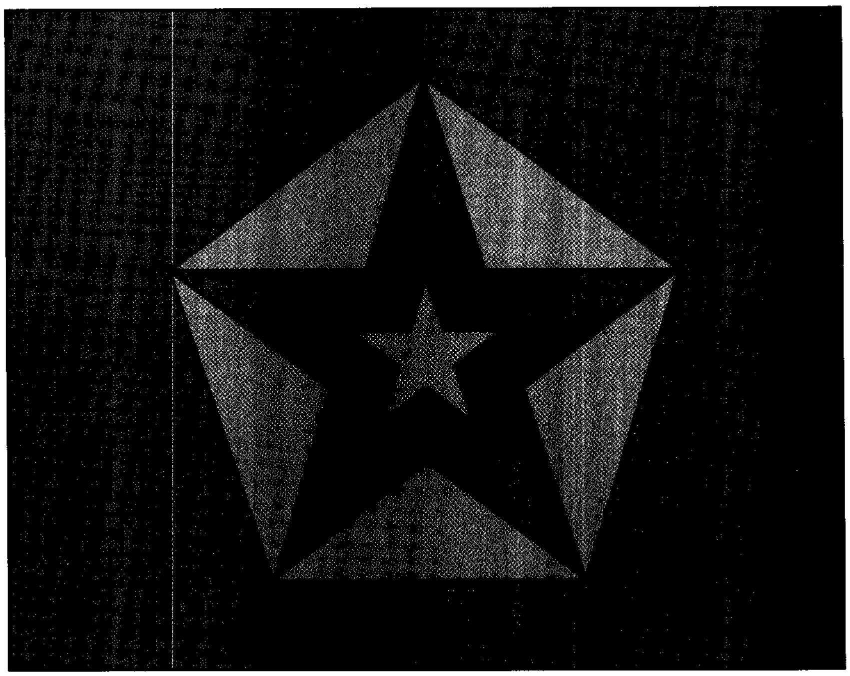


Figure 2.1 Color Graphics Sample

Color graphics capability provides a variety of ways to highlight graphics material. Reflection 7 allows you to emulate the HP 2627A terminal in all aspects on an IBM PC or compatible with an Enhanced Color Display and an Enhanced Graphics Adapter (EGA) card. Because color graphics are only possible when you are in Reflection 7, these features are dealt with separately here.

Basic Concepts

Graphics terminals produce color by overlaying three planes. These are either considered the colors red, green, and blue (RGB color method) or hue, saturation, and luminosity (HSL color method). For simplicity the RGB method will be used as the basis for discussion.

Because the graphics display is composed of three planes, what planes are on or off at any one time varies the color produced. The pixels within the plane can also be turned on and off. Varying the percentages of pixels that are on or off creates even more colors, called *dither patterns.*

For example, if all pixels of the blue plane are on, and the other planes are completely off, a blue display is presented.

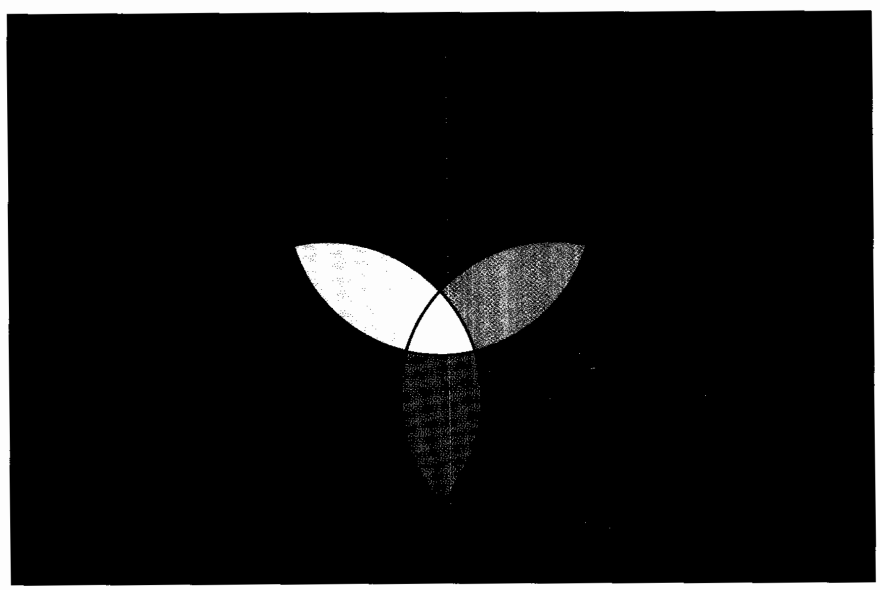


Figure 2.2 The Additive Color System

Pen Colors

When plotting graphics, *pens* (or colors) are selected through escape sequences. The colors selected to draw lines or fill in rectangle or polygon areas are considered pens. You can define primary and secondary pen colors as well as a boundary pen color. The primary pen color is used for most lines and area fills. The secondary pen is used in a drawing mode which needs a second color. The boundary pen is used to outline rectangles or polygons. You may also set a background color for the entire graphics display. Numbers associated with pen colors follow:

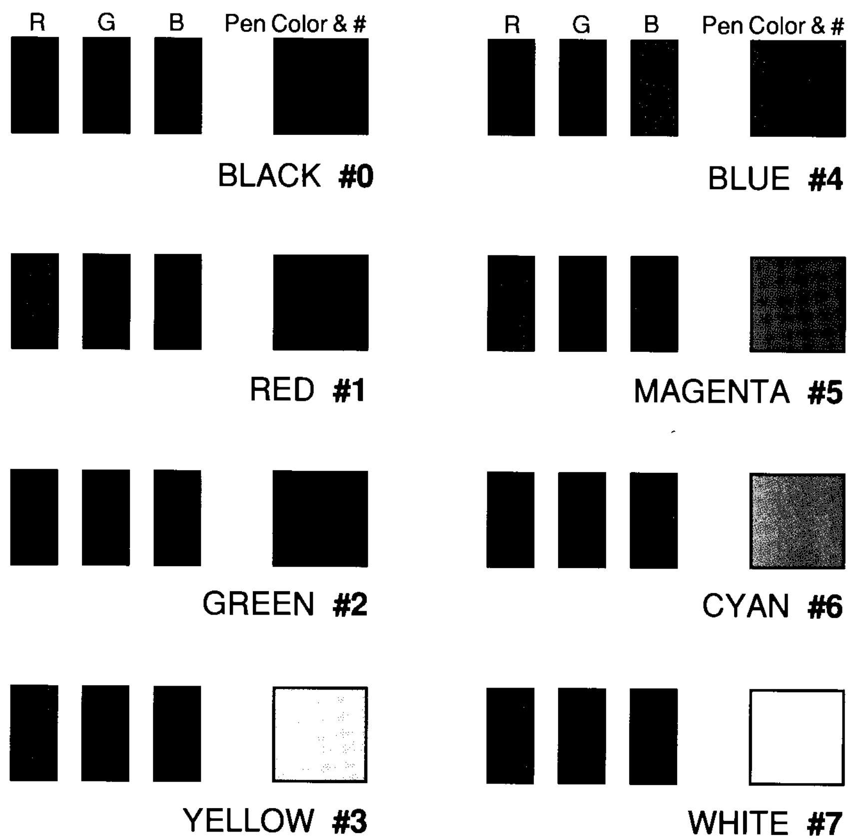


Figure 2.3 Basic Pen Colors

Selecting Colors for Plotting

Whenever *pen* is used, it can be replaced with the word color without any change in meaning.

- Select primary pen color with ^{E}c*m<pen#>X.
- Select secondary pen color with ^{E}c*m<pen#>Y. This is used for special line types and area fills according to the selected drawing mode.
- Select area boundary pen with ^{E}c*m<pen#>H.
- Select background color with ^{E}c*e<pen#>B. Black is the default. When a background pen is selected, the entire graphics memory is filled with the selected color.
- Select text color with ^{E}c*n<pen#>X.

Once selected, these pen color assignments are changed with another escape sequence or when the defaults are reset.

Selecting a Color for the Display

The entire display can be set or cleared to a color.

Clear graphics memory — ^{E}c*d<pen#>A
(0 is the default)
Set graphics memory — ^{E}c*d<pen#>B
(7 is the default)

To clear the graphics data and turn all pixels in graphics display to red, use ^{E}c*d1A. (When you use the keypad **Graph Clear** key, the screen is left black.)

Area Fill with Color

Areas can be filled with either an area fill pattern or a dither pattern. (Either method, not both, can be used at any one time.) Patterns use colors as described under drawing modes. Dithering has to do with the densities of the color planes regardless of the pen selections.

To select dithering as the area fill type, use the area pattern escape sequence with 0 as the selection as follows:

^{E}c*m0G

Whichever pattern has been selected as the current dither pattern is used.

As some resolution may be lost when using dither patterns, double width lines and text (or larger) are suggested for plotting over a dither pattern.

Selecting Dither Patterns

To select a pattern type, use the following sequence:

EC*m<dither pattern>W

where <dither pattern> is replaced by one of the following:

#1 User-defined dither pattern

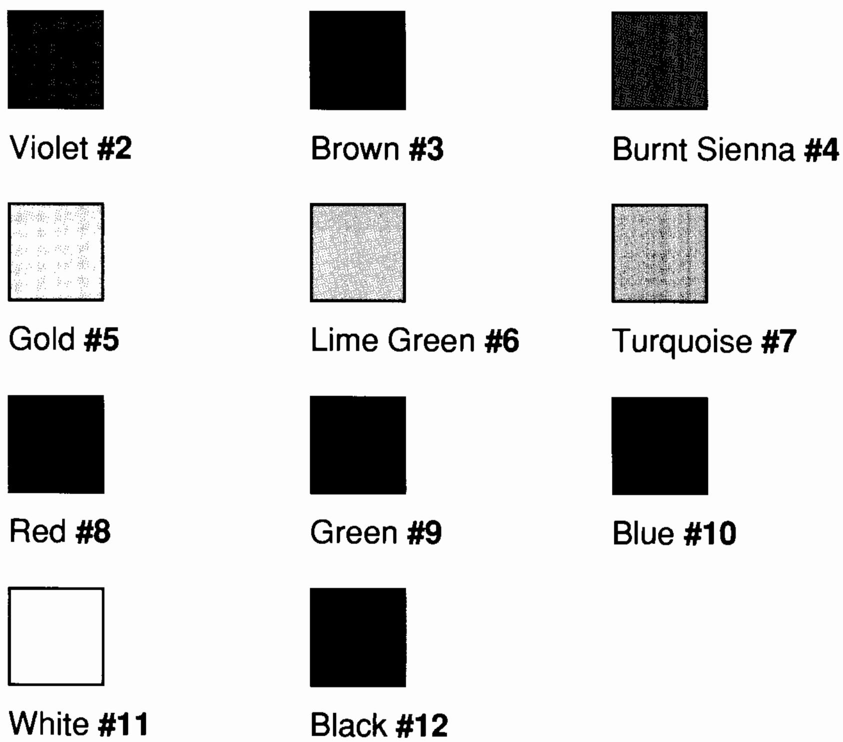


Figure 2.4 Predefined Dither Patterns

Creating a Dither Pattern

Additional colors can be created by mixing the amount of each plane used (or the amount of hue, saturation, and luminosity). These colors are called *dither patterns.* A defined number of pixels, in a specified order, are turned on and off in each plane to simulate a variety of colors.

1	13	4	16
9	5	12	8
3	15	2	14
11	7	10	6

How pixel percentage is mapped.

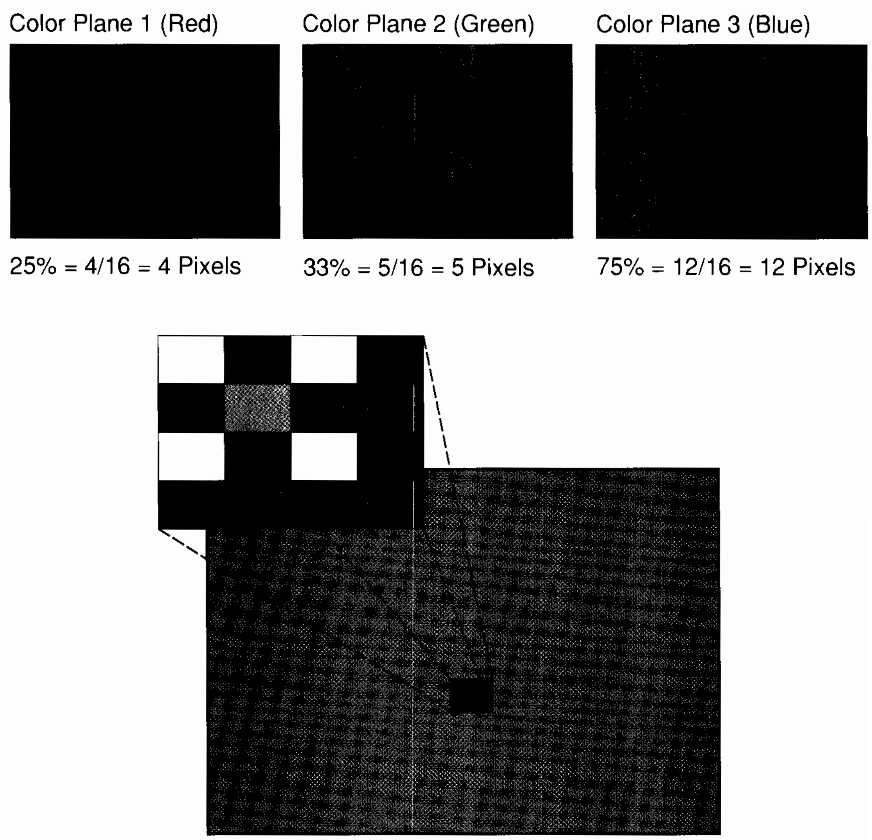


Figure 2.5 Dithering

To define your own dither pattern use the following steps:

1. First select the current dither pattern as the area fill type with Ec*m0G.
2. Select the user-defined dither pattern as the pattern to use with Ec*m1W.
3. Define the dither pattern by using decimal fractions which represent the density value for each color plane. The escape sequence is as follows:

   Ec*m<d1,d2,d3>V

   where d1-d3 define the percent of each plane to include. An example might be 25% red, 50% green, and 75% blue: the escape sequence would be Ec*m.25,.5,.75V. Note that these percentages represent how many pixels will be on in each plane.

Using an Area Boundary Pen

Both rectangle and polygon shapes can be outlined with a boundary pen color if desired. The boundary pen must be selected before drawing the area. Use Ec*m<pen#>H where one of the eight possible pen colors is selected.

Once the boundary pen is selected it is used to outline the area drawn. If you wish to *lift* the pen, use Ec*pU. The pen can then be lowered with Ec*pV at any point in the plotting.

To disable the boundary pen altogether use Ec*mH.

Drawing Modes

The drawing modes for the 2627A terminal are expansions of the 2623A drawing modes. The primary reason is to take advantage of the color options available.

The escape sequence Ec*m<mode>A selects the mode. These modes are explained in the following table. See the explanation and definitions of mode types on page TR-125 for more information. In most cases if a bit in the pattern is turned on, the primary color is used; if it is turned off, the pixel is not affected.

To select a drawing mode use Ec*m<mode>A.

Table 2.15
Drawing Modes - Reflection 7

Mode	*Name*	*Pattern*	*Effect*
0	*NOP*	*0,1*	*NOP*
1	*CLEAR1*	*0*	*NOP*
		1	*Pixel = Bpen*
2	*JAM1*	*0*	*NOP*
		1	*Pixel = Cpen*
3	*COMP1*	*0*	*NOP*
		1	*Pixel = NOT Pixel*
4	*JAM2*	*0*	*Pixel = Spen*
		1	*Pixel = Cpen*
5	*OR*	*0*	*NOP*
		1	*Pixel = Pixel OR Cpen*
6	*COMP2*	*0*	*NOP*
		1	*Pixel = Pixel XOR Cpen XOR Bpen*
7	*CLEAR2*	*0*	*NOP*
		1	*Pixel = Pixel AND NOT Cpen*

NOP No effect

Cpen Current pen (primary drawing color or text color) or dither pixel, in which case the color of the pixel for that dither pattern is the current color

Spen Secondary drawing pen (or color)

Bpen Background pen (or color)

Overstriking Graphics Text

In most drawing modes, overstriking a graphics text character produces an unreadable result, since only the foreground bits of the character cell are changed. To avoid this problem, use the JAM2 write mode. In JAM2 mode, all bits in the character cell are written. See page TR-139.

Underlining in Reflection 7

Normally with Reflection 7 in text mode, on-screen underlining is not supported. To enable underlining, start up Reflection 7 by typing **R7 /U** instead of the usual **R7**. With some EGA cards, this will cause *snow* to appear on the screen while scrolling.

Using Command Language

Contents

Reflection Command Mode

Reflection can be operated in command mode by means of an English-based command language. This is a language used to write commands, or instructions, to the computer to perform communications tasks. The commands in the language may be used singly, by entering them at the keyboard, or in combination, by joining a series of them in a command file.

A command file allows you to write down a list of operations – involving the sending and receiving of files, for example – which, when invoked, will be executed without further manual intervention. Walker Richer and Quinn provides several examples of useful command files in EXAMPLES.CMD. This file is provided on one of the support disks and can be typed into display memory or viewed with a PC editor.

Reflection's command language may be used to do the following:

- Access DOS-like commands such as DIR, CD, TYPE, PRINT, DEL, etc., without exiting Reflection
- Set configuration values
- Transmit and log data
- Send and receive files, with or without an error-checking protocol
- Synchronize transmissions in accordance with various conditions
- Design and manage local screens

This chapter explains how to use the command language and includes a number of useful examples. The keywords and command operators that constitute the command language are listed alphabetically and explained in detail in the *Command Language Keywords* appendix beginning on page TR-179.

Bringing Up the Command Line

The first thing you must do to operate in command mode is access the command line by pressing [F5] from the system keys, or [Alt]-[Y] elsewhere. The command line can only be accessed when Reflection is operating in terminal mode. When accessed, the following command prompt and a rectangular block of inverse video are displayed near the bottom of the screen.

Command (A):

Entering Commands

Any of Reflection's various commands may be entered and executed by typing the keyword and parameters in the command line, and then pressing [Return]. You can also type the name of a command file to execute all the commands contained in that file.

When you type a command on the command line, you need not begin with the first column; you can type the command anywhere on the command line. Make sure, however, that you enter only the command name and parameters. Enter spaces to separate each part of a command.

As with DOS commands, both upper case and lower case characters are acceptable. If the command is entered correctly, it will be executed automatically when you press [Return]. Reflection will then clear the command from the inverse video portion of the command line, but will continue to display the command prompt.

To enter a command:

1. Bring up the command line by pressing [F5] from the system keys or [Alt]-[Y].
2. Type one command keyword and any other required command elements (for example, HELP followed by a specific command).
3. Press [Return] when you've finished typing. Reflection reads the entire command line, even if the cursor is not at the end of the line.

Exiting Command Mode

When you have no further commands to enter, press [Esc] or [F10]. This returns the system to terminal mode. If the command line is blank, you may press [Return] to exit command mode.

Help in Command Mode

Should you find yourself at a loss in command mode, type **HELP** on the command line and press [Return]. A menu of help topics appears on the screen. Select the one you want, and then type **HELP <topic>** on the command line and press [Return]. An explanation of your selection appears on the screen.

Correcting Errors

If you enter an incorrect command on the command line, Reflection responds with an error message. Once an error message has appeared, there are two ways of proceeding.

- One way is to press [Return], thereby acknowledging the error. You may then retype the command correctly, and press [Return] a second time to execute it.
- A second way of responding is to press [F10] or [Esc], either of which gets you out of command mode and immediately returns you to terminal mode.

Editing the Command Line

Several keys may be used to enter, re-enter, or alter text within the command line window. Function keys [F1] through [F5] function just as when editing under DOS. When a command is typed on the command line and [Return] is pressed, the line is stored in a previous line buffer. [F3] allows you to copy the last line entered from the buffer back into the command line. Then you can edit it and re-enter it. [F5] erases the command line completely, and [F6] erases from the cursor position to the end of the line.

Editing Keystrokes

Each editing keystroke is discussed below. These may be used to alter the current logical position within the command line window, within the previous line buffer, or both. Reflection also has a user command line buffer that is implemented using function keys [F7] and [F8]. Note that [F9] may be used to cause the function keys to generate their user-key values. The keystroke after [F9] must be the user key you wish to use.

Command Language

Key	Function
F1	Displays the next character in the previous line buffer. The cursor advances one position on the command line and in the buffer as each character is displayed. Pressing F1 repeatedly, then, displays the previous line one character at a time.
F2	When followed by a character contained in the previously entered line, displays the previously entered line up to the first occurrence of the specified character. The cursor advances to the corresponding position on the command line (and in the previous line buffer).
F3	Displays the entire previous line or displays from the current position in the previous line buffer. The cursor advances to the corresponding position on the command line (and in the line buffer).
F4	When followed by a character from the previously entered line, moves the cursor position in the previous line buffer to the first occurrence of the specified character without displaying any portion of the line. Once this is done, pressing F3 will display the previously entered line from the specified character to the end of the line.
F5	Places the current contents of the command line into the previous line buffer, clears the command line window, and positions the cursor at the start of the command line.
F6	Erases any text from the cursor position to the end of the command line window. None of the line is stored.
F7	Stores the current contents of the command line in the user command line buffer. This buffer is initially empty when Reflection is executed. This enables temporary recording of a command or text string. (See F8.)

F8 — Displays the current contents of the user command line in the command line window beginning at the current cursor position.

F9 — Causes the next function key (F1 to F8) pressed to generate its user-key value instead of its editing value (see *Configuring User Keys* on page TR-103). The desired user function key must be the first key pressed after F9.

Del — Deletes the character at the current position in the command line window and skips over the next character in the previous line buffer (i.e., it advances position in the previous line buffer without displaying a character).

Ins — Turns on insert mode within the command line window and within the previous line buffer. This allows the insertion of characters into the current command line without changing position in the previous line buffer.

Backspace — Moves the cursor one position to the left in both the command line window and the previous line buffer. If **destructive bkspace** is configured to *YES*, the character at the new position is erased.

Left — Moves the cursor one position to the left in the command line. Unlike DOS, it does not delete a character and does not change position in the previous line buffer.

Right — Moves the cursor one position to the right in the command line. Unlike DOS, it does not overwrite the character at the new cursor position and does not change position in the previous line buffer.

Home — Moves the cursor to the beginning of the command line window. It does not change position in the previous line buffer.

End — Moves the cursor to the right of the last character in the command line window. It does not change position in the previous line buffer.

Interrupting Command Execution

You may occasionally find it necessary to interrupt a command – a command to transfer a file, for example – while it's being executed. Reflection has four ways to do this:

- The first method is to press [Ctrl]-[Y], which orders the system to interrupt or terminate execution in an orderly fashion – by seeking an agreement between PC and host, for example.
- A soft reset, [Alt]-[S] may also be used. This acts like the user STOP function in file transfer, but also removes any pending requests.
- Another method is to press [Ctrl]-[A], which orders the system to terminate execution at any cost – by unilaterally severing communication, for example.
- [Alt]-[R] is a slightly stronger means of terminating execution than [Ctrl]-[A].

NOTE: If DISABLE-INTERRUPT is set to *YES*, only a hard reset will stop the execution of a command file; see page TR-218.

Interrupting the Display

You can interrupt a scrolling display by pressing [ScrollLock] or [Ctrl]-[S]. Use of [ScrollLock] requires that **Receive pacing** have a value of *XON/XOFF*, and that SCROLL-LOCK=STOP be set to *YES* (see page TR-221). Press [ScrollLock] again or [Ctrl]-[Q] to resume. You might use this operation during execution of the TYPE command, when you want to stop a scrolling file.

Command Entry Format Requirements

Command keywords are arranged alphabetically beginning on page TR-179 and explained there in detail with examples. This section explains the general rules that govern the use of the command language keywords.

- Only one command at a time may be entered on any command line.
- All commands require a space after the keyword. Spell and space command keywords exactly as they are shown in the

command summary, except where abbreviations are clearly permitted. For example, you may enter **CD** instead of **CHDIR**, **DEL** instead of **ERASE**, **REN** instead of **RENAME** and S instead of **SEND**. The keyword descriptions in the command summary tell you when abbreviations are permitted.

- Parameters enclosed within angled brackets (<>) indicate that something must be added to complete the command like a test condition or a filename. For example, **TYPE <filename>** means you must follow the keyword TYPE with the name of the file you want to display on the screen.
- Parameters enclosed within standard brackets ({}) indicate a set of choices, one of which must be made to complete the command. For example, **CLOSE {PRINTER | DISK}** means you must follow the CLOSE keyword with one of the indicated parameters, either PRINTER or DISK.
- A vertical bar (|) separates exclusive options. For example, **CLOSE {PRINTER | DISK}** means you must follow the CLOSE keyword with either PRINTER or DISK, but not both.
- Parameters enclosed within square brackets ([]) indicate optional components of a command; that is, any information contained within a pair of such brackets may, but need not, be included in the command. For example,

  ERASE [<d:>][<path>]<filename>[<.ext>]

  means you must include a filename, but the drive specifier, path, and extension parameters are optional.
- In specifying parameters, do not type the braces, angle, or square brackets, but only the information inside them, unless otherwise instructed.
- You may, in some cases, use the global filename characters (wildcards), ? and *, to specify certain parameters. They have the same meanings in Reflection that they have in DOS. (For more information on these characters, consult the DOS manual.)

- You must enter commands and parameters with correct spacing and punctuation. All punctuation included in the command formats must be entered.
- Files need not have extensions when you create or rename them. However, you must include the extension when you refer to an already existing filename that has an extension.

Text Strings

Several of Reflection's commands contain text parameters that you complete, or specify, by means of *text strings*. A text string can be one of the following:

- A command variable (explained in the next section)
- A string of characters enclosed within quotes, including expressions that use string operators; see page TR-230.

Hence the term *quoted strings*. In a command, the quotes that surround a string of characters indicate to Reflection that the enclosed string is to be duplicated literally on the screen. For example, the quoted string in the command:

DISPLAY "Welcome to Reflection"

produces the following expression on the screen when the command is executed:

Welcome to Reflection

Reflection recognizes several commonly used quotation conventions:

- Single quotes
- Double quotes
- Backslashes

You may use the convention with which you feel most comfortable, or a combination of conventions, provided you are consistent in any one use. Thus, for example, a text string containing a word enclosed by double quotes may be entered as the value for a **<text>** parameter in several different ways.

'The word "impact" is not a verb.'

The above string uses the single-quote convention to delimit the text string. The string is displayed on the screen as:

The word "impact" is not a verb.

Alternatively, you may use the double-quote convention to demarcate a string containing a literal quote, by typing:

"The word ""impact"" is not a verb."

Notice that the literal quotation marks surrounding the internal quoted word are doubled to avoid Reflection's confusing them with the text-string delimiters. Similarly, if you want the string to read:

The word 'impact' is not a verb.

Use the double-quote convention, as follows:

'The word ''impact'' is not a verb.'

Finally, when you want any character to be taken literally by Reflection, precede it by a backslash, as follows:

"The word \"impact\" is not a verb."

The important thing to remember about these conventions is that they are all designed to help the system distinguish clearly between quotation marks that are intended as literal characters and those that delimit the string.

Control Characters

An ASCII control character is a non-text character that signals the computer to perform an action or acknowledge a condition. For example, to display a text string followed by a carriage return/line-feed, the string is followed in the typed command by the ASCII control characters for carriage return and linefeed, ^M and ^J.

To include a control character in a command line, type a circumflex symbol (^) and follow it with the appropriate letter. (On the IBM and most workalike PCs, the circumflex is Shift-6.) See *ASCII Control Characters* on page TR-323 for a list of their meanings, decimal values, abbreviations, and the keystrokes required to create them.

Command Language

The Literal Escape Character

Reflection understands that **Ctrl** is intended whenever it encounters the circumflex. In this connection, you should also note that the combination, ^^, has a special control-character meaning of its own. The easiest way to mention the circumflex itself is to use the combination, \^. To mention the backslash, use the combination, \\.

The \ character is called a *literal escape* and implies that the next character is to be read as a literal character, not with any control character meaning. Note that the backslash preceding DOS path names will be stripped from text enclosed in quotes unless you change the literal escape character. See page TR-218.

Command Files

The power and flexibility of Reflection's command language is most evident in its role as a programming language. Although virtually all of the commands, at least theoretically, may be entered directly at the keyboard, most of them find their greatest use in command files.

A command file is a collection of commands arranged sequentially in a single file, such as you might create with a text editor. More precisely, it is a series of lines, each of which is terminated by a carriage return/line feed combination, and the whole of which can be terminated by a Ctrl-Z.

Reflection's command language allows you to construct such files for a wide range of communications-related applications. Several examples of command files are provided later in this chapter.

Converting AdvanceLink Command Files

Reflection provides a utility program which allows you to convert AdvanceLink command files so that they will work with Reflection. At the DOS prompt, use the following steps:

1. Make sure the program AL2WRQ.EXE is available in the current directory.
2. At the DOS prompt type:

   AL2WRQ <ALfile> <Rfile>.CMD

where <ALfile> is the AdvanceLink file, and <Rfile> is the converted Reflection file. Reflection converts the AdvanceLink file to a compatible command file with the extension .CMD. Minor editing may be required after the conversion.

Protecting Command Files

Command files can be *encrypted* to protect them from being altered or even run by other users. Encrypted files cannot be viewed with the TYPE command or decoded. There are two basic reasons for encrypting files:

- Encrypt command files to prevent users from viewing their contents or altering them. This allows you to embed a host system password within a command file, for example.
- Encrypt files to prevent users from invoking them without the correct password. To prevent a logon command file from being used by anyone other than the owner, for example.

To encrypt a file, use the utility *CRYPT* at the DOS prompt. The syntax for this program is as follows:

CRYPT <input file> <output file> [<password>]

The input file is any command file which has not been encrypted previously. The output file must have a different filename or extension. Note that the password is optional. If a password is used for encryption, it must be supplied when the command file is run.

Using Encrypted Files

Encrypted command files take a slightly different syntax if a password was used when the file was encrypted. (The slash before the password is required.):

<filename/password> [<optional parameters>]

NOTE: Optional parameters can be used after any command file. When the program is executed, these parameters are contained in variables, beginning with V1. See *Command Variables* on page TR-169.

If a password is not required, use the following syntax at the command line:

<filename> [<optional parameters>]

Adding Remarks to a Command File

Comments may be added to command files which are not encrypted to explain what is happening. To insert a comment, begin a new line with a semicolon. Whenever the first non-blank character on a line is a semicolon, Reflection treats the rest of the line as a comment and ignores it when the file is executed. You may well appreciate this when you are trying to figure out, months later, what one of your files does or is supposed to do.

Creating and Saving a Command File

Creating a command file is much like creating any other file. Use a text editor by placing Reflection in the background and loading the editor. When ready to invoke the command file, save the file (but leave it in memory so that you can easily make changes to it if it doesn't work as expected), and toggle back into Reflection. (You can also type a program in Reflection from any terminal mode screen in local mode. Turn on auto linefeed to make text entry easier, but make sure to turn it off. Use MSAVE to save display memory to disk.) The following is a short program for dialing an HP 3000 and logging on. Lines beginning with a semicolon (;) are *remarks*.

```
;This command file dials a remote HP 3000
;via a Hayes modem
;Wait one second
WAIT 0:0:1
;Transmit Hayes commands to dial phone
TRANSMIT "ATDT6341234^M"
;Wait at most 1 minute for connection
WAIT 0:1 FOR "CONNECT"
WAIT 0:0:2
;Get colon prompt
TRANSMIT "^M"
WAIT 0:0:10 FOR "^Q"
TRANSMIT "^M"
;Transmit HP signon
PTRANSMIT "HELLO MGR.ACCOUNT/PASSWORD"
```

Select a name for the command file. The extension .CMD should be used to help you distinguish command files from other files on the disk. The sample program might be named LOGON.CMD.

To save the file to disk, enter command mode and type **MSAVE <filename>**. All of display memory will be saved to the file. You can recall the command file to the screen by using the TYPE command on the command line.

1. First, you might want to clear display memory by homing the cursor, [Ctrl]-[Home], and pressing [Alt]-[J]. Then when the file is typed into display memory, you can edit it and use MSAVE to resave it to disk, without saving any extra lines which may be in display memory.
2. Press [Alt]-[Y] or **COMMAND LINE** from the system keys.
3. Type **TYPE <filename>** [Return]. For example TYPE LOGON.CMD.

Disk Operations on page UM-69 contains more detailed information about saving disk files within Reflection.

Command Language

Command Variables

One of the most versatile and powerful features of Reflection's command language is the ability to use command variables. A command variable is any of the character combinations, V0 through V9, which take on as values quoted strings not exceeding 80 characters. Such variables have a variety of uses:

- They establish limits.
- They indicate that different responses are acceptable in commands asking for input.
- They program communications procedures that involve repeated operations.

NOTE: Three variable names have been reserved: **$SERIAL, $DATE,** and **$TIME**. These may be used as variables alone or with operators. The format for these commands is discussed under command language operators which begins on page TR-230. The MID operator allows you to select portions of the contents of these variables.

The following command file prompts the user to enter **Y** or **N**, and then checks what was entered.

```
DISPLAY "ENTER Y OR N^M^J"
ACCEPT V1 LIMIT 1
IF V1 <> "Y" AND V1 <> "N"
    DISPLAY "MUST BE 'Y' OR 'N'"
ENDIF
```


Using the Contents of Variables in Commands

The command interpreter preprocesses commands containing dollar-sign combinations by substituting the contents of the corresponding variable into the command. The resulting command string is then parsed and executed.

The dollar-sign combinations may, in a way, be considered *aliases* for the command variables. The following command file demonstrates *aliasing* of command variables using the dollar-sign combination. The command file prompts the user for a **Y** or **N**, and then jumps to **CASE_N** or **CASE_Y**, depending on what was entered. Note that an underscore can be used in the label name, but not a hyphen.

```
:ASK_AGAIN
DISPLAY "Enter Y or N: "
ACCEPT V1
IF V1 <> "Y" AND V1 <> "N"
    GOTO ASK_AGAIN
ENDIF
GOTO CASE_$1
:CASE_Y
<do stuff>
:CASE_N
<do stuff>
```

The following command file prompts the user and computes the area of a rectangle.

```
DISPLAY "Enter Length: "
ACCEPT V1
DISPLAY "Enter Width: "
ACCEPT V2
LET V3 = V1 * V2
```

DISPLAY "THE AREA IS $3"

In some situations, the direct use of command variables is inappropriate. For example, **DISPLAY "V1"**, displays the literal *V1* rather than the contents of the variable, **V1**. The command, **OPEN V1**, attempts to open a file named **V1** rather than a file with the name stored in the variable **V1**. In these cases, using the combination $<number> solves the problem. (In the example below, be sure to leave a space after the command name, and between DISCUSS and the double quotation mark.)

LET V1 = "POLITICS"
DISPLAY "I REFUSE TO DISCUSS " & V1 & "^M^J"
DISPLAY "V1 is my favorite variable^M^J"
DISPLAY "$1 is my favorite subject^M^J"

When you invoke the command file, you should see:

I REFUSE TO DISCUSS POLITICS
V1 is my favorite variable
POLITICS is my favorite subject

When you invoke the command file, the V1 in the statement, DISPLAY "V1 is my favorite variable" is taken literally by Reflection's command interpreter. Thus, Reflection displays **V1** instead of the contents of the variable that V1 represents. However, when $1 is used in the quoted string, "$1 is my favorite subject," Reflection's command interpreter knows that you want to display the contents of the variable represented by V1.

The following command file demonstrates how a string variable can contain a command.

LET V1 = "SEND"
$1 MYFILE

Note, however, that the $<number> combination produces a syntax error if used where a variable is required. For example, **ACCEPT $1** will produce an error because a text string cannot be used with ACCEPT.

Sample Command Files

The commands in a command file are executed sequentially and automatically when the file is initiated. The following command file invokes another command file called DOSTUFF.CMD ten times.

```
LET V1 = 1
:MY_LABEL
DOSTUFF.CMD
LET V1 = V1 + 1
IF V1 < 11
    GOTO MY_LABEL
ENDIF
```

The following is a program for dialing a VAX and logging on. Several lines deal exclusively with potential problems in making the connection.

```
;This command file dials a remote VAX via a modem
;Replace the phone number with the correct one
;
;Transmit an AT and carriage return
TRANSMIT "AT^M"
WAIT 0:0:4 FOR "OK"
IF NOT FOUND
;Clear the screen and display message
   DISPLAY "^[[H^[[JNO RESPONSE FROM MODEM"
   STOP
ENDIF
;Transmit modem commands to dial phone
TRANSMIT "ATDT<phone#>^M"
;Wait at most 1 minute for connection
;If no connection, end
WAIT 0:1 FOR "CONNECT"
IF NOT FOUND
;Clear the screen and display message
   DISPLAY "^[[H^[[JCALL NOT COMPLETED"
   STOP
ENDIF
WAIT 0:0:1
LET V0=1
:username
```

```
TRANSMIT "^M"
WAIT 0:0:05 FOR "Username"
IF FOUND
   TRANSMIT "<username>^M"
   GOTO pswd
;Checks value of V0, tries five times
ELSE
   IF V0=5
   DISPLAY "Timed out waiting for 'username' prompt"
   STOP
;Increments the value of V0 by one
ELSE
   LET V0=V0+1
   GOTO username
ENDIF
:pswd
WAIT 0:0:20 FOR "Password"
IF FOUND
   TRANSMIT "<password>^M"
   GOTO alldone
ELSE
   DISPLAY "Timed out waiting for 'password' prompt"
   STOP
ENDIF
:alldone
STOP
```

Initiating a Command File

These are the three ways of executing a command file from within Reflection:

```
INVOKE PROC.CMD
CHAIN MYFILE.CMD
MYFILE.CMD                    (implicit INVOKE)
```

See the explanations under INVOKE and CHAIN in the *Command Language Keywords* appendix which begins on page TR-179.

- The first way, which is available to you only when you are already in Reflection, is to use the INVOKE command. Any variables which have been defined during the current session with Reflection retain their values.

- The second way, which is also only available within Reflection is to use the CHAIN command. Any variables which have been defined during the current session with Reflection retain their values.
- The third method, called an *implicit* invoke, is available from both within Reflection and from the DOS prompt. You can pass information to an implicitly invoked command file by placing text after the command filename. The V1 and V0 variables are always reset under an implicit invoke and other variables may be.

The implicit invoke method allows you to execute a command file at the same time you start up Reflection. Enter **R1** followed by the appropriate configuration filename, followed by the name of the command file you want to execute (and any text which you wish to be stored as variable information), with one blank between each. Or, if you do not want to mention a configuration file, follow the **R1** with a comma, colon or semicolon, followed by the name of the command file. The comma, colon, or semicolon act here as a place holder for the configuration filename. Note that it is always the file in the second position that is executed. From DOS, the following line will cause Reflection to execute an initial command file and load V1 with the string 4. (V2 will be set to a null string.)

R1,START.TXT 4

The last line in the following command file is a CHAIN command, which begins execution of another file. There is no need to return to the initial command file, so CHAIN is appropriate.

SEND FILE1.DOC
SEND FILE2.DOC
CHAIN MORECMDS

Using Subroutines To reuse a series of lines in more than one file easily, create a file which can be chained to or invoked at any time. For example, the following lines are taken from the file on page TR-172; and modified to make a file that can be called:

;DIAL.CMD
;Invokable command file to dial a Hayes modem.

```
;Entry - Expects the phone number as the first
;        passed parameter
;Exit -  Terminates command processing if the call
;        cannot be completed or returns to the
;        invoking file

;Get response from modem
TRANSMIT "AT^M"
WAIT 0:0:4 FOR "OK"
IF NOT FOUND
    DISPLAY "No response from modem"
    STOP
ENDIF
;Transmit ATDT, the phone #, and a carriage return.
TRANSMIT "ATDT" & V1 & "^M"
WAIT 0:1 FOR "CONNECT"
IF NOT FOUND
    DISPLAY "Call not completed"
    STOP
ENDIF
```

These lines could be stored in the file DIAL.CMD, and replaced in the logon file by the following lines:

```
;Invoke command file to dial the modem
DIAL.CMD 5551234
:username
TRANSMIT "^M"
WAIT 0:0:5 FOR "Username"
etc. ...
```

Note that the phone number is passed as a parameter to the file DIAL.CMD and is placed in the variable V1.

Text String Operators

Reflection includes a number of text string operators that form strings and numbers, and which can be used in any context where text variables are allowed. For example, LENGTH (V1) returns the length of the sting contained in V1. These operators are listed alphabetically and explained, with examples, at the end of *Command Language Keywords*, which begins on page TR-179.

Arithmetic Expressions

Some of Reflection's commands contain numeric parameters that are completed, or specified, by means of an arithmetic expression. An arithmetic expression can be any of the following:

- A command variable
- A number
- A combination of a variable and a number by means of the arithmetic operators plus (+), minus (-), product (*), and quotient (/)

An arithmetic expression may contain numeric operators.

Normal precedence rules apply to operators (* and / are done before + and -), but parentheses may be used to force the order of execution. All arithmetic is performed with integer values (no decimal fractions, so 4/3 = 1) in the range -32767 through +32767. Reflection does not round off numbers.

Remote Use of Commands

To complete the discussion of Reflection's command language, one more important use needs to be mentioned. The language can be used by a remote computer to control the operations of a PC. A remote user transmits instructions to a PC, which the PC, when programmed with Reflection, can understand and execute; the remote computer prefaces each command it issues with the escape sequence, $^{E}{}_{C}$&oC, as follows:

$^{E}{}_{C}$&oC<Reflection command>$^{C}{}_{R}$

When Reflection receives a command language command from the host computer, it transmits a response (technically a device completion code) when the command has been executed. This response is either an **S** for success or **F** for failure, and is transmitted as a Type 3 block transfer. (See page TR-259 for more information about block transfers.)

Note that while a PC programmed with Reflection executes instructions transmitted to it in this way, it is not itself operating in command mode; its own command line has not been accessed and no command has been entered at the keyboard.

The disk of support files provided with Reflection contains examples of command files and host files which invoke Reflection commands. Also see *Configuring User Keys* on page TR-109 for more information.

The following example is a TDP *use* file which transfers two files from the host computer. (TDP is a text editor available on the HP 3000.)

Q "^{E}c&oCCONTINUE"
ZP::="ENTER COMPLETION CODE"
Q "^{E}c&oCRECEIVE AFILE ASCII"
ZP::="ENTER RUN COMMAND "
Z::
ZP::="ENTER COMPLETION CODE"
Q "^{E}c&oCRECEIVE AFILE2 DELETE"
ZP::="ENTER RUN COMMAND "
Z::
ZP::="ENTER COMPLETION CODE"

Note that ^{E}c in this example means to press [Esc] with display functions on.

Command Language Keywords

This appendix contains an alphabetical listing of Reflection's commands and a description of their various uses. Representative examples are given for most of the commands.

NOTE: Function keys, except F10, have special operations in command mode. See page TR-162 for more information.

- All command keywords require a space immediately after them.
- An alphabetical listing of text string operators, along with their descriptions and examples, is included at the end of this appendix.
- Parameters enclosed in angled brackets (<>) indicate that something must be added to complete the command, e.g., a test condition or a PC filename.

  The parameter **<text>** represents either literal text or a string expression.
- Parameters enclosed in standard brackets ({}) indicate a set of choices, one of which must be made in order to complete the command.
- Parameters enclosed in square brackets ([]) indicate optional components of a command; that is, any information contained within a pair of such brackets may, but need not, be included in the command.
- Parameter choices are separated by a vertical bar (|).

ACCEPT

ACCEPT orders Reflection to accept the value for a command variable entered at the keyboard.

Format: **ACCEPT V{0-9} [LIMIT <number>]**

Unless otherwise restricted by the LIMIT option, the value given to the variable may be any block of text less than or equal to 80 characters.

Use the LIMIT option to restrict the length of the value entered for the command variable.

When the ACCEPT command is processed the cursor remains in place until the user responds. Users are then able to interact with the terminal within (but only within) the character range of the command. $\boxed{\text{Return}}$ terminates the input string.

Example: This example instructs Reflection to accept as a value for the command variable V1 any string not exceeding two characters.

ACCEPT V1 LIMIT 2

BACKGROUND

Places Reflection into background mode from within a command file.

Example: The following lines begin a file transfer and then place Reflection in background mode. The file transfer continues uninterrupted. If a DOS batch file invoked the Reflection command file, the batch file will continue with its next command while Reflection continues to process the command file in the background. The FGD Boolean can be used to test Reflection's current state.

SEND LEDGER.DOC
BACKGROUND

BACKUP

BACKUP backs up PC files to an HP 3000, VAX/VMS, or UNIX-based host. See page UM-157 for more information.

Format: **BACKUP [<d:>][<path>]<filename>[<.ext>]**
[<hostfile>]
[/S] [/C] [/D:<date>] [/T:<time>]
[/L:<filename>]

/S
Requests that subdirectory files be included in the backup.

/C
Requests a complete backup, whether files have been modifed or not.

/D:MM-DD-YY
Only files created or modified after the given date will be backed up.

/T:00:00:00
Only files created or modified after the given time will be backed up.

/L:<filename>
Defines the file where the backup report will be stored; BACKUP.LOG is the default.

Example: The following line creates a sufficiently large file on the HP 3000 for the PC files (on a 20-megabyte hard disk). The next line backs up every file on the hard disk of the PC.

PTRANSMIT "FILE BACKUP;DISC=80000,32,8"
BACKUP C:*.* /C/S

BREAK

BREAK transmits a break signal of approximately 200 milliseconds to the host computer. The interpretation of this signal depends on the host to which it is sent. SET BREAK-LENGTH may be used to alter the length of the signal; see page TR-217.

Command Language

Command Language

Format: **BREAK**

In command mode BREAK performs the same action as pressing Ctrl-Break from the keyboard.

Example: This example shows how to break and abort a host program running on an HP 3000. After sending a break, Reflection waits for the host prompt. It then transmits an escape followed by a colon (turning on local echo), and aborts the program running on the host.

BREAK
WAIT 0:0:2 FOR "^Q"
TRANSMIT "^[:ABORT^M"

CHAIN

The CHAIN command begins, or starts up, a command file. It instructs Reflection to call up the file specified in the filename parameter and begin execution.

Format: **CHAIN [<d:>][<path>]<filename>[<.ext>]**

The CHAIN command is similar to INVOKE, except that there is no return to the control level issuing the CHAIN command.

The filename you use must obey DOS rules. If no drive or path is specified, Reflection searches the current directory for the file you name.

You may use the abbreviation, C, instead of the command name, CHAIN.

Example: To begin execution of the file MYCMD.CMD, type on the command line or include it in a command file:

CHAIN MYCMD.CMD

CHDIR

CHDIR displays the current directory. It may be used to change the current directory of the default drive or of a specified drive, or to display the current directory path of a drive.

Format: **CHDIR [[<d:>]<path>]**

If you enter CHDIR with no parameters specified, the current directory path of the default drive is displayed. Note that Reflection and DOS maintain separate directories when you are operating in background mode.

To change the current directory of any drive, you must complete the path option and, if appropriate, the drive option.

You may use the DOS abbreviation, CD, instead of the command name, CHDIR.

CLOSE

CLOSE disables access to the printer, remote device, or disk.

Format: **CLOSE {PRINTER | DISK | REMOTE}**

The "to" device function-key labels are updated.

CLOSE can also be used to close a file by its file number (see the OPEN command):

Format: **CLOSE <file number>**

;Comment

A semicolon (;) is used in command files to set off comments. Any text or command which occurs on a line beginning with a semicolon is ignored by Reflection; however, if an error condition exists from the execution of the previous command, it is removed.

CONTINUE

CONTINUE is used in command files to instruct Reflection to ignore an error in the following line, should it occur, and to proceed as if nothing had happened.

Format: **CONTINUE [ON | OFF]**

If CONTINUE is given without parameters, Reflection ignores an error only in the line immediately following an occurrence of CONTINUE.

If the optional ON clause is specified, Reflection continues past all errors. All error handling becomes your responsibility. The optional OFF clause reverses the ON clause.

Example: The following sequence in a command file instructs Reflection to ignore any error in the execution of SEND. The resulting error-code (0 for successful execution) is displayed.

CONTINUE
SEND FROM.FIL TO DEST.FIL
DISPLAY ERROR-CODE
STOP

COPY

COPY allows you to copy one or more files to a specified disk. Global characters (*wildcards*) are acceptable. It operates much like the DOS COPY command but excludes the append options and switches.

Format: **COPY <filename>[<.ext>] [<filename>[<.ext>]]**

Example: The following sequence copies SALES.NOV from disk drive A: to \SALES\NOV.DOC.

COPY A:SALES.NOV \SALES\NOV.DOC

DIR

DIR displays a list of all the disk files that match a name you specify. The command may thus be used to view all the directory entries or only those for specified files, such as those with a particular extension.

Format: **DIR [<d:>][<path>][<filename>][<.ext>]**

You may use the global characters (*wildcards*), ? and *, in specifying the filename and extension parameters. For information on these characters, consult your DOS manual.

If you want to list all the directory entries on the default drive, complete the path option alone. If you wish to view all the directory entries on anything but the current drive, you must enter both path and drive parameters.

If you do not specify any options, entering DIR lists all the files in the current directory.

DISPLAY

DISPLAY displays data on a terminal screen as if it had been received from the host.

Format: **DISPLAY {<text> | <arith. expression>}**

DISPLAY must be followed by either a string of characters surrounded by quotes or an arithmetic expression. A variable containing either of these may also be used, in which case the quotes are omitted. Control characters may be displayed by preceding the character with the ^ character.

Escape sequences may be sent to the terminal with DISPLAY, which can be used to configure Reflection, construct graphs and diagrams, or a variety of other things. To make full use of DISPLAY, however, you must understand the terminal Reflection is emulating. The more you know about the capabilities of that terminal, the more you can do with the command.

NOTE: Control and escape sequences are entered through use of ^ and ^[respectively.

Examples: In HP mode, to home the cursor, clear the screen and display a message, enter:

DISPLAY "^[H^[JWelcome to Reflection"

In this example, the internal square brackets are literal and must be included in the command.

In VT mode, to home the cursor, clear the screen and display a message, enter:

DISPLAY "^[[H^[[JWelcome to Reflection"

Here again the square brackets are literal.

Drive

Drive changes the default disk drive.

Format: **<drive letter>:**

The letter displayed in the command prompt designates the default, or current, drive. The Drive command allows you to change the default drive, and thus the drive letter in the command prompt, by entering a new drive letter followed by a colon on the command line.

Unlike other commands in Reflection's command language, the keyword, *Drive*, is not part of the command format, and so is not entered at the keyboard when the command is entered.

Example: If A is the default drive and you want to change it to B, enter:

B:

on the command line. When you execute the command, B becomes the default drive and the drive letter **B** is displayed in the command prompt.

ELSE

ELSE signals the alternative case, if any, associated with an **IF** command. It performs no operation by itself, but is followed by one or more executable commands.

Format: **ELSE**

ELSE must be terminated by an ENDIF command. Refer to ENDIF and IF command descriptions.

Example: The following lines could be used in a modem-dialing command file. If the literal CONNECT is not returned by the modem within 2 seconds, the message is displayed and the command line is exited.

```
WAIT 0:2 FOR "CONNECT"
IF FOUND
   INVOKE LOGON.CMD
ELSE
   DISPLAY "COULD NOT LOG ON^G^M^J"
   STOP
ENDIF
```

ENDIF

ENDIF terminates the execution of conditional statements initiated by a matching IF statement.

Format: **ENDIF**

When, in the running of a command file, an IF parameter's condition is satisfied, one or more associated commands are executed. Otherwise, commands are skipped until the matching ENDIF is encountered, which instructs Reflection to proceed to the next command in the file, or until ELSE is encountered, which instructs Reflection to execute an alternative to the original command. An ENDIF then terminates the ELSE sequence.

If a nested sequence of IF conditions is being tested, an occurrence of ENDIF completes the testing of the most recent, or innermost, condition.

Example: The following tests values for V1 and V2 before executing the LET statements. The inner IF condition will not be executed if V1 is greater than or equal to 10. If the first test succeeds, the variable V1 is incremented, even if the second test fails.

```
IF V1 < 10
    IF V2 < V1
        LET V2 = V2 * 2
    ENDIF
    LET V1 = V1 + 1
ENDIF
```

ENTER

ENTER has the same effect as pressing Enter (keypad + key).

Format: **ENTER**

ERASE

ERASE deletes the file with the specified filename from the specified directory or drive, or deletes the file from the current directory drive if no drive or path is specified.

Format: **ERASE [<d:>][<path>]<filename>[<.ext>]**

Be sure that the filename you enter obeys DOS rules. You may use the global characters (*wildcards*), ? and *, in specifying the filename and/or its extension. These should be used with care, however, since more than one file may be erased with a single command. Consult your DOS manual for more information.

You may use the DOS abbreviation, DEL, instead of the command name, ERASE.

Examples: To remove the file COM.DOC from the current directory of drive A, enter:

ERASE A:COM.DOC

To erase all the files in the current directory:

ERASE [d:]*.*

To erase all the files in a specific directory:

ERASE [<d:>]<path>*.*

EXIT

EXIT is used to exit Reflection and return to DOS. If Reflection has been placed in background mode (the letter **B** will be present between function key labels), EXIT performs the same function as the hot-key keystroke. A host connection is maintained in the background (if previously active), and the DOS prompt or DOS application is brought to the foreground.

Otherwise, if not in background mode, EXIT erases everything in display memory, returns the system to DOS, and leaves the PC logged on to the host computer if connection to the host was active. You may set DISCONNECT-ON-EXIT to *YES* to ensure that exiting Reflection logs you off the host.

Format: **EXIT [arithmetic expression]**

The **[arithmetic expression]** option sets the DOS error level. See the DOS manual's IF sub-command. If Reflection has been installed into memory as a pop-up, the EXIT command has the same function as the BACKGROUND command. You can uninstall Reflection only with the keystroke Alt-X.

Example: To log off an HP 3000 and return to DOS:

TRANSMIT "BYE^M"
EXIT

Example: To log off a VAX host and return to DOS:

TRANSMIT "LO^M"
EXIT

FOR

FOR causes Reflection to execute any other Reflection command upon a variable, which represents one or more specified PC filenames. Execution is similar to the DOS command of the same name.

Format: **FOR** V{0-9} **IN**
[<d:>][<path>]<filename>[<.ext>]

<executable command>

The specified string variable will take the value of each filename that matches the indicated file specification, at which time the **<executable command>** will be executed.

You may use the DOS global characters (*wildcards*), ? and *, in specifying the filename and/or its extension.

The executable command may be any Reflection command, including INVOKE.

Examples: To send every PC file on the B disk to the host, type:

FOR V1 IN B:*.* SEND V1

V1 is the reference variable. For each filename matching the specified pattern V1, V1 assumes the value of the filename.

B:*.* is the group of PC files the reference variable represents. In the example above, **SEND V1** is the Reflection command that FOR causes to be executed upon the files specified in **V1**.

In many cases, you may need to remove the PC extension filename before sending the file to the host computer.

FOR V1 IN B:*.TXT SEND V1
TO MID(V1,3,FIND(".",V1)-1)

In the above example (contained on a single line), all the PC files in drive B with the .TXT extension are sent to the host. The **TO** portion of the command specifies the host filename under which the PC files will be saved.

The **MID(V1,3,FIND(".",V1)-1)** portion of the command specifies that the .TXT extension of the PC filenames will be dropped for the host filenames. MID indicates that only part of the string V1 with be used. V1 is the complete

filespec, including the drive designator (B:). The <
3 indicates the first letter of the filename, and the <
FIND string indicates the last letter of the <
filename, up to the letter before the period. See <
notes on the MID and FIND operators, in the final section of this reference.

When FOR appears with commands which do not accept variable arguments (for example, TYPE), you may still use macro variables (*$1*). The result may be undesirable, however.

LET V1 = "TEXT.DOC"
FOR V1 IN *.TXT TYPE $1

The above example would type TEXT.DOC repeatedly for every occurrence of a *.TXT file. Use $$1 after TYPE to correct this. When the line is initially processed, the $$1 is stripped to $1. Then the appropriate value is used for V1 when TYPE $1 is executed.

FOREGROUND FOREGROUND forces Reflection into the foreground. It can be used in conjunction with a variety of commands, such as WAIT, HOLD, and IF.

Example: The following example forces Reflection to the foreground if an error occurs.

```
CONTINUE
SEND MYFILE
IF ERROR
    FOREGROUND
ENDIF
```

GOTO GOTO is used in command files to transfer control to the line following the one with the specified label.

Format: **GOTO <labelname>**

The **<labelname>** is entered in a command file as a colon followed by a label name, thus:

Command Language

:<labelname>

If **:<labelname>** is not defined, the command file currently running terminates with the message, **No target label**.

NOTE: Beware of nesting a labelname within an IF sequence, as it will likely not have the effect you intend.

Example: The following attempts to send a file a maximum of five times.

```
LET V1 = 1
:try_again
CONTINUE
SEND AFILE
IF ERROR
    LET V1 = V1 + 1
    IF V1 <= 5
        GOTO try_again
    ELSE
        EXIT
    ENDIF
ENDIF
```

HELP

HELP calls up information that explains command keywords.

Format: **HELP [keyword]**

Entering HELP by itself on the command line produces a help menu listing Reflection command language terms for which there is assistance. To get more detailed information about a command or term, type HELP followed by a **<keyword>** from the listing.

Example: If you need help with the various options of WAIT, enter on the command line:

```
HELP WAIT
```

HOLD

HOLD allows a command file to suspend operations until a specified time, until an information condition has been met, or until a defined text string has been received from the host. This allows the user to have access to the keyboard while a command file has been set in motion.

Format: **HOLD [[UNTIL]<hh:mm:ss.cc>]**
[FOR {<text>}]

Once HOLD is executed, the command line is no longer available until the time or FOR parameters have been satisfied. Alt-R, a hard reset, disables HOLD and returns Reflection to normal operation.

The parameter **<hh:mm:ss.cc>** stands for hours, minutes, seconds, and hundredths of a second. UNTIL is optional. Complete the time parameter if you want the command file to suspend operations for a certain period of time. Use UNTIL if you want the system to wait until a given clock time before proceeding with the next command.

The FOR option instructs the command file to wait for a given response to come through the communications port before proceeding. Complete the **<text>** parameter to suspend operations until a given text string (password, filename, message, or other block of text) is received.

The FOR clause of the HOLD command determines the value of the condition, FOUND. Initially, HOLD sets the value to *false*. If and when the condition specified in the FOR clause is satisfied, HOLD sets FOUND to *true*.

Example: The following example completely controls a user session.

```
HOLD UNTIL 8:00:00
TRANSMIT "RUN TDP^M"
HOLD FOR "END OF PROGRAM"
WAIT FOR "^Q"
TRANSMIT "RUN QUERY.PUB.SYS^M"
HOLD FOR "END OF PROGRAM"
WAIT FOR "^Q"
TRANSMIT "BYE^M"
```

IF

IF tests a condition. If the condition is satisfied the associated command (or commands) is executed. If the condition is not satisfied, subsequent commands are ignored by Reflection until a matching ELSE or ENDIF is encountered.

Format: **IF <condition>**

Use IF to have Reflection perform any of the following simple tests:

- Test the ERROR condition.
- Test the FOUND condition.
- Test the DCD condition.
- Test the EOF condition.
- Test the existence of a string.
- Test the value of a variable.
- Perform algebraic or textual comparisons.

See page TR-230 for syntax and more information on the following operators.

The ERROR condition is set every time a command is executed. It has the value *true* or *false*, depending on whether an error has occurred in the command's execution. The command, IF ERROR, instructs Reflection to determine the value of the error setting and to take action based on its findings. If the condition fails, execution is suspended until an ENDIF or ELSE is encountered.

The FOUND condition is set by the FOR clause of the WAIT or HOLD command. If the FOR clause is satisfied, the FOUND condition has the value *true*; if not, it has the value *false*. The command, IF FOUND, instructs Reflection to determine the value of the FOUND setting and to take action based on its findings.

The DCD condition tests whether the **Data Carrier Detect** status of the datacomm port is *true* or *false*. This is useful for determining when a connection is made when dialing a remote computer via modem. Note that some modems always assert DCD.

The EOF condition tests whether or not the end of the file has been reached. The file number of a currently open file must be provided. *True* means that the indicated file is at its end.

The EXIST condition indicates the existence of a DOS file. *True* means that the indicated file exists.

IF may also be used to compare two algebraic expressions or two blocks of text, and to have Reflection take action based on the result. Reflection recognizes six operators for making comparisons:

1. Equal to (=)
2. Not equal to (<>)
3. Less than (<)
4. Less than or equal to (<=)
5. Greater than (>)
6. Greater than or equal to (>=)

Reflection is able to compare any two blocks of text not exceeding 80 characters each, by comparing the letters of which they are composed.

IF should be used in conjunction with one or more executable commands – with SEND or DISPLAY, for example. Note, further, that an IF is generally based on the result of a prior command – WAIT or ACCEPT, for example.

IF is often used to test the primitive, or simple, conditions described above and to specify associated actions. But it may also be used to test compound conditions and to specify actions associated with these conditions. Compound conditions are constructed from one or more simple conditions using AND, OR, and NOT.

Finally, you may test a nested array of conditions by making several uses of IF. However, an occurrence of ENDIF must complete each conditional sequence.

Examples: The first IF statement determines whether PROC1 or PROC2 is invoked based on the values of various variables. The second IF statement chains to ERROR.CMD depending on variable values or an error condition.

```
IF V1*V2 <> V3/12 AND LENGTH(V5) < 7
    INVOKE PROC1.CMD
    LET V1 = V1 * V1
ELSE
    INVOKE PROC2.CMD
ENDIF
IF (V1=2 AND V5 <= 12) OR ERROR
    DISPLAY "ERROR^G"
    CHAIN ERROR.CMD
ENDIF
```

INVOKE

INVOKE instructs Reflection to call up the file specified in the filename parameter and begin executing it. When the invoked command file is completed, control returns to the invoking level (the caller).

The INVOKE command starts executing a command file, preserving any existing variable values. INVOKE instructs Reflection to call up the file specified in the filename parameter and begin executing it. When the invoked command file is completed, control returns to the invoking level (the caller).

Format: **[INVOKE] [<d:>][<path>]<filename>[<.ext>]**

The INVOKE command is similar to the CHAIN command, but implies that control will return after the invoked command file is completed. An invoked command file may invoke yet another command file; however, this can only be done to a depth of five levels. In practice, you may be limited to fewer levels owing to the DOS maximum number of open files. The DOS limit can be adjusted by use of the DOS configuration file (CONFIG.SYS); consult your DOS manual. FILES=20 is recommended.

If no directory is specified, the current directory is searched. Otherwise, if you are using DOS 3.0 or greater, the directory where the Reflection program resides is searched for the command file. You can also use the DOS SET environment command to indicate where Reflection related files reside; use SET REFLECT1= (or REFLECT3=/REFLECT7=) followed by the path name.

Existing variables are unaffected when INVOKE precedes the command filename; if any variables have a current value, they are maintained during the execution of the command file.

Example: To invoke the command files named PHASE1.CMD and PHASE2.CMD, type the following on the command line. Note that the invocation of PHASE2.CMD will immediately follow PHASE1.CMD.

INVOKE PHASE1.CMD
INVOKE PHASE2.CMD

You may use the abbreviation, X, instead of the command name, INVOKE.

Implicit Invoke You may also use the command filename alone as a command; termed *implicit invocation*. This can be done either on the command line or when loading Reflection at the DOS prompt. Whenever a filename is mentioned in place of a normal command, Reflection assumes that you wish to invoke the file as a set of commands.

Any strings mentioned after the filename are assumed to be parameters to be passed to the command file. You may pass parameters to command files invoked from the DOS prompt when loading Reflection as well. For example:

R1, TRIAL.CMD MAY.TXT 25

As soon as Reflection 1 is loaded, TRAIL.CMD will be invoked. The variable **V1** is loaded with the first parameter delimited by a blank or tab (in the example, MAY.TXT). **V2** is loaded with the second parameter delimited by a blank or tab (25 in the example) and so forth, until **V<n>**, which becomes a null variable. Variables which follow the null variable are unaffected.

The command variable, **V0**, captures all of the text entered after the filename. If you displayed the contents of V0 from the example, it would be MAY.TXT 25, including any preceding or following spaces. If no strings follow the command filename, V1 and V0 are set to null strings.

Example: The following example might be called *SENDIT.CMD*; it allows users to enter filenames as they invoke the command file.

```
IF V1 = ""
    DISPLAY "Transfer what file?"
    ACCEPT V1
    DISPLAY "^M^J"
ENDIF
SEND V1
```

If a filename is placed after the command filename, as in **SENDIT.CMD MYFILE**, *myfile* is stored in V1. If V1 is empty, the user is prompted for a filename. (If INVOKE was used with this command file instead of the command filename alone, the command file would use any value previously stored in V1 for the filename.)

KBYE

Allows Reflection to tell a remote KERMIT running in server mode to exit from KERMIT and, if applicable, logout or terminate its job or process. It does not take any parameters.

KFINISH

Allows Reflection to tell a remote KERMIT to exit from KERMIT. It does not take any parameters.

KGET

Allows Reflection to receive a file from a remote KERMIT running in server mode.

Format: **KGET <filename>[<.ext>]**

KRECEIVE

Receives a file from the host using KERMIT protocol. The filename may be a string expression. You may also use KR.

Format: **KRECEIVE [<filename>[<.ext>]**
[FROM <hostfile>]
[APPEND | DELETE]]

If all parameters are omitted, the filename used will be that which has been specified by the sender. Omitting parameters allows the sender to use wild cards and send multiple files; furthermore, DELETE is assumed in this case. See KSEND and page UM-143.

Example: The following command receives a file under the name AFILE, once KERMIT has been invoked on the host.

KRECEIVE AFILE

KSEND

Sends a file to the host using KERMIT protocol. The filename may be a string expression. You may also use KS.

KSEND <filename>[<.ext>] [TO <hostfile>]

If the optional parameter is omitted, you may use wild cards in the required parameter to send multiple files in a single command. See KRECEIVE and page UM-143.

Example: The following command sends a group of files to the host which all have the extension .DOC, once KERMIT has been invoked on the host. Files maintain the same name on the host. DELETE is assumed.

KSEND *.DOC

Label

Label inserts a place marker in a command file. It performs no operation by itself, but allows another command to refer by name to a certain place in a file.

Format: **:<labelname>**

The command **GOTO <labelname>** causes commands to be executed beginning with the line immediately following **:<labelname>**. A label name can be no more than 16 characters.

Refer to the description of GOTO.

LET

LET assigns a quoted string or arithmetic expression to a command variable.

Format: **LET V{0-9} = {<text> |**

<arithmetic expression>}

You must complete the parameters on both sides of the equal sign.

The **<text>** option may be completed with a command variable, password, message, or other block of text not exceeding 80 characters. When you enter a value for this parameter, it must be enclosed within quotes, either double or single.

The **<arithmetic expression>** option may be completed by any well-formed expression containing numerals, variable letters, and the operators, plus (+), minus (-), product (*), and quotient (/).

Examples: Some uses of LET follow:

```
;The result of this display is ABCXYZ.
LET V1 = "ABC"
LET V1 = V1 & "XYZ"
DISPLAY V1
```

```
;In this example, V1 receives the value 12.
LET V1 = 3
LET V1 = ((V1 + 5)*V1) / LENGTH("XY")
```

Since V1 is equal to 3, the first part of the equation is equal to 24. The length of XY is 2, so the result is 12.

LOAD

LOAD loads and activates a configuration data file. CFG is the recommended filename extension.

Format: **LOAD <filename>[<.ext>]**

LOG

LOG saves, or *logs*, incoming data to a file or to the printer.

Format: **LOG [OFF]**

Use of LOG is equivalent to pressing the LOG BOTTOM function key. It toggles on the **LOG BOTTOM** label of the device modes keys.

Command Language

If the optional OFF parameter is included, logging is disabled; otherwise logging is enabled.

Example: The following commands generate a directory listing on disk.

CLOSE PRINTER
OPEN AFILE
LOG
DIR *.*
LOG OFF
CLOSE DISK

MSAVE

MSAVE allows you to save all of display memory to a file on the PC or other specified device. The current settings of the REMOTE/PRINTER labels are ignored and unchanged.

Format: **MSAVE {PRINTER | REMOTE | <filename>[<.ext>] [APPEND | DELETE]}**

You must complete the APPEND or DELETE parameter if the file exists.

Example: The following command saves all of display memory to the file *update.dat*, and appends to the file if it already exists.

MSAVE UPDATE.DAT APPEND

OPEN

OPEN is used to allow output to be directed to a printer, disk device, or character device.

Format: **OPEN {REMOTE | PRINTER | <filename>[<.ext>] [APPEND | DELETE]}**

The command should appear on a single line, with no spaces between the path and filename. The filename you use must obey DOS rules. If the file exists, you must complete the APPEND or DELETE parameter. Use APPEND when you want to add data to an existing file, and use DELETE when you want to overwrite an existing file.

The **"to" device** labels are updated.

Example: This example opens the configured printer device and writes a line to the printer.

OPEN PRINTER
LOG
DISPLAY "Beginning file transfer sequence^M"

The following alternate format is used to read input or write output to or from a file.

Format: **OPEN <filename>[<.ext>]**
{APPEND | DELETE | OUTPUT | INPUT}
AS <file number>

Complete the DELETE option when you want to completely overwrite an existing file.

Complete the APPEND option when you want to add data to an existing file.

Complete the OUTPUT option when you want to write to a new file.

Complete the INPUT option when you want to read from an existing file into a variable.

Example: The following commands open a file under number 1 for input, write a line of text to it, and close the file.

OPEN NEWF OUTPUT AS 1
WRITE 1 "This is a new file"
CLOSE 1

NOTE: File numbers cannot be used where a "to" device is expected; i.e., you cannot log to a file number.

PRINT

PRINT sends to the printer the quoted string or contents of the file specified in the print parameter.

Format: **PRINT {<text> |**
[<d:>][<path>]<filename>[<.ext>]}

Command Language

You may complete the **<text>** option with any quoted string not exceeding 80 characters. If you have configured a print buffer large enough for the file you wish to print, this command will provide print spooling.

The filename you specify must obey DOS rules.

Example: To make a printed copy of the contents of COMMAND.DOC, type the following on the command line:

PRINT COMMAND.DOC

Example: You can print a literal string as follows:

PRINT "This is a line.^M^J"

PTRANSMIT

PTRANSMIT sends a string to a host, including a carriage return, and then waits for either a host response or a period of time. A carriage return is automatically transmitted following the **<text>** parameter (a linefeed is included if **AUTO LF** is on).

PTRANSMIT is similar to TRANSMIT except that the TRANSMIT command does not include a carriage return after transmitting. Also, PTRANSMIT cannot be used to send PC files to the host. The PROMPT and PAUSE parameters are discussed separately below.

Format: **PTRANSMIT <text>**
[PROMPT [<ptext>]

The optional PROMPT parameter specifies that Reflection should test datacomm for the associated **<ptext>** after transmitting the text data. When the **<ptext>** parameter is omitted from the command line, Reflection waits for the host prompt character (see Terminal Configuration, Page 2).

The file transfer configuration parameter, **Receive timeout**, specifies the time allowed before Reflection will give up an unsatisfied wait and the error message **Timeout on receive** appears. This prompting mechanism should be used with host machines that begin each terminal read with a specific character – IBM/TSO and HP 3000 computers use "^Q". With line editors and command interpreters on DEC, DG, and Prime computers, a carriage return or carriage return/linefeed prompt can be used for approximate pacing.

Format: **PTRANSMIT <text>**
[PAUSE [<tenths>]]

The optional PAUSE parameter specifies that Reflection should wait the associated number of tenths of a second after transmitting the text data. If the **<tenths>** parameter is omitted from the command, the **Line xmit delay** is used.

If you transmit a string and don't specify either the PROMPT or PAUSE option, the PROMPT option is used. If there is no host prompt character, then the PAUSE option is used with the line transmit delay. However, if the value of the **Line xmit delay** configuration parameter is set to zero, the PROMPT option is used with a carriage return/linefeed prompt.

Example: This command sequence issues various host commands. Note that a carriage return is not included in the command strings because PTRANSMIT automatically includes a Return. Note also that all pacing considerations are handled by default.

PTRANSMIT "PURGE FILE1"
PTRANSMIT "PURGE FILE2"
PTRANSMIT "STREAM AJOB"

VAX examples follow:

PTRANSMIT "SHOW TERM"
PTRANSMIT "DEL FILE2.TXT"

QUIET

QUIET prevents information from appearing on the terminal as a command or command file is being executed.

Format: **QUIET {COMMAND | DISPLAY | STATUS} {ON | OFF}**

Information is suppressed only when QUIET is in the ON position.

To prevent information from being displayed on the command line, complete the COMMAND option in the quiet parameter.

To prevent the file transfer menu from being displayed, complete the STATUS option.

To prevent other information that would normally appear during a host session or during the execution of a command or command file (data that is being logged to a file, for example) from being displayed, complete the DISPLAY option.

You may select only one option at a time for the information parameter. If you want to suppress both DISPLAY and STATUS information, or DISPLAY and COMMAND information, use QUIET twice.

You may use the abbreviation, Q, instead of the command name, QUIET.

Examples: This command sequence disables the display of commands during the signon procedure.

Q COMMAND ON
INVOKE SIGNON
QUIET COMMAND OFF

READ

READ reads a line from a file opened for input into a variable. Data up to and including a linefeed or a maximum of 80 characters is read into the variable. There is no data editing.

Format: **READ <file number> V{0-9}**

Examples: In order to read from a file it must first be opened for input. Then data may be read into a variable.

OPEN NEW.TXT INPUT AS 1
READ 1 V1
CLOSE 1

READHOST

READHOST stores data received through the communications connection into a specified variable. Data is captured until a carriage return is received, unless an optional terminating character or list of characters is specified. A timeout may be included as well.

Format: **READHOST [<hh:mm:ss.cc>]**
V{0-9} [UNTIL <text>]
[LIMIT <n>]
[TERMINATOR V{0-9}]

When this command is executed, all data received from the host is stored until a carriage return is received. Note that data that results from a previous command may be included.

Example: The following lines execute the SHOWME command on an HP 3000. The first READHOST captures SHOWME as it is echoed back. The last line overwrites the contents of V1 with the first line of the executed command.

TRANSMIT "SHOWME^M"
READHOST V1
READHOST V1

Use a variable for each line you want to capture. If the received data exceeds 80 characters on a line, only the most recent 80 characters are retained. See the LIMIT option below.

If the optional TIMEOUT parameter is included, the command concludes when the time has elapsed, even though no terminator has been found. If the terminator is found, the condition name, FOUND, is set to *true*; otherwise it is *false*.

The optional **<text>** parameter specifies a list of terminating characters (not a string). If any one of these characters is detected, the command terminates.

If the **<text>** parameter is omitted, the carriage return is used as a termination character.

Example: In this example, all data received by the terminal is stored in **V2** until the first occurrence of a period, a carriage return, or a DC1.

READHOST V2 UNTIL ".^M^Q"

The LIMIT option allows you to capture only the specified number of characters from the host. Any characters which occur after the limit is reached are discarded. Otherwise, the last 80 characters are returned.

The TERMINATOR parameter allows you to indicate a variable which will contain the terminating character (Return in the default case). This is useful if several terminators are possible. For example:

READHOST V1 TERMINATOR V2
DISPLAY V2

The character which terminated the READHOST command is displayed.

RECALL

Recall the last stored user-key labels. See STORE.

Format: **RECALL**

This command performs the same function as Alt-F.

RECEIVE

RECEIVE transfers a file from a host computer to a PC, using Reflection's error-checking protocol.

Format: **RECEIVE <filename>[<.ext>]**
[FROM <hostfile> [;L]]
[ASCII | BINARY]
[APPEND | DELETE]

RECEIVE and its parameters may not exceed 80 characters and must be typed as one line.

The PC filename must obey DOS rules and may include a drive/path specification. The host filename must obey rules that apply on the host. Both PC and host filenames can be literal phrases or string expressions.

If **FROM <hostfile>** is omitted, the host filename is assumed to be the same as the PC filename.

No global characters (*wildcards*) are permitted in the filename specifications.

You must complete the APPEND or DELETE parameter if the file exists. Use APPEND when you are adding a file to an existing file, and use DELETE when you want to delete any existing PC file with the same name.

It is advisable always to complete the ASCII or BINARY option, but especially when using command files. The current file transfer value is used if this parameter is omitted.

You may use the abbreviation, R, instead of the command name, RECEIVE.

The ;L option is only available with HP 3000 hosts, and should be used when receiving a file which must be returned to the host with all of its label information intact.

Example: The following command transfers the file HOSTFILE from the host to the B drive of the PC.

R B:MY.TXT FROM HOST.PUB ASCII

RENAME

RENAME changes a file's specifications – either its filename, or extension, or both. Specifically, it changes the name of the file in the first parameter to the name specified in the second parameter.

Format: **RENAME [<d:>][<path>]<filename>[<.ext>] [<path>]<filename>[<.ext>]**

Filenames must obey DOS rules. No global characters (*wildcards*) are permitted in the filename specifications. See FOR on page TR-189.

If you specify a drive which is not the default, you must also use the same drive specifier with the new filename.

Unlike DOS, the path may be specified in both filenames, so RENAME may be used to switch a file to another directory.

Example: The following command renames the file, OLDFILE, on drive B, to NEWFILE, and also moves it to drive A.

RENAME B:OLDFILE A:NEWFILE

You may use the DOS abbreviation, REN, instead of the command name, RENAME.

RESTORE

RESTORE restores PC files from a host backup file. See page UM-165 for more information.

Format: **RESTORE [<hostfile>] [<d:>][<path>] <filename>[<.ext>] [/S][/K][/H]**

/S Restores all subdirectory files.

/K Requests that files modified since the last backup not be overwritten.

/H Restores hidden files.

Example: To restore all files from the default backup file, BACKUP, use the following command:

RESTORE C:*.* /S

RETURN

RETURN exits an invoked file and returns to the file or program which called the current file.

Format: **RETURN**

SAVE

Saves the current configuration to a disk file. If the filename is omitted, the default is used.

Format: **SAVE [<filename>[<.ext>] [DELETE]]**

SEND

SEND transfers a file from a PC to a host computer using Reflection's error-checking protocol. Supported hosts are discussed separately below. You may use the abbreviation, S, instead of the command name, SEND.

The PC filename must obey DOS rules and may include drive/path specifications. The host filename must obey host computer rules. Both PC and host filenames may be literal phrases or string expressions; e.g., **SEND V1 TO V2** is acceptable. No global characters (*wildcards*) are permitted in the filename specifications. See FOR on page TR-189.

If **TO <hostfile>** is omitted, the host filename is assumed to be the same as the PC filename.

Complete the DELETE option to overwrite an existing host file.

HP 3000 Hosts Syntax for HP 3000 file transfers follows:

Format: **SEND <filename>[<.ext>]**
[TO <hostfile>[;P][;L][;F][;Q]]
[ASCII | BINARY]
[DELETE] [REC=<number>]

Use **;P**, the ;Purge option, as part of the filename (instead of using the parameter DELETE) to completely overwrite an existing HP 3000 file, including file characteristics. This is especially important if the existing file is smaller than the one being transferred. See page UM-129.

To maintain HP 3000 label information use **;L**, the ;Label option. This is required for programs which are transferred to the PC and then back to the host to be executed. See page UM-129.

To produce a fixed-length file on the HP 3000, use the **;F**, or ;Fixed option. This is required for files such as *Charting Gallery* graphics files. See page UM-130.

If you are using QEDIT on the HP 3000, you can use **;Q** to ensure that work files are transferred to the host in the correct format for QEDIT. The ;Q also maintains label information so that files transferred to the PC and then to the HP 3000 retain their correct attributes. Reflection recognizes QEDIT files which are transferred to the PC without ;Q, and treats them as standard ASCII files. See page UM-105.

Example: The following command sends a fixed-length binary PC file to the HP 3000.

SEND SAMP.DAT TO GRAPH;F BINARY

VAX Hosts

If you are transferring files to a VAX/VMS host with VAXLINK, the special options differ from the HP 3000 parameters.

Format: **SEND <filename>[<.ext>]**
[TO {<hostfile> | <string>}]

[ASCII | BINARY]
[DELETE] [REC=<number>]

Two host filename switches can only be used during **ASCII** file transfer: /C causes the file on the VAX to be submitted as a batch DCL file upon completion of the transfer. /S causes the file on the VAX to be submitted to the spooler for printing upon completion of the transfer.

The following host filename switches can only be used during **binary** file transfers: /F causes the file on the VAX to be created with fixed-length records. The record length is that which is specified in the host record size field. /I allows a user to move a file from the VAX to a PC and back again while still allowing the file to be usable by VAX applications. /V causes the file on the VAX to be created with 512-byte, fixed-length records, the format needed for an .EXE file.

See page UM-136 for more information.

Example: The following command sends a binary, fixed-length PC file to a VAX/VMS host with a record size of 80.

SEND COMMAND.COM TO
MSDOS /F BINARY REC=80

UNIX and DOS Use the following syntax when transferring files to UNIX hosts or PCs running under DOS:

Format: **SEND <filename>[<.ext>]**
[TO <hostfile>]
[ASCII | BINARY] [DELETE]

It is advisable to always use ASCII with these hosts. If you are using the SEND command in command files include this option to ensure the correct setting; the current value is used otherwise.

Example: The following command sends an ASCII PC file to a UNIX host, and uses the PC filename on the host.

SEND CREATE.TXT ASCII

SET

SET is used to set various communication parameter values. Some are grouped below according to configuration menus. If the parameter doesn't appear on a configuration menu, it is discussed after the tables which follow.

NOTE: SET command values are saved when a configuration file is saved unless otherwise indicated, even if that value does not appear on a configuration menu.

VERIFY lists these commands alphabetically. See escape sequences for other configuration setting strings. Also see the VALUE operator on page TR-234.

Format: **SET <parameter name> <parameter value>**

Use the following parameter names and values:

Table A.1
Datacomm Configuration Commands

BAUD	*110 - 19200*
CHECK-PARITY	*YES | NO*
CONNECTION	*DIRECT | MODEM*
CTS-REQUIRED	*YES | NO*
DATACOMM-PORT	*<datacomm value>*
DSR-REQUIRED	*YES | NO*
ENQ-ACK	*YES | NO*
PARITY	*NONE | EVEN | ODD | 0s | 1s*
RECEIVE-PACING	*NONE | XON/XOFF*
SESSION#	*1-255*
STOPBITS	*1 | 2*
TRANSMIT-PACING	*NONE | XON/XOFF*

Table A.2
Terminal Configuration Commands

CHARACTER-DELAY	*0-225*
HOST-PROMPT	*(ASCII character in quotes; e.g., "^Q")*
LABEL-LINES	*NONE | 1 | 2*
LINE-DELAY	*0-225*
TERMINAL-TYPE	*<terminal>*
TYPE-AHEAD	*YES | NO*

Table A.3
Printer Configuration Commands

PRINTER-CONTROL	*(***Control Codes values***)*
PRINTER-INTERFACE	*<value>*

Table A.4
Global Configuration

HOT-KEY	*(see below)*
STRIP-TEXT	*YES | NO*
TERMINAL-CLASS	*HP | DEC*

(unavailable after background)

Table A.5
File Transfer Commands

BLOCK-SIZE	*64 - 512*
CR/LF=SEPARATOR	*YES | NO*
CTRL-Z=EOF	*YES | NO*
DELETE-TRAILING-SPACES	*YES | NO*
HOST-STARTUP	*(string in quotes)*
ISO7-TO-ROMAN8	*YES | NO*
ROMAN8-TO-ISO7	*YES | NO*
RECEIVE-TIMEOUT	*1 - 9999*
RETRY-LIMIT	*1 - 9999*
SEPARATOR=CR/LF	*YES | NO*
SPACES-TO-TABS	*YES | NO*
TABS-TO-SPACES	*YES | NO*
WRITE-CTRLZ	*YES | NO*

The following SET commands do not appear on any configuration screens. All but VIDEO-BUFFER and STRIP-NULLS-DELS are saved when a configuration file is saved. VERIFY entered on the command line allows you to check the settings for all SET commands, or create a command file composed of SET commands.

ALERT

When Reflection is operating is background mode, messages are sent to the foreground screen concerning errors or the completion of file transfers. If you do not want Reflection messages displayed, set ALERT to *NO*. *YES* is the default.

Values: YES* | NO

AUTO-ALPHA

With AUTO-ALPHA set to *YES*, the graphics display adapter (such as the EGA) is switched out of graphics mode when Reflection's graphics display is turned off. Reflection's graphics display is toggled on or off by escape sequence or by Alt-7 (keypad). The *YES* setting makes Reflection scroll much faster when the graphics display is off, but the screen does flash during mode changes. If the flashing is irritating, try changing AUTO-ALPHA to *NO*. Alt-8 (keypad) can be used to force the display to alpha mode if necessary.

Values: YES* | NO

BREAK-LENGTH

This command allows you to adjust the length of the break signal. Breaks entered by pressing Break or with the BREAK command are governed by this setting. Initially, breaks are 200 milliseconds long. Use SET BREAK-LENGTH <number>, where **<number>** determines the number of milliseconds of the break up to 999.

Values: 0-999
Default: **200**

CAPTURE

When CAPTURE is set to YES, all data coming from the host is logged to a disk file, including escape sequences and control codes. Flow control such as XON/XOFF is omitted. You must enter the disk filename from the "to" device labels by pressing TO DISK, or use the OPEN <filename> command to open a file. It is not necessary to toggle TO PRINTER off. All data will be written to the file until the file is closed by pressing TO DISK again.

This command is typically used for debugging purposes. It can also be used for capturing graphics plotting sequences.

Values: YES | NO*

DISABLE-COMP-CODES

DISABLE-COMP-CODES disables or enables completion codes. It should be used with caution. SET DISABLE-COMP-CODES YES *never* returns a completion code. SET DISABLE-COMP-CODES NO, the default, *always* returns a completion code. Reflection and printer/plotter escape sequences always return completion codes.

Values: YES | NO*

DISABLE-INTERRUPT

Under the default, *NO*, [Ctrl]-[Y], [Ctrl]-[A], [Ctrl]-[S] will abort the execution of a command file. Reflection can prevent the interruption of a command file if DISABLE-INTERRUPT is set to *YES*.

Values: YES | NO*

DISCONNECT-ON-EXIT

The host connection is not normally broken when you exit Reflection. If this value is set to *YES*, a disconnect will be sent when Reflection is exited. The default is *NO*. This may, depending on your host, log you off automatically whenever you exit Reflection or disconnect you from the LAN; see page TR-243. The SHELL command and HOT-KEY are not affected by this setting.

Values: YES | NO*

DO-FORM-FEEDS

Reflection allows you to configure whether form feeds entering display memory are kept. This controls logging and COPY operations which send data to the "to" devices. The default is *NO*, which causes form feeds to be discarded as they enter display memory.

If you want form feeds in host data to be kept and passed on to the printer or a disk file, use SET DO-FORM-FEEDS YES.

Values: YES | NO*

EXITS-DISABLED

When set to *YES*, [Alt]-[R] (hard reset) and [Alt]-[X] (exit to DOS) are disabled. Only the EXIT command (unless Reflection has been installed as a pop-up) or [Ctrl]-[Alt]-[Del] exits Reflection.

Values: YES | NO*

LITERAL-ESCAPE

In order to change the escape character from a backslash (\), use this set command with any quoted character. See page TR-165 for an example of how this character is used. SET LITERAL-ESCAPE "^@" sets the escape character to NULL. To set the escape character back to a backslash, use SET LITERAL-ESCAPE "\".

Values: "<any character>"

MAX-XOFF-TIME

MAX-XOFF-TIME 22 specifies that only 22 seconds should elapse before receiving an XON after receiving an XOFF. If an XON is not received within this time limit, Reflection will assume one has been sent. This command is recommended in unattended operations involving file transfers to host systems which use XON/XOFF pacing. (The default value of 0 means Reflection will wait unconditionally for an XON.)

Values: 0 - 9999
Default: **0**

POPUP-ONLY

This SET command affects the function of the hot-key. To make Reflection act like a simple pop-up program and disable background multiprocessing, set this value to *YES*. The default allows background processing. *YES* may be necessary to provide compatibility with other memory-resident programs.

Values: YES | NO*

QUIET-<parameter>

A series of commands are provided so that the current setting for the QUIET command can be checked with the text operator VALUE. The value may also be set with these commands; they are identical to the

QUIET command keyword.

QUIET-COMMAND YES/NO*
QUIET-DISPLAY YES/NO*
QUIET-STATUS YES/NO*

REC-MODE-CLOSES-DISK

This SET command determines whether a disk file remains open or closed when data is sent to disk and record mode ends. Reflection will always close the file, or always leave it open, depending on the setting of REC-MODE-CLOSES-DISK. *YES* will close the file when record mode finishes, and *NO* will leave the file open. All open files are closed when Reflection is exited, regardless of how this command is set.

Values: YES* | NO

RETURN=ENTER

If you prefer to use Return rather than Enter in block mode applications, set RETURN=ENTER to *YES*. This setting may cause problems if your host system expects a Return and you have configured Return to act as Enter. The default is *NO*, which means that only the keypad + key or Shift-F10 may be used as Enter.

Values: YES | NO*

SCROLL-LOCK=STOP

ScrollLock may be configured to act as the Stop key found on HP 239x series terminals by setting this value to *YES*. **Receive pacing** must be set to *XON/XOFF*. The default is *NO*. See page UM-40.

Values: YES | NO*

STRIP-NULLS-DELS

NULLS and DELS are usually stripped from files during transfers to or from the host. To disable this function use **SET STRIP-NULLS-DELS NO**. The default is *YES*. This value is not saved when a configuration file is saved; it must be reset when Reflection is loaded again.

Values: YES* | NO

SUB-BLOCK-LENGTH

Some UNIX-based systems require that a SUB-BLOCK-LENGTH be set. It can range from 40 to 240. This allows the host to flush any internal input buffer, and send the data to the file transfer software before the buffer overflows. Then the **Block size** value can often be set to 512.

Values: 0 - 256
Default: **0**

TRACE

When TRACE is set to YES, all data coming from the host or going to the host is logged to a disk file, including escape sequences and control codes. ENQ/ACK flow control is included but XON/XOFF is omitted. You must enter the disk filename from the "to" device labels by pressing TO DISK, or using the OPEN <filename> command to open a file. It is not necessary to close TO PRINTER. All incoming data will be written to the file until the file is closed by pressing TO DISK again.

This command is typically used for debugging purposes.

Values: YES | NO*

VERTICAL-FILL

To prevent gaps in the mapping of vertical lines in graphics mode, set this command to *YES*. Lines next to gap areas are drawn with a width of two dots under this setting, eliminating gaps. The default setting is *NO*. Graphics text characters may be adversely affected when vertical fill is on.

Values: YES | NO*

VIDEO-BUFFER

Use this SET command to specify the size of the buffer for saving the screen image of the program you are toggling in and out of with the hot-key. Space for the buffer is taken from the **Total available memory** when the hot-key is first used. The default value of 8K works with any alphanumeric display. 16K is sufficient for the IBM CGA in graphics mode. This value is not saved when a configuration file is saved; it must be reset when Reflection is loaded again.

Values: 8K, 16K, 32K, or 64K
Default: **8K**

XOFF-REPEAT

If characters are still received from the host after an initial XOFF, an XOFF will be repeated. This command allows you to define the number of characters received between XOFFs. The range is 0 to 9999. The default is 32, which means that only one XOFF will be sent until the receive buffer completely fills up. If 9999 is set only one XOFF will *ever* be sent.

Values: 0 - 9999
Default: **32**

Example: The following commands initialize file transfer parameters:

SET HOST-STARTUP "RUN PCLINK"
SET BLOCK-SIZE 116
SET HOST-PROMPT ""
SET RECEIVE-TIMEOUT 10
SET RETRY-LIMIT 5
SET TABS-TO-SPACES NO

The parameters **Host prompt character**, **Character xmit delay**, and **Line xmit delay** are explained under the HP Terminal Configuration, Page 2 menu. All of the other parameters are explained in *File Transfer Configuration* on page TR-88 or *Changing Global Characteristics* on page TR-97.

HP 2624B, HP 2393A, HP 2394A, HP 2397A
SET commands follow. See *Format Mode* on pages TR-269 and page TR-284.

AUTOTERM

Sets the 2624B feature *autoterm* on or off. The default is *NO*.

CLEARTERM
Sets the 2624B feature *clearterm* on or off. The default is *NO*.

TRANSMIT
Sets the 2624B feature *transmit* to *ALL* or *MODIFIED*. The default is *ALL*.

SHELL

SHELL allows you to issue DOS commands or run other programs from within Reflection.

Format: **SHELL [<command name> [<parameters>]]**

Memory must be available in your PC for SHELL to work properly; see page TR-100. About 30K is sufficient for DOS 2.1 commands.

NOTE: SHELL is disabled once Reflection has been placed in background mode.

SHELL entered at the command line exits Reflection and places you at the DOS prompt. Typing **EXIT** returns you to Reflection. SHELL entered with a DOS command executes the command, and then returns to Reflection.

SHELL entered with a PC program name executes the program. You are placed at the program prompt. When you are finished, the program's exit command returns you to Reflection.

NOTE: Do not **load** memory resident programs from within Reflection. The format of the SHELL command allows this, but it drastically reduces the amount of memory available.

STOP

STOP stops the execution of a command file. An encountered occurrence of STOP in a command file prevents anything following that command from being executed. It also automatically returns Reflection to terminal mode.

Format: **STOP**

STOP has the same effect programmatically that pressing [Esc] from the command line has directly.

Example: In this example, command file execution ceases if the word *CONTINUE* is not received within 5 seconds.

```
WAIT 0:0:5 FOR "CONTINUE"
IF NOT FOUND
    STOP
ENDIF
```

STORE

STORE saves the current user-key values for later recall.

Format: **STORE**

Reflection stores user-key values in configuration files. When STORE is issued on the command line or in a command file it keeps the current user-key values for later recall. These user-key definitions are not written to disk. They can be recalled by using either the RECALL command or [Alt]-[F].

TRANSFER

This command allows the host to request information from the PC running Reflection. As the PC can be in any mode when the request is made, TRANSFER does a *type 3* block transfer to ensure that the correct handshaking occurs between the host and PC. For example, a host might want to change a Reflection configuration value and then reset it at a later time.

EC&oCTRANSFER VALUE(TYPE-AHEAD)CR

Reflection responds with the current TYPE-AHEAD value, which can be stored so that a Reflection SET command can reset the original value.

TRANSMIT

TRANSMIT sends data in the form of text or files to a host without the use of an error-checking protocol. If none of the supported protocol options is available to you for file transfers, try this method of transferring files. See page UM-153 for more information. Various transmission pacing options are included for transmitting files. The three possible formats are discussed separately below.

Format: **TRANSMIT <text>**

The **<text>** parameter must be enclosed in quotes and may not exceed 80 characters. The options PROMPT and PAUSE cannot be used with the **<text>** parameter, only with the **<filename>** parameter.

Format: **TRANSMIT <filename>[<.ext>]**
[PROMPT [<ptext>]]

The PROMPT option specifies that Reflection should test datacomm for the associated **<ptext>** after transmitting the text data. If the **<ptext>** parameter is omitted from the command line, Reflection waits for the host prompt character (see the Terminal Configuration, Page 2 menu).

The file transfer configuration parameter, **Receive timeout**, specifies the time allowed before Reflection gives up an unsatisfied wait and the error message, **Timeout on receive**, appears. This prompting mechanism should be used for host machines that begin each terminal read with a specific character (IBM/TSO and HP 3000 computers use $\boxed{\text{Ctrl}}$-$\boxed{\text{Q}}$).

Format: **TRANSMIT <filename>[<.ext>]**
[PAUSE [<tenths>]]

The PAUSE option specifies that Reflection should wait the associated number of tenths of a second after transmitting the text data. If the **<tenths>** parameter is omitted from the command, the **Line xmit delay** (see the Terminal Configuration, Page 2 menu) will be used. To transmit data with no pacing, use PAUSE 0.

If you transmit a file and don't specify either the PROMPT or PAUSE option, the PROMPT option is used. If there is no host prompt character, the PAUSE option is used with the line transmit delay. However, if the value of the **Line xmit delay** configuration parameter is set to zero, the PROMPT option is used with a carriage return/linefeed prompt. When a **TRANSMIT <filename>** is initiated from the host, Reflection appends a Ctrl-Z character to the end of the transmitted file.

Example: The first line transmits a text line and the second transmits a file with pacing.

TRANSMIT "Hello!"
TRANSMIT B:MYFILE.TXT PAUSE 5

TYPE

TYPE displays on the screen the contents of the file specified in the filename parameter.

Format: **TYPE [<d:>][<path>]<filename>[<.ext>]**

The filename must obey DOS rules. Global filename characters (*wildcards*) are not permitted in the filename specification or extension.

If you do not complete the drive and path parameters, Reflection searches the current directory path of the default drive for the specified filename.

You may use the abbreviation **T** instead of the command name **TYPE**.

Example: To display on the screen the contents of the file COM.DOC on the disk in drive B, type on the command line:

TYPE B:COM.DOC

VERIFY

VERIFY displays the values of all the parameters that can be set using the SET command. You may use V.

Format: **VERIFY [COMMANDS | <SET parameter>]**

If you specify a particular SET parameter after the VERIFY command, the value for that parameter is listed.

The COMMANDS option allows you to generate SET commands with the VERIFY command. All current values will be displayed in the format SET <parameter> <value>.

Example: The following command displays the current baud rate setting:

VERIFY BAUD

WAIT

WAIT orders Reflection to suspend operations until a specified time or information condition has been met. See the HOLD command.

Format: **WAIT {[[UNTIL]<hh:mm:ss.cc>] [FOR {<text> | <hh:mm:ss.cc> SILENCE}]}**

WAIT cannot be used by itself; you must complete one or more of the parameters that follow it. UNTIL is optional.

The parameter **<hh:mm:ss.cc>** stands for hours, minutes, seconds, and hundredths of a second. Omit UNTIL if you want Reflection to suspend operations for a certain period of time. Use the UNTIL option if you want the system to wait until a given clock time before proceeding with whatever operation it is performing.

The FOR option instructs Reflection to wait for a specific response to come through the communications port before proceeding. Complete the **<text>** parameter if you want the system to suspend operations until a given password, filename, message, or other block of text is received. Complete the SILENCE option to instruct Reflection to wait for a period of silence before proceeding with its operations.

The FOR clause of the WAIT command determines the value of the condition, FOUND. Initially, WAIT sets the value to *false*. If and when the condition specified in the FOR clause is satisfied, WAIT sets the value to *true*.

Example: These commands attempt to dial (via Hayes modem) a remote computer. If a connection is not made after 6 seconds, an error message is displayed.

```
TRANSMIT "ATDT12068345555^M"
WAIT 0:0:6 FOR "CONNECT"
IF NOT FOUND
    DISPLAY "CALL NOT COMPLETED"
    STOP
ENDIF
```

WRITE

You may write a text string or a line from a variable into a file opened for output. There is no data editing. See the OPEN command.

Format: **WRITE <file number>**
{V{0-9} | <text in quotes>}

Example: The following lines open a file, write a line of text and the contents of a variable to it, and close the file.

```
OPEN TEST.TXT OUTPUT AS 1
WRITE 1 "The following text is from variable 2"
WRITE 1 V2
CLOSE 1
```

XRECEIVE

XRECEIVE receives a file from the host using XMODEM protocol. The filename may be a string expression. See page UM-147.

Format: **XRECEIVE <filename>[<.ext>]**
[APPEND | DELETE]

Note that you must complete the APPEND or DELETE parameter if the file exists. XR may be used instead of XRECEIVE.

XSEND XSEND sends a file to the host using XMODEM protocol. The filename may be a string expression. See page UM-147.

Format: **XSEND <filename>[<.ext>]**

XS may be used instead of XSEND.

Operators

Obtaining True/False Values

Some operators have been provided which indicate true/false values of variables; these are called *Boolean* values.

DCD
The DCD operator requests data carrier detect. It is typically used with an IF statement.

EOF (<file number>)
The EOF operator can be used in an IF statement to indicate the end of the file. The file number must be enclosed in parentheses and refer to a currently open file. It uses an integer as an input string and a *true/false* value as an output. *True* means that the indicated file is at its end. For example, EOF(1) is true if the end of file number 1 has been reached.

ERROR
The ERROR operator is used with an IF statement. It can be used to indicate the action to be taken in case an error is encountered in the processing of a command. The value *true* indicates that the previous command has failed.

EXIST("<string>")
The EXIST operator is used to indicate the existence of a DOS file. For example, EXIST("R1.COM") verifies the existence of R1.COM. EXIST (V1) may also be used.

FGD
The FGD operator tests to see if Reflection is in foreground

mode. *True* indicates Reflection is currently in the foreground.

FOUND

The FOUND operator tests whether the WAIT or READHOST condition has been met. *True* means that the indicated text was found.

Text String Operators

Reflection includes a number of text string operators that form strings and numbers; they can be used in any context where text variables are allowed.

& The & operator returns a string that is the joining of a pair of strings.

Format: **<string1> & <string2>**

The resulting string is limited to 80 characters.

Example:

DISPLAY "HELLO " & "WORLD" & "!"

The result of this display is **HELLO WORLD!**.

$DATE $DATE may be used as a variable alone or with other operators. The format is MMDDYY.

Examples:

DISPLAY $DATE
DISPLAY MID($DATE,4,5)

The result returned for the first would be the current PC date. The second example would display the day of the current system date.

$TIME $TIME may be used as a variable alone or with other operators. The format is hours, minutes, seconds.

Examples:

DISPLAY $TIME

DISPLAY MID($TIME,4,5)

The result returned for the first would be the current PC time. The second example would display the minutes only.

$SERIAL This reserved variable is used to provide information on the Reflection product being used. There are 6 parts to the 14-digit field.

x00-VVVySSSSSS

x	Provides information on the type of personal computer being used; e.g., **I** for IBM or **H** for HP 150.
00	The product number (01 for Reflection 1)
-	A + in this field indicates a PLUS version.
VVV	The version number of the product.
y	The type of environment that this product supports: **I** for individual, **L** for LAN, and **S** for server.
SSSSSS	A six digit-code for the unique serial number of the product.

The following example displays the unique number identification for the product (SSSSSS):

DISPLAY MID($SERIAL, 9, 14)

LOWER The LOWER operator returns a string where all upper case characters are converted to lower case.

Format: **LOWER(<string>)**

Only upper case characters are changed.

Example:

DISPLAY LOWER("123aABC!")

The result of this display is **123aabc!**.

MID The MID operator extracts characters from within a given string by specifying the starting and ending positions.

Format: **MID(<string>,<start>,<end>)**

The **<start>** parameter must not be zero.

If **<end>** is negative or exceeds the length of the string, then the actual length will be used instead.

Example:

DISPLAY MID("123aABC!",3,5)

The result of this display is **3aA**.

PACK The PACK operator produces a string with all leading, trailing, and redundant spaces removed.

Format: **PACK(<string>)**

Example:

DISPLAY PACK(" AX B C   ")

The result of this display is **AX B C**.

UPPER The UPPER operator returns a string in which all lower case characters are converted to upper case.

Format: **UPPER(<string>)**

Only lower case characters are changed.

Example:

DISPLAY UPPER("123aAbC!")

The result of this display is **123AABC!**.

VALUE The VALUE operator returns a string indicating the value of the specified *SET* parameter.

Format: **VALUE (<SET-parameter>)**

Example:

DISPLAY VALUE (TRANSMIT-PACING)

The value displayed will be the current TRANSMIT-PACING setting.

Integer Operators

Reflection includes a number of integer operators that form numbers which can be used in any context where text variables are allowed.

ERROR-CODE The ERROR-CODE operator returns the error code associated with the last command processed. A value of **0** indicates successful completion. See page TR-236 for a complete list of error-codes.

Example:

```
IF ERROR-CODE=10
    GOTO end
ENDIF
```

Note that if the line immediately following the command which produced an error is a comment line, the error-code value is reset to 0.

FIND The FIND operator finds the first occurrence of string1 within string2 and returns the position at which the match is found, expressed as a numeric value.

Format: **FIND(<string1>,<string2>)**

If the string is not found, the result is zero.

Examples:

LET V1 = FIND("AB","XYAABC")

DISPLAY V1

The result of this display is the value 4.

LET V1 = "C:DOC.TXT"
LET V2 = MID(V1,3,FIND(".",V1) -1)
DISPLAY V2

The result of this display is *DOC*.

LENGTH

The LENGTH operator returns the number of characters in a string.

Format: **LENGTH(<string>)**

Blanks and control characters are included in the count of the number of characters.

Example:

DISPLAY LENGTH("Your move:")

The result of this display is the value 10.

Command Language

Error Code Summary

The following error codes may be referenced with the ERROR-CODE Boolean:

0	Normal completion of any command.
1	Transmit error. CTS or DSR required; if modem connection, no DCD present.
2	EOF on file read.
3	Timeout while waiting for expected or specified data.
4	Received unexpected data from host.
5	File transfer aborted by user.
6	Fatal block number sequence error on a file transfer.
7	Received a nonprocessable message during file transfer.
8	Data message received during file transfer was not the correct packet type.
9	File transfer aborted by host program.
10	Local file does not exist.
12	PC file transfer software is incompatible with the host file transfer software.
13	DOS disk error detected.
14	Too many consecutive naks received or sent during a file transfer.
15	Could not create local file.
16	Could not execute host file transfer program.
17	User requested termination of file transfer with **STOP** (not **ABORT**).
18	Local filename already exists.
19	Unexpected error detected.
97	Encryption version mismatch.
101	No files were found to back up.

103 Backup terminated by user.

104 Backup terminated due to error.

160 Bad command.

161 Bad syntax.

162 Error detected while executing command.

163 Encryption version mismatch.

HP Keyboard Functions

Quick Reference

Table B.1
HP Keyboard Functions

Function	*Keystroke*
Break	*Ctrl*-*Break*
Clear alpha display	*Alt*-*J*
Clear line	*Alt*-*K*
Clear print buffer	*Ctrl*-*F1*
Command line	*Alt*-*Y*
Config keys	*Alt*-*C*
Define user keys	*Ctrl*-*F9*
Delete character	*Del*
Delete line	*Alt*-*D*
Disconnect	*Ctrl*-*F10*
Enter	+ *(keypad) or* *Shift*-*F10*
Error recap	*Alt*-*E*
Exit to DOS	*Alt*-*X*
Extend key	*Alt*-*Z*
Function key recall	*Alt*-*F*
Hard reset	*Alt*-*R*
Help	*Alt*-*H*
Home down	*Ctrl*-*End*
Home up	*Ctrl*-*Home*
Insert character	*Ins*
Insert line	*Alt*-*I*
Keypad	*NumLock*
Modes keys	*Alt*-*M*

Function	*Keystroke*
Move cursor down	*Down*
Move cursor left	*Left*
Move cursor right	*Right*
Move cursor up	*Up*
Move to bottom row	*Ctrl*-*PgDn*
Move to line end	*End*
Move to line start	*Home*
Move to top row	*Ctrl*-*PgUp*
Next page	*PgDn*
Previous page	*PgUp*
Print screen	*Alt*-*PrtSc*
Print screen with labels (DOS)	*Shift*-*PrtSc*
Reset handshake	*Alt*-*Q*
Reset Hayes internal modem	*Ctrl*-*F8*
Reset typeahead	*Alt*-*T*
Roll down	*Ctrl*-*Up*
Roll left	*Ctrl*-*Right*
Roll right	*Ctrl*-*Left*
Roll up	*Ctrl* *Down*
Select key	*PrtSc*
Soft reset	*Alt*-*S*
*Stop key**	*ScrollLock*
System keys	*F10*
Toggle user labels	*F9*
User keys	*F9* or *Alt*-*U*

Function Description

Break (Ctrl-Break)

Esc sequence: ^{E}c&oCBREAK^{C}R
Availability: Always

Transmits a break signal of approximately 200 milliseconds to the host computer, assuming that the PC is connected to a host. The break signal is sent even if Reflection is in local mode.

---

* Only when **Receive pacing** is XON/XOFF and SET SCROLL-LOCK=STOP is *YES*.

The length of the break signal can be configured; see page TR-216.

This use of Ctrl-Break differs from some other programs' break function. Many programs use this key combination to exit to the DOS command prompt. Compare *Exit to DOS*, (Alt-X). Note that ScrollLock on the PC keyboard is Break without Ctrl.

Clear alpha display (Alt-J)

Esc sequence: ^{E}cJ
Availability: In terminal mode

Erases all of the alpha display memory beginning with the alpha cursor position. In format mode, only the characters within unprotected fields are erased, beginning with the alpha cursor location.

Clear line (Alt-K)

Esc sequence: ^{E}cK
Availability: In terminal mode, the user-key definition menu, any file transfer menu and the command line

Erases all characters from the alpha cursor through the end of the line. In format mode, this key works only if the alpha cursor is within an unprotected field. All characters from the alpha cursor position through the end of the field are erased.

Clear print buffer (Ctrl-F1)

Esc sequence: None
Availability: Always

Clears the contents of Reflection's internal printer buffer. The printer continues to print if it has a built-in buffer. To stop sending data to the printer buffer, use the Soft Reset (Alt-S) function.

Command line (Alt-Y)

Esc sequence: None
Availability: In terminal mode

Causes Reflection to enter command mode and display the command entry line. Exit command mode by pressing [Esc], or by leaving the command line blank and pressing [Return].

Config keys ([Alt]-[C])

Esc sequence: None
Availability: Always, except when disabled by the ^{E}c&jS escape sequence, or when the configuration is locked

Accesses configuration keys. If the selection of system, modes, and user keys has been disabled by the escape sequence ^{E}c&jS, pressing this key causes a beep and the function is not performed.

Define user keys ([Ctrl]-[F9])

Esc sequence: ^{E}cj
Availability: In terminal mode

Brings up the user-key menu for you to define user keys.

Delete character ([Del])

Esc sequence: ^{E}cP
Availability: In terminal mode, the user-key definition menu, file transfer menus and the command line

With format mode off, [Del] deletes the character at the alpha cursor position and moves all following characters on the line one position to the left.

When the alpha cursor is to the left of the right margin, only the characters up to the right margin are moved left, and a space is inserted at the right margin.

When the alpha cursor is to the right of the right margin, all characters to the right of the alpha cursor are moved left, and the line is shortened by one column. This also applies to characters that cannot be displayed, such as display enhancements and unprotected field delimiters. When the alpha cursor is at the first column of an unprotected field, [Del] changes the field from unprotected to protected. When the alpha cursor is at the first column of a display enhancement, however, the display enhancement is not changed.

With format mode on, [Del] operates only within an unprotected field. Further, only characters that can be displayed within the unprotected field are affected. The size of the field remains unchanged, and any display enhancements or character set changes within the field are unchanged.

Delete line ([Alt]-[D])

Esc sequence: ^{E}cM
Availability: In any terminal mode, except format mode

Deletes the alpha cursor line and moves all subsequent lines in alpha display memory up one row. The alpha cursor then moves to the left margin.

Within a locked area of the screen, the size of the locked area is not changed; instead, the alpha cursor line is deleted and the line immediately following the locked area is moved up into the locked area.

Disconnect ([Ctrl]-[F10])

Esc sequence: ^{E}cf
Availability: During datacomm, when the PC and the host are connected

Disconnect functions differently depending on the type of host connection. If you are connected via modem, Reflection lowers the Data Terminal Ready signal (DTR) for 2 seconds. This function disconnects your modem if it disconnects when DTR is lowered. See *Using an Auto-Dial Modem with Reflection* on page UM-55.

If you are connected via LAN, the appropriate LAN disconnect signal is put into effect with this keystroke. See page TR-22. Also see DISCONNECT-ON-EXIT in the SET command section on page TR-217.

Enter ([+] (keypad) or [Shift]-[F10])

Esc sequence: ${}^{E}C$&oCENTER${}^{C}R$
Availability: Always

In remote mode, [Enter] transfers a block of data to the host. The specifications of this block depend on various mode settings such as *block mode, format mode,* and *line/page* mode. If you are operating in character mode, the current line is transferred. If you are operating in block mode, the current line or all pages in memory depending on how the **Line/Page** field on Terminal Configuration, Page 1 is set.

In local mode, [Enter] homes the cursor and sends all of display memory to a local printer.

Error recap ([Alt]-[E])

Esc sequence: None
Availability: In terminal mode (or from file transfer)

Displays a menu of errors that can occur in data communication. A count is displayed for each type of error on the current datacomm port. See page UM-201 for instructions for using this screen to diagnose and correct datacomm problems.

Exit to DOS ([Alt]-[X])

Esc sequence: ${}^{E}C$&oCEXIT${}^{C}R$
Availability: Always

Terminates Reflection and exits to DOS. The host session is not terminated. (See DISCONNECT-ON-EXIT on page TR-217.) If the PC is online over a modem, Reflection does not automatically hang up the phone. To hang up before exiting Reflection, press [Ctrl]-[F10], disconnect.

Some host systems will automatically disconnect when you log off. With an external modem, notice that the Carrier Detect (CD) light goes off when you log off such a host system.

If you are in background mode, [Alt]-[X] will unload Reflection. You should only attempt this when Reflection is in the foreground and no other programs are in background. It is best when first loading Reflection to load it before most

PC applications so that when you unload Reflection you free up the maximum amount of memory.

Extend key (Alt-Z)

Esc sequence: None
Availability: Whenever data can be entered

Gives access to the Roman 8 character set as shown on page TR-328. Also see *National Keyboards and Characters* on page UM-97.

Function key recall (Alt-F)

Esc sequence: EC&oCRECALLCR
Availability: In terminal mode

Restores the configuration file values of user-key definitions if they have been cleared by the host software or a hard reset.

Hard reset (Alt-R)

Esc sequence: ECE
Availability: Always, except during file transfer

A hard reset does the following:

- Emits a beep.
- Unlocks the keyboard if it was locked.
- Turns memory lock off if it was on.
- Clears alpha display memory.
- Sets the margins to columns 1 and 80.
- Clears all tab stops.
- In Reflection 3 and 7, the graphics display is cleared and turned off along with the graphics cursor.
- Sets all terminal configuration parameters to their last activated values.

- Sets the following modes:

  Format mode off
  Display functions off
  Insert character mode off
  Caps lock off
  Report mode off
  Metric mode off
  Record mode off
  Log top and log bottom off

- Does not change remote, block, auto linefeed, and modify all modes settings.
- Initializes serial communications ports to their last activated values and clears the receive buffer.
- Clears the printer buffer, and sends a reset command to the printer.
- Sets user keys to their default values.
- Sends the alpha cursor home.
- Sets the *host prompt received* switch on.
- Transmits an XON if *receive pacing* is enabled.
- Displays modes key labels and enters terminal mode.

Help ([Alt]-[H])

Esc sequence: None
Availability: In terminal mode

Displays the Help screen. Return to the exact point at which you entered the Help screen with **RESUME** ([F8]).

Home down ([Ctrl]-[End])

Esc sequence: ^{E}cF
Availability: In terminal mode

Causes alpha display memory to roll up, so the last line in display memory occupies the next-to-last row of the screen. The alpha cursor moves to the left margin of the last row of the screen.

If the last row in alpha display memory is already on the screen in a row other than the last row, the display is not rolled up; the alpha cursor moves to the left margin of the row below the last line.

When memory lock is enabled, the alpha cursor moves to the left margin of the first row below the locked area. If the cursor is within the locked area when the home down function is performed, it moves to the first row below the locked area.

Home up (Ctrl-Home)

Esc sequence: ^{E}cH
Availability: In terminal mode, user-key definition menu, and file transfer

With format mode off, alpha display memory rolls down so that the first line of display memory occupies the first row of the screen. The alpha cursor moves to the left margin of the top row.

With memory lock enabled, the alpha cursor moves to the left margin of the first row below the locked area. If the cursor is within the locked area when the home up function is performed, it moves to the first row below the locked area.

With format mode on, the text rolls down as described above, but the alpha cursor moves to the first column of the first unprotected field on the screen. If there are no unprotected fields, the alpha cursor moves to the upper left corner (row 1, column 1). If memory lock is enabled, the cursor moves to the first field, even if that field is within the locked area.

Insert character (Ins)

Esc sequence: ^{E}cQ enables insert mode
^{E}cR disables insert mode
Availability: Always

Toggles insert character mode on and off. When on, the bottom line of the screen contains a letter **I**.

The alpha cursor becomes a blinking block when insert character mode is on.

When format mode is off, characters are inserted at the alpha cursor location, and all characters to the right of the cursor move right one column. The character at the right margin is deleted from the line to make room for the other characters to be shifted right.

If the alpha cursor is beyond the right margin when the inserted character is typed, the character at the right margin is deleted.

Any nondisplaying characters to the right of the alpha cursor position are shifted along with the displaying characters, so that the range of unprotected fields and display enhancements may be extended.

When format mode is on, insert character mode works only within the field that contains the alpha cursor. If the cursor is within a protected area, it moves to the first column of the next field before the character is inserted. Only displaying characters are moved; nondisplaying characters such as display enhancements, character set changes, and field definitions remain fixed.

Insert line ([Alt]-[I])

Esc sequence: ${}^{E}{}_{C}L$
Availability: In terminal mode, except format mode

Inserts a blank line in the screen row that contains the alpha cursor. The old contents of that row and all the following rows in alpha display memory are pushed down one row to make room for the new row. The cursor then moves to the left margin of the new (blank) row.

If alpha display memory is full when this function is performed, then either the first row or the last row of display memory must be deleted to make room for the new row. If the first row is not currently displayed on the screen, it is deleted; otherwise, the last row of display memory is deleted.

If memory lock is on and the alpha cursor is within the locked area, then the last line of the locked area is pushed down and becomes unlocked.

Keypad (NumLock)

Esc sequence: None
Availability: Always

When NumLock is on, the keypad generates numbers. When it's off, the keypad keys are the alpha cursor movement keys. The letter N is displayed between F4 and F5 when NumLock is on.

When the graphics keypad is enabled, the letter G appears, and the cursor keys move the graphics cursor.

Modes keys (Alt-M)

Esc sequence: ^{E}c&jA
Availability: Always, except when disabled by the escape sequence ^{E}c&jS

Brings up the modes keys and switches Reflection to terminal mode. If the system, modes, and user keys are disabled by the escape sequence ^{E}c&jS, pressing this key causes a beep, and the function is not performed.

Move cursor down (Down, Shift-Down with NumLock on)

Esc sequence: ^{E}cB
Availability: In terminal and menu modes

Moves the alpha cursor down one row on the screen without changing the contents of the screen. If the cursor is already in the bottom row of the screen, it moves to the same column in the top row. In menu mode the cursor moves between fields.

Move cursor left (Left, Shift-Left with NumLock on)

Esc sequence: ^{E}cD
Availability: When the alpha cursor is visible

Moves the alpha cursor one column to the left, without changing the contents of the screen. If the cursor is already at the left column of the screen, it moves to the rightmost column of the next higher row. If the cursor is at the upper left corner of the screen, it moves to the lower right corner.

HP Reference

Move cursor right ([Right], [Shift]-[Right] with [NumLock] on)

Esc sequence: ^{E}cC
Availability: When the alpha cursor is visible

Moves the alpha cursor one column to the right, without changing the contents of the screen. If the cursor is already at the right column of the screen, it moves to the leftmost column of the next row. If the cursor is at the lower right corner of the screen, it moves to the upper left corner.

Move cursor up ([Up], [Shift]-[Up] with [NumLock] on)

Esc sequence: ^{E}cA
Availability: In terminal and menu modes

Moves the alpha cursor up one row on the screen, without changing the contents of the screen. If the cursor is already in the top row of the screen, it moves to the same column in the bottom row. In menu mode the cursor moves to the previous field.

Move to bottom row ([Ctrl]-[PgDn])

Esc sequence: One occurrence of the ^{E}cB (cursor-down) sequence for each row that the alpha cursor is moved
Availability: In terminal mode

Moves the alpha cursor to the bottom row of the screen, without changing its column.

This function is implemented as a series of cursor-down functions. When [Ctrl]-[PgDn] is entered with the **Transmit functions** parameter configured to *YES*, Reflection transmits one cursor-down escape sequence for each row that the alpha cursor is moved.

Move to line end ([End], [Shift]-[Right] with [NumLock] on)

Esc sequence: One occurrence of the cursor-right or cursor-left sequence (^{E}cC or ^{E}cD) for each column the alpha cursor is moved
Availability: In terminal mode

Moves the alpha cursor one position to the right of the last column containing data in the current row. If the right margin is greater than column 80, and if the last column that contains data is the right margin, then the cursor is moved only as far as the right margin.

If the new cursor position is either to the left or right of the current window, the alpha display is scrolled left or right so that the alpha cursor is on the screen.

Move to line start ([Home], [Shift]-[Right] with [NumLock] on)

Esc sequence: One occurrence of the cursor-left sequence (^{E}cD) for each column the alpha cursor is moved
Availability: In terminal mode

Moves the alpha cursor to column 1 on the screen. If necessary, the text is rolled right as far as possible, so that column 1 appears on the screen.

Move to top row ([Ctrl]-[PgUp])

Esc sequence: One occurrence of the cursor-up (^{E}cA) sequence for each row the alpha cursor is moved
Availability: In terminal mode

Moves the alpha cursor to the top row of the screen, without changing its column.

This function is implemented as a series of cursor-up functions. When [Ctrl]-[PgUp] is entered with the **Transmit functions** value configured to *YES*, Reflection transmits one cursor-up escape sequence for each row the alpha cursor is moved. If the cursor is not moved, nothing is transmitted.

Next page ([PgDn], [Shift]-[PgDn] with [NumLock] on)

Esc sequence: ^{E}cU
Availability: In terminal mode

Displays the next page of alpha display memory. A page consists of 24 lines of display memory, unless memory lock is enabled. When memory lock is enabled, a page is 24 lines minus the number of locked screen rows. For example, when the top 6 rows of the screen are locked, the page consists of 18 (24 - 6) rows. If the last line of alpha display memory is already on the screen when the Next page function is performed, the display is rolled up until the last line of display memory appears in the first unlocked screen row. Further Next page operations have no effect.

Previous page (PgUp, Shift-PgUp with NumLock on)

Esc sequence: ^{E}cV
Availability: In terminal mode

Displays the previous page of alpha display memory. A page consists of 24 lines of display memory, unless memory lock is enabled. When memory lock is enabled, a page is 24 lines minus the number of locked screen rows. The Previous page function leaves the locked rows unchanged and rolls the unlocked rows down. If the first line of alpha display memory is already on the screen when the Previous page operation is performed, the operation has no effect.

HP Reference

Print screen (Alt-PrtSc)

Esc sequence: None
Availability: Always

Prints a copy of the currently displayed screen (alpha only), not including the function key labels.

Print screen with labels (DOS) (Shift-PrtSc)

Esc sequence: None
Availability: Always

Prints a copy of the currently displayed screen (alpha only), including the function key labels. No form feed is performed after the print screen, and only 25 lines are printed. This method is unreliable with the graphics display on.

Reset handshake (Alt-Q)

Esc sequence: None
Availability: Always

If Reflection is configured for a host prompt character other than NONE (null), under certain circumstances Reflection will wait until it receives that character from the host computer before it transmits the next line or block of data. (On a LAN the result varies; see page TR-22.) Occasionally, however, the host prompt character may not be received, or it may be garbled on reception, so that Reflection goes into an endless wait. The Reset handshake function tells the program to go ahead and transmit the next block of data.

Reset connection ([Ctrl]-[F8])

Esc sequence: Ec&bR
Availability: Always

- Resets a Hayes Smartmodem 1200B to its power-on state. Sets the *out 1* bit of the modem control register to 1 for at least 50 milliseconds, as described in the Hayes manual.
- Suspends the current LAN session.
- Powers down the communications ports on the HP Portable Plus.

Reset typeahead ([Alt]-[T])

Esc sequence: None
Availability: Always

Typeahead allows you to type ahead of the HP 3000. When typeahead is enabled, Reflection will buffer each line of keystrokes until it sees a DC1, which indicates that the HP 3000 is ready to receive. If you have logged off the system, the host will not send a DC1, and thus Reflection will go into an endless wait. The DC1 can also be lost or garbled in transmission.

The Reset typeahead function temporarily disables typeahead and forces Reflection to transmit the contents of the typeahead buffer immediately. Typeahead resumes when the next DC1 is seen, indicating a resumption of an HP 3000 conversation.

Roll down ([Ctrl]-[Up])

Esc sequence: EcT or Ec&r<x>D
Availability: In terminal mode

Rolls the text in alpha display memory down one line and brings the previous line of text onto the top row of the screen. If the first line of display memory is already in the top row of the screen, then roll down has no effect. When using the escape sequence, a number may follow the **r** to specify the number of lines to roll.

When memory lock is enabled, only the unlocked portion of the alpha display rolls down; the locked portion remains fixed in the upper part of the screen. If the first line of display memory is already on the screen, roll down has no effect.

Roll left (Ctrl-Right)

Esc sequence: ^{E}C&r<x>L
Availability: When the right margin is beyond 80

If the right margin is set beyond column 80, roll left brings text from beyond the right border of the screen into view. Each instance shifts a specified number of columns off the left side of the screen and the same number of columns onto the right side. The alpha cursor remains in the same position on the screen. The number of columns scrolled is specified in the **Columns per horizontal scroll** field of the Terminal Configuration, Page 2 menu screen.

When using the escape sequence, a number may follow the **r** to specify the number of columns to roll.

If the right margin column is already on the screen, this function has no effect.

Roll right (Ctrl-Left)

Esc sequence: ^{E}C&r<x>R
Availability: In terminal mode, when the right margin is beyond column 80

When the right margin is set beyond column 80, roll right brings text from beyond the left border of the screen into view. Each instance shifts a specified number of columns off the right side of the screen and the same number of columns onto the left. The alpha cursor remains in the same position on the screen. The number of columns scrolled is specified in the **Columns per horizontal scroll** field of the Terminal Configuration, Page 2 menu screen.

When using the escape sequence, a number may follow the **r** to specify the number of columns to roll.

When column 1 is already on the screen, this function has no effect.

Roll up (Ctrl-Down)

Esc sequence: ^{E}cS or ^{E}c&r<x>U
Availability: In terminal mode

Rolls the text in alpha display memory up one line and brings the next line of text onto the bottom row of the screen. If the last line of display memory is already in the top row of the screen, roll up has no effect.

When using the escape sequence, a number may follow the **r** to specify the number of lines to roll.

When memory lock is enabled, only the unlocked portion of the alpha display rolls up; the locked portion remains fixed in the upper part of the screen. If the last line of display memory is already in the screen row immediately below the locked portion, roll up has no effect.

Select key (unshifted PrtSc)

Esc sequence: ^{E}c&P
Availability: In terminal mode

Transmits ^{E}c&P over the datacomm. The interpretation of this sequence depends on the application program. See page TR-259 for handshaking considerations.

Soft reset (Alt-S)

Esc sequence: ^{E}cg
Availability: Always, except in file transfer

A soft reset does the following:

- Emits a beep.
- Unlocks the keyboard, if it was locked.

- Turns display functions off, if it was on.
- Initializes the serial communications ports to the last activated values, and clears the receive buffer.
- Clears the printer buffer, and sends a reset command to the printer.
- Sets the *host prompt received* switch on.

Stop key (ScrollLock)

Esc sequence: None
Availability: In terminal mode, when **Receive pacing** has a value of *XON/XOFF* and SCROLL-LOCK=STOP has a value of *YES*

The Stop (or *Hold*) key alternately stops and starts the process of taking data from the receive buffer and displaying it on the screen. It differs from the Ctrl-Q/Ctrl-S combination in that it does not immediately send the XON and XOFF characters to the host. Instead, it depends on the receive pacing mechanism to send the XON/XOFF as they are needed so that the receive buffer does not overflow. An **H** appears between the F4 and F5 labels when the hold key is active.

System keys (F10)

Esc sequence: None
Availability: Always, except when disabled by the ^{E}c&jS escape sequence

Brings up system keys. F10 also places Reflection in terminal mode, erases any menu screen, and displays the most recent page of alpha display memory.

If the selection of system, modes, and user keys has been disabled by the escape sequence ^{E}c&jS, pressing this key causes a beep, and the function is not performed.

HP Reference

Toggle user labels ([F9])

Esc sequence: None
Availability: When user keys are enabled

Changes the number of screen label rows to one, zero, or two. The effect of this function is temporary; the next time the user keys are displayed, the labels revert to their configured height.

The initial height of the user-key labels and the ability to present them as the startup labels can be configured. See *HP Terminal Configuration* which begins on page TR-35. This field is on the Page 2 menu.

User keys ([F9] or [Alt]-[U])

Esc sequence: ^{E}C&jB
Availability: Always, except when disabled by the escape sequence ^{E}C&jS

Brings user keys to the screen. [F9] may then be used to toggle the height of the user-key labels. When the selection of system, modes, and user keys is disabled by the escape sequence ^{E}C&jS, pressing this key causes a beep, and the function is not performed.

HP Block Transfers

Table C.1
Block Transfer Handshaking Protocol

TRANSFER TYPE	*INH HNDSHK*	*INH DC2*	*HANDSHAKE*
1	*No*	*No*	*DC1/DC2/DC1*
1	*No*	*Yes*	*None*
1	*Yes*	*No*	*DC1/DC2/DC1*
1	*Yes*	*Yes*	*None*
2	*No*	*No*	*None*
2	*No*	*Yes*	*None*
2	*Yes*	*No*	*DC1/DC2/DC1*
2	*Yes*	*Yes*	*None*
3	*No*	*No*	*DC1*
3	*No*	*Yes*	*DC1*
3	*Yes*	*No*	*DC1/DC2/DC1*
3	*Yes*	*Yes*	*None*
4	*N/A*	*N/A*	*None**

* With an HP 2622A, you must set the **Inhibit DC2** parameter to *YES* to avoid sending a DC2 character when you press $\boxed{\text{Enter}}$ or $\boxed{\text{Return}}$. Reflection assumes you always want to send the data block, not a DC2.

Block Transfer Handshaking

Reflection uses three types of handshaking:

- No handshake: The block of data is sent immediately.
- DC1: Reflection sends the block after receiving the host prompt character.
- DC1/DC2/DC1: Reflection sends an ASCII DC2 character after receiving a host prompt character. After receiving another host prompt, it sends the block.

Following is a list of the types of block transfers, classified according to the handshaking rules that apply:

1. Type 1: Press [Enter] in block mode. In block/page mode, press a transmit-only user function key or [Select].
2. Type 2: Press [Enter] in character mode.
3. Type 3: Press a transmit-only user function key in character or block/line mode, or [Select].
   a) Reflection responds to a primary status request, secondary status request, terminal ID request, serial number request, device status request, or cursor sense escape sequence.
   b) Reflection sends a block of data from display memory or from the user-key definition menu, in response to an ^{E}cd sequence from the host computer.
   c) Reflection responds to a device control operation with a device control completion code (**S**,**F**, or **U**).
4. Type 4: Press [Enter] or [Return] in character mode with **LINE MODIFY** or **MODIFY ALL** enabled.

Block Transfer Terminators

Reflection appends one or more characters to the end of each line or field that is sent as part of a block transfer. It also appends one or more characters to the end of the entire block.

In character mode or block/line mode, Reflection appends a carriage return (or carriage return/linefeed if **AUTO LF** is enabled) to the end of the data. If data is not being sent, or the transmission is terminated by encountering a block terminator in display memory, Reflection sends a block terminator followed by a carriage return.

In block/page (non-format) mode, Reflection appends a carriage return (or carriage return/linefeed if **AUTO LF** is enabled) to the end of each line except the last one. If a block terminator character in display memory ends the transfer, then only the block terminator is sent after the last line. If the operation ends because of encountering the end of display memory, then Reflection appends a carriage return, followed by a block terminator to the last line.

In block/page/format mode, Reflection appends a field separator character to the end of each transmitted field except the last one. Reflection appends a block terminator to the last field. If there are no fields to be sent, Reflection just sends a block terminator.

HP Format Mode

Reflection has a *format mode* that allows the use of protected and unprotected fields. By using combinations of unprotected fields, display enhancements, and the line drawing character set, you can create forms for *fill in the blanks* data entry using an HP 3000 forms program such as *HP VPLUS/3000*.

Format Mode Fields

In format mode, the screen is divided into protected and unprotected fields. Data can be entered only in the unprotected fields. If the cursor is in a protected area of the screen and a character is typed, the cursor advances to the next unprotected field before the character is displayed on the screen.

Enable and disable format mode through the modes keys ([Alt]-[M]) by pressing [F5], **FORMAT MODE**. Format mode can also be enabled and disabled by the following escape sequences, either from the keyboard or from the host computer:

$^{E}_{C}$W enables format mode.
$^{E}_{C}$X disables format mode.

When format mode is enabled, a home up function is performed; the cursor moves to the first column of the first unprotected field in display memory. All tab stops are cleared, and the margins are set to columns 1 and 80. Tab stops are ignored in format mode. If there are no unprotected fields in display memory, the cursor moves to row 1, column 1.

Format mode affects keyboard functions as listed in the table in the next section.

HP Reference

Designing Forms

Forms can be drawn on the screen manually through the keyboard, or by escape sequences and text received from the host computer. With the HP VPLUS/3000 forms handling system, forms are initially designed from the keyboard; then, when the application is run, the form descriptions are sent to the terminal from the HP 3000 to draw the forms.

Table D.1
Keyboard Functions in HP Format Mode

Function	*Keystroke*	*Effect in Format Mode*
Insert line	*Alt*-*I*	*Disabled.*
Delete line	*Alt*-*D*	*Disabled.*
Insert char	*Ins*	*Works within unprotected field boundaries.*
Delete char	*Del*	*Works within unprotected field boundaries.*
Clear line	*Alt*-*K*	*Clears to end of unprotected field.*
Clear display	*Alt*-*J*	*Clears only unprotected fields from cursor to end of screen.*
Tab	*Tab*	*Advances cursor to next unprotected field.*
Backtab	*Shift*-*Tab*	*Moves cursor back to start of current field, or to previous unprotected field.*
Home up	*Ctrl*-*Home*	*Moves cursor to first column of first unprotected field in display memory.*

The steps involved in drawing a form, either from the keyboard or from the host computer, are as follows:

1. Make sure that format mode is not on. The ^{E}cX sequence turns it off.
2. Home the cursor (^{E}cH) and clear display memory (^{E}cJ).
3. For each unprotected field, move the cursor to the row and column at which you want the field to start. If you want the

field to be enhanced (inverse video, underlined, etc.), enter the ^{E}c&d<x> escape sequence to start the display enhancement. Refer to page TR-281 for a table of display enhancement escape sequences.

4. Enter the ^{E}c[escape sequence to start the unprotected field.
5. Move the cursor one column beyond the last column that is to be part of the unprotected field. Enter the escape sequence ^{E}c] to terminate the field. If you want a field to extend through column 80, do not enter an *end of field* escape sequence. If two fields are to be contiguous, entering ^{E}c[will end the previous field and start a new one.
6. To clear a *start of field* indicator, position the cursor at the first column of the field and press Del.
7. To define a field that wraps around from the end of one screen row to the next, do not enter an *end of field* escape sequence. Move the cursor to the first column of the following row, and then enter an ^{E}c[sequence. This method may be used to define a field to span any number of screen rows.
8. To end a display enhancement, move the cursor one column beyond the last column that is to be enhanced, and enter the sequence ^{E}c&d@.
9. When the form has been drawn, you can enable format mode with the escape sequence ^{E}cW, or by pressing F5, the **FORMAT MODE** key, from the modes labels.

Forms generation can be simplified by loading escape sequences into the user keys. Assign the attribute **N** (normal keyboard input) to each key's definition. See pages TR-103 and TR-109 for detailed information about user keys.

HP Reference

Forms Cache

Forms cache is a feature of Reflection that can be used to reduce the amount of data transmission from the host computer to Reflection when performing applications that use forms. In a typical format mode application, most of the data that is sent to the terminal merely defines the forms to be displayed on the screen. If a given form is to be displayed more than once, a lot of time is wasted by redefining the same form to the terminal each time it is needed.

With the forms cache feature, the programmer can send a number of forms definitions to Reflection for local storage and cause a given form to be displayed by sending a single short escape sequence.

In order to use the forms cache feature with VPLUS/3000, and perhaps with other HP 3000 software as well, you must change the **Terminal ID response** field in the Terminal Configuration, Page 2 menu to *2624B*.

Allocating the Forms Buffer

Before any forms can be stored in the PC, a memory buffer must be allocated. This can be done manually through the Terminal Configuration, Page 2 field, **Forms buffer size**, or by the escape sequence:

^{E}c&q4te2{<blocks>L

where <blocks> is the number of 256-byte blocks to be allocated.

If the requested number of blocks cannot be allocated, then the size of the forms buffer is not changed. The maximum number of blocks that can be requested is 255. A value of zero deletes the forms buffer.

Any attempt to change the forms buffer size causes memory to be re-allocated, so that the contents of display memory and the printer buffer are lost. Note that the forms buffer size directly relates to the amount of free memory available for running other applications with Reflection; see page TR-100.

In order for a host program to determine if the requested allocation was successful, the programmer must issue a status request, as described below.

Reading the Forms Buffer Status

A program on the host computer can request Reflection to send the status of the forms buffer, which may include the following:

- Number of 256-byte blocks of memory in the forms buffer

- Number of unused 256-byte blocks remaining in the buffer

The status request escape sequence takes the form:

^{E}c&p9^ or ^{E}c&p<form#>p9^

If the **<form#>** parameter is missing, or if **<form#>** is zero, then the total size of the forms buffer is returned. Otherwise, the number of unused blocks and the presence or absence of the specified form number is returned.

The form of the response is as follows:

^{E}c\p9<xyz>

where **<xyz>** are three ASCII characters for which the high-order four bits are 0011 and the low-order four bits are as defined below.

- The low-order four bits of the first byte, **<x>**, contain the most significant four bits of the number of blocks (either in total or unused) in the forms buffer.
- The low-order four bits of the second byte, **<y>**, contain the least significant four bits of the number of 256-byte blocks.
- If a non-zero form number is specified in the status request, the low-order four bits of the third byte, **<z>**, indicate the presence (0001) or absence (0000) of the form in the buffer.

In the examples in the following table, 50 blocks with 256 characters each have been allocated for the forms buffer, and form numbers 1 through 15 have been stored, leaving 30 blocks unused.

Table D.2
Sample Requests and Responses

Status Request	Status Response
^{E}c&p9^	^{E}c\p9320
^{E}c&p0p9^	^{E}c\p9320
^{E}c&p7p9^	^{E}c\p91>1
^{E}c&p19p9^	^{E}c\p91>0

Storing a Form

There are two escape sequences that may be used to store a form in the forms buffer. In either sequence, the form number may be any number from 1 through 255. If the specified form number already exists in the buffer, it is overwritten with the new form's contents. The first escape sequence follows:

Ec&p9u<form#>p<length>L<contents>

It causes the next **<length>** characters following the **L** to be stored. Note that the < and > characters delimit variables, and are not part of the escape sequence. The second escape sequence takes the form:

Ec&p9u<form#>p<<contents>>L

In this escape sequence, the < and > characters before and after the **<contents>** are part of the sequence, and are used to delimit the **<contents>**. Contrary to what is documented in the HP 2624B manual, the **<contents>** can contain any ASCII characters except NULL, DEL, ENQ, and DC1. This escape sequence is useful if you don't know the length of the **<contents>** in advance.

Displaying a Form

To display a form that has previously been stored in the forms buffer, send the following escape sequence to Reflection:

Ec&p9u<form#>pF

A completion code of S (success) or F (failure) is returned to the host by Reflection after the appropriate handshaking.

Purging a Form

The following escape sequences purge a form from the forms buffer:

Ec&p9u<form#>pL or
Ec&p9u<form#>p0L

If the form number is found, the amount of space that it used in the buffer is released.

Special Format Mode Features

Reflection must be configured as a terminal which supports these features (HP 2393A, HP 2394A, HP 2397A, or HP 2624B). See pages TR-284 and TR-223.

Transmit-only fields

Transmit-only fields are similar to unprotected fields, although data is not usually entered into transmit-only fields. (Tab and Ctrl-Home skip over transmit-only fields.)

Ec{ is used to begin a transmit-only field.

Nondisplaying terminators

Reflection supports nondisplaying terminators and the Autoterm and Clearterm configuration options when configured as a terminal which supports this feature. Ec_ and Ec| will, respectively, insert and delete a non-displaying terminator at the current cursor position. Autoterm and Clearterm can be enabled by escape sequence or by using the Reflection SET command.

Modified data tags

With the appropriate configuration, Reflection supports the modified data tag feature. When an HP 2624B (HP 2393A, HP 2394A, or HP 2397A) terminal is configured to Transmit Modified rather than All fields, trailing blanks are eliminated from the contents of unprotected fields and, if in Block Page mode, only modified fields are sent. Reflection supports the escape sequences to change the Transmit option and also provides a SET command to do so.

HP Escape Sequences

An escape sequence is a character string beginning with the ASCII escape character (decimal 27). Such strings constitute a command, usually to the terminal, to perform some action. Escape sequences are usually sent to a terminal from a host computer, but they can also be entered through the keyboard.

A Word about Notation

In this reference, escape sequence parameters you must supply are enclosed in angled brackets. For example, the **<x>** parameter in the escape sequence ^{E}c&d<x> asks you to supply one of several codes specifying the display enhancement you want. The parameter **<n>** or **<num>** usually represents a repetition count or a screen coordinate.

When coding escape sequences, take care to distinguish between upper and lower case characters.

Terminal Control

Block transfer request
^{E}cd — Requests a block transfer from display memory.

Delay one second
^{E}c@ — Delays one second.

Disconnect
^{E}cf — Disconnects (lower DTR for 2 seconds).

Display functions on/off
^{E}cY — Turns display functions mode on.

^{E}cZ — Turns display functions mode off.

Format mode on/off

$^{E}{}_{C}$W — Turns format mode on.

$^{E}{}_{C}$X — Turns format mode off.

Hard reset

$^{E}{}_{C}$E — Performs a hard reset.

Home up and copy

$^{E}{}_{C}$0 — Homes cursor and copies from memory to the currently selected "to" devices.

Lock/Unlock keyboard

$^{E}{}_{C}$c — Locks the keyboard.

$^{E}{}_{C}$b — Unlocks the keyboard.

Memory lock on/off

$^{E}{}_{C}$l — Turns memory lock mode on.

$^{E}{}_{C}$m — Turns memory lock mode off.

Primary terminal status

$^{E}{}_{C}$^ — Requests primary terminal status.

Secondary terminal status

$^{E}{}_{C}$~ — Requests secondary terminal status.

Self test

$^{E}{}_{C}$z — Initiates terminal self-test (always successful).

Soft reset

$^{E}{}_{C}$g — Performs a soft reset.

Start/End unprotected field

$^{E}{}_{C}$[— Starts an unprotected field.

$^{E}{}_{C}$] — Ends an unprotected field.

Terminal class

$^{E}{}_{C}$&k<n>\ — Switches between HP and VT terminal class, as follows:

<n> = 0 change to HP
<n> = 1 change to VT

Changing terminal class reactivates the last saved or last activated configuration values for the new terminal class.

Transmit-only field

^{E}c{ — Start a *transmit only* field. This sequence is treated by Reflection as only a *start an unprotected field* unless the terminal emulation is HP 2624B, HP 2393A, HP 2397A. Otherwise it is treated as if an ^{E}c[is sent.

Setting Margins and Tabs

Clear all tabs

^{E}c3 — Clears all tab stops.

Clear/Set tab at cursor

^{E}c2 — Clears the tab stop at the current cursor position.

^{E}c1 — Sets a tab stop at the current cursor position.

Reset default margins

^{E}c9 — Sets the left margin in column 1, and the right margin in column 80.

Set left/right margin

^{E}c4 — Sets the left margin at the current cursor position.

^{E}c5 — Sets the right margin at the current cursor position.

Editing Display Memory

Clear display/line

^{E}cJ — Clears display from cursor through end of memory.

^{E}cK — Clears display from cursor through end of line.

Delete character

^{E}cP — Deletes character.

Insert/Delete line

^{E}cM — Deletes line.

^{E}cL — Inserts line.

Insert mode on/off

^{E}cQ — Starts insert character mode.

^{E}cR — Ends insert character mode.

User Key Control

Default F1
^{E}cp — Default definition of user key F1.

Default F2
^{E}cq — Default definition of user key F2.

Default F3
^{E}cr — Default definition of user key F3.

Default F4
^{E}cs — Default definition of user key F4.

Default F5
^{E}ct — Default definition of user key F5.

Default F6
^{E}cu — Default definition of user key F6.

Default F7
^{E}cv — Default definition of user key F7.

Default F8
^{E}cw — Default definition of user key F8.

Define user keys
^{E}c&f... — Defines user keys (see page TR-109 for detailed instructions).

Display user-key labels
^{E}c&jB — Displays user-key labels and enables user keys.

Trigger user-key definition
^{E}c&f<x>E — Triggers function key <x>. When received, Reflection acts exactly as if the function key <x> had been pressed. Only the current user function keys are triggered.

User keys enabled, no labels
^{E}c&j@ — Enables user keys, remove labels.

User key menu displayed
^{E}cj — Displays the user-key definition menu.

User key menu removed
^{E}ck — Removes the user-key definition menu.

Cursor Positioning

NOTE: See *Reflection Escape Sequences* on page TR-288 for a 5-digit cursor position response.

Backtab
^{E}ci — Performs a Backtab.

Cusor position absolute
^{E}ca — Senses cursor position, absolute.

Cusor position relative
^{E}c' — Senses cursor position, relative to the current display.

Cursor position mode
^{E}c&x<x>C — <x> = 1 sets; <x> = 0 clears. Sends cursor position mode. When this mode is set and [Enter], [Select], or a transmit-only user key is pressed, the cursor position report escape sequence is added to the beginning of the transmitted block.

Home cursor
^{E}ch — Homes the cursor. In format mode the cursor is positioned in the first unprotected field. *Transmit only* fields are ignored if the terminal emulation is HP 2624B, HP 2393A, or HP 2397A.

Home down/up
^{E}cF — Homes down the cursor.

^{E}cH — Homes up the cursor. In format mode the cursor is positioned in the first unprotected or transmit-only field.

Move cursor up/down
^{E}cA — Moves the cursor up one row.

^{E}cB — Moves the cursor down one row.

Move cursor right/left
^{E}cC — Moves the cursor right one column.

^{E}cD — Moves the cursor left one column.

Move cursor to left margin
^{E}cG — Moves cursor to left margin.

Tab

^{E}cI — Performs a horizontal tab.

NOTE: Row and column numbers are expressed relative to zero in the following sequences. The column and row commands can be switched, in which case the C would be in upper case and the other lower.

Position cursor relative to screen

^{E}c&a<col>c<row>Y — Moves the cursor to **<col>** and **<row>**, relative to the currently displayed screen.

Position cursor in display memory

^{E}c&a<col>c<row>R — Moves the cursor to **<col>** and **<row>**, relative to the start of display memory.

Position cursor relative

^{E}c&a+<col>c+<row>Y — Moves the cursor **<col>** columns (left or right) and **<row>** rows (up or down), relative to the current cursor position. The **+** may be either a **+** or **-** sign.

Position cursor relative

^{E}c&a+<col>c+<row>R — Moves the cursor **<col>** columns (left or right) and **<row>** rows (up or down), relative to the current cursor position in memory. The **+** may be either a **+** or **-** sign.

Configuration Control

The escape sequences that begin with ^{E}c&k... and ^{E}c&s... can be of any length, as long as the last character is a capital letter. This allows a number of parameters to be specified in one sequence. Sequences are listed here with an ending capital letter as if they will be issued alone. Note that 0 is the number zero.

Auto linefeed off/on

^{E}c&k0A — Turns auto linefeed off.

^{E}c&k1A — Turns auto linefeed on.

Bell off/on

^{E}c&k0D — Turns bell off.

^{E}c&k1D — Turns bell on.

Block mode off/on

^{E}C&k0B — Turns block mode off.

^{E}C&k1B — Turns block mode on.

Caps lock off/on

^{E}C&k0C — Turns caps lock off.

^{E}C&k1C — Turns caps lock on.

Caps mode off/on

^{E}C&k0P — Turns caps mode off.

^{E}C&k1P — Turns caps mode on.

Check parity YES/NO

^{E}C&s0Z — Sets **Check parity** to NO.

^{E}C&s1Z — Sets **Check parity** to YES.

Escape transfer to printer YES/NO

^{E}C&s0N — Sets **Esc Xfer** to NO.

^{E}C&s1N — Sets **Esc Xfer** to YES.

Frame rate = 60/50

^{E}C&k0J — Sets frame rate = 60 (ignored by Reflection).

^{E}C&k1J — Sets frame rate = 50 (ignored by Reflection).

Inhibit DC2 NO/YES

^{E}C&s0H — Sets **Inhibit DC2** to NO.

^{E}C&s1H — Sets **Inhibit DC2** to YES.

Inhibit EOL wrap NO/YES

^{E}C&s0C — Sets **Inhibit EOL wrap** to NO.

^{E}C&s1C — Sets **Inhibit EOL wrap** to YES.

Inhibit handshake NO/YES

^{E}C&s0G — Sets **Inhibit handshake** NO.

^{E}C&s1G — Sets **Inhibit handshake** YES.

Keyboard lock off/on

^{E}C&k0K — Turns auto keyboard lock off.

^{E}C&k1K — Turns auto keyboard lock on.

Line/Page mode

EC&s0D — Sets line mode.

EC&s1D — Sets page mode.

Local echo off/on

EC&k0L — Turns local echo off.

EC&k1L — Turns local echo on.

Lock/Unlock configuration

EC&q1L — Locks the configuration menu; also locks modify all, block mode, remote mode, and auto linefeed.

EC&q0L — Unlocks the configuration.

Modify all off/on

EC&k0M — Turns modify all mode off.

EC&k1M — Turns modify all mode on.

No parity

EC&k1I — Sets no parity (8-bit).

Previous parity

EC&k0I — Sets previous parity (7-bit).

Remote mode off/on

EC&k0R — Turns remote mode off.

EC&k1R — Turns remote mode on.

SPOW (space overwrite) latch off/on

EC&k0N — Turns the SPOW latch to off.

EC&k1N — Turns the SPOW latch to on.

SPOW NO/YES

EC&s0B — Sets **SPOW** to NO.

EC&s1B — Sets **SPOW** to YES.

Transmit functions NO/YES

EC&s0A — Sets **Transmit functions** to NO.

EC&s1A — Sets **Transmit functions** to YES.

Zentec mode 132/80

^{E}C! — Sets 132-column mode.

^{E}C" — Sets 80-column mode.

Printer and Data Operations

Copy to destination device

^{E}C&p<a>d<Y> — Copies the amount of data implied by **<Y>** to the destination device **<a>**. If no destination device is specified, the currently selected "to" devices are assumed. Only numbers 4 and 6 are recognized; both refer to a printer. **<Y>**, may be one of the following:

B = the line containing the cursor
F = the current screen page, from the cursor
M = all of display memory, from the cursor

Printer control

^{E}C&p<x>p<y>u<z>C — Performs the action specified by **<z>** on the device(s) specified by **<y>**. The device codes of 4 or 6 indicate the printer.

The **<z>** action codes are as follows:

0	Generates 1 form feed.
1	Skips **<x>** lines.
2-10	Generates 1 form feed.
11	Turns on log bottom mode.
12	Turns on log top mode.
13	Turns off either logging mode.
14	Prints normal characters.
15	Prints expanded characters.
16	Prints compressed characters.
17	Turns on report mode.
18	Turns on metric report mode.
19	Turns off any report mode.
20	Turns on record mode.

If the action is *skip line*, the **<x>** value can be entered to specify the number of lines to skip. If the action is record mode, the **<x>** value defines the character that will turn off record mode. The **<x>** value is the decimal equivalent of the

ASCII character, in the range 0-127.

Request printer status

Ec&p4^ or Ec&p6^ — Either of these escape sequences requests the status of the printer. Reflection returns its *device status* response (see page TR-295).

Record mode

Ec&p<num>p20C — Record mode. In record mode, data received from the host is passed through to the specified "to" devices without appearing on the screen. The **<num>p** parameter is optional. It is the decimal equivalent of an ASCII character that will terminate record mode (the terminator cannot be a NULL, but can be a control character).

Data received while in record mode should consist of lines ending in linefeeds. The terminator character, if specified, must follow a linefeed or a 256-character block without linefeeds; otherwise, the terminator will be ignored by Reflection, which will remain in record mode.

If a terminator is not specified, record mode can be switched off only from the terminal. When record mode is switched off (either from the terminal or upon recognizing a terminator), Reflection transmits **S** (Success) or **F** (Failure), unless DISABLE-COMP-CODES has been set to *YES*; see page TR-217. This transmission is a type 3 block transfer (see page TR-259).

Transfer ASCII string

Ec&p<a>d<b>d<c>dW<data string> — Transfer the ASCII data string from the host computer to the destination devices **<a>**, **<b>**, and **<c>**.

Any number of destination devices can be specified. If no device is specified, the currently selected "to" devices are assumed. The data string is terminated by either the 256th character or by an ASCII linefeed character.

Transfer binary string

Ec&p<a>d<b>d<c>d<x>W<data string> — Transfers the **<data string>**, which is **<x>** bytes in length, from the host computer to the devices specified by **<a>**, **<b>**, and **<c>**. This is a binary transfer, so the data string can contain non-ASCII characters.

Only the printer may be specified; the currently selected "to" devices are assumed.

Display Enhancements

Display enhancements

^{E}c&d<x> — Begins and ends display enhancements, where <x> is a character from the table below.

Security display enhancements

^{E}c&ds<x> — Begins and ends display enhancements, where <x> is a character from the table below, with the security enhancement.

Table E.1
Values for <x>, Display Enhancement

Value	*Half Bright*	*Under-line*	*Inverse Video*	*Blink-ing*	*Security*	*End en-hancement*
A				*A*		
B			*B*			
C			*C*	*C*		
D		*D*				
E		*E*		*E*		
F		*F*	*F*			
G		*G*	*G*	*G*		
H	*H*					
I	*I*			*I*		
J	*J*		*J*			
K	*K*		*K*	*K*		
L	*L*	*L*				
M	*M*	*M*		*M*		
N	*N*	*N*	*N*			
O	*O*	*O*	*O*	*O*		
S					*S*	
@						@

The security enhancement (S) may be combined with another enhancement in the same escape sequence by using the second form shown above.

Some of the combinations are not supported by the monochrome display adapter. The underline and inverse video attributes cannot be combined.

Screen Blanking

Display on/off

Ec&w12F — Turns the display on.

Ec&w13F — Turns the display off.

Display Window

Assign page width

Ec&q3t0{<x>W — Assigns **<x>** columns as the page width, where **<x>** is a number in the range 1-9999.

Assign page width

Ec&w5f<x>W — Assigns **<x>** columns as the page width, where **<x>** is a number in the range 1-9999.

Assign 132/80 column mode

Ec! — Assigns 132-column mode.

Ec" — Assigns 80-column mode.

Modem Control

Disconnect

Ecf — Disconnects (lowers DTR line for approximately 2 seconds). Drops the virtual connection on a LAN connection.

Display Control Operations

Next/Previous page

EcU — Displays next page of display memory.

EcV — Displays previous page of display memory.

Roll display down/up

EcT — Rolls display down one line.

EcS — Rolls display up one line.

Roll display down/up <n> lines

Ec&r<n>D — Rolls the display down **<n>** lines.

EC&r<n>U — Rolls the display up **<n>** lines.

Roll display left/right

EC&r<x>L — Rolls the display left **<x>** columns, if the page width is greater than 80 columns.

EC&r<x>R — Rolls the display right **<x>** columns, if the page width is greater than 80 columns.

Alternate Character Set Selection

Select base character set

EC)@ — Selects the base character set as the alternate character set.

Select line drawing character set

EC)B — Selects the line drawing character set as the alternate character set.

Select space character set

EC)A — Selects an alternate character set that is all spaces.

EC)C — Selects an alternate character set that is all spaces. (Same as EC)A.)

Function Key and Message Operations

Disable system, modes and user keys

EC&jS — Disables system, modes, and user keys.

Display modes labels

EC&jA — Displays modes labels.

Enable system, modes and user keys

EC&jR — Enables system, modes, and user keys.

Remove message, replace labels

EC&jC — Removes a message, replaces screen labels.

Replace labels with message

EC&j<x>L<string> — Replaces the screen labels with a message on the bottom two lines of the screen, where **<x>** is a number from 0 to 160, and **<string>** is a message of **<x>** number of characters.

Forms Cache

See *HP Format Mode* on page TR-263 for more detailed information about using forms cache escape sequences.

> **NOTE:** The innermost set of angled brackets are literal; be sure to enclose the **<contents>** parameter in angled brackets.

Allocate forms buffer (HP 2394, 2624)

EC&q4te2{<blocks>L — Allocates a forms buffer of **<blocks>** by 256 bytes.

Display form

EC&p9u<form#>pF — Displays a form.

Forms buffer status request

EC&p9^ — Requests the status of the forms buffer.

Forms buffer status request

EC&p<form#>p9^ — Requests the status of the forms buffer.

Purge form

EC&p9u<form#>pL — Purges a form from the buffer.

Store form

EC&p9u<form#>p<length>L<contents> — Stores a form of known length.

Store form, length unknown

EC&p9u<form#>p<<contents>>L — Stores a form of unknown length.

Restricted Escape Sequences

> **NOTE:** The following escape sequences require that one of these terminals be selected: HP 2624B, HP 2393A, HP 2394A, HP 2397A.

Autoterm off/on

EC&s0J — Turns off Autoterm.

EC&s1J — Turns on Autoterm.

Clearterm off/on

Ec&s0K — Turns Clearterm off.

Ec&s1K — Turns Clearterm on.

Erase non-displaying terminator

Ec| — Erases non-displaying terminators.

Set receive pacing

Ec&q1te1{0H — Same as SET RECEIVE-PACING NONE.

Ec&q1te1{1H — Same as SET RECEIVE-PACING XON/XOFF.

Set transmit pacing

Ec&q1te1{0G — Same as SET TRANSMIT-PACING NONE.

Ec&q1te1{1G — Same as SET TRANSMIT-PACING XON/XOFF.

Set transmit to All/Modified

Ec&k0Z — Turns modified data tags off.

Ec&k1Z — Turns modified data tags on.

Start transmit-only field

Ec{ — Starts a transmit-only field.

Write nondisplaying terminator

Ec_ — Writes a nondisplaying terminator at the cursor position.

HP 2624B and 2394A

Auto linefeed on/off

Ec&q<m>te1{0A — Turns Auto Linefeed off (m = 4-7).

Ec&q<m>te1{1A — Turns Auto Linefeed on (m = 4-7).

Autoterm on/off

Ec&q<m>te0{0J — Turns Autoterm off (m = 4-7).

Ec&q<m>te0{1J — Turns Autoterm on (m = 4-7).

Bell off/on

Ec&q8te0{0D — Turns Bell off.

Ec&q8te0{1D — Turns Bell on.

Block mode off/on

Ec&q<m>te1{0B — Turns block mode off (m = 4-7).

Ec&q<m>te1{1B — Turns block mode on (m = 4-7).

Block terminator character

Ec&q<m>te2{<x>R — Sets Block Terminator character where m = 4-7 and x = 0-127.

Caps lock off/on

Ec&q<m>te1{0C — Turns Caps Lock off (m = 4-7).

Ec&q<m>te1{<1>C — Turns Caps Lock on (m = 4-7).

Clearterm off/on

Ec&q<m>te0{0K — Turns Clearterm off (m = 4-7).

Ec&q<m>te0{1K — Turns Clearterm on (m = 4-7).

Field separator character

Ec&q<m>te2{<x>F — Sets field separator character where m = 4-7 and x = 0-127.

Forms buffer size

Ec&q<m>te2{<x>L — Sets the forms buffer size where m = 4-7 and x = 0-255.

Inhibit DC2 NO/YES

Ec&q<m>te0{0H — Sets **Inhibit DC2** to NO (m = 4-7).

Ec&q<m>te0{1H — Sets **Inhibit DC2** to YES (m = 4-7).

Inhibit EOL Wrap NO/YES

Ec&q<m>te0{0C — Sets **Inhibit EOL Wrap** to NO (m = 4-7).

Ec&q<m>te0{1C — Sets **Inhibit EOL Wrap** to YES (m = 4-7).

Inhibit handshake NO/YES

Ec&q<m>te0{0G — Sets **Inhibit Handshake** to NO (m = 4-7).

Ec&q<m>te0{1G — Sets **Inhibit Handshake** to YES (m = 4-7).

Line/Page mode

Ec&q<m>te0{0D — Sets Line mode (m = 4-7).

Ec&q<m>te0{1D — Sets Page mode (m = 4-7).

Local echo off/on

^{E}c&q<m>te1{0L — Turns Local Echo off (m = 4-7).

^{E}c&q<m>te1{1L — Turns Local Echo on (m = 4-7).

Modify all off/on

^{E}c&q<m>te1{0M — Turns Modify All mode off (m = 4-7).

^{E}c&q<m>te1{1M — Turns Modify All mode on (m = 4-7).

Remote mode off/on

^{E}c&q<m>te1{0R — Turns Remote mode off (m = 4-7).

^{E}c&q<m>te1{1R — Turns Remote mode on (m = 4-7).

Return definition

^{E}c&q8te1{<x>A — Sets Return definition first character where x = 0-127.

^{E}c&q8te1{<x>B — Sets Return definition second character where x = 0-127.

Return equals Enter

^{E}c&q8te1{0R — Same as SET RETURN=ENTER NO.

^{E}c&q8te1{1R — Same as SET RETURN=ENTER YES.

Set alternative character set

^{E}c&q<m>te2{<x>D — Sets alternate character set where m = 4-7 and x = 0-3.

SPOW NO/YES

^{E}c&q<m>te0{0B — Sets **SPOW** to NO (m = 4-7).

^{E}c&q<m>te0{1B — Sets **SPOW** to YES (m = 4-7).

TAB=SPACES NO/YES

^{E}c&q8te1{0T — Sets TAB=SPACES to NO.

^{E}c&q8te1{1T — Sets TAB=SPACES to YES.

Transmit ALL/MODIFIED

^{E}c&q<m>te2{0Z — Sets **TRANSMIT** to ALL (m = 4-7).

^{E}c&q<m>te2{1Z — Sets **TRANSMIT** to MODIFIED (m = 4-7).

Transmit functions NO/YES

^{E}c&q<m>te0{0A — Sets **Transmit Functions** to NO (m = 4-7).

Ec&q<m>te0{1A — Sets **Transmit Functions** to YES (m = 4-7).

Reflection Escape Sequences

Clear typeahead buffer

Ec&oX — Clears the contents of the typeahead buffer.

Connection reset

Ec&bR — Performs the following:

- Resets an internal Hayes Smartmodem 1200B modem card by setting the *out 1* bit of the modem control register to 1 for about 50 milliseconds.
- Powers down the communications ports on an HP Portable Plus.
- Suspends a LAN session.

Five-Digit absolute cursor position

Ec&oA — Returns a 5-digit cursor position, reflecting the absolute value of the current position.

Reflection command

Ec&oC<command>CR — Where **<command>** is any Reflection command. The **<command>** is passed to the command language processor as if it had been entered from the keyboard or a batch file. (See *Command Mode* on page TR-157.)

Reflection indicates the success or failure of the command by returning a completion code of **S** or **F** (unless DISABLE-COMP-CODES is set to YES; see page TR-217). The **S** or **F** is returned according to the rules of a type 3 block transfer (see page TR-259).

Serial Number Request

The host computer can request the serial number and version of the Reflection program running on the PC by sending the escape sequence EC*s12347^. After exchanging the appropriate block transfer handshake, Reflection sends a 14-character serial number in the form I01+200I123456.* See $SERIAL on page 232.

NOTE: If this sequence is sent to a terminal, the terminal ID response is sent in place of the Reflection escape sequence.

To return a sequence of the form WRQ-12345, use EC*s12345^. Previous versions of Reflection may use this exclusively. A nine-character serial number is returned of the form, WRQ-12345, where the *WRQ-* is fixed and the numeric portion is variable.

---

* EC*S12346^ and EC*S33^ are obsolete versions of this sequence.

HP Cursor Positioning and Sensing

The cursor can be placed anywhere in Reflection's display memory by means of an escape sequence issued from the host computer or from the keyboard. There are three ways to specify cursor position within display memory:

- Screen-relative
- Cursor-relative
- Absolute

In the escape sequences, row and column numbers are always expressed relative to zero. In other words, the first row and column position is row 0, column 0. The row and column numbers within the escape sequences are expressed as ASCII decimal digits. If a number is preceded or followed by spaces, the spaces are ignored.

Screen-Relative Cursor Positioning

In this form of cursor positioning, the new position is specified relative to the upper left corner of the current screen. The escape sequences take one of the following forms:

- ^{E}C&a<row>y<column>C
- ^{E}C&a<column>c<row>Y
- ^{E}C&a<row>Y
- ^{E}C&a<column>C

If only a row is specified, the cursor is moved to the current column in the specified row. If only a column is specified, the cursor is moved to the specified column in the current row. If the value of either row or column is so large that it implies moving the cursor off the screen, the maximum on-screen value is used instead of the specified value. In other words, if you try to place the cursor in column 85 using this form, Reflection places it in column 79 (relative to zero).

For example, the following sequence places the cursor in the 12th screen row and the 10th column:

    Ec&a11y009C

Note that there must be no space between the Ec and the **&**.

Absolute Cursor Positioning

In this form of cursor positioning, the new position is expressed relative to the first row (row 0) in display memory. If the new position is outside the bounds of the current screen window, the display is rolled in the direction required to bring the target position onto the screen.

The escape sequence takes one of the following forms:

- Ec&a<row>r<column>C
- Ec&a<column>c<row>R
- Ec&a<row>R

Note that only the **<row>** portion of the escape sequence is *absolute,* since column numbers are always relative to column 0.

The maximum value of **<column>** is either 79 or the right margin column, whichever is greater. If the specified value of **<column>** is greater than the limit, the limit is used instead of the specified value. This prevents the cursor from being placed in a column that is not accessible by horizontal scrolling.

The maximum allowable **<row>** value is 32767. If the **<row>** value is larger than Reflection's display memory can hold, then rows are deleted off the top of display memory and added to the bottom until the cursor has been moved the number of rows that are implied by the escape sequence.

For example, the following escape sequence moves the cursor to the 103rd column of the 1451st row of display memory:

    Ec&a102c1450R

Cursor-Relative Positioning

In this form of cursor positioning, the new position is specified relative to the current cursor position. If necessary, the display is rolled in whatever direction is required to bring the new position within the screen window.

Cursor relative positioning is specified by placing a + or - sign in front of the row or column number. For example, the following escape sequence moves the cursor up 2 rows and right 7 columns from its current position:

    Ec&a-2r+07C

The following escape sequence moves the cursor left 20 columns within the current row.

    Ec&a-20C

Cursor Sensing

The host computer can request and receive the current cursor position. The position can be expressed relative to the current screen or relative to the first row in display memory.

The cursor sensing response is a block transfer, so the rules of block transfers apply, (see page TR-259, *Block Transfers*).

Screen-Relative Sensing

The host computer sends the escape sequence Ec`. Reflection responds with a cursor positioning escape sequence of the form Ec&a<column>c<row>Y, where **<column>** and **<row>** are both three-digit ASCII decimal numbers. The maximum value of **<column>** is 79; the maximum value of **<row>** is 23.

Absolute Sensing The host computer sends the escape sequence ^{E}ca. Reflection responds with a cursor positioning escape sequence of the form, ^{E}c&a<column>c<row>R, where **<column>** and **<row>** are both three-digit ASCII decimal numbers. The maximum value of either number is 999. Note that the actual row and column numbers may be larger than 999; in that case, Reflection reports only the low-order three digits.

Five-Digit Cursor Positioning Reflection provides an escape sequence which will return a 5-digit cursor position, reflecting the absolute value of the current position.

^{E}c&oA

HP Status Requests

A program running on the host computer can request status and terminal information from Reflection. There are five types of status requests:

- Primary status
- Secondary status
- Terminal ID
- Device status
- Serial number request

Reflection responds to each status request with a block transfer. The rules for block transfer handshaking and the terminator characters that Reflection appends to the end of any block transfer are explained in *HP Block Transfers* on page TR-259.

Primary Status Request

To request the primary status from the terminal, the host computer sends the escape sequence ^{E}C^.

Reflection responds by performing a block transfer handshake and then sends the response followed by a block transfer terminator. The response consists of nine characters, as follows:

- The first two characters are ^{E}C\.
- The remaining characters are all 8-bit binary quantities, where the high-order 4 bits of each byte are 0011. This ensures that all responses can be read as ASCII characters. The low-order four bits of each byte are described below, where the bits are numbered as follows:

```
7 6 5 4  3 2 1 0 - bit numbers
| | | |  | | | |
0 0 1 1  ? ? ? ? - bit values
```

Byte 0: (indicates the amount of display memory)

Bit 3: 1 = 8K bytes
Bit 2: 1 = 4K bytes
Bit 1: 1 = 2K bytes
Bit 0: 1 = 1K bytes

Reflection responds with the value for 4K bytes, 8K, 12K, or 15K, depending on the value configured for **Display memory status response**.

Byte 1: (configuration straps A-D)

Bit 3: Strap D, page/line mode 0 = line, 1 = page
Bit 2: Strap C, inhibit end-of-line wrap 0 = no, 1 = yes
Bit 1: Strap B, space overwrite 0 = no, 1 = yes
Bit 0: Strap A, transmit functions 0 = no, 1 = yes

Byte 2: (configuration straps E-H)

Bit 3: Strap H, inhibit DC2 0 = no, 1 = yes
Bit 2: Strap G, inhibit DC1 handshake 0 = no, 1 = yes
Bit 1: Always 0
Bit 0: Always 0

Byte 3: (latching keys)

Bit 3: Always 1, indicating this model of terminal is capable of sending secondary status
Bit 2: Auto linefeed 0 = off, 1 = on
Bit 1: Block mode 0 = character mode, 1 = block mode
Bit 0: CapsLock status 0 = off, 1 = on

Byte 4: Pending transfer flags

Bit 3: Secondary status pending 0 = no, 1 = yes
Bit 2: Enter pending 0 = no, 1 = yes
Bit 1: User function key pending 0 = no, 1 = yes

Bit 0: Cursor sense pending 0 = no, 1 = yes

Byte 5: Error flags

Bit 3: Printer error 0 = no error, 1 = last printer operation failed
Bit 2: Always 0
Bit 1: Self-test result 0 = Error or self-test not requested
1 = No error
Bit 0: Datacomm error status 0 = no error,

1 = framing, overrun, or parity error has occurred since the last status request

Byte 6: Device transfer pending flags. This byte reports on the **S**, **F**, or **U** completion codes associated with the Ec&p device control escape sequences.

Bit 3: Always 0
Bit 2: Always 0
Bit 1: Device operation status pending 0 = no, 1 = yes
Bit 0: Device status pending 0 = no, 1 = yes

Secondary Status Request

To request the secondary status response from the terminal, the host computer sends the escape sequence Ec~. Reflection responds by performing a block transfer handshake and then sends the response followed by a block transfer terminator. The response consists of 9 characters, as follows:

- The first two characters are Ec|. (The ASCII vertical slash character, decimal 124.)
- The remaining characters are all 8-bit binary quantities, where the high-order 4 bits of each byte are 0011. This ensures that all responses can be read as ASCII characters. The low-order four bits of each byte are described below, where the bits are numbered as follows:

```
7 6 5 4  3 2 1 0 - bit numbers
| | | |  | | | |
0 0 1 1  ? ? ? ? - bit values
```

Byte 0: Reports the amount of available terminal memory, besides display memory, available for data buffers.

Bit 3: 1 = 8K bytes
Bit 2: 1 = 4K bytes
Bit 1: 1 = 2K bytes
Bit 0: 1 = 1K bytes

Reflection always returns a value of zero in this byte.

Byte 1: Terminal firmware configuration

Bit 3: Always 0, non-programmable terminal
Bit 2: Always 1, terminal identifies self
Bit 1: Always 0, no APL firmware
Bit 0: Always 1, I/O firmware installed

Byte 2: Configuration straps J-M. These straps do not apply to the HP 2622A terminal. Reflection returns a 0 in this byte.

Byte 3: Keyboard interface keys (N-R). Reflection returns a 0 in this byte.

Byte 4: Configuration straps S-V. These straps do not apply to the HP 2622A terminal. Reflection returns a value of 0 in this byte.

Byte 5: Configuration straps W-Z. These straps do not apply to the HP 2622A terminal. Reflection returns a value of 0 in this byte.

Byte 6: Memory lock mode status

Bit 3: Always 0
Bit 2: 0 = memory is not full, 1 = memory is full
Bit 1: 0 = memory lock not on, 1 = memory lock on
Bit 0: 0 = locked in row 0 (i.e., overflow protect),
    1 = not locked in row 0

Terminal ID

The host computer can request the terminal ID with the escape sequence $^E c$*s^.

After exchanging the appropriate block transfer handshake, Reflection sends the five-character string, **2392A**, **2622A**, **2623A**, **2624B**, **2627A**, or **2626A**, depending on the value configured for **Terminal ID response.**

Device Status

The host computer can request the status of the attached printer by issuing the escape sequence $^E c$&p <device#>^, where **<device#>** is either 4 or 6. Both 4 and 6 indicate external printers in Reflection, so there is no difference.

Reflection responds by sending the string $^E c$\p<device#>, followed by three bytes of status information, in which the high-order four bits of each byte contain 0011.

Byte 0:

Bit 3: Always 0
Bit 2: Always 0
Bit 1: Printer error report 0 = no error, 1 = printer error
Bit 0: Tracks S or **F** response of device operation

0 = last command was successful (S),
1 = last command failed (**F**)

Byte 1:

Bit 3: Tracks **U** response of device operation

0 = last command was interrupted
1 = last command was performed

Bit 2: Always 0
Bit 1: Always 0
Bit 0: Printer status 0 = not busy, 1 = busy

Byte 2: Bits 3 - 0 = 0001 (printer present)

HP Graphics Escape Sequences

NOTE: Escape sequences listed below only relate to Reflection 3 and Reflection 7, which emulate HP graphics terminals 2623A and 2627A respectively. Sequences which refer to choosing pen colors only relate to 2627A emulation.

Table H.1
Graphics Sequence Types

Ec*d	*Display and Cursor Control*
Ec*e	*Image Control*
Ec*l	*Labeling*
Ec*m	*Drawing Mode*
Ec*n	*Graphics Text*
Ec*p	*Line Plotting*
Ec*s	*Graphics Status*
Ec*t	*Compatibility Mode*
Ec*w	*Graphics initialization*

Graphics Defaults

Graphics hard reset

Ec*wR — Sets all defaults, clears raster memory buffer, and sets the primary drawing pen to 0,0.

Restoring graphics defaults

Ec*mR — Restores all graphics defaults.

Restoring selected graphics defaults

Ec*m<default flag>R — If a 1 is used as the <default flag>, a subset of default values are set. Otherwise all defaults are set. See page TR-118.

Reading Graphics Status

Read area shading capability

^{E}C*s11^ — Area shading capabilities are fixed for the HP 2627A. The terminal always responds as follows, (shading can be a polygon 8 pixels wide and 8 pixels high):

Value returned: **2,8,8, <terminator>**

Read current pen position

^{E}C*s2^ — Both the pen position and whether it is up or down are returned as a string of ASCII characters.

Value returned: **<X> <Y> <terminator>**

Read device capabilities

^{E}C*s6^ — Returns a list of graphics and plotting features. If the capability is not available, a zero is returned. The capabilities of Reflection 7 are marked with an asterisk.

Clear display	*b1*	*0 = no clear* *1 = paper advance* *2 = clear (total erase)* *3 = partial clear by area**
Number of pens	*b2*	*(8*)*
Color capability	*b3*	*0 = black or white* *1 = gray levels* *2 = color**
Color levels capability	*b4*	*(2*)* *2 equals the number of color levels for each plane.*
Area shading	*b5*	*0 = no* *1 = yes**
	b6 and b7 — Not used.	
Dynamic modification	*b8*	*0 = no* *1 = yes**

Graphics character size	*b9*	*0 = fixed* *1* = integer multiples of the basic cell size* *2 = any size*
Graphics character angles	*b10*	*0 = fixed* *1* = multiples of 90°* *2 = multiples of 45°* *3 = any angle*
Graphics character slant	*b11*	*0 = fixed* *1* = 45°* *2 = any angle*
Dot-Dash line patterns	*b12*	*0 = none* *1 = predefined only* *2* = user-defined and predefined*

Reflection 7 responds as follows:

3,8,2,2,1,0,0,1,1,1,1,2,0,0,0,0,<terminator>

Read device ID

^{E}c*s1^ — Requests the device ID of the terminal. The response depends on the current emulation selected.

Value returned: **2627A <terminator>**

Read display size

^{E}c*s5^ — Returns the number of pixels which can be displayed per millimeter available on the graphics display. If you will be addressing graphics devices with varying display areas, this sequence allows you to scale data.

<LLX>,<URX>	The lower left and upper right X coordinates.
<LLY>,<URY>	The lower left and upper right Y coordinates.
<MMX>,<MMY>	The number of pixels per millimeter in both axes. (Five digits and a decimal point)

Value returned: **+00000,+00000,+00511,
+00389, 00002.,00002.,
<terminator>**

Read graphics cursor position

Ec*s3^ — The position of the graphics cursor is returned.

Value returned: **<X> <Y> <terminator>**

Read graphics cursor position with wait

Ec*s4^ — The cursor can be positioned by the user before the position coordinates are returned. The arrow keys can be used to position the cursor, which is turned on if not previously on. When a character key is struck, the position of the cursor is given. If another escape sequence is received before a key is struck, the read sequence is ignored. Along with coordinates, the ASCII value of the key struck is given.

Value returned: **<X> <Y> <key code>**

Read graphics modification

Ec*s12^ — The terminal's ability to change selected portions of the display are read. The terminal's fixed response is selective erase and complement capabilities, expressed as follows:

Value returned: **1,1 <terminator>**

Read graphics text status

Ec*s7^ — The current text size, orientation, slant and type of justification are returned. Values are returned in the following order.

<X size>	The <x> dimension of the character cell. (Sign plus 5 digits)
<Y size>	The <y> dimension of the character cell. (Sign plus 5 digits)
<origin>	Relative position of the text to the graphics cursor. (1 digit)
<angle>	Text angle 0,90,180, or 270. (Five digits and a decimal point)

<slant> 00000. or 00045. degrees.

Read relocatable origin

Ec*s9^ — Reads the position of the relocatable origin.

Value returned: **<X> <Y> <terminator>**

Read reset status

Ec*s10^ — Checks to see if a reset was done since the last reset status was done. 1 indicates a full reset has been done; 0 indicates no reset has been done since last check. (Bytes 1 to 7 are returned but not used.)

Read zoom status

Ec*s8^ — Reads the zoom status. When emulating an HP 2627A, the zoom status is always a constant value.

Value returned: **001.,0CR**

NOTE: If any other parameter is sent to the terminal which does not have an assigned value, the terminal's ID is returned.

Graphics Keypad Control

Graphics keypad off/on

Ec&k0O — Turns the graphics keypad off.

Ec&k1O — Turns the graphics keypad on.

Graphics Cursor Control

Alphanumeric cursor

Ec*dQ — Turns on the alphanumeric cursor.

Ec*dR — Turns off the alphanumeric cursor.

Graphics cursor

Ec*dK — Turns on the graphics cursor.

Ec*dL — Turns off the graphics cursor (the default).

Position graphics cursor, absolute

Ec*d<X,Y>O — Positions the graphics cursor to absolute screen coordinates.

Position graphics cursor, relative

EC*d<X,Y>P — Positions the graphics cursor relative to its current position.

Display Control

Alphanumeric display

EC*dE — Turns on the alpha display.

EC*dF — Turns off the alpha display.

Clear/Set graphics memory

EC*d<pen#>A — Turns all graphics memory pixels off, or to the selected pen number, if given. The default pen color is 0 (black).

EC*d<pen#>B — Turns all graphics memory pixels on, or to the selected pen number, if given. The default pen color is 7 (white).

Graphics display

EC*dC — Turns on the graphics display.

EC*dD — Turns off the graphics display (the default).

Print graphics memory

EC&p7sF — Prints graphics memory.

Rubberband line

EC*dM — Turns on rubberband mode.

EC*dN — Turns off rubberband mode.

Graphics Text Escape Sequences

Graphics labels

EC*l<text string>CR (or LF) — Creates a label in graphics memory using current text settings. Must end with a CR, LF, or both. The <text string> may not exceed 73 characters.

Graphics text colors

EC*n<pen>X — Selects the color for graphics text. The default is the primary pen color.

Justifying graphics Text

Ec*m<origin>Q — 0-9 can be used as origin points for placement of text in relation to the current pen position.

Graphics text mode

Ec*dS — Turns on graphics text mode.

Ec*dT — Turns off graphics text mode.

Table H.2
Graphics Text Escape Sequences

Parameter	*Sequence*	*Action*
Size:	Ec*m<x>M	*Character size (1-8)*
Slant:	Ec*mO	*Slanted characters*
	Ec*mP	*Upright characters*
Justification:	Ec*m<origin>Q	*Origin (0-9)*
Orientation:	Ec*m<n>N	*Orientation*
where	*1 = default*	
	2 = rotated 90 degrees counter-clockwise	
	3 = rotated 180 degrees, upside down	
	4 = rotated 270 degrees counter-clockwise	

Selecting Drawing Modes

NOTE: See page TR-314 for HP 2627A drawing modes.

Select drawing mode

Ec*m<x>A — <x> defines which mode is selected:

Table H.3
Drawing Modes - Reflection 3

1	*Clear (turn off graphics bits)*
2	*Set (turn on graphics bits)*
3	*Complement (toggle graphics bits)*
4	*Jam (turn bits on or off according to data)*

Plotting Lines

Lift/Lower pen.
Ec*pA Ec*pB

Graphics cursor as new point.
Ec*pC

Draw point at pen position and lift pen.
Ec*pD

Set relocatable origin at pen position.
Ec*pE

Data is ASCII absolute.
Ec*pF

Data is ASCII incremental.
Ec*pG

Data is ASCII relocatable.
Ec*pH

Data is binary absolute.
Ec*pI

Data is binary short incremental.
Ec*pJ

Data is binary incremental.
Ec*pK

Data is binary relocatable.
Ec*pL

No operation is performed.
Ec*pZ

Drawing Patterns

Table H.4
Line Types

x	*Line Type*	
Ec*m1B	Solid line (default)	______________________
Ec*m2B	User defined line pattern	
Ec*m3B	Current area pattern	
Ec*m4B	Line #1	– – – – – – – – – – – – – –
Ec*m5B	Line #2	— — — — — — — — — — — —
Ec*m6B	Line #3	- - - - - - - - - - - - - - - - -
Ec*m7B	Line #4	....
Ec*m8B	Line #5	— - — - — - — - — - — - — -
Ec*m9B	Line #6	--- --- --- --- --- --- --- -
Ec*m10B	Line #7	----------------------
Ec*m11B	Point plot*	. (point plot) .

Defining Line Patterns

Define a line pattern

Ec*m<x><y>C — Defines an 8-bit segment of line pattern and a scale according to the following:

<x> is a number from 0 to 255 which, when converted to its binary form, illustrates the segment of line pattern.

<y> is a number from 1 to 16 which indicates the number of times each dot of the line pattern should be repeated.

Area Patterns

Define an area pattern

Ec*m<row 0> <row 1>...<row 7>D — Where each decimal number represents one row of the pattern.

---

* Point plot causes a single point to be plotted at the coordinates specified. This is useful for *scattergram* graphs.

Polygon area fill

Ec*pS — Starts filling polygon area.

Ec*pT — Stops polygon area fill.

Ec*pA — When the pen is lifted in the middle of a polygon fill, the previous polygon is closed and a new polygon is begun.

Rectangle fill, absolute

Ec*m<x1> <y1> <x2> <y2>E — <x1> and <y1> are the lower left corner coordinates; <x2> and <y2> are the upper right corner coordinates.

Rectangle fill, relocatable

Ec*m<x1> <y1> <x2> <y2>F — <x1> and <y1> are the relocatable coordinates of the lower left corner; <x2> and <y2> are the relocatable coordinates of the upper right corner.

Select area pattern

Ec*m<area pattern>G — where the area pattern is one of the following:

0	Current dither pattern (Reflection 7 only)
1	Solid area fill pattern
2	User-defined area fill pattern
3-10	Predefined area pattern

Relocatable Origins

Relocatable origin absolute

Ec*m<X,Y>J — X and Y are ASCII numbers between -16348 and 16383.

Relocatable origin at pen position

Ec*m<X,Y>K — X and Y are the current pen position.

Relocatable origin at graphics cursor

Ec*m<X,Y>L — The graphics cursor position is the new origin.

Setting Compatibility Mode

Compatibility mode

EC&s1p0Q — Sets scaled mode on.

EC&s0p1Q — Sets unscaled mode on.

EC&s0p0Q — Turns off compatibility mode (the default).

Straps

Disable all terminators
EC*t2A — Sets nothing as the terminator.

Page full busy, normal
EC*t0B — Normal indicates there is no keyboard locking.

Page full busy, lock
EC*t1B — The keyboard locks on a full page.

Page full break, no break
EC*t0C — No break occurs.

Page full break, break
EC*t1C — Sends a break signal to host after the 35th line.

Set carriage return terminator
EC*t0A — Sets a carriage return as the terminator (the default).

Set carriage return and EOT terminator
EC*t1A — Sets both the carriage return and EOT as terminators.

Color Graphics Escape Sequences

Color features are effective in Reflection 7 only.

Pen colors

Numbers associated with pen colors:

0 Black

1 Red

2 Green

3 Yellow

4 Blue

5 Magenta

6 Cyan

7 White

Selecting dither patterns

0 User defined dither pattern

1 Predefined dither pattern (violet)

2 Predefined dither pattern (brown)

3 Predefined dither pattern (burnt orange)

4 Predefined dither pattern (gold)

5 Predefined dither pattern (lime green)

6 Predefined dither pattern (turquoise)

7 Predefined dither pattern (red)

8 Predefined dither pattern (green)

9 Predefined dither pattern (blue)

10 Predefined dither pattern (white)

11 Predefined dither pattern (black)

Boundary pen

Ec*m<pen#>H — Selects the boundary pen color, where <pen#> is replaced with one of the eight possible pen colors.

Ec*pU — Lifts boundary pen.

Ec*pV — Lowers boundary pen.

Clearing display with background color

Ec*eb — Sets background pen and clears graphics display to background pen color.

Dither pattern

Ec*mW — Selects the predefined dither pattern.

Ec*m<d1,d2,d3>V — Defines the dither pattern, where d1-d3 define the percent of each plane to include.

Graphics text colors

Ec*n<pen#>X — Selects the color for graphics text.

Select primary pen color

Ec*m<pen#>X — Selects primary pen color.

Select secondary pen color

Ec*m<pen#>Y — Selects secondary pen color.

Select background pen color

Ec*e<pen#>B — Sets the background pen color. Black is the default.

Selecting color pairs

Ec&v<parameter> — where <parameter> can be replaced by one of the list below. The defaults (except for status) are zero. All decimal values must be in the range of 0 to 1, but are truncated to two decimal places. Other values are ignored.

0m	Red, Green, Blue color method.
1m	Hue, Saturation, Luminosity color method.
<decimal>a	Red (or hue) color value for foreground.
<decimal>b	Green (or saturation) color value for foreground.
<decimal>c	Blue (or luminosity) color value for foreground.
<decimal>x	Red (or hue) color value for background.
<decimal>y	Green (or saturation) color value for background.
<decimal>z	Blue (or luminosity) color value for background.
<0-7>i	Color pair number to be initialized.
<0-7>s	Color pair number to be selected.
<0-7>^	Color pair definition status. Because this character is considered a *capital* this must be the last parameter of the escape sequence where more than one parameter is used.

Select drawing mode

Ec*m<mode>A — Selects one of seven drawing modes. The default is JAM1.

Table H.5
Drawing Modes - Reflection 7

Mode	*Name*	*Value*	*Effect*
0	NOP	0,1	NOP
1	CLEAR1	0	NOP
		1	Pixel = Bpen
2	JAM1	0	NOP
		1	Pixel = Cpen
3	COMP1	0	NOP
		1	Pixel = NOT Pixel
4	JAM2	0	Pixel = Spen
		1	Pixel = Cpen
5	OR	0	NOP
		1	Pixel = Pixel or Cpen
6	COMP2	0	NOP
		1	Pixel = Pixel XOR Cpen XOR Bpen
7	CLEAR2	0	NOP
		1	Pixel = Pixel AND NOT Cpen

NOP No effect.

Cpen Current pen (primary drawing color or text color) or dither pixel, in which case the color of the pixel for that dither pattern is the current color.

Spen Secondary drawing pen (or color).

Bpen Background pen (or color).

Summary of HP Escape Sequences

HP Escape sequence		Function	Page
Terminal Control:			
Ec0		Home up and copy	TR-272
Ec1	Ec2	Set/Clear tab	TR-273
Ec3		Clear all tabs	TR-273
Ec4	Ec5	Set left/right margin	TR-273
Ec9		Reset margin defaults	TR-273
Ec@		Delay one second	TR-271
EcA	EcB	Move cursor up/down	TR-275
EcC	EcD	Move cursor right/left	TR-275
EcE		Hard rest	TR-272
EcF	EcH	Home down/up	TR-275
EcG		Cursor to left margin	TR-275
EcI		Horizontal tab	TR-276
EcJ	EcK	Clear display/line	TR-273
EcL	EcM	Insert/Delete line	TR-273
EcP		Delete character	TR-273
EcQ	EcR	Insert mode on/off	TR-273
EcS	EcT	Roll display up/down	TR-282
EcU	EcV	Next/Previous page	TR-282
EcW	EcX	Format mode on/off	TR-272
EcY	EcZ	Display functions on/off	TR-271
Ec[	Ec]	Start/End unprotected field	TR-272
Ec^	Ec~	Prim/Sec status request	TR-272
Ec`		Sense cursor position, relative	TR-275
Eca		Sense cursor position, absolute	TR-275
Ecb	Ecc	Unlock/Lock keyboard	TR-272
Ecd		Block transfer request	TR-271
Ecf		Disconnect	TR-271
Ecg		Soft reset	TR-272

HP Escape Sequences

HP Escape sequence		Function	Page
Ech		Home cursor	TR-275
Eci		Backtab	TR-275
Ecj	Eck	Display/Remove user-key menu	TR-274
Ec&f<parameters>		User key control	TR-109
Ec&f<color parameters>		User key control, Reflection 7	TR-111
Ecl	Ecm	Memory lock mode on/off	TR-272
Ecz		Terminal self-test	TR-272
Ec!	Ec"	132/80 column mode, (Zentec)	TR-279
Cursor Positioning:			
Ec&x<x>C		Cursor position mode	TR-275
Ec&a<col>c<row>Y		Move cursor, screen relative	TR-276
Ec&a<col>c<row>R		Move cursor, display relative	TR-276
Ec&a+/-<col>c+/-<row>R		Move cursor, display relative	TR-276
Ec&a+/-<col>c+/-<row>Y		Move cursor, cursor relative	TR-276
Configuration Control:			
Ec&k0A	Ec&k1A	Auto linefeed off/on	TR-276
Ec&k0B	Ec&k1B	Block mode off/on	TR-277
Ec&k0C	Ec&k1C	Caps lock off/on	TR-277
Ec&k0D	Ec&k1D	Bell off/on	TR-276
Ec&k0I	Ec&k1I	Previous/No parity	TR-278
Ec&k0K	Ec&k1K	Keyboard lock off/on	TR-277
Ec&k0J	Ec&k1J	Set frame rate, 60/50	TR-277
Ec&k0L	Ec&k1L	Local echo off/on	TR-278
Ec&k0M	Ec&k1M	Modify all mode off/on	TR-278
Ec&k0N	Ec&k1N	SPOW latch off/on	TR-278
Ec&k0P	Ec&k1P	Caps mode off/on	TR-277
Ec&k0R	Ec&k1R	Remote mode off/on	TR-278
Ec&k0\	Ec&k1\	Switch to HP/VT mode	TR-272
Ec&q0L	Ec&q1L	Unlock/Lock configuration	TR-278
Ec&s0A	Ec&s1A	Transmit functions NO/YES	TR-278
Ec&s0B	Ec&s1B	SPOW NO/YES	TR-278
Ec&s0C	Ec&s1C	Inhibit EOL wrap NO/YES	TR-277
Ec&s0D	Ec&s1D	Line/Page mode	TR-278
Ec&s0G	Ec&s1G	Inhibit handshake off/on	TR-277
Ec&s0H	Ec&s1H	Inhibit DC2 NO/YES	TR-277
Ec&s0N	Ec&s1N	Esc xfer to printer NO/YES	TR-277
Ec&s0Z	Ec&s1Z	Check parity NO/YES	TR-277

HP Escape sequence	Function	Page
Data Operations:		
Ec&p<a>d<Y>	Copy data	TR-279
Ec&p4^ or Ec&p6^	Printer status	TR-280
Ec&p<x>p<y>u<z>C	Printer control action	TR-279
Ec&p<num>p20C	Record mode	TR-280
Ec&p<a>d<b>d<c>d<x>W<string>	Transfer binary data	TR-280
Ec&p<a>d<b>d<c>dW<data string>	Transfer ASCII data	TR-280
Forms Cache Sequences:		
Ec&p9^	Forms buffer status request	TR-284
Ec&p<form#>p9^	Forms buffer status request	TR-284
Ec&p9u<f#>p<len>L<con>	Store form	TR-284
Ec&p9u<f#>p<<con>>L	Store form, length unknown	TR-284
Ec&p9u<form#>pF	Display form	TR-284
Ec&p9u<form#>pL	Purge form	TR-284
Ec&q4te2{<blocks>L	Allocate forms buf.(restricted)	TR-284
HP 2393, 2394, 2397, 2624 Emulation Only:		
Ec_	Write nondisplaying terminator	TR-285
Ech	Home cursor (ignore xmit-only)	TR-275
Ec{	Start transmit-only field	TR-273
Ec|	Erase nondisplaying terminator	TR-285
Ec&k0Z	Set TRANSMIT to ALL	TR-285
Ec&k1Z	Set TRANSMIT to MODIFIED	TR-285
Ec&q8te1{0R	Set RETURN=ENTER NO	TR-287
Ec&q8te1{1R	Set RETURN=ENTER YES	TR-287
Ec&q1te1{0G	Set xmit pacing NONE	TR-285
Ec&q1te1{1G	Set xmit pacing XON/XOFF	TR-285
Ec&q1te1{0H	Set receive pacing NONE	TR-285
Ec&q1te1{1H	Set receive pacing XON/XOFF	TR-285
Ec&s0J	Set Autoterm off	TR-284
Ec&s1J	Set Autoterm on	TR-284
Ec&s0K	Set Clearterm off	TR-285
Ec&s1K	Set Clearterm on	TR-285

HP Escape sequence	Function	Page
2624B and 2394A Only:		
Ec&q<m>te0{<0/1>A	Set transmit functions NO/YES	TR-287
Ec&q<m>te0{<0/1>B	Set SPOW NO/YES	TR-287
Ec&q<m>te0{<0/1>C	Set Inhibit EOL Wrap NO/YES	TR-286
Ec&q<m>te0{<0/1>D	Set Line/Page mode	TR-286
Ec&q<m>te0{<0/1>G	Set Inhibit Handshake NO/YES	TR-286
Ec&q<m>te0{<0/1>H	Set Inhibit DC2 NO/YES	TR-286
Ec&q<m>te0{<0/1>J	Turn autoterm off/on	TR-285
Ec&q<m>te0{<0/1>K	Turn clearterm off/on	TR-286
Ec&q<m>te1{<0/1>A	Turn auto linefeed off/on	TR-285
Ec&q<m>te1{<0/1>B	Turn block mode off/on	TR-286
Ec&q<m>te1{<0/1>C	Turn caps lock off/on	TR-285
Ec&q<m>te1{<0/1>L	Turn local echo off/on	TR-287
Ec&q<m>te1{<0/1>M	Turn modify all mode off/on	TR-287
Ec&q<m>te1{<0/1>R	Turn remote mode off/on	TR-287
Ec&q<m>te2{<x>D	Set alternate character set	TR-287
Ec&q<m>te2{<x>F	Field separator character	TR-286
Ec&q<m>te2{<x>L	Set forms buffer size	TR-286
Ec&q<m>te2{<x>R	Set block terminator character	TR-286
Ec&q<m>te2{<x>Z	Set transmit ALL/MODIFIED	TR-287
Ec&q8te0{<0>D	Turn bell off	TR-285
Ec&q8te0{<1>D	Turn bell on	TR-285
Ec&q8te1{<x>A	Set Return definition 1st char	TR-287
Ec&q8te1{<x>B	Set Return definition 2nd char	TR-287
Ec&q8te1{<0>R	Set RETURN=ENTER NO	TR-287
Ec&q8te1{<1>R	Set RETURN=ENTER YES	TR-287
Ec&q8te1{<0>T	Set TAB=SPACES NO	TR-287
Ec&q8te1{<1>T	Set TAB=SPACES YES	TR-287
Display Control and Enhancements:		
Ec&d<x>	Display enhancements	TR-281
Ec&ds<x>	Display enhance., security	TR-281
Ec&r<x>U Ec&r<x>D	Roll up/down <x> lines	TR-282
Ec&r<x>L Ec&r<x>R	Roll left/right <x> columns	TR-283
Ec&q3t0{<x>W	Page width	TR-282
Ec&w5f<x>W	Page width (active value only)	TR-282
Ec&w12F Ec&w13F	Display on/off	TR-282

HP Escape sequence		Function	Page
Character Sets and Labels:			
Ec)B		Select line drawing set	TR-283
Ec)@		Select base set	TR-283
Ec)A	Ec)C	Select space set	TR-283
Ec&jA		Display modes keys	TR-283
Ec&jB		Display and enable user keys	TR-274
Ec&j@		Enable user keys, no labels	TR-274
Ec&jR	Ec&jS	Enable/Disable sys, modes, user	TR-283
Ec&j<x>L<string>		Replace labels with message	TR-283
Ec&jC		Remove message, replace labels	TR-283
Ec&f...	Ec&f<x>E	Define/Trigger user keys	TR-274
Reflection Escape Sequences:			
Ec&bR		Connection reset	TR-288
Ec&oA		5-digit absolute cursor position	TR-288
Ec&oC<command>CR		Reflection command	TR-288
Ec&oX		Clear typeahead buffer	TR-288
Ec*s12345^		Request 9-char serial number	TR-288
Ec*s12346^		Request 11-char serial number	TR-288
Ec*s12347^		Request 14-char serial number	TR-288
Ec*s33^		Request serial number (PC 2622)	TR-288
Graphics sequences:			
Ec&k0O	Ec&k1O	Graphics keypad off/on	TR-306
Ec&p7sF		Print graphics memory	TR-306
Ec*dA	Ec*dB	Clears/Sets graphics memory	TR-306
Ec*dC	Ec*dD	Graphics display on/off	TR-306
Ec*dE	Ec*dF	Turns alpha display on/off	TR-306
Ec*dK	Ec*dL	Turns graphics cursor on/off	TR-307
Ec*dM	Ec*dN	Turn on/off rubberband line	TR-306
Ec*d<X,Y>O		Move graphics cursor absolute	TR-306
Ec*d<X,Y>P		Move graphics cursor relative	TR-306
Ec*dQ	Ec*dR	Turns on/off alpha cursor	TR-302
Ec*dS	Ec*dT	Turns on/off graphic text mode	TR-306

HP Escape sequence	Function	Page
Drawing Modes and Patterns:		
$^E{}_C$*l<text>$^C{}_R$	Graphics labels	TR-306
$^E{}_C$*m0A	No effect	TR-307
$^E{}_C$*m1A	Clear (turn off graphics bits)	TR-307
$^E{}_C$*m2A	Set (turn on graphics bits)	TR-307
$^E{}_C$*m3A	Complement (toggle bits)	TR-307
$^E{}_C$*m4A	Jam (according to data)	TR-307
$^E{}_C$*m1B	Solid line (default)	TR-309
$^E{}_C$*m2B	User-defined line pattern	TR-309
$^E{}_C$*m3B	Current area pattern	TR-309
$^E{}_C$*m<n>B	Select line type (4-10)	TR-309
$^E{}_C$*m11B	Point plot	TR-309
$^E{}_C$*m<x><y>C	Define a line pattern	TR-309
$^E{}_C$*m<row 0>...<row 7>D	Define an area pattern	TR-309
$^E{}_C$*m<lower> <upper>E	Area fill absolute	TR-310
$^E{}_C$*m<lower> <upper>F	Area fill relocatable	TR-310
$^E{}_C$*m1G	Solid area pattern	TR-310
$^E{}_C$*m2G	User-defined area fill pattern	TR-310
$^E{}_C$*m<n>G	Predefined area pattern (3-10)	TR-310
$^E{}_C$*m<X,Y>J	Relocatable origin absolute	TR-310
$^E{}_C$*mK	Relocatable origin at pen	TR-310
$^E{}_C$*mL	Reloc. origin at graphics cursor	TR-310
Graphics text and drawing modes:		
$^E{}_C$*m<x>M	Character size (1-8)	TR-307
$^E{}_C$*m<origin>Q	Justification (0-9)	TR-307
$^E{}_C$*m1N	Rotated 0 degrees	TR-307
$^E{}_C$*m2N	Rotated 90 degrees	TR-307
$^E{}_C$*m3N	Rotated 180 degrees	TR-307
$^E{}_C$*m4N	Rotated 270 degrees	TR-307
$^E{}_C$*mO	Slanted characters	TR-307
$^E{}_C$*mP	Upright characters	TR-307
$^E{}_C$*m<default flag>R	Restores graphics defaults	TR-301
$^E{}_C$*pA $^E{}_C$*pB	Lifts/Lowers pen	TR-308
$^E{}_C$*pC	Uses graphics cursor as point	TR-308
$^E{}_C$*pD	Draws point at pen, lifts pen	TR-308
$^E{}_C$*pE	Sets relocatable origin at pen	TR-308

HP Escape sequence	Function	Page
Ec*pF	Data is ASCII absolute	TR-308
Ec*pG	Data is ASCII incremental	TR-308
Ec*pH	Data is ASCII relocatable	TR-308
Ec*pI	Data is binary absolute	TR-308
Ec*pJ	Data is binary short incremental	TR-308
Ec*pK	Data is binary incremental	TR-308
Ec*pL	Data is binary relocatable	TR-308
Ec*pS Ec*pT	Start/End polygon area fill	TR-310
Ec*pZ	No operation is performed	TR-308
Graphics status:		
Ec*s1^	Read terminal ID	TR-303
Ec*s2^	Read pen position	TR-303
Ec*s3^	Read graphics cursor position	TR-303
Ec*s4^	Read position, wait for key	TR-303
Ec*s5^	Read display size	TR-303
Ec*s6^	Read graphics capabilities	TR-302
Ec*s7^	Read graphics text status	TR-304
Ec*s8^	Read zoom status	TR-304
Ec*s9^	Read relocatable origin	TR-304
Ec*s10^	Read reset status	TR-302
Ec*s11^	Read area shading	TR-302
Ec*s12^	Read modifications	TR-302
Compatibility mode:		
Ec&s1p0Q	Scaled compatibility mode on	TR-311
Ec&s0p1Q	Unscaled compatibility mode on	TR-311
Ec&s0p0Q	Compatibility mode off	TR-311
Ec*t0A	Set carriage return terminator	TR-311
Ec*t1A	Set return and EOT terminator	TR-311
Ec*t2A	Disable all terminators	TR-311
Ec*t0B	Page full busy, normal	TR-311
Ec*t1B	Page full busy, lock	TR-311
Ec*t0C	Page full break, no break	TR-311
Ec*t1C	Page full break, break	TR-311
Ec*wR	Graphics hard reset	TR-301

HP Escape sequence	Function	Page
Color graphics sequences:		
Ec*d<pen#>A	Clears graphics memory	TR-306
Ec*e<pen#>B	Select background pen color	TR-312
Ec*m<mode>A	Select drawing mode	TR-313
Ec*m<pen#>H	Set area boundary pen (1-8)	TR-312
Ec*m<d1,d2,d3>V	Define dither pattern	TR-312
Ec*m0G	Dither pattern as area pattern	TR-310
Ec*m1V	User-defined dither pattern	TR-312
Ec*m<x>V	Predef. dither patterns (2-12)	TR-312
Ec*mW	Select predef. dither pattern	TR-312
Ec*m<pen#>X	Select primary pen color	TR-313
Ec*m<pen#>Y	Select secondary pen color	TR-313
Ec*n<pen#>X	Graphics text colors	TR-314
Ec*pS Ec*pT	Begin/End area fill	TR-310
Ec*pU Ec*pV	Lift/Lower boundary pen	TR-312
Color methods:		
Ec&v0m Ec&v1m	Selecting HGB/HSL method	TR-313
Ec&v<decimal><a,b, or c>	Red(a), Green(b), Blue(c) or Hue(a), Saturation(b), or Luminosity(c) foreground	TR-313
Ec&v<decimal><x,y, or z>	Red(x), Green(y), Blue(z) or Hue(x), Saturation(y), or Luminosity(z) background	TR-313
Ec&v<0-7>i	Color pair to be initialized	TR-313
Ec&v<0-7>s	Color pair to be selected	TR-313
Ec&v<0-7>^	Color pair definition status	TR-313

HP Character Sets

The ASCII Character Set

DECIMAL VALUE	⇨	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	‼	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	|
13	D	♪	↔	-	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

DECIMAL VALUE	KEYSTROKE	SCREEN DISPLAY	DEFINITION
0	^@		Null
1	^a	☺	Start of heading
2	^b	☻	Start of text
3	^c	♥	End of text
4	^d	♦	End of transmission
5	^e	♣	Enquiry
6	^f	♠	Acknowledge
7	^g	•	Bell
8	^h	◘	Backspace
9	^i	○	Horizontal tabulation
10	^j	◙	Line feed
11	^k	♂	Vertical tabulation
12	^l	♀	Form feed
13	^m	♪	Carriage return
14	^n	♫	Shift out
15	^o	☼	Shift in
16	^p	►	Data link escape
17	^q	◄	Device control 1 (XON)
18	^r	↕	Device control 2
19	^s	‼	Device control 3 (XOFF)
20	^t	¶	Device control 4
21	^u	§	Negative acknowledge
22	^v	▬	Synchronous idle
23	^w	↨	End of transmission block
24	^x	↑	Cancel
25	^y	↓	End of medium
26	^z	→	Substitute
27	^[	←	Escape
28	^\	∟	File Separator
29	^]	↔	Group separator
30	^^	▲	Record separator
31	^_	▼	Unit Separator

DECIMAL VALUE	SCREEN DISPLAY	DEFINITION
32		Space (blank)
33	!	Exclamation point
34	"	Quotation mark
35	#	Number sign
36	$	Dollar sign
37	%	Percent sign
38	&	Ampersand
39	'	Apostrophe
40	(	Opening parenthesis
41	)	Closing parenthesis
42	*	Asterisk
43	+	Plus
44	,	Comma
45	-	Hyphen and minus sign
46	.	Period and decimal point
47	/	Slant
48	0	Zero
49	1	One
50	2	Two
51	3	Three
52	4	Four
53	5	Five
54	6	Six
55	7	Seven
56	8	Eight
57	9	Nine
58	:	Colon
59	;	Semicolon
60	<	Less than
61	=	Equals
62	>	Greater than
63	?	Question mark

DECIMAL VALUE	SCREEN DISPLAY	DEFINITION
64	@	Commercial "at"
65	A	Uppercase A
66	B	Uppercase B
67	C	Uppercase C
68	D	Uppercase D
69	E	Uppercase E
70	F	Uppercase F
71	G	Uppercase G
72	H	Uppercase H
73	I	Uppercase I
74	J	Uppercase J
75	K	Uppercase K
76	L	Uppercase L
77	M	Uppercase M
78	N	Uppercase N
79	O	Uppercase O
80	P	Uppercase P
81	Q	Uppercase Q
82	R	Uppercase R
83	S	Uppercase S
84	T	Uppercase T
85	U	Uppercase U
86	V	Uppercase V
87	W	Uppercase W
88	X	Uppercase X
89	Y	Uppercase Y
90	Z	Uppercase Z
91	[	Opening bracket
92	\	Reverse slant
93	]	Closing bracket
94	^	Circumflex
95	_	Underscore

DECIMAL VALUE	SCREEN DISPLAY	DEFINITION
96	`	Grave accent
97	a	Lowercase a
98	b	Lowercase b
99	c	Lowercase c
100	d	Lowercase d
101	e	Lowercase e
102	f	Lowercase f
103	g	Lowercase g
104	h	Lowercase h
105	i	Lowercase i
106	j	Lowercase j
107	k	Lowercase k
108	l	Lowercase l
109	m	Lowercase m
110	n	Lowercase n
111	o	Lowercase o
112	p	Lowercase p
113	q	Lowercase q
114	r	Lowercase r
115	s	Lowercase s
116	t	Lowercase t
117	u	Lowercase u
118	v	Lowercase v
119	w	Lowercase w
120	x	Lowercase x
121	y	Lowercase y
122	z	Lowercase z
123	{	Opening (left) brace
124	|	Vertical line
125	}	Closing (right) brace
126	~	Tilde
127	△	Delete

Roman 8 Character Set

Roman 8 characters cannot all be displayed on the IBM PC screen. The following table shows the graphic character which is actually displayed on the PC screen.

Character Code	*ROMAN8 Graphic*	*IBM PC Graphic*	*Character Code*	*ROMAN8 Graphic*	*IBM PC Graphic*
161	*À*	*A*	*182*	*Ñ*	*Ñ*
162	*Â*	*A*	*183*	*ñ*	*ñ*
163	*È*	*E*	*184*	*¡*	*¡*
164	*Ê*	*E*	*185*	*¿*	*¿*
165	*Ë*	*E*	*186*	¤	*ø*
166	*Î*	*I*	*187*	*£*	*£*
167	*Ï*	*I*	*188*	*¥*	*¥*
168	´	’	*189*	*§*	*§*
169	`	‘	*190*	*f*	*f*
170	ˆ	^	*191*	*¢*	*¢*
171	¨	■	*192*	*â*	*â*
172	˜	˜	*193*	*ê*	*ê*
173	*Ù*	*U*	*194*	*ô*	*ô*
174	*Û*	*U*	*195*	*û*	*û*
175	*₤*	*£*	*196*	*á*	*á*
176	—	—	*197*	*é*	*é*
177			*198*	*ó*	*ó*
178			*199*	*ú*	*ú*
179	°	°	*200*	*à*	*à*
180	*Ç*	*Ç*	*201*	*è*	*è*
181	*ç*	*ç*	*202*	*ò*	*ò*

Character Code	ROMAN8 Graphic	IBM PC Graphic	Character Code	ROMAN8 Graphic	IBM PC Graphic
203	ù	ù	229	Í	I
204	ä	ä	230	Ì	I
205	ë	ë	231	Ó	O
206	ö	ö	232	Ò	O
207	ü	ü	233	Õ	O
208	Å	Å	234	õ	o
209	î	î	235	Š	S
210	Ø	Ö	236	š	s
211	Æ	Æ	237	Ú	U
212	å	å	238	Ÿ	Y
213	í	í	239	ÿ	ÿ
214	ø	ö	240	Þ	Θ
215	æ	æ	241	þ	Θ
216	Ä	Ä	242		
217	ì	ì	243		
218	Ö	Ö	244		
219	Ü	Ü	245		
220	É	É	246	—	—
221	ï	ï	247	¼	¼
222	ß	ß	248	½	½
223	Ô	O	249	ª	ª
224	Á	A	250	º	º
225	Ã	A	251	«	«
226	ã	a	252	■	■
227	Đ	D	253	»	»
228	đ	d	254	±	±

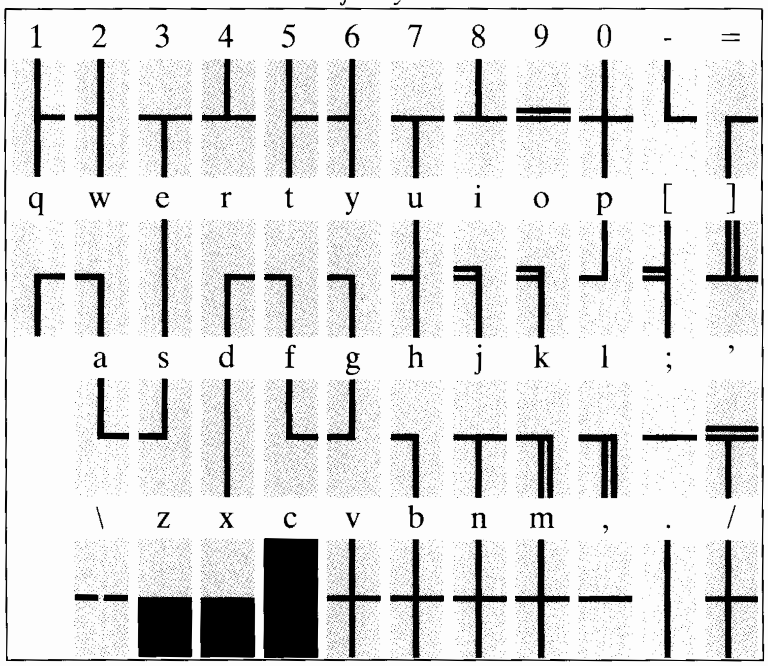
The HP Line Drawing Character Set

Reflection has an alternate line drawing character set, which is useful for creating forms. The HP line drawing character set is a group of 28 graphic elements that can be used in combination to create forms and other lined images on the screen. The charts in this reference show an enlarged version of each of the 28 characters in the set, along with its corresponding keystrokes. Each letter key generates only one line drawing character, whether Shift is depressed or not. However, the same character may be generated by more than one letter key. Most of the numeric and punctuation keys can generate two line drawing characters: one with Shift and a different one when only the character key is pressed.

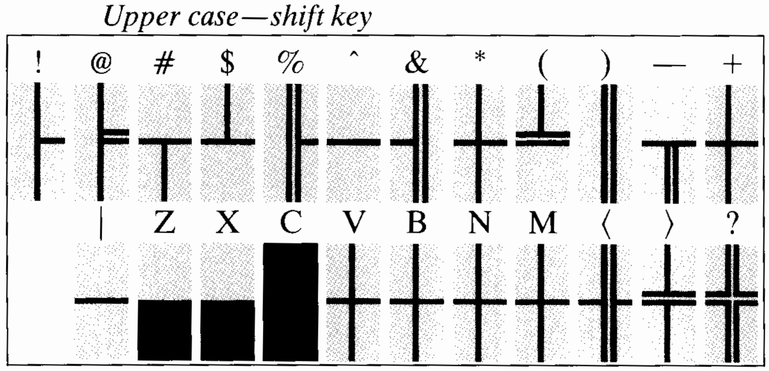
Press Ctrl-N to access the line drawing character set; press Ctrl-O to return to normal characters. Reflection reverts to the normal character set when the cursor moves to a new row.

Table J.1
HP Line Drawing Character Set

Lower case—no shift key



Upper case—shift key



VT Keyboard Functions

Quick Reference

Table K.1
VT Keyboard Functions

Function	*Keystroke*
Clear display	*Alt*-*J*
Clear line	*Alt*-*K*
Clear print buffer	*Ctrl*-*F1*
Command line	*Alt*-*Y*
Config keys	*Alt*-*C*
Define user keys	*Ctrl*-*F9*
Delete character	*Del*
Delete line	*Alt*-*D*
Disconnect	*Ctrl*-*F10*
Error recap	*Alt*-*E*
Exit to DOS	*Alt*-*X*
Extend key	*Alt*-*Z*
Hard reset	*Alt*-*R*
Help	*Alt*-*H*
Hold screen	*ScrollLock*
Home down	*Ctrl*-*End*
Home up	*Ctrl*-*Home*

Function	*Keystroke*
Insert character	*Ins*
Insert line	*Alt*-*I*
Keypad	*NumLock*
Modes keys	*Alt*-*M*
Next page	*Ctrl*-*PgDn*
Previous page	*Ctrl*-*PgUp*
Print screen	*Alt*-*PrtSc*
Reset handshake	*Alt*-*Q*
Reset Hayes internal modem	*Ctrl*-*F8*
Roll down	*Ctrl*-*Up*
Roll left	*Ctrl*-*Right*
Roll right	*Ctrl*-*Left*
Roll up	*Ctrl* *Down*
Soft reset	*Alt*-*S*
System keys	*F10*
Toggle print logging	*Ctrl*-*PrtSc*
User keys	*F9*

Function Descriptions

This reference describes local keyboard functions in VT mode. These functions are also listed on the VT Help screen, accessed by Alt-H. None of the functions described here transmits characters to the host computer; for a description of keystrokes that transmit characters to a host, see page TR-365, *Keyboard Generated Codes*.

Clear display (Alt-J)

Availability: In local mode, except at a menu

Erases display memory from the cursor position through the end of display memory.

Clear line (Alt-K)

Availability: In local mode, except at a menu

Erases all characters from the cursor through the end of the line.

VT Reference

Clear print buffer (Ctrl-F1)

Availability: Always

Clears the contents of Reflection's internal printer buffer. The printer will continue to print if it has a built-in buffer.

Command mode (Alt-Y)

Availability: Always, except at a menu

Causes Reflection to enter command mode and display the command entry line in row 25. Exit command mode by pressing Esc or by pressing Return with the command line empty.

Config keys (Alt-C)

Availability: Always

Brings up the configuration keys.

Connection Reset (Ctrl-F8)

Availability: In terminal mode

- Resets a Hayes Smartmodem 1200B to its power-on state. Sets the *out 1* bit of the modem control register to 1 for at least 50 milliseconds, as described in the Hayes manual.
- Suspends the current LAN session.
- Powers down the communications ports on the HP Portable Plus.

Define user keys (Ctrl-F9)

Availability: In terminal mode

Brings up the user-key definition menu.

Delete character (Del)

Availability: In local mode, except at a menu

Deletes the character at the cursor position and moves all following characters on the line one position to the left.

When the cursor is to the left of the right margin, only the characters up to the right margin are moved left, and a space is inserted at the right margin.

When the cursor is to the right of the right margin, all characters to the right of the cursor are moved left, and the line is shortened by one column.

This key does not transmit anything. Its action is entirely local to the PC.

Delete line ([Alt]-[D])

Availability: In local mode

Deletes the cursor line and moves all subsequent lines in display memory up one row. The cursor then moves to the left margin.

Disconnect ([Ctrl]-[F10])

Availability: During datacomm, when the PC and the host are connected

Disconnect functions differently depending on the type of host connection you have. If you are connected via a serial port, Reflection lowers DTR for 200 milliseconds. This function disconnects a properly configured modem. Note that some modems can be configured not to respond to the disconnect signal.

If you are connected via LAN, the appropriate LAN disconnect signal is transmitted with this keystroke. Also see DISCONNECT-ON-EXIT in appendix A, *Command Language Keywords* on page TR-179.

Error recap ([Alt]-[E])

Availability: In terminal mode (and file transfer)

Displays a menu of errors that can occur in data communication. A count is displayed for each type of error on the current datacomm port. See page UM-201 for information on error diagnosing.

Exit to DOS ([F8])

Availability: From system keys

Terminates Reflection and exits to DOS. If the hot-key has been used, [F8] puts Reflection into the background; you must use [Alt]-[X] to unload Reflection from memory. Also see *Background Operations* on page 23.

When the PC is online over a modem, Reflection does not automatically disconnect. To disconnect before exiting Reflection, enter [Ctrl]-[F10], power off, or unplug your modem. See *Using an Auto-Dial Modem with Reflection* on page UM-55.

When the PC is online over a modem, Reflection does not automatically disconnect. To disconnect before exiting Reflection, enter [Ctrl]-[F8], power off, or unplug your modem. See *Using an Auto-Dial Modem with Reflection* on page UM-55. Also see DISCONNECT-ON-EXIT under SET.

Extend key ([Alt]-[Z])

Availability: Whenever data can be entered

The character typed after the extend sequence matches the keyboard mapping shown on page UM-99. Gives access to the Roman 8 character set as shown on page TR-328. Also see *National Keyboards and Characters* on page TR-97.

Hard reset ([Alt]-[R])

Availability: Always, except during file transfer

A hard reset does the following:

- Emits a beep.
- Cancels any escape sequence processing, including record mode.
- Clears display memory and homes the cursor.
- Sets all terminal configuration parameters to the last activated values.
- Initializes the serial communications ports and clears the receive buffer.

- Clears the keyboard buffer and sets the *host prompt received* switch on.
- Switches off display functions, insert, and caps lock.
- Sets character sets, display enhancements, and selective erase to their default values.
- Sends an XON to the host when the **Receive pacing** parameter has a value of *XON/XOFF*.
- Displays the **pf keys** labels and enters terminal mode.

Help (Alt-H)

Availability: In terminal mode

Displays the Help screen. Return to the exact point at which you entered the Help screen with **RESUME** (F8).

Hold screen (ScrollLock)

Availability: In terminal mode, when **Receive pacing** has a value of *XON/XOFF*

Hold screen alternately stops and starts the process of taking data from the receive buffer and displaying it on the screen. It differs from the Ctrl-Q/Ctrl-S combination in that it does not immediately send the XON and XOFF characters to the host. Instead, it depends on the receive pacing mechanism to send the XON/XOFF as they are needed so that the receive buffer does not overflow. An H appears between the F4 and F5 labels when hold screen is active.

Home up/down (Ctrl-Home and Ctrl-End)

Availability: In terminal mode

Home up rolls display memory down so that the first line of display memory occupies the first row of the screen. The cursor moves to the left margin of the top row.

Home down causes display memory to roll up, so the last line in display memory is visible on the screen. The cursor moves to the last position in display memory. When top and bottom margins are specified, or **Multi-page** mode is *NO*, Home up and Home down are disabled.

Insert char (Ins)

Availability: In local mode

Toggles insert character mode on and off. When on, an I appears between the F4 and F5 labels. The cursor becomes a blinking block when insert mode is on.

Characters are inserted at the cursor location, and all characters to the right of the cursor move right one column. The character at the right margin is deleted from the line to make room for the other characters to be shifted right.

If the cursor is beyond the right margin when the inserted character is typed, the character in column 80 is deleted.

Insert line (Alt-I)

Availability: In local mode

Inserts a blank line in the screen row that contains the cursor. The old contents of that row and all the following rows in display memory are pushed down one row to make room for the new row. The cursor then moves to the left margin of the new (blank) row.

If **Multi-page** mode is *YES*, and top and bottom margins are not defined, lines rolled off the screen are pushed down into display memory. Otherwise, lines are deleted.

Keypad (NumLock)

Availability: Always

When NumLock is on, the keypad generates numbers. When it's off, the keypad keys control cursor movement.

Modes keys (Alt-M)

Availability: Always

Brings up the modes keys and switches Reflection to local mode.

Next/Prev page (Ctrl-PgDn, Ctrl-PgUp)

Availability: In terminal mode

Next page displays the next page of display memory. A page consists of 24 lines of display memory. If the last line of display memory is on the screen when the Next page function is performed, the operation has no effect.

Prev page displays the previous page of display memory. If the first line of display memory is on the screen when the Prev page operation is performed, the operation has no effect.

When **Multi-page** mode is *NO*, or top and bottom margins are specified, Next page and Prev page are disabled.

Print screen (Alt-PrtSc)

Availability: Always

Prints a copy of the currently displayed screen, not including the function-key labels, and sends a form feed to the printer.

Roll screen left/right (Ctrl-Left/Right)

Availability: In terminal mode

When the right margin is set beyond column 80, rolling left brings text from beyond the right border of the screen into view. Each instance shifts a specified number of columns off the left side of the screen, and the same number of columns onto the right side. The number of columns scrolled is specified in the **Columns per horizontal scroll** field of the Terminal Configuration, Page 2 menu.

When the right margin column is visible on the screen, rolling left has no effect.

When the right margin is set beyond column 80, rolling right brings text from beyond the left border of the screen into view. Each instance shifts a specified number of columns off the right side of the screen and the same number of columns onto the left. When column 1 is on the screen, rolling right has no effect.

Roll screen up/down (Ctrl-Up/Down)

Rolling down moves the text in display memory down one line and brings the previous line of text onto the top row of the screen. When the first line of display memory is in the top row of the screen, rolling down has no effect.

Rolling up moves the text in display memory up one line and brings the next line of text onto the bottom row of the screen. When the last line of display memory is in the top row of the screen, rolling up has no effect.

Rolling up/down is disabled if **Multi-page** mode is *NO* or top or bottom margins have been set.

Soft reset (Alt-S)

Availability: Always, except in file transfer

A soft reset, also called *clear communications*, does the following:

- Emits a beep.
- Cancels any escape sequence processing, including record mode.
- Initializes the serial communications ports to their last activated values and clears the receive buffer.
- Cancels any print operation in progress and clears the printer buffer.
- Clears the keyboard buffer and sets the *host prompt received* switch on.
- Sends an XON to the host if the **Receive pacing** configuration parameter has a value of *XON/XOFF*.

System keys (F10)

Availability: Always

Brings up the system keys. F10 also places Reflection in terminal mode, erases any menu screen, and displays the most recent page of display memory.

Toggle print logging (Ctrl-PrtSc)

Availability: In terminal mode

Toggles **TO PRINTER** and **LOG BOTTOM** on and off. Press Ctrl-PrtSc to route data entered at the keyboard and received from datacomm immediately to the printer. Make sure that **AUTO LF** is on: press F4, **modes keys,** from the system menu and press F8 until an asterisk appears. Press Ctrl-PrtSc again to toggle print logging off.

User keys (Alt-U or F9)

Availability: Always

Brings user keys to the screen. Subsequent pressing of F9 toggles the height of user keys. The effect of this function is temporary; the next time the user keys are displayed, the labels revert to the configured number of lines.

The initial height of the user-key labels and the ability to present them as the startup labels can be configured. See *VT Terminal Configuration* on page TR-57.

VT Escape Sequences

An *escape sequence* is a sequence of ASCII characters preceded by the C0 control character ^{E}C (18 hex). An escape sequence uses only 7-bit characters and is therefore applicable in both a 7-bit and 8-bit environment.

A *control sequence* is a sequence of ASCII characters preceded by the C1 control character CSI (98 hex). A control sequence can be used in a 7-bit environment by replacing the CSI character with its 7-bit eqivalent ^{E}C[.

A *device control string* is a string of characters used for control purposes which is preceded by the C1 control character DCS (90 hex) and terminated by C1 control character ST (9C hex). The 7-bit equivalents of DCS and ST are ^{E}CP and ^{E}C/, respectively.

A Word about Notation

In this reference, escape sequence parameters which you must supply are enclosed in angled brackets. For example, the **<n>** parameter in the escape sequence, **CSI <n>**A, specifies the number of lines you want the cursor to move. The parameter **<n>** or **<num>** usually represents a repetition count or a screen coordinate.

Some sequences have been combined to ease locating related ones. When coding escape sequences, take care to distinguish between upper and lower case characters.

Cursor and Screen Control

Cursor Backtab

CSI <n>Z — Moves the cursor backward along the active line to the **<n>th** preceding tab position. The cursor stops at column 1 if the **<n>th** tab stop is not found.

Cursor up/down

CSI <n>A — Moves cursor up **<n>** lines in the same column. If the cursor is within the scrolling region, it stops at the top margin. If the cursor is above the top margin, it stops at the top of the screen.

CSI <n>B — Moves the cursor down **<n>** lines in the same column. If the cursor is within the scrolling region, it stops at the bottom margin. If the cursor is below the bottom margin, it stops at the bottom of the screen.

Cursor right/left

CSI <n>C — Moves the cursor right **<n>** columns. If the screen width is greater than 80 columns, the display is scrolled, if necessary, to bring the cursor into view. If the cursor is at the right margin, it does not move.

CSI <n>D — Moves the cursor left **<n>** columns. If the screen width is greater than 80 columns, the display is scrolled, if necessary, to bring the cursor into view. If the cursor is at the left margin, it does not move.

Cursor position

CSI <r>;<c>H — **CSI <r>;<c>f** may also be used. Moves the cursor to row **<r>** and column **<c>**. Normally row 1, column 1 is the upper left corner of the screen. However, if top and bottom margins have been defined and origin mode is set, then the top row of the scrolling region is row 1. The cursor is never positioned beyond the bottom margin or right margin. If the screen width is greater than 80 columns, the display is scrolled horizontally, if necessary to bring the cursor into view.

Horizontal Position Absolute

CSI <n>‘ — Moves the cursor to the <n>th column without changing the current row.

Horizontal Position Relative

CSI <n>a — Moves the cursor right <n> columns without changing the current row.

Index down

$^{E}{}_{C}$D — Moves the cursor down one row. If the cursor is at the bottom margin, the display scrolls up.

Next line

^{E}cE and CSI <n>E — Moves the cursor to column 1 of the next (or **<n>th**) row. If the cursor is at the bottom margin, the display scrolls up.

Previous line

CSI <n>F — Moves the cursor to the first position of the **<n>th** previous line. If line **<n>** is above the first line, a roll down is performed.

Restore/Save cursor

^{E}c8 — Restores the states saved by ^{E}c7. If nothing has been saved, the power on defaults are used.

^{E}c7 — Saves cursor position, display enhancements, character set, end-of-line wrap state, selective erase state, and origin mode.

Reverse index

^{E}cM — Moves the cursor up one row. If the cursor is at the top margin, the display scrolls down and a blank line is inserted.

Vertical position absolute

CSI <n>d — Moves the cursor to the <n>th line without changing the column.

Vertical position relative

CSI <n>e — Moves the cursor down <n> lines without changing the column.

Top and Bottom Margins

NOTE: The scrolling region is the area between the top and bottom margins. This is the area that moves during vertical scrolling. When the margins are selected, the cursor moves to the home position as determined by the origin mode. Rows are counted from 1.

Set top and bottom margins (scrolling region)

CSI <t>;<b>r — Sets the top margin to row <t> and the bottom margin to row **<b>**. The **<t>** parameter must be at least 2 less than **<b>**.

Setting a top or bottom margin has the same effect as setting **Multi-page** mode to *NO*; i.e., lines that are rolled off the screen are deleted from display memory. In addition, the following functions are disabled:

Next/Prev page
Scroll up/down
Home up/down

Terminal Modes

NOTE: The last character in every other sequence is a lower-case L, not a number 1.

Auto linefeed off/on

CSI 20h — Return and keypad Enter send both a CR and a LF. A received LF, FF, or VT character moves the cursor to the first column of the next line. Auto linefeed is turned OFF.

CSI 20l — Return and keypad Enter send only a CR character. A received LF, FF, or VT character moves the cursor down one line in the current column. Auto linefeed is turned ON.

Columns 132/80

CSI ?3h — Sets the left margin to 1 and the right margin to 132.

CSI ?3l — Sets the left margin to 1 and the right margin to 80.

Cursor keys – application

CSI ?1h — The cursor keys send special *application* escape sequences. See page TR-365.

CSI ?1l — The cursor keys send normal cursor positioning escape sequences.

Cursor on/off

CSI ?25h — Displays the cursor as a blinking underline or box.

CSI ?25l — The cursor is no longer visible on the screen.

Display functions (controls) on/off

CSI 3h — Control codes such as escape and linefeed are displayed but not acted on.

CSI 3l — Control codes function normally.

End-of-line wrap on/off

CSI ?7h — When the cursor reaches the right margin, it is automatically returned to the left margin in the next line.

CSI ?7l — When the cursor reaches the right margin, it does not move. Additional characters overwrite the character at the right margin until the cursor is explicitly moved.

Form feed after PrtSc yes/no

CSI ?18h — A form feed is sent to the printer after either of the following print-screen operations: CSI i or Alt-PrtSc.

CSI ?18l — No form feed is sent to the printer after a print-screen operation.

Insert mode/replace mode

CSI 4h — Characters are inserted at the cursor position, and all characters to the right of the cursor move one column to the right. The insert indicator is turned on.

CSI 4l — New characters replace characters at the cursor position. The insert indicator is turned off.

Inverse/Normal video

CSI ?5h — Sets display enhancements to inverse video. The screen background is bright.

CSI ?5l — Sets display enhancements to normal. The screen background is dark.

Keyboard auto repeat on/off

CSI ?8h = on   CSI ?8l = off

Keyboard lock/unlock

CSI 2h = on   CSI 2l = off

Keypad mode application/normal

Ec= — The numeric keypad keys send special *application* escape sequences. See page TR-365.

Ec> — The numeric keypad keys send the normal numeric values. The gray + key acts as Return, and PrtSc sends a comma.

Local echo off/on

CSI 12h — Turns off local echo. Characters entered on the keyboard are not sent to the display. In remote mode the characters are sent to the host and the host must echo them back to the display.

CSI 12l — Characters entered on the keyboard are sent directly to the display. In remote mode, if the host is echoing characters, the characters will appear twice on the display.

Multi-page/single-page mode

CSI >1h — Data that scrolls off the screen is retained in display memory. This data can be brought back into view by the Scroll up, Scroll down, Prev page, Next page, Home up, and Home down functions.

CSI >1l — Data that scrolls off the screen is erased from display memory.

Origin mode set/reset

CSI ?6h — Selects the first row in the scrolling region as the home position. The first row in the scrolling region is now row 1.

CSI ?6l — Selects the upper left corner of the screen as the home position. The first row on the screen is now row 1.

Print extent full screen/scrolling region

CSI ?19h — Print-screen prints the entire screen.

CSI ?19l — Print-screen prints only the scrolling region.

Terminal class

Ec&k<n>\— Switches between VT and HP terminal class, as follows:

<n> = 0 change to HP
<n> = 1 change to VT

Changing terminal class reactivates the last saved or last activated configuration values for the new terminal class. It will also cause an error if background mode has been entered.

VT52 mode

CSI ?2l — Causes the emulator to enter VT52 emulation mode. Only VT52 escape sequences will be recognized.

VT102 mode

CSI 61"p — Sets the emulator to VT102 mode.

VT220 mode

CSI 62;0"p or CSI 62;2"p — Sets Reflection to VT220 mode with 8-bit controls (not in VT52 mode).

$^{E}{}_{C}{}^{S}{}_{P}$G — Same but recognized only in VT220 mode.

CSI 62;1"p — Sets Reflection to VT220 mode with 7-bit controls. All 8-bit controls are converted to their 7-bit equivalent before transmitting (not recognized in VT52 mode).

$^{E}{}_{C}{}^{S}{}_{P}$F — Same but recognized only in VT220 mode.

Display Control

NOTE: The following escape sequences control the position of the screen window into display memory. They are disabled if **Multi-page** mode is *NO* or if top and bottom margins are set.

Home up/down

CSI >0s = up   CSI >1s = down

Next page forward <n> pages

CSI <n>U — (**<n>** = 0 or missing equals 1).

Prev page backward <n> pages

CSI <n>V — (**<n>** = 0 or missing equals 1).

Scroll down <n> lines

CSI <n>T — (**<n>** = 0 or missing equals 1).

Scroll up <n> lines

CSI <n>S — (**<n>** = 0 or missing equals 1).

Editing and Erasing

Delete characters

CSI <n>P — Deletes **<n>** characters starting with the cursor position.

Erasing:

Beginning of line to cursor

CSI 1K

Cursor to end of line

CSI 0K — (Note: 0 is the number zero.)

Cursor to end of screen

CSI 0J — (Note: 0 is the number zero.)

Entire line

CSI 2K

Entire screen

CSI 2J — Display memory above the top of the screen is retained.

Top of screen to cursor

CSI 1J — Display memory above the top of the screen is retained.

Insert/delete line

CSI <n>L — Inserts **<n>** lines at the cursor position.

CSI <n>M — Delete line. Deletes **<n>** lines at the cursor position.

Editing and Erasing (VT220 Mode Only)

NOTE: 0 is the number zero in the following escape sequences.

Erase characters

CSI <n>X — Erases **<n>** characters starting with the character in the cursor position. The cursor does not move.

Insert blanks

CSI <n>@ — Inserts **<n>** blank characters at the cursor position. The line is shifted right and characters that go beyond the right margin are lost. The cursor does not move.

Select erasable
CSI 0"q or use CSI 2"q — All following characters are not protected from erasure by the selective erase escape sequence.

Selective erase of *erasable* characters:

Cursor to end-of-line
CSI ?0K - (Note: 0 is zero.)

Beginning of line to cursor
CSI ?1K

Entire line
CSI ?2K

Cursor to end of screen
CSI ?0J - (Note: 0 is zero.)

Top of screen to cursor
CSI ?1J — Display memory above the top of the screen is retained.

Entire screen
CSI ?2J— Display memory above the top of the screen is retained.

Select protected
CSI 1"q — All following characters are protected from erasure by the selective erase escape sequences. However, they can still be erased by a normal erase request.

Tab Stops

Clear all tabs
CSI 3g — Clears all tab stops.

Tab, set/clear
^{E}CH — Sets a tab stop at the current column.

CSI g — Clears tab at the current position.

Display Enhancements

Double-width bottom half
^{E}C#4 — Similar to ^{E}C#3.

VT Reference

Double-width line

$^{E}{}_{C}$#3 or $^{E}{}_{C}$#6 — Characters already on the line are spaced to every other position. Characters to the right of the center of the line are lost. The cursor moves two positions at a time when it is on the line.

Select display enhancements

CSI <n>;. . .<n>m — The **<n>** parameter is selected from the attributes listed in the following table. If more than one attribute is present, the attributes must be separated by a semicolon (;). For example **CSI 5;7m** selects blinking inverse video.

Table L.1
Values for Display Enhancements (<n>)

Enhancement	*On*	*Off*
Bold	*1*	*22*
Underline	*4*	*24*
Blink	*5*	*25*
Inverse video	*7*	*27*
Clear all attributes		*0*

Single-width line

$^{E}{}_{C}$#5 Changes a double-width line back to single width.

Selecting and Invoking Character Sets

Reflection supports the VT special graphics, supplemental graphics, and UKASCII character sets. Character sets are selected by escape sequences and the [Ctrl]-[N], [Ctrl]-[O] keystrokes. For example, in local mode enter $^{E}{}_{C}$)0 (zero), then [Ctrl]-[N], and type **abcd**. Notice the results: the characters on the screen are not the usual display representation for these keys.

Now enter [Ctrl]-[O] and type **abcd** again. The characters on the screen are the normal *abcd* characters. The codes stored in display memory and transmitted to the host are the same in both cases; it is only the representation of the code on the screen that is different. The IBM PC comes with 256 characters defined in alphanumeric mode. Reflection divides these characters into four character sets (see page TR-371):

- ASCII
- UKASCII
- Special graphics
- Supplemental graphics

Each character set contains 96 characters which can be displayed. A given character may appear in more than one character set. These character sets can be designated as G0, G1, G2, or G3 by the *select character set* escape sequences as shown below. Each of these may in turn be invoked into GL by control codes or escape sequences. The current value of GL is used to translate all incoming codes into a display representation on the screen. The GL character set translates codes 32 to 127, and codes greater than 128 are mapped according to the HP Roman 8 character set. When using the select character set escape sequences, the first part specifies G0, G1, G2, or G3 and the final character indicates which character set is being designated.

$^{E}_{C}$(<final char>	selects G0	
$^{E}_{C}$)<final char>	selects G1	
$^{E}_{C}$*<final char>	selects G2	VT220 mode only
$^{E}_{C}$+<final char>	selects G3	VT220 mode only

The <final> characters are as follows:

ASCII	B
Special Graphics	0
Supplemental Graphics	<
British	A

NOTE: You can use only one national character set at a time.

After designating character sets as above, they may be invoked into GL as follows:

Ctrl-O	SI	Invoke G0 into GL	
Ctrl-N	SO	Invoke G1 into GL	
Lock Shift G2	$^{E}_{C}$n	Invoke G2 into GL	VT200
Lock Shift G3	$^{E}_{C}$o	Invoke G3 into GL	VT200
Single shift G2	$^{E}_{C}$N	Invoke G2 into GL for	VT200

		the next character only	
Single shift G3	^{E}cO	Invoke G3 into GL for the next character only	VT200

Printer and Disk Control

If a printer is configured, the following escape sequences will send data to the printer. If a disk file has been selected as a "To" device, data is sent to both the printer and the disk file. If the **Printer interface** field is configured to *NONE*, data is sent only to the selected disk file.

Log to printer on/off

CSI ?5i — Also called *auto-print mode*. Data received into display memory, either from datacomm or from the keyboard, is sent to the printer one line at a time.

Lines must be delimited by ^{L}F, ^{F}F, or ^{V}T to be recognized as lines. The delimiting ^{L}F, ^{F}F, or ^{V}T is also sent to the printer.

CSI ?4i — Turns off logging to the printer.

Pass-through mode on/off

CSI 5i — Also called *printer controller mode*. Data received from datacomm is passed through to the printer without appearing on the screen. All codes except NUL, XON, XOFF, CSI 5i, and CSI 4i are passed to the printer.

CSI 4i — Turns off pass-through mode.

Print line

CSI ?1i — Prints the line that currently contains the cursor.

Print screen

CSI i — If the print extent is set to the *scrolling region*, only the scrolling region is printed. If the **Form feed after PrtSc** field is set to *YES*, a ^{F}F is sent to the printer after the print operation.

Reflection Escape Sequences

Capture on/off

CSI?7i — Turns capture on. When CAPTURE is on, all data coming from the host is logged to a disk file, including escape

VT Reference

sequences and control codes. Flow control such as XON/XOFF is omitted. The escape sequence can also be used for capturing graphics plotting sequences.

CSI?6i — Turns capture off.

Connection reset

EC&bR — This sequence does the following:

- Resets an internal Hayes Smartmodem 1200B modem card by setting the *out 1* bit of the modem control register to 1 for about 50 milliseconds.
- Suspends a LAN session.
- Powers down the communications ports on the HP Portable Plus.

Invoke command

EC& oC<command>CR — The parameter CR represents a carriage return. The **<command>** parameter is passed to the command language processor as if it had been entered from a batch file. Reflection indicates the success or failure of the command by returning a completion code of **S** or **F**. The **S** or **F** is followed by a carriage return. See page TR-217 to disable completion codes.

Serial number request

The host computer can request the serial number and version of the Reflection program running on the PC by sending the escape sequence EC*s12347^. After exchanging the appropriate block transfer handshake, Reflection sends a 14-character serial number in the form I01+200I123456. See $SERIAL on page 232.

NOTE: If sent to a terminal, the terminal ID response is returned. sequence.

To return a sequence of the form WRQ-12345, use EC*s12345^. Previous versions of Reflection may use this exclusively. A nine-character serial number is returned of the form, WRQ-12345, where the *WRQ-* is fixed and the numeric portion is variable.

---

* EC*s12346^ and EC*s33^ are obsolete versions of this sequence.

Reports

Cursor position request
CSI 6n — Reflection replies with **CSI <r>;<c>R** where **<r>** is the row and **<c>** is the column.

Device status request
CSI 5n — Reflection replies with **CSI 0n** (ready).

Primary device attributes request
CSI c and ^{E}cZ — Reflection replies according to the setting of the **DA response** field on the Terminal Configuration, Page 1 menu. See the table below.

Table L.2
Device Attributes Request Replies

Value	*Reply*	*Meaning*
VT102	*CSI ?6c*	*VT102*
VT100	*CSI ?1;0c*	*VT100*
VT100av	*CSI ?1;2c*	*VT100 advan. video*
VT100p	*CSI ?1;11c*	*VT100 w/printer*
VT220	*CSI ?62;1;2;6;7;8c*	*normal VT220*
VT220x	*CSI ?62;1;2;6c*	*Reflection*

Printer status request
CSI ?15n — Reflection replies, depending on the **Printer interface** configuration value, either **CSI ?10n** (printer ready) or **CSI ?13n** (no printer).

Secondary device attributes request
CSI >c — Reflection replies with **CSI >1;11;0c**.

Reset, Tests, LEDs

Control LEDs
CSI ?<l>q — Controls the simulated LEDs that appear on the **pf key** label line, in the $\boxed{\text{F8}}$ position, as follows:

Table L.3
Values for Controlling LEDs (<l>)

Value	*Meaning*
0	*Clear all LEDs.*
1	*Turn on LED 1.*
2	*Turn on LED 2.*
3	*Turn on LED 3.*
4	*Turn on LED 4.*

Hard reset

^{E}cc — Reset to initial state. Identical to the keyboard functions, hard reset ([Alt]-[R]), and disconnect ([Ctrl]-[F10]).

Reset to defaults

CSI !p — (VT220 mode only). Resets many Reflection features to the *default state* as follows:

Table L.4
Reset Features

Feature	*Value*
Character sets	*defaults*
Character attributes	*normal*
Cursor key mode	*normal*
Cursor style	*underline*
End-of-line wrap	*off*
Insert mode	*off*
Keyboard	*unlocked*
Keypad mode	*normal*
Origin mode	*off*
Top/Bottom margins	*whole screen*
Saved cursor state	*defaults*
Selective erase	*normal*

Terminal tests

CSI y — Causes a communications line disconnect.

Test pattern

^{E}c#8 — Displays a screen full of **E**s.

VT52 Escape Sequences

ANSI mode
$^{E}{}_{C}$<

Character set graphics/ASCII
$^{E}{}_{C}$F = graphics $^{E}{}_{C}$G = ASCII

Cursor up/down/right/left
$^{E}{}_{C}$A = up $^{E}{}_{C}$B = down
$^{E}{}_{C}$C = right $^{E}{}_{C}$D = left

Erase to end of screen
$^{E}{}_{C}$J

Erase to end-of-line
$^{E}{}_{C}$K

Home cursor
$^{E}{}_{C}$H

Identification request
$^{E}{}_{C}$Z — Reflection replies $^{E}{}_{C}$/Z.

Keypad mode application/normal
$^{E}{}_{C}$= — application mode $^{E}{}_{C}$> — normal mode

Log to printer on/off
$^{E}{}_{C}$^ = on $^{E}{}_{C}$_ = off - Also called *auto-print mode.*

Pass-through mode start/stop
$^{E}{}_{C}$W = start $^{E}{}_{C}$X = stop

Position cursor
$^{E}{}_{C}$Y<r><c> — The parameters **<r>** and **<c>** are equal to the character whose value is the desired row or column value plus 31. For example, row 2 and column 4 would be $^{E}{}_{C}$Y!#.

Print cursor line
$^{E}{}_{C}$V

Print screen
$^{E}{}_{C}$]

Reverse linefeed
$^{E}{}_{C}$I

VT Escape Sequence Summary

Table L.5
Sequential List of Sequences

Sequence	Function	Page
Character Attributes		
CSI 1"q	*Select protected*	*TR-351*
CSI 0"q CSI 2"q	*Select erasable*	*TR-351*
CSI <n>;...<n>m	*Select display enhancements*	*TR-352*

Enhancement	*On*	*Off*	
Clear all attributes		*0*	
Bold	*1*	*22*	*Normal*
Underline	*4*	*24*	*Not underlined*
Blink	*5*	*25*	*Not blinking*
Inverse video	*7*	*27*	*Not reverse*

Select Character Sets

$^{E}{}_{C}$*(<final>*	*Selects G0*
$^{E}{}_{C}$*)<final>*	*Selects G1*
$^{E}{}_{C}$*<final> (with *)	*Selects G2 (VT220)*
$^{E}{}_{C}$*+<final>*	*Selects G3 (VT220)*

The <final> characters are as follows:

ASCII	*B*
Special Graphics	*0*
Supplemental Graphics	*<*
British	*A*

Invoke Character Sets

SI	*Lock shift G0*	*TR-353*
SO	*Lock shift G1*	*TR-353*
$^{E}{}_{C}$~	*Lock shift G1 right*	*TR-353*
$^{E}{}_{C}$}	*Lock shift G2 right*	*TR-353*
$^{E}{}_{C}$|	*Lock shift G3 right*	*TR-353*

VT Reference

Sequence		Function	Page
$^{E}{}_{C}n$		*Lock shift G2*	*TR-353*
$^{E}{}_{C}N$		*Single shift G2*	*TR-353*
$^{E}{}_{C}o$		*Lock shift G3*	*TR-353*
$^{E}{}_{C}O$		*Single shift G3*	*TR-354*

Display and Cursor Control

$^{E}{}_{C}7$	$^{E}{}_{C}8$	*Save/Restore cursor*	*TR-345*
$^{E}{}_{C}D$		*Index down*	*TR-344*
$^{E}{}_{C}E$		*Next line*	*TR-345*
$^{E}{}_{C}M$		*Reverse index*	*TR-345*
CSI <n>A	*CSI <n>B*	*Cursor up/down*	*TR-344*
CSI <n>C	*CSI <n>D*	*Cursor right/left*	*TR-344*
CSI <n>E	*CSI <n>F*	*Next/Previous line*	*TR-345*
CSI <n>G		*Cursor position absolute*	*TR-344*
CSI <r>;<c>H		*Cursor position*	*TR-344*
CSI <t>;<b>r		*Set scrolling region*	*TR-345*
CSI <n>S	*CSI <n>T*	*Scroll up/down* **<n>***lines*	*TR-349*
CSI <n>U	*CSI <n>V*	*Next/Prev page* **<n>** *pages*	*TR-349*
CSI <n>Z		*Cursor backtab*	*TR-343*
CSI <n>'		*Cursor horizontal absolute*	*TR-344*
CSI <n>a		*Cursor horizontal relative*	*TR-344*
CSI <n>d		*Cursor vertical absolute*	*TR-345*
CSI <n>e		*Cursor vertical relative*	*TR-345*
CSI <r>;<c>f		*Cursor position absolute*	*TR-344*
CSI >0s	*CSI >1s*	*Home up/down*	*TR-349*

Line Attributes

$^{E}{}_{C}\#3$	$^{E}{}_{C}\#6$	*Double line width*	*TR-352*
$^{E}{}_{C}\#4$		*Double line width (blank)*	*TR-352*
$^{E}{}_{C}\#5$		*Single line width*	*TR-352*
$^{E}{}_{C}\#8$		*Test pattern*	*TR-357*

Operating modes

CSI 2h	*CSI 2l*	*Keyboard lock/unlock*	*TR-347*
CSI 3h	*CSI 3l*	*Display controls on/off*	*TR-347*

Sequence		Function	Page
CSI 4h	*CSI 4l*	*Insert/Replace mode*	*TR-347*
CSI 12h	*CSI 12l*	*Local echo off/on*	*TR-348*
CSI 20h	*CSI 20l*	*Auto linefeed off/on*	*TR-346*
CSI ?1h	*CSI ?1l*	*Cursor keys application/normal*	*TR-346*
CSI ?2l		*VT52 mode*	*TR-349*
CSI ?3h	*CSI ?3l*	*Columns 132/80*	*TR-346*
CSI ?5h	*CSI ?5l*	*Video inverse/normal*	*TR-347*
CSI ?6h	*CSI ?6l*	*Origin mode set/reset*	*TR-348*
CSI ?7h	*CSI ?7l*	*End-of-line wrap on/off*	*TR-347*
CSI ?8h	*CSI ?8l*	*Keyboard repeat on/off*	*TR-347*
CSI ?18h	*CSI ?18l*	*Form feed after* $\boxed{PrtSc}$ *yes/no*	*TR-347*
CSI ?19h		*Print ext — screen/scrl region*	*TR-348*
CSI ?25h	*CSI ?25l*	*Cursor visible/invisible*	*TR-346*
CSI >1h	*CSI >1l*	*Page mode - multi/single*	*TR-348*
E_C=	E_C>	*Keypad - appl/normal*	*TR-348*
CSI 61"p		*VT100 mode*	*TR-349*
CSI 62;0"p		*VT200 mode 8-bit controls*	*TR-349*
CSI 62;1"p		*VT200 mode 7-bit controls*	*TR-349*
CSI 62;2"p		*VT200 mode 8-bit controls*	*TR-349*
$E_C S_P$*F*		*Transmit 7-bit controls*	*TR-349*
$E_C S_P$*G*		*Transmit 8-bit controls*	*TR-349*

Editing

CSI <n>M	*CSI <n>P*	*Delete line/characters*	*TR-350*
CSI <n>@	*CSI <n>L*	*Insert blanks/line*	*TR-350*

Selective Erase

CSI ?1J	*Beginning of screen to cursor*	*TR-351*
CSI ?2J	*Entire screen*	*TR-351*
CSI ?0J	*Cursor to end of screen*	*TR-351*
CSI ?1K	*Beginning of line to cursor*	*TR-351*
CSI ?0K	*Cursor to end of screen*	*TR-351*
CSI ?2K	*Entire line*	*TR-351*

Sequence		Function	Page
Erase			
CSI <n>X		*Erase characters*	*TR-350*
CSI 0K		*Cursor to end-of-line*	*TR-350*
CSI 1K		*Beginning of line to cursor*	*TR-350*
CSI 2K		*Complete line*	*TR-350*
CSI 0J		*Cursor to end of screen*	*TR-350*
CSI 1J		*Top of screen to cursor*	*TR-350*
CSI 2J		*Entire screen*	*TR-350*
Printer			
CSI ?1i	*CSI i*	*Print line/screen*	*TR-354*
CSI ?5i	*CSI ?4i*	*Log to printer/stop*	*TR-354*
CSI 5i	*CSI 4i*	*Pass-through mode on/off*	*TR-354*
Reflection Sequences			
EC*&oC<cmd>*CR		*Reflection command*	*TR-355*
CSI ?7i	*CSI ?6i*	*Set capture yes/no*	*TR-354*
EC*&bR*		*Connection reset*	*TR-355*
EC*s12345^		*Serial number request (9 digit)*	*TR-355*
EC*s12347^		*Serial number request (14 digit)*	*TR-355*
Tabs and Margins			
CSI 0g	EC*H*	*Tab clear/set*	*TR-351*
CSI 3g		*Tabs clear all*	*TR-351*
CSI <t>;<b>r		*Set top and bottom margins*	*TR-345*
Reports, Resets, and LEDS			
CSI c	EC*Z*	*Primary device attr request*	*TR-356*
CSI >c		*Sec device attributes request*	*TR-356*
CSI 5n		*Device status request*	*TR-356*
CSI 6n		*Cursor position request*	*TR-356*
CSI ?15n		*Printer status request*	*TR-356*
EC*c*		*Hard reset*	*TR-357*
CSI !p		*Reset to defaults (VT220)*	*TR-357*

CSI <n>q	*Control LEDs*	*TR-356*
CSI <n>...;<n>y	*Terminal tests (disconnect)*	*TR-357*

VT102/220 mode supports HP escape sequences which begin with the following prefixes:

$^{E}_{C}$&f	$^{E}_{C}$&q
$^{E}_{C}$&j	$^{E}_{C}$&s
$^{E}_{C}$&k	$^{E}_{C}$&w
$^{E}_{C}$&p	$^{E}_{C}$&K

VT Keyboard Generated Codes

In VT mode, the typewriter keyboard generates the normal ASCII characters. Using [Ctrl] in combination with the character keys generates ASCII control characters (see page TR-323 for a table of ASCII control characters). This reference describes additional codes that can be generated from the keyboard.

Keyboard Keys

Table M.1
Keyboard Key Codes

VT 102	*Reflection*	*Generated Code*
Backspace	[Backspace] or [Ctrl]-[Backspace]	BS or DEL (configurable)
Break	[Ctrl]-[ScrollLock]	Break
Delete	[Ctrl]-[Backspace] or [Backspace]	DEL or BS (configurable)
Linefeed	[Ctrl]-[Return]	LF
Disconnect	[Ctrl]-[F10]	Lowers DTR and RTS for 2 seconds
Esc	[Esc]	ESC
Return	[Return]	CR
Send answerback message	[Ctrl]-[F2]	Answerback

Cursor Control Keys

The VT terminal cursor control keys are available in two ways on the IBM PC keyboard. While [NumLock] is off, the cursor control keys on the PC keypad generate cursor control. When [NumLock] is on, [Shift] plus the cursor control keys on the PC keyboard generate cursor control. In addition, while the **pf keys** labels are displayed ([Alt]-[P]), [F5] through [F8] generate cursor control.

Table M.2
Cursor Control Keys

Keystroke		Generated Codes		
		VT 102 Mode Cursor Key Mode		VT 52
VT 102	Reflection	Normal	Application	Mode
up	[Up]	^{E}c[A	^{E}cOA	^{E}cA
down	[Down]	^{E}c[B	^{E}cOB	^{E}cB
right	[Right]	^{E}c[C	^{E}cOC	^{E}cC
left	[Left]	^{E}c[D	^{E}cOD	^{E}cD

Numeric Keypad

The VT numeric keypad keys are available on the IBM PC keypad when [NumLock] is on. When [NumLock] is off, the keypad generates the cursor control escape sequences. While [NumLock] is on, the keypad may be in either normal or application mode. Normal or application mode is usually selected by host software using escape sequences.

Table M.3
Numeric Keypad Codes

Keystroke			Generated Codes		
			VT 52 Mode		VT 102 Mode
VT 102	Reflection	Normal	Appl	Normal	Appl
0	[0]	0	EC?p	0	ECOp
1	[1]	1	EC?q	1	ECOq
2	[2]	2	EC?r	2	ECOr
3	[3]	3	EC?s	3	ECOs
4	[4]	4	EC?t	4	ECOt
5	[5]	5	EC?u	5	ECOu
6	[6]	6	EC?v	6	ECOv
7	[7]	7	EC?w	7	ECOw
8	[8]	8	EC?x	8	ECOx
9	[9]	9	EC?y	9	ECOy
-	[-]	-	EC?m	-	ECOm
,	[PrtSc]	,	EC?l	,	ECOl
Enter	[+]	Return	EC?M	Return	ECOM

PF Keys

[F2] from the system keys or [Alt]-[P] accesses the PF function-key labels. When the **pf keys** labels are on the screen, [F1] through [F4] act as PF1 through PF4. [F5] through [F8] act as cursor control keys. The rightmost label includes the LED indicators. The PF keys are available as [F1] through [F4] while the **pf key** labels are displayed.

Table M.4
PF Key Codes

Keystroke		Generated Code	
VT 102	Reflection	VT 102 mode	VT 52 mode
PF1	[F1]	ECOP	ECP
PF2	[F2]	ECOQ	ECQ
PF3	[F3]	ECOR	ECR
PF4	[F4]	ECOS	ECS

VT Reference

VT 220 Keys

Additional VT 220 keys available in VT 220 mode only.

Table M.5
VT 220 Key Codes

VT 220 Keystroke	Reflection Keystroke	Generated Code
Find	Alt-1	^{E}c[1~
Insert Here	Alt-2	^{E}c[2~
Remove	Alt-3	^{E}c[3~
Select	Alt-4	^{E}c[4~
Prev Screen	Alt-5	^{E}c[5~
Next Screen	Alt-6	^{E}c[6~
F6	Shift-F6	^{E}c[17~
F7	Shift-F7	^{E}c[18~
F8	Shift-F8	^{E}c[19~
F9	Shift-F9	^{E}c[20~
F10	Shift-F10	^{E}c[21~
F11	Alt-F1	^{E}c[23~
F12	Alt-F2	^{E}c[24~
F13	Alt-F3	^{E}c[25~
F14	Alt-F4	^{E}c[26~
F15 (help)	Alt-F5	^{E}c[28~
F16 (do)	Alt-F6	^{E}c[29~
F17	Alt-F7	^{E}c[31~
F18	Alt-F8	^{E}c[32~
F19	Alt-F9	^{E}c[33~
F20	Alt-F10	^{E}c[34~

VT Control Characters

Decimal	*Mnemonic*	*Name*	*Action*
5	ENQ	Enquiry	The answerback message is transmitted when **Auto answerback** is configured to *YES*.
7	BEL	Bell	The bell is sounded.
8	BS	Backspace	Moves cursor left one position on the current line.
9	HT	Tab	Moves cursor to the next tab stop. If there are no more tab stops, the cursor moves to end of line.
10	LF	Linefeed	Moves the cursor to the next line in same column. When the auto linefeed field is on, the cursor moves to the first column of the next line.

Decimal	*Mnemonic*	*Name*	*Action*
11	$^{V}{}_{T}$	Vertical tab	Same as linefeed.
12	$^{F}{}_{F}$	Form feed	Same as linefeed.
13	$^{C}{}_{R}$	Carriage return	Moves the cursor to the left margin.
14	$^{S}{}_{O}$	Shift out	Invokes the character set in table G1 to GL.
15	$^{S}{}_{I}$	Shift in	Invokes the character set in table G0 to GL.
17	$^{D}{}_{C1}$	Device control 1 (XON)	When **Transmit pacing** is set to *XON/XOFF*, a DC1 causes Reflection to resume transmitting.
19	$^{D}{}_{C3}$	Device control 3 (XOFF)	When **Transmit pacing** is set to *XON/XOFF*, a DC3 causes Reflection to stop transmitting.
24	$^{C}{}_{AN}$	Cancel	When received during an escape sequence, the sequence is cancelled.
26	$^{S}{}_{UB}$	Substitute	Same as Cancel.
27	$^{E}{}_{C}$	Escape	Introduces an escape sequence. Cancels any escape sequence that is in progress.

VT Character Sets

In VT mode, Reflection supports the VT special graphics, supplemental graphics, and UKASCII character sets. Character sets are selected by escape sequences and the [Ctrl]-[N], [Ctrl]-[O] keystrokes.

The IBM PC comes with 256 characters defined in alphanumeric mode. Reflection divides these characters into five character sets:

ASCII
UKASCII
Display Function Graphics
Special graphics
Supplemental graphics

Each character set contains 96 characters which can be displayed. Note that the same character may appear in more than one character set. These character sets can be loaded into any of four tables, named G0, G1, G2, and G3, by the *select character set* escape sequences. Each of these tables may in turn be invoked into a register, called GL by control codes or escape sequences. The current value of GL is used to translate all incoming codes into a display representation on the screen. (See page TR-353 for a description of these escape sequences.)

For example, in local mode enter ^{E}c)0 (zero), then [Ctrl]-[N], and type **abcd**. Notice the results: the characters on the screen are not the usual display representation for these keys. Now enter [Ctrl]-[O] and type **abcd** again. The characters on the screen are the normal *abcd* characters. The codes stored in display memory and transmitted to the host are the same in both cases; it is only the representation of the code on the screen that is different.

ASCII Character Set

DECIMAL VALUE	⇒	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	|
13	D	♪	↔	–	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

UKASCII Character Set

DECIMAL VALUE	⇨	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	NUL	DLE	BLANK (SPACE)	0	@	P	`	p
1	1	SOH	DC1	!	1	A	Q	a	q
2	2	STX	DC2	”	2	B	R	b	r
3	3	ETX	DC3	£	3	C	S	c	s
4	4	EOT	DC4	$	4	D	T	d	t
5	5	ENQ	NAK	%	5	E	U	e	u
6	6	ACK	SYN	&	6	F	V	f	v
7	7	BEL	ETB	'	7	G	W	g	w
8	8	BS	CAN	(	8	H	X	h	x
9	9	HT	EM	)	9	I	Y	i	y
10	A	LF	SUB	*	:	J	Z	j	z
11	B	VT	ESC	+	;	K	[	k	{
12	C	FF	FS	,	<	L	\	l	|
13	D	CR	GS	–	=	M	]	m	}
14	E	SO	RS	.	>	N	^	n	~
15	F	SI	US	/	?	O	_	o	△

VT Display Function Graphics

DECIMAL VALUE	⇒	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	‼	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	|
13	D	♪	↔	-	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

VT Reference

VT Special Graphics

DECIMAL VALUE	⇨	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	NUL	DLE	BLANK (SPACE)	0	@	P	♦	⎻
1	1	SOH	DC1	!	1	A	Q	▒	─
2	2	STX	DC2	”	2	B	R	○	─
3	3	ETX	DC3	#	3	C	S	♀	⎼
4	4	EOT	DC4	$	4	D	T	♪	├
5	5	ENQ	NAK	%	5	E	U	♬	┤
6	6	ACK	SYN	&	6	F	V	°	┴
7	7	BEL	ETB	'	7	G	W	±	┬
8	8	BS	CAN	(	8	H	X	→	│
9	9	HT	EM	)	9	I	Y	♂	≤
10	A	LF	SUB	*	:	J	Z	┘	≥
11	B	VT	ESC	+	;	K	[	┐	π
12	C	FF	FS	,	<	L	\	┌	╪
13	D	CR	GS	–	=	M	]	└	£
14	E	SO	RS	.	>	N	^	┼	•
15	F	SI	US	/	?	O		─	DEL

VT Reference

VT Supplemental Graphics

DECIMAL VALUE	⇨	0	16	32	48	64	80	96	112
⇩	HEXA DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	NUL	DLE		°	A	¿	à	¿
1	1	SOH	DC1	¡	±	A	Ñ	á	ñ
2	2	STX	DC2	¢	²	A	O	â	ò
3	3	ETX	DC3	£	↑	A	O	a	ó
4	4	EOT	DC4	¿	¿	Ä	O	ä	ô
5	5	ENQ	NAK	¥	µ	Å	O	å	o
6	6	ACK	SYN	¿	¶	Æ	Ö	æ	ö
7	7	BEL	ETB	§	•	Ç	O	ç	o
8	8	BS	CAN	☼	¿	E	O	è	ö
9	9	HT	EM	c	↓	É	U	é	ù
10	A	LF	SUB	ª	º	E	U	ê	ú
11	B	VT	ESC	«	»	E	U	ë	û
12	C	FF	FS	¿	¼	I	Ü	ì	ü
13	D	CR	GS	¿	½	I	Y	í	ÿ
14	E	SO	RS	¿	¿	I	¿	î	¿
15	F	SI	US	¿	¿	I	β	ï	

VT Reference

Index

A

B

C

D

E

F

G

H

J

K

L

M

N

O

Q

T

U

V

Y

Z