Intermec

Programmer's Guide

TE 2000™ Terminal Emulation

Intermec

Programmer's Guide

TE 2000™ Terminal Emulation

Intermec Technologies Corporation

Worldwide Headquarters
6001 36th Ave. W
Everett, WA 98203
U.S.A.

www.intermec.com

Document Change Record

This page records changes to this document. The document was originally released as Revision A.

Revision Letter	Date	Description of Change
B	06/2006	Written for software version 8.12. Added information about CK61 mobile computers.
C	02/2007	Written for software versions 8.21 and 8.25. Added information about the CV30 fixed mount computer, the CN3 mobile computer, Syvox speech for 700 Color mobile computers, and SSH sessions.
D	06/2007	Written for software version 8.25+. Added information about the CK32 handheld computer.

Contents

Before You Begin

This section provides you with safety information, technical support information, and sources for additional product information.

Safety Information

This section explains how to identify and understand notes that are in this document.

Note: Notes provide extra information about a topic or contain special instructions for handling a particular condition or set of circumstances.

Global Services and Support

Warranty Information

To understand the warranty for your Intermec product, visit the Intermec web site at www.intermec.com and click **Service & Support**. The Intermec Global Sales & Service page appears. From the Service & Support menu, move your pointer over **Support**, and then click **Warranty**.

Disclaimer of warranties: The sample code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided “as is with all faults.” All warranties are expressly disclaimed, including the implied warranties of merchantability and fitness for a particular purpose.

Web Support

Visit the Intermec web site at www.intermec.com to download our current manuals (in PDF). To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

Visit the Intermec technical knowledge base (Knowledge Central) at intermec.custhelp.com to review technical information or to request technical support for your Intermec product.

Telephone Support

These services are available from Intermec Technologies Corporation.

Service	Description	In the U.S.A. and Canada call 1-800-755-5505 and choose this option
Order Intermec products	• Place an order. • Ask about an existing order.	1 and then choose 2
Order Intermec media	Order printer labels and ribbons.	1 and then choose 1
Order spare parts	Order spare parts.	1 or 2 and then choose 4
Technical Support	Talk to technical support about your Intermec product.	2 and then choose 2

Service	Description	In the U.S.A. and Canada call 1-800-755-5505 and choose this option
Service	• Get a return authorization number for authorized service center repair. • Request an on-site repair technician.	2 and then choose 1
Service contracts	• Ask about an existing contract. • Renew a contract. • Inquire about repair billing or other service invoicing questions.	1 or 2 and then choose 3

Outside the U.S.A. and Canada, contact your local Intermec representative. To search for your local representative, from the Intermec web site, click **Contact.**

Who Should Read This Manual

This manual is written for the person who is responsible for installing, configuring, and maintaining the TE 2000 terminal emulation application.

This document provides you with information about the features of the TE 2000 application, and how to install, configure, operate, maintain, and troubleshoot it.

Before you work with the TE 2000 terminal emulation application, you should be familiar with your network and with general networking terms, such as IP address.

Related Documents

This table contains a list of related Intermec documents and part numbers.

Manual	Part Number
700 Series Color Mobile Computer User's Manual	961-054-031
CK30 Handheld Computer User's Manual	073528
CK31 Handheld Computer User's Manual	075207
CK32 Handheld Computer User's Manual	935-006-xxx
CK60 Mobile Computer with Windows Mobile User's Manual	935-000-xxx
CN3 Mobile Computer User's Manual	935-003-xxx
CV30 Fixed Mount Computer User's Manual	935-005-xxx
CV60 Vehicle Mount Computer User's Manual	934-004-xxx
Intermec Computer Command Reference Manual	073529
Intermec Gateway User's Guide	072245
IP4 Handheld Reader Instructions	943-002-xxx
MobileLAN™ access 21xx System Manual	067150

The Intermec web site at www.intermec.com contains our documents (as PDF files) that you can download for free.

To download documents

1. Visit the Intermec web site at www.intermec.com.
2. Click **Service & Support** > **Manuals**.
3. In the **Select a Product** field, choose the product whose documentation you want to download.

To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

Software Support

The following table shows which TE software versions support which computers at the time of this publication. See the applicable user's manual for information on learning what OS IVA build is on your computer. Contact your Intermec representative for more information.

Computer	TE 2000	Operating System	Build Version	Intermec Content
700 Color	v8.25	Windows Mobile 2003	v4.95.1	PSM v.4.06
CK30/CK31	v8.12	Windows CE 4.2	v4.00.00.0807 (RoHS)	IVA v4.03.37.1125 (RoHS)
CK32	v8.25	Windows Mobile 5.0	v3.10.08.0035	SSPB 5.00.00.0191
CK61	v8.28	Windows CE 5.0	v3.10.8.0057	IVA 5.00.00.223 SSPB 5.00.42.1490
CK61	v8.25	Windows Mobile 5.0	v3.1.05.0045	SSPB 5.00.07.0183 IVA 5.00.42.1431
CN3	v8.28	Windows Mobile 5.0	v3.10.09.0057	IVA 5.00.51.1456 SSPB 5.00.00.0213
CV30	v8.28	Windows CE 5.0	v3.00.02.0031	IVA 5.00.42.1413 SSPB 5.00.06.0170
CV30	v8.20	Windows Mobile 5.0	v3.00.01.0038	SSPB v.5.00.01.0068
CV60	v8.12	Windows CE 4.2	v2.00.00.0237 (RoHS)	IVA v4.03.38.1184 (RoHS)

1 Getting Started

This chapter introduces the TE 2000™ terminal emulation application.

- Understanding Network Protocol Options
- Setting Up the Computer and the Network
- Installation and Setup Instructions
- Starting the TE 2000 Application
- Autostarting TE 2000 on 700 Colors
- Quick Configuration to a Standard Telnet Host
- Configuring the TE 2000 Application
- Program Names

Understanding Network Protocol Options

TE 2000 applications for the Enterprise Wireless LAN system use one of the following network protocol options. For network configuration options, refer to your computer's user manual.

Network Protocol Options

TCP/IP	The computer communicates through access points directly connected to the host computer on an Ethernet or a token ring network
TGAP (Telnet Gateway Appliance)	TGAP is a function of the WA2x Intermec access point, which allows a client session persistence using RTC (Real-Time Control) over TCP on computer clients running TE 2000 version 7.49 or later. It is required for Norand Native emulation and supported in VT/ANSI, 5250, and 3270 emulations. If a TE 2000 client loses connectivity for any reason (such as roams out of range, powered off, or lost battery power), the gateway keeps the client's session alive to its TCP/IP host. For more information on how session persistence is achieved and system requirements for using a TGAP, refer to the TGAP section of the manuals for the MobileLAN Access WA2x family of access points.
UDP Plus	The computer communicates with the host computer through the Intermec® Application Server and an access point

Setting Up the Computer and the Network

Before you can start using the TE 2000 application on your computer, you must first perform these steps:

1 Set up your computer.

Setup includes charging and installing the battery pack and turning on the computer for the first time. For instructions, refer to your computer's user manual.

Note: Battery pack instructions do not apply to all computers. Vehicle mounted and stationary computers are powered via an external source.

2 Configure your computer and the network.

To use RF communications on the computer, you need to:

a Configure the Intermec Application Server (UDP Plus), other Intermec gateways, or host (TCP/IP).

b Configure the access point.

c Configure the network parameters on the computer.

For instructions, see your computer’s user’s manual.

3 Verify that your computer is communicating correctly with the access point and Intermec Application Server or the host.

To verify that your computer is communicating correctly, refer to the computer's user manual for instructions.

Installation and Setup Instructions

See the *TE 2000 Installation and Setup Instructions* (P/N 962-055-010) for the latest instructions on installing the TE 2000 application and configuring your computers.

Starting the TE 2000 Application

TE 2000 starts automatically by default after every reboot on all Intermec computers except the 700 Color Mobile Computer.

Note: “700 Color” refers to Intermec 730, 740, 741, 750, 751, 760, and 761 Mobile Computers, unless otherwise noted.

If TE 2000 is not running, you can launch TE 2000 from the Start menu or the desktop PC (if applicable). While TE 2000 is starting, the screen will display the logo for up to 30 seconds while setup parameters are checked, keyboard layouts are loaded, and peripherals are initialized. The TE 2000 application begins after the following initialization screens (shown below) are shown.

COPR. 1991-2007
INTERMEC. ALL
RIGHTS RESERVED
V<version number>
<release date>

TE2000(TM)

<program name> <version>
Session: <current session number>
<session name>
<emulation type>
Host:
<host name>
<device name>

Autostarting TE 2000 on 700 Color Series Computers

By default, the TE 2000 application does not launch after a reboot without user intervention when installed on a 700 Color series computer. You can automate launching the TE 2000 application on every startup.

To launch the TE 2000 application automatically at startup

1 On a desktop or laptop PC, create a file named AutoUser.dat with this entry: `RUN=Flash File Store\te2000\fwp700h0.exe`

Note: If TE 2000 is installed to the Secure Digital storage card, substitute the RUN command with this line:
`RUN=SDMMC Disk\TE2000\fwp700h0.exe`

2 Copy the file to the \Flash File Store\2577 folder on the 700 Color, then restart the computer.

Quick Configuration to a Standard Telnet Host

1 While TE 2000 is running, access the TE 2000 setup menu using one of the following methods:
   - Double-tap the upper right corner of the display.
   - Tap [Shift] [Mn] on the Soft Input Panel (SIP).
   - Press [Alt] [M] on the SIP.
   - Scan this bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

     %TECFG

2 Press **[1]**, then type “cr52401” at the **Enter Password** prompt.

3 From the Set-up Parms menu, select **Communication > Direct > Host Setup > Host A**.

4 Type the IP address (or DNS name if the computer is configured to use DNS) of your host. You can enter up to 64 alphanumeric characters. Press **Enter** twice.

5 Select the TE 2000 emulation mode to use.

6 Press **Enter** until you return to the Main Menu, then select **Exit Menus**.

7 Press **[Y]**, then type “cr52401” at the **Enter Password** prompt to exit the TE 2000 Set-up menu.

Configuring the TE 2000 Application

You can use the computer's TE 2000 configuration menus to configure site-specific operational parameters, including UDP Plus and TCP/IP communications, terminal emulation options, and the Main Menu password. For information about configuring the computer, see “Using theTerminal Emulation Menus” on page 97.

Other methods you can use to configure your parameters include the Intermec Settings applet in your computer, the Wavelink Avalanche application, and the SmartSystems™ Foundation application. Contact your Intermec representative for information about these other methods.

Program Names

The following chart lists computers and related program names. TE/IP/802.11 or TE/UDP Plus/802.11 are the feature options.

Model	Program Name
700 Color	FWP700H0
CK30, CK31	FWPCK3H0
CK32	TECCK32R
CK61	TECCK60R
CN3	TECCN3_R
CV30	TECCV30R
CV60	FWPCV6H0

2 Using Terminal Emulation Applications

This chapter describes how to use the TE 2000™ terminal emulation application for your particular computer.

For the TE 2000 3270 application, Intermec emulates an IBM-3278-2 computer.

For the TE 2000 5250 application, Intermec emulates the following:

- IBM-5291-1 computer
- IBM-5555-B01 and IBM-5555-C01 computers (If a double-byte CAB file is installed, a localized operating system is running for a double-byte country, or depending on the Use Color setting).
- IBM-5292-02 computer (If the 5250 color option is enabled).
- IBM-3477-FG and IBM-3477-FC computers (If 132 column mode is selected, depending on the Use Color setting).

Annunciators

The computer's display reserves a location for annunciators (icons) that monitor RF and network communications or alert you to a condition that requires action. For information about annunciators that indicate battery condition and general operational status, refer to the computer's user manual.

3270 Annunciators

Icon Name	Icon	Position	Description
Session number	#	1	The session number of the TE 2000 application.
Input inhibit	X	2	The keyboard has accepted enough information for the defined input field. The “key-ahead” feature stores keystrokes after the “input inhibited” annunciator appears. These are saved for the next field. This overrides Insert Mode if both are active.
Insert mode	^	2	The keyboard inserts characters instead of overwriting them.

5250 Annunciators

Icon Name	Icon	Position	Description
Message waiting	M	1	The host has a message waiting for the operator. This overrides Session Number if both are active.
Session number	#	1	The session number of the TE 2000 application.
Input inhibit	X	2	The keyboard has accepted enough information for the defined input field. The “key-ahead” feature stores keystrokes after the “input inhibited” annunciator appears. These are saved for the next field. This overrides Insert Mode if both are active.
Insert mode	^	2	The keyboard inserts characters instead of overwriting them.
Hebrew mode	H	3	If the start of header is set for right to left data input and a 5250 bidirectional Hebrew screen was received.
RTL mode	<	4	If the data input mode when operating in Hebrew is set for right to left, then the cursor is set in a right-to-left mode.

Native Annunciators

Icon Name	Icon	Position	Description
Session number	#	1	The session number of the TE 2000 application.

Native Annunciators (continued)

Input inhibit	X	2	Keyboard action mode (KAM) was set. The computer ignores all keystrokes that send characters to the host. This state stays on until KAM is reset. This overrides Insert Mode if both are active.
Insert mode	^	2	The keyboard inserts characters instead of overwriting them.

VT/ANSI Annunciators

Icon Name	Icon	Position	Description
Session number	#	1	The session number of the TE 2000 application.
Input inhibit	X	2	Keyboard action mode (KAM) was set. The computer ignores all keystrokes that send characters to the host. This state stays on until KAM is reset. This overrides Insert Mode if both are active.
Keypad mode	K	2	The computer is in Keypad mode.
Character mode	C	2	Computer is in Character mode, sending each character as pressed.
Line Edit (block) mode	B	2	The computer is in Line Edit (block) mode. When you press a terminating key, the computer sends a block of characters to the host.
Local Edit mode	e	2	The computer is in Local Edit mode, which is a feature of the VT330/VT340 computer.

Applications

The following information pertains to applications as used by your computers running 3270, 5250, or VT/ANSI mode.

3270 Applications

The following information pertains to 3270 computers.

3278 SNA Keys

The following describes 3278 SNA keys. To enter an SNA Key, press the keys listed in the chapter for the computer. Or, scan the bar codes. For instructions, see “Bar Code Scanning” on page 391 in Appendix A.

3278 SNA Keys

Key	Description
Clr	Erases the current unprotected field. Also sets the MDT bit and does a reverse tab. A beep means the field is protected and cannot be erased.

3278 SNA Keys (continued)

Del	Deletes the character over the cursor in the current unprotected field. Data to the right of the cursor shifts left one position. A beep indicates the character is in a protected field and cannot be erased.
Enter	Transmits all modified data fields to the host.
EOF	Erases all data from the position of the cursor to the end of the unprotected field. The cursor remains in the same location. A beep indicates that the field is protected.
Home	Sends the cursor to the unprotected field in the display buffer. The first unprotected field is determined by the Insert Cursor order.
Insert	Toggles between insert and normal mode. In insert mode, characters are inserted instead of overwritten.
Reset	Resets from an error condition.

AID-Generating Keys

An AID-generating key is any key that causes a data transmission to the host system. They alert the host system via an AID code that the current session requires some action. TE 2000 emulates all of the AID-generating keys on a 5291 Display Station.

AID-Generating Keys

Key	Description
Clear	This key clears the data buffer but leaves the keyboard unlocked. It sends the Clear AID key value to the host.
Programmable function keys F1-F24	These keys send modified input fields and AID key values to the host. The keys lock the keyboard until the host unlocks it. The function keys are used exclusively for 3270 AID key emulation. When you press a programmable function key or scan its bar code, you send the data on the screen to the host, and the function you specified is performed on this data. Each function is determined by the application you use with your system. See the application's user's manual for details.
Program Access (PA) keys 1-3	PA1, PA2, and PA3 send the AID key value to the host but leave the keyboard unlocked. When an operator presses a PA key, one of the following AID codes is returned along with the current cursor address on the normal LU-LU (logical unit) flow. No data is returned to the AS/400 system with any PA key. PA AID X6C; PA AID X6E; PA AID X6B

5250 Applications

The following information pertains to 5250 computers.

Special Function Keys

Special function keys are the following. To enter a special function key, press the keys listed in the chapter for the computer or scan the bar codes listed in Appendix A, "Bar Code Scanning" on page 391.

The following pages describe the special functions keys. For complete descriptions, refer to the appropriate IBM 5250 reference manual.

AID-Generating Keys

AID-generating keys generate AID codes that go in the display data stream to the host system. They alert the host system that the Intermec Application Server or controller requires some action.

TE 2000 emulates all of the AID-generating keys on a 5291 Display Station.

AID-Generating Keys

Key	Description
Clear	The system environment determines the results of this key. If the computer is in session, [CLEAR] issues the AID code hex BD, which requests the host system issue a Clear Unit command to the computer to clear the display. If not in session, [CLEAR] clears the entire display regeneration buffer.
Enter/Rec Adv	Enters information.
F1-F24	User-defined command functions. Refer to your application's user manual for detail on the functions.
Help (nonerror state)	Issues a hex F3 AID byte to the host system.
Print	Tells the controller that the operator wants to print the contents of the present display. Issues hex F6 to the host system.
Record Backspace (Home)	When pressed with the cursor in the home position, a record backspace is requested. The AID code hex F8 and cursor address are sent to the host system.
Roll Up and Roll Down	Roll display up or down one page. **Roll Up** issues AID code hex F5. **Roll Down** issues AID code hex F4.

Roll Keys

Roll Up and **Roll Down** are AID keys the computer sends to the host to request and display additional screens. The host transmits a new screen in response to this command. The new screens allow you to view data either above or below what appears on the current screen.

5250 computers support the **Roll** command (hex 23) received from a host application. Using this command, a host application can roll an area of the screen up or down. The direction of the roll and number of lines to roll are specified in the command.

Do not confuse the **Roll Up** and **Roll Down** AID keys with the **Roll** command. The **Roll** keys cause the host to send down additional screens when you are at a **Roll** screen. A **Roll** screen typically has text in the lower right-hand corner of the screen to indicate additional screens to view.

Note the following:

- A **Roll** command received from a host application moves the screen, but not the window/viewport. The screen scrolls through the window/

viewport when you roll up or down, but the window/viewport remains stationary.

- Paging keys (window/viewport page up, window/viewport page down, window/viewport page right, window/viewport page left) move the viewport within one screen. They do not move the screen itself.

Cursor Keys

You can manually move the computer's window/viewport by using the cursor keys and paging keys. For more information about the window/ viewport, refer to the computer's user manual.

Field Exit Key

Field Exit exits an input field and moves the cursor to the beginning of the next input field. If you press this key while the cursor is between characters, all characters in the field to the right of the cursor are erased.

Signal Keys

Signal keys cause a Signal command to go from the controller to the host system.

Signal Keys

Key	Description
Attn	The operator presses this key to alert the host system the function request (such as [ENTER]) is not honored. **Attn** is valid when the keyboard is locked or unlocked. It does not change the keyboard state or the cursor location.
Help (from error state)	Operator uses this key to request the host system send data about the error to the display.

Special Control Keys

Use special control keys to change operator-generated information in the display. These keys do not work when the keyboard is locked.

Special Control Keys

Key	Description
Del	Deletes the character in the position where the cursor was located. All remaining characters in the field shift to the left to fill the column.
Erase Input	Clears all fields to nulls, and cursor moves to the first input position on screen. This command does not erase protected fields. If you press this key when the screen shows only protected fields, the cursor returns to the home position.
Error Reset	Restores the original data on the error line of the display and resets the state.
Hex	Enters hexadecimal codes from the keypad to generate EBCDIC characters needed for input or display.
Home	Moves the cursor to the position specified by the insert cursor (IC) address.

Special Control Keys (continued)

Insert	Sets or turns off the insert mode for the input field the operator has the cursor in. The operator must reset the insert state before exiting it, by either pressing [RESET] or [INSERT] again.
Shift Lock	Puts the keyboard into shift lock mode.

Special Host Key

The special 5250 host key is System Request.

Special Host Key

Key	**Description**
Sys Req	Data on the error line is saved, the error line is cleared, a column separator and underscore field attribute are supplied to column 1 of the error line, and the cursor is located under column 2 to begin polling keystrokes.

5250 Additional Functions

This table lists additional operations you can perform on your computer. To enter an operation, press the keys listed in the chapter for the computer or scan the bar code listed in Appendix A, "Bar Code Scanning" on page 391."

5250 Additional Functions

Function	**Description**
¢ (cent sign)	Enters a cent sign.
↵ (New Line)	Moves the cursor to the first unprotected character position of the first line in the screen. If the screen is a protected field, the cursor returns to the home position.
¬ (Not symbol)	Enters a Not symbol.
Back Tab	Moves the cursor back to the most recent first field position. If in the middle of a field, it moves it to the first position of the same field. If the cursor is at the first position of a field, it moves it to the first position of the preceding input field.
Dup (duplicate enabled fields only)	Controller repeats hex "1C" from the cursor position to the end of the field. This shows in the display as an overstruck asterisk.
Field-	Advances cursor to the previous input field. For numeric fields, makes the input a negative number.
Field+	Advances cursor to the next input field. For numeric fields, makes the input a positive number.
Field Mark	This is valid within any unprotected entry input field in which the **Dup** or **Field Mark** key is allowed (FFW bit 3 set to on). The **Field Mark** character (X`1E') is displayed as an overscore on IBM 5251 Display Stations and as a space on all other supported workstations. If an operator presses the **Field Mark** key in an entry field that does not allow the **Dup** or **Field Mark** key, operator error 0019 is posted. The **Field Mark** code point is allowed in an outbound data stream.
Forward Tab	Moves the cursor to the first position in the next input field.

System Messages

The computer's display reserves a line for status information. The status line can display system (non-local) information such as a message waiting from the host computer, help messages in response to the **[Help]** key, or the system request state of the computer.

VT/ANSI Applications

The following information pertains to VT/ANSI computers.

VT/ANSI Main Keypad

The VT/ANSI computer's main keypad consists of standard keys and function keys. Standard keys generate letters, numbers, and symbols. Function keys generate special function codes. The following table describes the keys.

VT/ANSI Main Keypad

Keys	Description
Compose character	The computer does not support this function, which starts a compose sequence that creates characters that cannot be typed directly from the keyboard.
Ctrl	The Ctrl key is used with another key to send a control code.
Delete	Operation depends on how the DEL to BS option is set in the TE configuration menus. The key either sends a delete (DEL, 7F hexadecimal) or a backspace (BS, 08 hexadecimal).
Lock	The Lock key alone does not send a code. It is used with shift-lock, which sets/clears shift-lock.
Return	Sends either a CR character (0D hexadecimal) or a CR character (0D hexadecimal) and an LF character (0A hexadecimal), depending on the set or reset state of line feed or new line mode (LNM).
Shift	The Shift key alone does not send a code. It is used with other standard keys to send uppercase characters.
Space bar	Sends an SP character (20 hexadecimal).
Tab	Sends an HT character (09 hexadecimal).

VT/ANSI Editing Keypad

The computer's editing keypad has editing keys and cursor (arrow) keys.

Cursor Keys

You can use cursor keys and paging keys to manually move the computer's window/viewport. See the computer's user's manual for more information.

Editing Keys

Editing keys have functions assigned to them by the application software in use. Refer to your application's software manual for information about editing key functions. Editing keys are **Find**, **Insert**, **Next Screen**, **Previous Screen**, **Remove**, and **Select**.

To enter an editing key

- Press the keys listed in the section for the computer or scan bar codes in Appendix A, “Bar Code Scanning.”

VT/ANSI Auxiliary Keys

The VT/ANSI computer's auxiliary keypad consists of numeric keys (which enter numeric data) and programmable function (PF) keys. The following chart describes VT/ANSI computer auxiliary keypad operations.

VT/ANSI Auxiliary Keys

Key	Description
0-9	Enters numeric data.
- (hyphen)	Enters a hyphen character.
, (comma)	Enters a comma character.
. (period)	Enters a period character.
Enter	Sends CR, CRLF, or SS# M, depending on the mode settings.
PF1-PF4	The application software in use assign operations to these PF keys. See the application’s software manual for programmed uses of these keys.

To enter an auxiliary key

- Press the keys while the computer is in Keypad mode or scan the bar code in Appendix A, “Bar Code Scanning.”

VT/ANSI Top-Row Function Keys

VT220/320/340 computers support function keys **[F1]**-**[F20]**. Keys **[F1]**-**[F5]** are used for hold screen, print screen, set-up, data/talk, and break. The computer supports only the break function. For VT220/320/340, **[F1]**-**[F4]** are PF1-PF4.

VT/ANSI Top-Row Function Keys

Key	Description
F5 (Break)	Sends a break function to the host.
F6-F20	User-defined keys (UDKs) that have operations assigned to them by the application software in use. Refer to your application's software manual for their uses.

Note: VT100 computers only support top-row function keys **[F11]** (Escape), **[F12]** (Backspace), and **[F13]** (Line feed).

To enter a top-row function key

- Press the keys listed in the section for the computer or scan the bar code in Appendix A, “Bar Code Scanning.”

VT/ANSI Transmission Mode

Use the transmission mode (labeled “Mode” on the overlay) to toggle between Line Edit (block) mode and Character mode. These modes are described in Chapter 6, “Programming” on page 197.

When Lock mode is disabled, you can press the Mode key to toggle between Line Edit (block) mode and Character mode. When Lock mode is enabled, you cannot toggle between the modes. By default, Lock mode is disabled. You can configure Lock mode through the TE configuration menus. See “Using theTerminal Emulation Menus” on page 97 for information about the menus.

VT/ANSI Local Edit Mode

If your application software program supports local editing, you can use the computer in Local Edit Mode, a feature of the VT330/ VT340 computer. Local Edit Mode is described in Chapter 6, “Local Edit Mode” on page 335.

VT/ANSI Printing and Serial Scanning

You can print data from a VT/ANSI host. To connect your computer to a printer, refer to the computer's user manual for instructions.

Using the Print Modes

The following chart defines the print modes you can use with the TE 2000 VT/ANSI application.

Print Modes

Print Mode	Description
Auto print	Prints each line after the cursor leaves that line using a carriage return or when auto-advancing through fields. This mode can be turned on and off from a VT/ANSI host.
Printer controller	Prints all data from a VT/ANSI host. Turn this mode on or off from the host as all host screens are printed without allowing the user to respond. You cannot log on or off while in this mode.
Print cursor line	Prints the line that the cursor is on. This mode can only be turned on from a VT/ANSI host and turns off after the line prints.
Print form feed	After a screen is printed, the printer advances the printed screen out of the printer. This mode can be turned on and off from a VT/ANSI host.

To send commands from the host

- Refer to the programmer's guide for your VT/ANSI host for help.

Configuring Printing and Serial Scanning Options

To set printing and serial scanning options

1. Connect your Intermec computer to a printer.
2. Configure the serial port on the computer to match the parameters set for the serial port on the printer. From the Main Menu, select **1) Set-up Parms**, then enter “cr52401” at the Enter Password prompt.
3. Select **3) Protocol Opts, 6) VT/ANSI**, then **7) More**.
4. Select **4) RS232 Setup**, then set the baud rate. For help, see the User’s Manual for your Intermec computer.

3 Using the Computer Keypad

Your computer has a special keypad that contains most of the keys available on your applicable computer. Use the keypad to enter data in the TE 2000 screens.

The keys on the keypad have their main character or operation marked directly on the key itself. To access that character or option, just press the key.

This chapter is broken down by computer model, with respective terminal emulations listed beneath.

For help with using your computer, refer to any of the documents listed in the table of "Related Documents" on page xix.

Launching TE 2000

Note: See "Starting the TE 2000 Application" on page 3 for more detailed information on how the TE 2000 application is launched.

The TE 2000 application automatically starts when the computer is booted except on the 700 Color Mobile Computer and the CN3 terminal. On these computers, tap **Start > TE2000** to start the TE 2000 application.

To disable automatic startup of the TE 2000 application

1 Start Windows Explorer and navigate to the folder where the TE 2000 application was installed by default.

2 Move the te2000.exe application to a different location or rename the exe file.

To manually launch the TE 2000 application

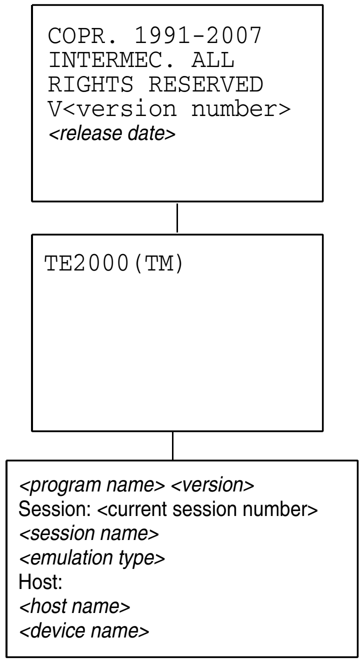
1 Click or tap **Start > TE2000**

2 Navigate to the folder where the TE 2000 application is.

3 Launch the appropriate program (default is te2000.exe).

Note: "700 Color" refers to Intermec 730, 740, 741, 750, 751, 760, or 761 Mobile Computers unless otherwise noted.

If you are running the TE 2000 application for the first time, you will see a logo for about 35 seconds while the TE 2000 application creates the default TE_SETTINGS.INI file. Then the default toolbar is displayed, followed by the default Soft Input Panel (SIP), ending with a program screen like the following.



Soft Input Panels (700 Color, CK31, CK32, CK61, CN3, CV30, CV60)

Note: For CK61s, CK32s, CN3s, and CV30s using Windows Mobile 2005, if the SIP is onscreen and you press a key on the physical keypad, the SIP will exit the screen and the TE 2000 application will go to full screen. Press **T** (toggle) on the TE 2000 toolbar to return the SIP to the screen.

The illustrations on the next two pages show the Soft Input Panels (SIPs) for the respective TE 2000 3270, 5250, Native, or VT/ANSI application.

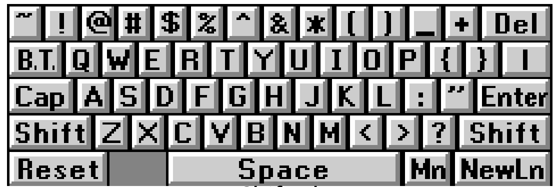
Using the SIPs

Observe the following rules when using these SIPs:

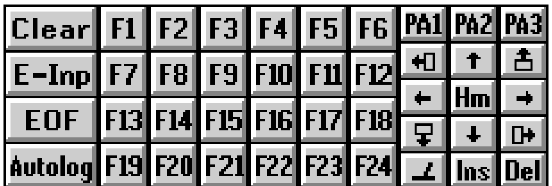
- Tap the **Mn** key off the Shifted keypad to get to the TE 2000 Setup Menus.
- Tap the **Shift** key to toggle between the Main and Shifted keypads.
- Tap the toggle key to toggle between the Function and Main keypads.
- Tap the **Cap** key to use uppercase keys with numbers.
- Tap the **Cap** key, then the **Shift** key to use lowercase keys with shifted characters.

3270/Native Keypads

Default

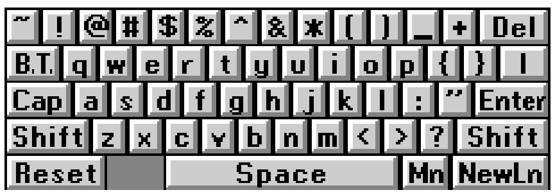


Shifted



Function Toggled

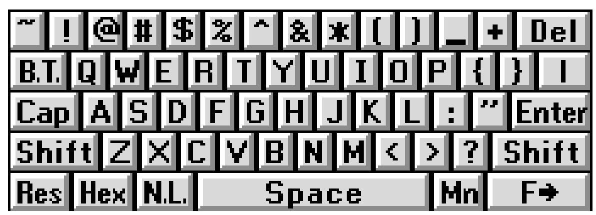
Caps Locked



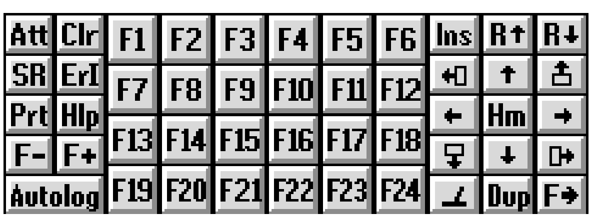
Caps Locked Shifted

5250 Keypads

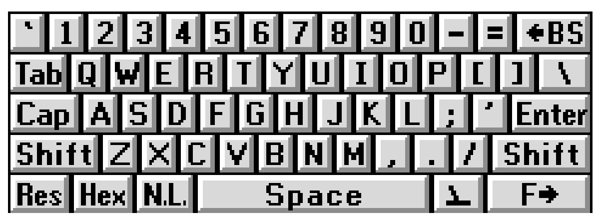
Default



Shifted



Function Toggled

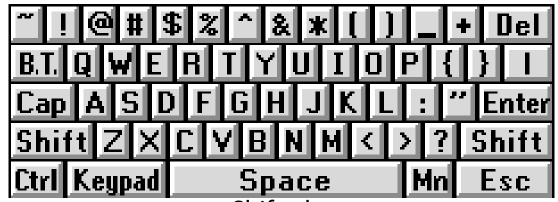


Caps Locked

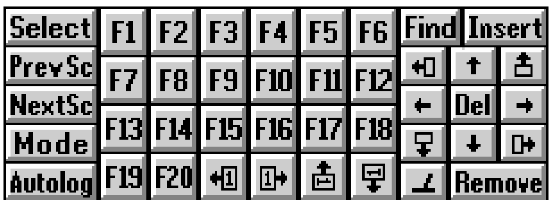
Caps Locked Shifted

VT/ANSI Keypads

Default

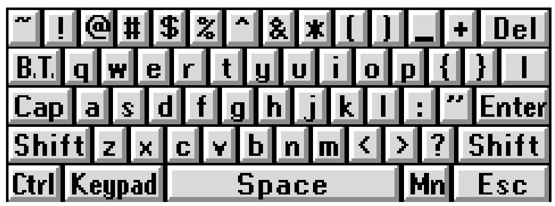


Shifted



Function Toggled

Caps Locked



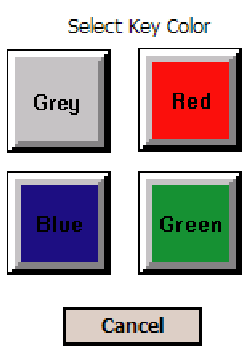
Caps Locked Shifted

Changing the SIP Key Color

With the Soft Input Panels (SIPs), you can change the color of up to ten keys, given four color choices (including the original gray).

To change the color of the keys *(only up to 10 keys)*

1. Hold the left Shift key down until it reverses back to normal.
2. Hold the right Shift key down until it reverses back to normal.

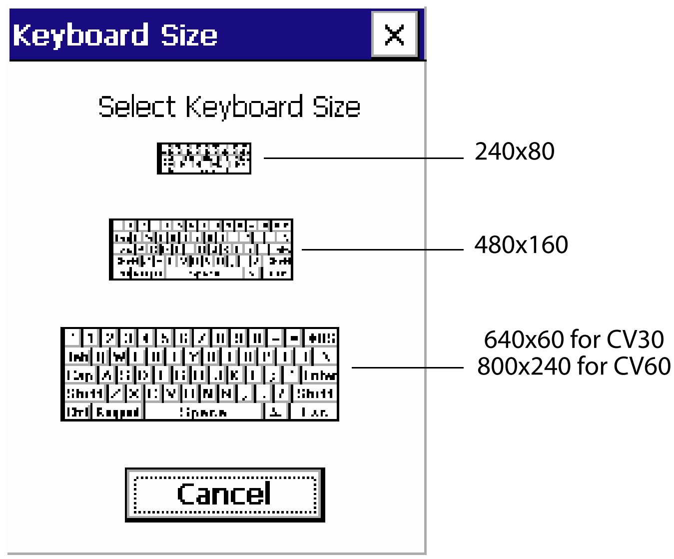


Resizing the SIPs

(CV30, CV60 only) CV30s and CV60s have the ability to resize their SIPs (w x h) to 240x80, 480x160, 640x160 for CV30, or 800x240 for CV60.

To change the size of the SIP keys

1. Hold the left [Shift] key down until it reverses back to normal.
2. Press and hold the minus (-) key until the following screen appears.
3. Select which size you want to use.
4. Tap the **X** to close this screen.

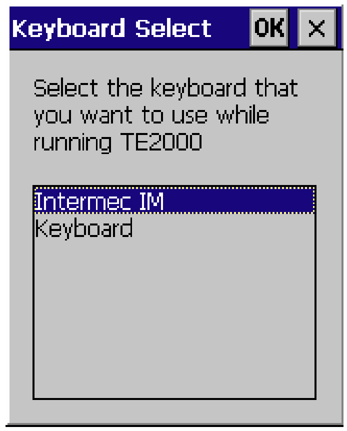


Creating Custom SIPs

You can create a custom SIP with the Intermec SIP Designer application. See its online help for instructions on installing the custom SIP to your computer. Contact your Intermec representative for more information.

To switch SIPs

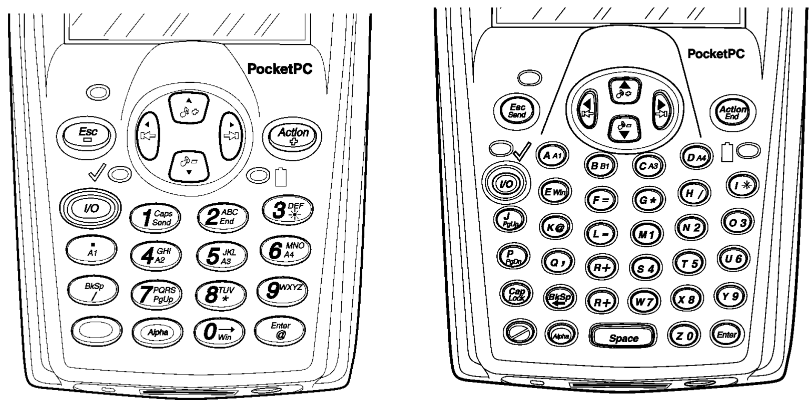
1. Enter the TE 2000 Setup Menu.
2. Select 7) **More** > **1) Keyboard Opts** > **2) SIP Settings** to launch a window of available SIPs with the active SIP highlighted. *The default SIP for TE 2000 is the Intermec IM keyboard.*



3. To switch to a different SIP, select another SIP in the list, then press **Enter** or tap **OK**. Press **[Esc]** or tap **X** to cancel your selection and return to the Keyboard Opts menu.

700 Color Mobile Computers

The following information pertains to 700 Colors.



Numeric keypad   Alphanumeric keypad

Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

700 Color Cursor Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up			
Window/viewport down			
Window/viewport right			
Window/viewport left			

700 Color Paging Keys

To Enter	3270/Native	5250	VT/ANSI
Page up			
Page down			
Page right			
Page left			

700 Color VT/ANSI Host Cursor Keys

To Enter	Press the Computer Keys	Tap the SIP Keys
Host cursor up key	[▲]	
Host cursor left key	[▶]	
Host cursor right key	[▶]	
Host cursor down key	[▼]	

700 Color Standard Keys

Note: See the *700 Series Mobile Computer User's Manual* for information on how to enter alpha characters using the 700 Color keypads. Keep in mind the [Alpha] key and the [Caps] key are toggle keys – these remain on until pressed again to turn them off.

700 Color Alphanumeric Characters

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
a	[Alpha] [2]	[Alpha] [A]	[a]
b	[Alpha] [2] [2]	[Alpha] [B]	[b]
c	[Alpha] [2] [2] [2]	[Alpha] [C]	[c]
d	[Alpha] [3]	[Alpha] [D]	[d]
e	[Alpha] [3] [3]	[Alpha] [E]	[e]
f	[Alpha] [3] [3] [3]	[Alpha] [F]	[f]
g	[Alpha] [4]	[Alpha] [G]	[g]
h	[Alpha] [4] [4]	[Alpha] [H]	[h]
i	[Alpha] [4] [4] [4]	[Alpha] [I]	[i]
j	[Alpha] [5]	[Alpha] [J]	[j]
k	[Alpha] [5] [5]	[Alpha] [K]	[k]
l	[Alpha] [5] [5] [5]	[Alpha] [L]	[l]
m	[Alpha] [6]	[Alpha] [M]	[m]
n	[Alpha] [6] [6]	[Alpha] [N]	[n]
o	[Alpha] [6] [6] [6]	[Alpha] [O]	[o]
p	[Alpha] [7]	[Alpha] [P]	[p]
q	[Alpha] [7] [7]	[Alpha] [Q]	[q]
r	[Alpha] [7] [7] [7]	[Alpha] [R]	[r]
s	[Alpha] [7] [7] [7] [7]	[Alpha] [S]	[s]
t	[Alpha] [8]	[Alpha] [T]	[t]
u	[Alpha] [8] [8]	[Alpha] [U]	[u]
v	[Alpha] [8] [8] [8]	[Alpha] [V]	[v]
w	[Alpha] [9]	[Alpha] [W]	[w]
x	[Alpha] [9] [9]	[Alpha] [X]	[x]
y	[Alpha] [9] [9] [9]	[Alpha] [Y]	[y]
z	[Alpha] [9] [9] [9] [9]	[Alpha] [Z]	[z]
A	[Alpha] [1] [2]	[Alpha] [CapLock] [A]	Shift [A]
B	[Alpha] [1] [2] [2]	[Alpha] [CapLock] [B]	Shift [B]
C	[Alpha] [1] [2] [2] [2]	[Alpha] [CapLock] [C]	Shift [C]
D	[Alpha] [1] [3]	[Alpha] [CapLock] [D]	Shift [D]
E	[Alpha] [1] [3] [3]	[Alpha] [CapLock] [E]	Shift [E]
F	[Alpha] [1] [3] [3] [3]	[Alpha] [CapLock] [F]	Shift [F]
G	[Alpha] [1] [4]	[Alpha] [CapLock] [G]	Shift [G]
H	[Alpha] [1] [4] [4]	[Alpha] [CapLock] [H]	Shift [H]

700 Color Alphanumeric Characters (continued)

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
I	[Alpha] [1] [4] [4] [4]	[Alpha] [CapLock] [I]	Shift [I]
J	[Alpha] [1] [5]	[Alpha] [CapLock] [J]	Shift [J]
K	[Alpha] [1] [5] [5]	[Alpha] [CapLock] [K]	Shift [K]
L	[Alpha] [1] [5] [5] [5]	[Alpha] [CapLock] [L]	Shift [L]
M	[Alpha] [1] [6]	[Alpha] [CapLock] [M]	Shift [M]
N	[Alpha] [1] [6] [6]	[Alpha] [CapLock] [N]	Shift [N]
O	[Alpha] [1] [6] [6] [6]	[Alpha] [CapLock] [O]	Shift [O]
P	[Alpha] [1] [7]	[Alpha] [CapLock] [P]	Shift [P]
Q	[Alpha] [1] [7] [7]	[Alpha] [CapLock] [Q]	Shift [Q]
R	[Alpha] [1] [7] [7] [7]	[Alpha] [CapLock] [R]	Shift [R]
S	[Alpha] [1] [7] [7] [7] [7]	[Alpha] [CapLock] [S]	Shift [S]
T	[Alpha] [1] [8]	[Alpha] [CapLock] [T]	Shift [T]
U	[Alpha] [1] [8] [8]	[Alpha] [CapLock] [U]	Shift [U]
V	[Alpha] [1] [8] [8] [8]	[Alpha] [CapLock] [V]	Shift [V]
W	[Alpha] [1] [9]	[Alpha] [CapLock] [W]	Shift [W]
X	[Alpha] [1] [9] [9]	[Alpha] [CapLock] [X]	Shift [X]
Y	[Alpha] [1] [9] [9] [9]	[Alpha] [CapLock] [Y]	Shift [Y]
Z	[Alpha] [1] [9] [9] [9] [9]	[Alpha] [CapLock] [Z]	Shift [Z]
0 - 9	[0] - [9]	[Z] - [Y]	[0] - [9]
Symbols	Symbol key, or Shift plus corresponding key.		

700 Color Function Keys

To Enter	3270/Native	5250	VT/ANSI
Back Tab	Shift B.T.	Shift B.T.	Shift B.T.
Ctrl	Not Applicable	Not Applicable	Ctrl
Delete	Shift Del	Shift Del	Shift Del
Forward Tab	Tab	Tab	Tab
Lock	Cap	Cap	Cap
Return	Enter	Enter	Enter
Shift	Shift	Shift	Shift
Space bar	Space	Space	Space
Clear	Clear	Clear	Not Applicable
PA1	PA1	Not Applicable	Not Applicable
PA2	PA2	Not Applicable	Not Applicable
PA3	PA3	Not Applicable	Not Applicable

700 Color Editing Keys

To Enter	3270/Native	5250	VT/ANSI
E-Inp	[⌙] [E-Inp]	Not Applicable	Not Applicable
Clr	Not Applicable	[⌙] [Erl]	Not Applicable
Del	[⌙] [Del]	[⌙] [Del]	Not Applicable
Enter	[Enter]	[Enter]	[Enter]
EOF	[⌙] [EOF]	Not Applicable	Not Applicable
Home	[⌙] [Hm]	[⌙] [Hm]	Not Applicable
Insert	[⌙] [Ins]	[⌙] [Ins]	Not Applicable
New Line (Return)	[NewLn]	[⌙] [N.L.]	Not Applicable
Reset	[Reset]	[Reset]	Not Applicable
Find	Not Applicable	Not Applicable	[⌙] [Find]
Insert here	Not Applicable	Not Applicable	[⌙] [Insert]
Next screen	Not Applicable	Not Applicable	[⌙] [NextSc]
Prev screen	Not Applicable	Not Applicable	[⌙] [PrevSc]
Remove	Not Applicable	Not Applicable	[⌙] [Remove]
Select	Not Applicable	Not Applicable	[⌙] [Select]

Note: For 3270 and 5250 computers, pressing **Esc** on the 700 Color numeric keypad or pressing [E] on the 700 Color alphanumeric keypad also performs the Reset function. Intermec Application Servers do not support the **F5** (break) function.

700 Color Auxiliary Keys

To Enter	Tap the SIP Keys
- (hyphen)	[-]
, (comma)	[,]
. (period)	[.]
_ (underscore)	[Shift] [-]
Enter	[Enter]

Note: 3270, 5250, Native computers: for the F11-F24 keys
VT/ANSI computers: for the F11-F20 keys

Press either the uppercase sequence or the lowercase sequence (separated by the "or" conjunction), but *not* both. Keep in mind the [Alpha] key and the [Caps] or [CapLock] key are toggle keys – these remain on until pressed again to turn them off.

700 Color Function Keys

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
F1 or PF1	[Action] [1]	[Action] [M]	[⌙] [F1]
F2 or PF2	[Action] [2]	[Action] [N]	[⌙] [F2]

700 Color Function Keys (continued)

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
F3 or PF3	[Action] [3]	[Action] [O]	F3
F4 or PF4	[Action] [4]	[Action] [S]	F4
F5	[Action] [5]	[Action] [T]	F5
F6	[Action] [6]	[Action] [U]	F6
F7	[Action] [7]	[Action] [W]	F7
F8	[Action] [8]	[Action] [X]	F8
F9	[Action] [9]	[Action] [Y]	F9
F10	[Action] [0]	[Action] [Z]	F10
F11	[Action] [Alpha] [1] [2] or [Action] [Alpha] [2]	[Action] [Alpha] [CapLock] [A] or [Action] [Alpha] [A]	F11
F12	[Action] [Alpha] [1] [2] [2] or [Action] [Alpha] [2] [2]	[Action] [Alpha] [CapLock] [B] or [Action] [Alpha] [B]	F12
F13	[Action] [Alpha] [1] [2] [2] [2] or [Action] [Alpha] [2] [2] [2]	[Action] [Alpha] [CapLock] [C] or [Action] [Alpha] [C]	F13
F14	[Action] [Alpha] [1] [3] or [Action] [Alpha] [3]	[Action] [Alpha] [CapLock] [D] or [Action] [Alpha] [D]	F14
F15	[Action] [Alpha] [1] [3] [3] or [Action] [Alpha] [3] [3]	[Action] [Alpha] [CapLock] [E] or [Action] [Alpha] [E]	F15
F16	[Action] [Alpha] [1] [3] [3] [3] or [Action] [Alpha] [3] [3] [3]	[Action] [Alpha] [CapLock] [F] or [Action] [Alpha] [F]	F16
F17	[Action] [Alpha] [1] [4] or [Action] [Alpha] [4]	[Action] [Alpha] [CapLock] [G] or [Action] [Alpha] [G]	F17
F18	[Action] [Alpha] [1] [4] [4] or [Action] [Alpha] [4] [4]	[Action] [Alpha] [CapLock] [H] or [Action] [Alpha] [H]	F18
F19	[Action] [Alpha] [1] [4] [4] [4] or [Action] [Alpha] [4] [4] [4]	[Action] [Alpha] [CapLock] [I] or [Action] [Alpha] [I]	F19
F20	[Action] [Alpha] [1] [5] or [Action] [Alpha] [5]	[Action] [Alpha] [CapLock] [J] or [Action] [Alpha] [J]	F20
For 3270, 5250, Native computers:			
F21	[Action] [Alpha] [1] [5] [5] or [Action] [Alpha] [5] [5]	[Action] [Alpha] [CapLock] [K] or [Action] [Alpha] [K]	F21

700 Color Function Keys (continued)

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
F22	[Action] [Alpha] [1] [5] [5] [5] or [Action] [Alpha] [5] [5] [5]	[Action] [Alpha] [CapLock] [L] or [Action] [Alpha] [L]	F22
F23	[Action] [Alpha] [1] [6] or [Action] [Alpha] [6]	[Action] [Alpha] [CapLock] [M] or [Action] [Alpha] [M]	F23
F24	[Action] [Alpha] [1] [6] [6] or [Action] [Alpha] [6] [6]	[Action] [Alpha] [CapLock] [N] or [Action] [Alpha] [N]	F24

Note: For 3270 and 5250 computers, pressing **Esc** on the 700 Color numeric keypad or pressing **E** on the 700 Color alphanumeric keypad also performs the Reset function. Intermec Application Servers do not support the **F5** (break) function.

700 Color Auto-Login Restart

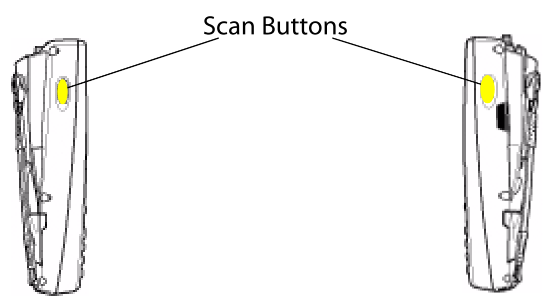
To enter Auto-Login Restart, tap the SIP Autolog keys or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

700 Color Scan Buttons

The scan buttons, or scanner triggers, on the 700 Color are the yellow buttons on either side of the computer.



700 Color 5250 Field Exit Characters

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
Field Exit	Not supported.	[Gold/White] [Enter]	[F→]

700 Color 5250 Signal Keys

To Enter	Tap the SIP Keys
Attn	 [Att]
Help *(from error state)*	 [Hlp]

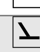
700 Color 5250 Special Control Keys

To Enter	Tap the SIP Keys
Del	[Shift] [Del]
Erase Input	 [Erl]
Error Reset	[Res]
Hex	[Hex]
Home	 [Hm]
Insert	 [Ins]
Shift Lock	[Cap]

700 Color 5250 Special Host Keys

To Enter	Tap the SIP Keys
Sys Req	 [Shift]

700 Color 5250 Additional Functions

To Enter	Tap the SIP Keys
↵ (New Line)	[N.L.]
Dup *(duplicate enabled fields only)*	 [Dup]
Field-	 [F-]
Field+	 [F+]
Field Mark	Not supported.

700 Color 5250 Additional Functions

To Enter	Press the Numeric Keys	Press the Alphanumeric Keys	Tap the SIP Keys
¬ (Not symbol)	 [Shift] [6]	[Shift] [6]	[Shift] [^]

700 Color VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press Mode from the SIPs.

700 Color VT/ANSI Control Keys

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
SOH	Ctrl [A]	DC1, X-ON	Ctrl [Q]
STX	Ctrl [B]	DC2	Ctrl [R]
ETX	Ctrl [C]	DC3, X-OFF	Ctrl [S]
EOT	Ctrl [D]	DC4	Ctrl [T]
ENQ	Ctrl [E]	NAK	Ctrl [U]
ACK	Ctrl [F]	SYN	Ctrl [V]
BEL	Ctrl [G]	ETB	Ctrl [W]
BS	Ctrl [H]	CAN	Ctrl [X]
HT	Ctrl [I]	EM	Ctrl [Y]
LF	Ctrl [J]	SUB	Ctrl [Z]
VT	Ctrl [K]	ESC	Esc
FF	Ctrl [L]	FS	Ctrl [1]
CR	Ctrl [M]	GS	Ctrl [2]
SO	Ctrl [N]	RS	Ctrl [3]
SI	Ctrl [O]	US	Ctrl [4]
DLE	Ctrl [P]	DEL	Shift Del

700 Color VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	Shift Mn on the SIP, on the keypads, or double-tap the upper-right corner of the display.

CK30/CK31 Handheld Computers

This information pertains to CK30 and CK31 Handheld Computers.

Characters on the CK30/CK31 Keypads

The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys. The shift keys are:

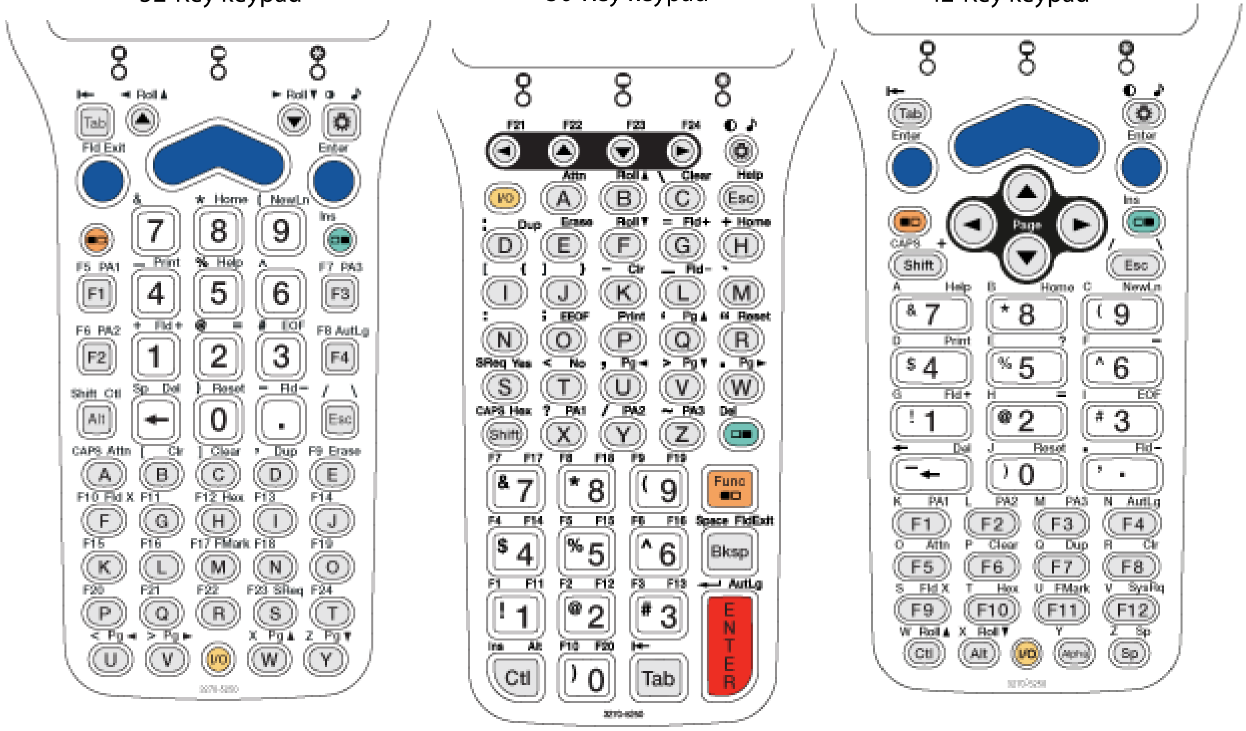
CK30/CK31 Shift Keys

Shift Key	Function
[Orange]	Press plus a key to type a character or do an operation printed in orange on the overlay.
[Green]	Press plus a key to type a character or do an operation printed in green on the overlay.

CK30/CK31 Keypads

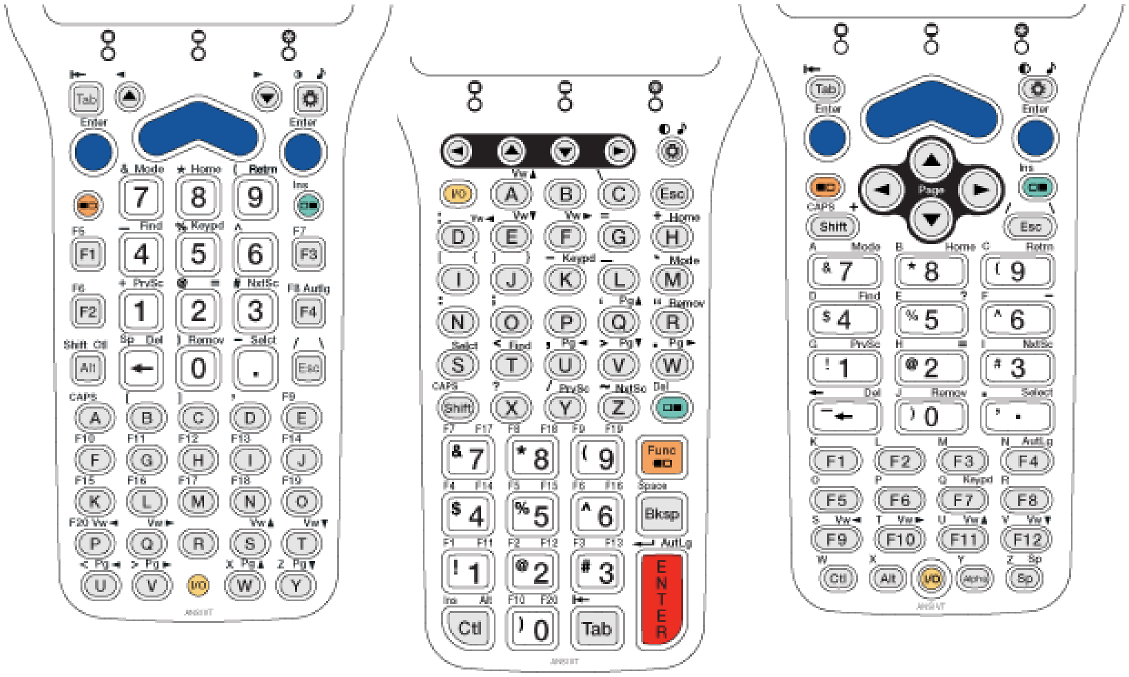
Your CK30/CK31 has a 52-key, a 50-key, or a 42-key keypad.

52-Key keypad 50-Key keypad 42-Key keypad



3270, 5250, and Native Terminal Emulation keypads for the CK30/CK31

52-Key keypad 50-Key keypad 42-Key keypad



VT/ANSI Terminal Emulation keypads for the CK30/CK31

CK30/CK31 52-Key Cursor Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	[▲]	[▲]	◨ [S]
Window/viewport down	[▼]	[▼]	◨ [T]
Window/viewport right	[Func] [▲]	◧ [U]	◨ [Q]
Window/viewport left	[Func] [▼]	◧ [D]	◨ [P]

CK30/CK31 50-Key Cursor Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	[▲]	[▲]	◨ [A]
Window/viewport down	[▼]	[▼]	◨ [E]
Window/viewport right	[▶]	[▶]	◨ [F]
Window/viewport left	[◀]	[◀]	◨ [D]

CK30/CK31 42-Key Cursor Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	[▲]	[▲]	◨ [F11]
Window/viewport down	[▼]	[▼]	◨ [F12]
Window/viewport right	[▶]	[▶]	◨ [F10]
Window/viewport left	[◀]	[◀]	◨ [F9]

CK30/CK31 52-Key Paging Keys

To Enter	3270/Native	5250	VT/ANSI
Page up	[W]	[W]	[W]
Page down	[Y]	[Y]	[Y]
Page right	[V]	[V]	[V]
Page left	[U]	[U]	[U]

CK30/CK31 50-Key Paging Keys

To Enter	3270/Native	5250	VT/ANSI
Page up	[Q]	[Q]	[Q]
Page down	[V]	[V]	[V]
Page right	[W]	[W]	[W]
Page left	[U]	[U]	[U]

CK30/CK31 42-Key Paging Keys

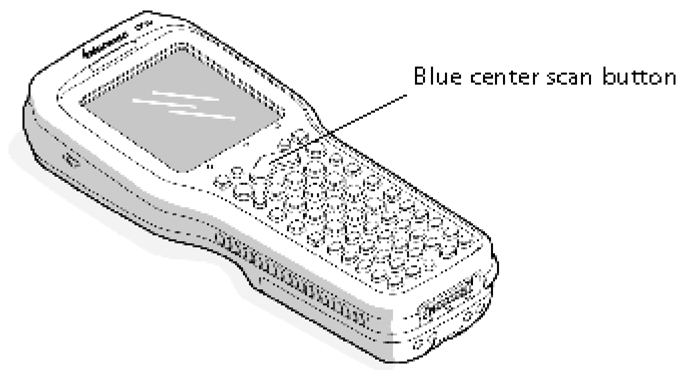
To Enter	3270/Native	5250	VT/ANSI
Page up	[▲]	[▲]	[▲]
Page down	[▼]	[▼]	[▼]
Page right	[▶]	[▶]	[▶]
Page left	[◀]	[◀]	[◀]

CK30/CK31 VT/ANSI Host Cursor Keys

	Press the Computer Keys			
To Enter	**52-Key**	**50-Key**	**42-Key**	**Tap the SIP Keys**
Host cursor up key	[▲]	[▲]	[▲]	↑
Host cursor down key	[▼]	[▼]	[▼]	↓
Host cursor right key	[Shift] [D]	[▶]	[▶]	→
Host cursor left key	[Shift] [U]	[◀]	[◀]	←

CK30/CK31 Scan Button

The scan button on the CK30/CK31 is the blue center button on the 52-key and 42-key keypads. The 50-key keypad does not have a scan button as its handle has a built-in scanner trigger. See the applicable user's manual for more information.



Note: See the *CK30 Handheld Computer User's Manual* or the *CK31 Handheld Computer User's Manual* for information on how to enter alpha characters using the CK30/CK31 keypads. Keep in mind the ■□ [Alt] and ■□ [A] keys are toggle keys for the 52-key keypad and the [Shift] key and the ■□ [Shift] keys are toggle keys for the 50-key and 42-key keypads – these remain on until pressed again to turn them off.

CK30/CK31 Standard Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
a	[A]	[A]	■□ [7]
b	[B]	[B]	■□ [8]
c	[C]	[C]	■□ [9]
d	[D]	[D]	■□ [4]
e	[E]	[E]	■□ [5]
f	[F]	[F]	■□ [6]
g	[G]	[G]	■□ [1]
h	[H]	[H]	■□ [2]
i	[I]	[I]	■□ [3]
j	[J]	[J]	■□ [0]
k	[K]	[K]	■□ [F1]
l	[L]	[L]	■□ [F2]
m	[M]	[M]	■□ [F3]
n	[N]	[N]	■□ [F4]
o	[O]	[O]	■□ [F5]
p	[P]	[P]	■□ [F6]

CK30/CK31 Standard Keys (continued)

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
q	[Q]	[Q]	[F7]
r	[R]	[R]	[F8]
s	[S]	[S]	[F9]
t	[T]	[T]	[F10]
u	[U]	[U]	[F11]
v	[V]	[V]	[F12]
w	[W]	[W]	[Ctl]
x	[W]	[X]	[Alt]
y	[Y]	[Y]	[Alpha]
z	[Y]	[Z]	[Sp]
A	[Alt] [A]	[Shift] [A]	[Shift] [7]
B	[Alt] [B]	[Shift] [B]	[Shift] [8]
C	[Alt] [C]	[Shift] [C]	[Shift] [9]
D	[Alt] [D]	[Shift] [D]	[Shift] [4]
E	[Alt] [E]	[Shift] [E]	[Shift] [5]
F	[Alt] [F]	[Shift] [F]	[Shift] [6]
G	[Alt] [G]	[Shift] [G]	[Shift] [1]
H	[Alt] [H]	[Shift] [H]	[Shift] [2]
I	[Alt] [I]	[Shift] [I]	[Shift] [3]
J	[Alt] [J]	[Shift] [J]	[Shift] [0]
K	[Alt] [K]	[Shift] [K]	[Shift] [F1]
L	[Alt] [L]	[Shift] [L]	[Shift] [F2]
M	[Alt] [M]	[Shift] [M]	[Shift] [F3]
N	[Alt] [N]	[Shift] [N]	[Shift] [F4]
O	[Alt] [O]	[Shift] [O]	[Shift] [F5]
P	[Alt] [P]	[Shift] [P]	[Shift] [F6]
Q	[Alt] [Q]	[Shift] [Q]	[Shift] [F7]
R	[Alt] [R]	[Shift] [R]	[Shift] [F8]
S	[Alt] [S]	[Shift] [S]	[Shift] [F9]
T	[Alt] [T]	[Shift] [T]	[Shift] [F10]
U	[Alt] [U]	[Shift] [U]	[Shift] [F11]
V	[Alt] [V]	[Shift] [V]	[Shift] [F12]
W	[Alt] [W]	[Shift] [W]	[Shift] [Ctl]
X	[Alt] [W]	[Shift] [X]	[Shift] [Alt]
Y	[Alt] [Y]	[Shift] [Y]	[Shift] [Alpha]
Z	[A] [Y]	[Shift] [Z]	[Shift] [Sp]
0-9	[0] - [9]	[0] - [9]	[0] - [9]

CK30/CK31 Standard Keys (continued)

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Symbols	[Shift] plus corresponding key *(3270, Native)* ■□ plus corresponding key *(VT/ANSI)*	[Shift] plus corresponding key *(3270, Native, VT/ANSI)*	[Shift] plus corresponding key *(3270, Native, VT/ANSI)*

CK30/CK31 Function Keys – 3270, Native, VT/ANSI

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Backspace	←	[Bksp]	←
Caps Lock	■□ [A]	[Func] [Shift]	■□ [Shift]
Ctrl	□■ [Alt]	[Ctl]	[Ctl]
Delete	□■ ←	[Func] □■	□■ ←
Forward Tab	[Tab]	[Tab]	[Tab]
Return	[Enter]	[Enter]	[Enter]
Shift	■□ [Alt]	[Shift]	[Shift]
Space bar	■□ ←	[Func] [Bksp]	[Sp]

CK30/CK31 Additional Functions

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Access TE configuration menus	[Alt] [M] or □■ [K]	□■ [Ctl] [M] or □■ [K]	[Alt] ■□ [F3] or □■ [K]

CK30/CK31 Special Characters

Note: Some of these characters are not shown on the keypad; these are considered “hidden” but can still be entered via the following key presses:

CK30/CK31 Special Characters – 3270/5250/Native Keypads

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
` *(grave accent)*	□■ [L]	■□ [Q]	Not Applicable
<	■□ [U]	■□ [T]	■□ [L]
>	■□ [V]	■□ [V]	■□ [R]
]	■□ [C]	■□ [J]	□■ [Alpha]
[	■□ [B]	■□ [I]	□■ [Tab]
` *(single quote)*	□■ [G]	■□ [M]	Not Applicable
"	□■ [K]	■□ [R]	Not Applicable
{	□■ [N]	□■ [I]	■□ [U]
}	□■ [O]	□■ [J]	■□ [D]
:	□■ [P]	■□ [N]	Not Applicable

CK30/CK31 Special Characters – 3270/5250/Native Keypads

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
;	□■ [Q]	■□ [O]	Not Applicable
?	□■ [R]	■□ [X]	□■ [5]
~	□■ [T]	■□ [Z]	Not Applicable
,	■□ [D]	■□ [U]	□■ Right [Enter]
!	[Shift] [1]	[Shift] [1]	[Shift] [1]
@	[Shift] [2]	[Shift] [2]	[Shift] [2]
#	[Shift] [3]	[Shift] [3]	[Shift] [3]
$	[Shift] [4]	[Shift] [4]	[Shift] [4]
%	[Shift] [5]	[Shift] [5]	[Shift] [5]
^	[Shift] [6]	[Shift] [6]	[Shift] [6]
&	[Shift] [7]	[Shift] [7]	[Shift] [7]
*	[Shift] [8]	[Shift] [8]	[Shift] [8]
(	[Shift] [9]	[Shift] [9]	[Shift] [9]
)	[Shift] [0]	[Shift] [0]	[Shift] [0]

CK30/CK31 Special Characters – VT/ANSI Keypads

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
,	■□ [D]	■□ [U]	Not Applicable
` *(grave accent)*	□■ [L]	■□ [Q]	Not Applicable
` *(single quote)*	□■ [G]	■□ [M]	Not Applicable
"	□■ [K]	■□ [R]	Not Applicable
{	□■ [N]	□■ [I]	■□ [U]
}	□■ [O]	□■ [J]	■□ [D]
:	□■ [P]	■□ [N]	Not Applicable
;	□■ [Q]	■□ [O]	Not Applicable
?	□■ [R]	■□ [X]	□■ [5]
]	■□ [C]	■□ [J]	□■ [Alpha]
[	■□ [B]	■□ [I]	□■ [Tab]
<	■□ [U]	■□ [T]	■□ [L]
>	■□ [V]	■□ [V]	■□ [R]
!	[Shift] [1]	[Shift] [1]	[Shift] [1]
@	[Shift] [2]	[Shift] [2]	[Shift] [2]
#	[Shift] [3]	[Shift] [3]	[Shift] [3]
$	[Shift] [4]	[Shift] [4]	[Shift] [4]
%	[Shift] [5]	[Shift] [5]	[Shift] [5]
^	[Shift] [6]	[Shift] [6]	[Shift] [6]
&	[Shift] [7]	[Shift] [7]	[Shift] [7]
*	[Shift] [8]	[Shift] [8]	[Shift] [8]
(	[Shift] [9]	[Shift] [9]	[Shift] [9]
)	[Shift] [0]	[Shift] [0]	[Shift] [0]

CK30/CK31 Auto-Login Restart

To enter Auto-Login Restart

- Press □■ **[F4]** on the 52-key and 42-key keypads
  or
- Press □■ **[Enter]** on the 50-key keypad or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CK30/CK31 AID-Generating Function Keys

("N/A" = Not Applicable)

	52-Key Keypad		50-Key Keypad		42-Key Keypad	
To Enter	**3270**	**Native**	**3270**	**Native**	**3270**	**Native**
F1	[F1]	[F1]	[Func] [1]	[Func] [1]	[F1]	[F1]
F2	[F2]	[F2]	[Func] [2]	[Func] [2]	[F2]	[F2]
F3	[F3]	[F3]	[Func] [3]	[Func] [3]	[F3]	[F3]
F4	[F4]	[F4]	[Func] [4]	[Func] [4]	[F4]	[F4]
F5	■□ [F1]	■□ [F1]	[Func] [5]	[Func] [5]	[F5]	[F5]
F6	■□ [F2]	■□ [F2]	[Func] [6]	[Func] [6]	[F6]	[F6]
F7	■□ [F3]	■□ [F3]	[Func] [7]	[Func] [7]	[F7]	[F7]
F8	■□ [F4]	■□ [F4]	[Func] [8]	[Func] [8]	[F8]	[F8]
F9	■□ [E]	■□ [E]	[Func] [9]	[Func] [9]	[F9]	[F9]
F10	■□ [F]	■□ [F]	[Func] [0]	[Func] [0]	[F10]	[F10]
F11	■□ [G]	■□ [G]	□■ [1]	□■ [1]	[F11]	[F11]
F12	■□ [H]	■□ [H]	□■ [2]	□■ [2]	[F12]	[F12]
F13	■□ [I]	■□ [I]	□■ [3]	□■ [3]	N/A	N/A
F14	■□ [J]	■□ [J]	□■ [4]	□■ [4]	N/A	N/A
F15	■□ [K]	■□ [K]	□■ [5]	□■ [5]	N/A	N/A
F16	■□ [L]	■□ [L]	□■ [6]	□■ [6]	N/A	N/A
F17	■□ [M]	■□ [M]	□■ [7]	□■ [7]	N/A	N/A
F18	■□ [N]	■□ [N]	□■ [8]	□■ [8]	N/A	N/A
F19	■□ [O]	■□ [O]	□■ [9]	□■ [9]	N/A	N/A
F20	■□ [P]	■□ [P]	□■ [0]	□■ [0]	N/A	N/A
F21	■□ [Q]	■□ [Q]	□■ L	□■ L	N/A	N/A
F22	■□ [R]	■□ [R]	□■ [U]	□■ [U]	N/A	N/A
F23	■□ [S]	■□ [S]	□■ [D]	□■ [D]	N/A	N/A
F24	■□ [T]	■□ [T]	□■ [R]	□■ [R]	N/A	N/A
Clear	□■ [C]	□■ [C]	□■ [C]	□■ [C]	□■ [F6]	□■ [F6]
PA1	□■ [F1]	N/A	□■ [X]	N/A	□■ [F1]	N/A
PA2	□■ [F2]	N/A	□■ [Y]	N/A	□■ [F2]	N/A

CK30/CK31 AID-Generating Function Keys (continued)
("N/A" = Not Applicable)

	52-Key Keypad		50-Key Keypad		42-Key Keypad	
To Enter	**3270**	**Native**	**3270**	**Native**	**3270**	**Native**
PA3	□■ [F3]	N/A	□■ [E]	N/A	□■ [F3]	N/A

CK30/CK31 Top-Row Function Keys *(N/A = Not Applicable)*

	52-Key Keypad		50-Key Keypad		42-Key Keypad	
To Enter	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**
F1	[F1]	[F1]	[Func] [1]	[Func] [1]	[F1]	[F1]
F2	[F2]	[F2]	[Func] [2]	[Func] [2]	[F2]	[F2]
F3	[F3]	[F3]	[Func] [3]	[Func] [3]	[F3]	[F3]
F4	[F4]	[F4]	[Func] [4]	[Func] [4]	[F4]	[F4]
F5	■□ [F1]	■□ [F1]	[Func] [5]	[Func] [5]	[F5]	[F5]
F6	■□ [F2]	■□ [F2]	[Func] [6]	[Func] [6]	[F6]	[F6]
F7	■□ [F3]	■□ [F3]	[Func] [7]	[Func] [7]	[F7]	[F7]
F8	■□ [F4]	■□ [F4]	[Func] [8]	[Func] [8]	[F8]	[F8]
F9	■□ [E]	■□ [E]	[Func] [9]	[Func] [9]	[F9]	[F9]
F10	■□ [F]	■□ [F]	[Func] [0]	[Func] [0]	[F10]	[F10]
F11	■□ [G]	■□ [G]	□■ [1]	□■ [1]	[F11]	[F11]
F12	■□ [H]	■□ [H]	□■ [2]	□■ [2]	[F12]	[F12]
F13	■□ [I]	■□ [I]	□■ [3]	□■ [3]	N/A	N/A
F14	■□ [J]	■□ [J]	□■ [4]	□■ [4]	N/A	N/A
F15	■□ [K]	■□ [K]	□■ [5]	□■ [5]	N/A	N/A
F16	■□ [L]	■□ [L]	□■ [6]	□■ [6]	N/A	N/A
F17	■□ [M]	■□ [M]	□■ [7]	□■ [7]	N/A	N/A
F18	■□ [N]	■□ [N]	□■ [8]	□■ [8]	N/A	N/A
F19	■□ [O]	■□ [O]	□■ [9]	□■ [9]	N/A	N/A
F20	■□ [P]	■□ [P]	□■ [0]	□■ [0]	N/A	N/A
F21	■□ [Q]	N/A	□■ [L]	N/A	□■ [L]	N/A
F22	■□ [R]	N/A	□■ [U]	N/A	□■ [U]	N/A
F23	■□ [S]	N/A	□■ [D]	N/A	□■ [D]	N/A
F24	■□ [T]	N/A	□■ [R]	N/A	□■ [R]	N/A
Clear	□■ [C]	N/A	□■ [C]	N/A	□■ [F6]	N/A
Enter	[Enter]	[Enter]	[Enter]	[Enter]	[Enter]	[Enter]
Help *(non-error state)*	□■ [5]	N/A	□■ [Esc]	N/A	□■ [7]	N/A
Print	□■ [4]	N/A	□■ [P]	N/A	□■ [4]	N/A
Record Backspace *(Home)*	□■ [8]	N/A	□■ [H]	N/A	□■ [8]	N/A

CK30/CK31 Top-Row Function Keys (continued) *(N/A = Not Applicable)*

	52-Key Keypad		50-Key Keypad		42-Key Keypad	
To Enter	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**
Roll Down	◻◼ [D]	N/A	◻◼ [F]	N/A	◻◼ [Alt]	N/A
Roll Up	◻◼ [U]	N/A	◻◼ [B]	N/A	◻◼ [Ctl]	N/A

CK30/CK31 3278 SNA Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Clr (E-Inp)	◻◼ [B]	◻◼ [K]	◻◼ [F8]
Del	◻◼ A	[Func] ◻◼	◻◼ A
Enter	[Enter]	[Enter]	[Enter]
EOF	◻◼ [3]	◻◼ [0]	◻◼ [3]
Home	◻◼ [8]	◻◼ [H]	◻◼ [8]
Insert	◼◻, ◻◼	[Func] [Ctl]	◼◻, ◻◼
New Line (Return)	◻◼ [9]	[Func] [Enter]	◻◼ [9]
Reset	◻◼ [0]	◻◼ [R]	◻◼ [0]

CK30/CK31 5250 Tab Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Forward Tab	[Tab]	[Tab]	[Tab]
Back Tab	[Shift] [Tab]	[Shift] [Tab]	[Shift] [Tab]

CK30/CK31 5250 Field Exit Key

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Field Exit	◻◼ [F]	◻◼ [Bksp]	◻◼ [F9]

CK30/CK31 5250 Signal Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Attn	◻◼ [A]	◻◼ [A]	◻◼ [F5]
Help *(from error state)*	◻◼ [5]	◻◼ [Esc]	◻◼ [7]

CK30/CK31 5250 Special Control Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Del	◻◼ ←	[Func] ◻◼	◻◼ ←
Erase Input	◻◼ [3]	◻◼ [E]	◻◼ [F8]
Error Reset	◻◼ [0]	◻◼ [R]	◻◼ [0]
Hex	◻◼ [1]	◻◼ [Shift]	◻◼ [F10]
Home	◻◼ [8]	◻◼ [H]	◻◼ [8]
Insert	◼◻, ◻◼	[Func] [Ctl]	◼◻, ◻◼
Shift Lock	◼◻ [A]	[Func] [Shift]	[Shift] [Shift]

CK30/CK31 5250 Additional Functions

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
¢ (cent sign)	Not supported	Not supported	Not supported
↲ (New Line)	[9]	[Enter]	[9]
¬ (Not symbol)	[6]	[Shift] [6]	[Shift] [6]
Dup *(duplicate enabled fields only)*	[D]	[D]	[F7]
Field-	[.] (period)	[L]	[.] (period)
Field+	[1]	[G]	[1]
Field Mark	[M]	Not supported	[F11]

CK30/CK31 VT/ANSI Editing Keys

To Enter	52-Key Keypad	50-Key Keypad	42-Key Keypad
Find	[4]	[T]	[4]
Insert here	,	[Func] [Ctl]	,
Next screen	[3]	[Z]	[3]
Prev screen	[1]	[Y]	[1]
Remove	[0]	[R]	[0]
Select	[.] (period)	[S]	[.] (period)

CK30/CK31 VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press [7] on the 52-key or 42-key; or press **[M]** on the 50-key keypad.

Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CK31 Cursor SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up			
Window/viewport down			
Window/viewport right			
Window/viewport left			

CK31 SIP Paging Keys

To Enter	3270/Native	5250	VT/ANSI
Page up			
Page down			
Page right			
Page left			

CK31 SIP Alphanumeric Characters

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
a	[a]	A	Shift [A]
b	[b]	B	Shift [B]
c	[c]	C	Shift [C]
d	[d]	D	Shift [D]
e	[e]	E	Shift [E]
f	[f]	F	Shift [F]
g	[g]	G	Shift [G]
h	[h]	H	Shift [H]
i	[i]	I	Shift [I]
j	[j]	J	Shift [J]
k	[k]	K	Shift [K]
l	[l]	L	Shift [L]
m	[m]	M	Shift [M]
n	[n]	N	Shift [N]
o	[o]	O	Shift [O]
p	[p]	P	Shift [P]
q	[q]	Q	Shift [Q]
r	[r]	R	Shift [R]
s	[s]	S	Shift [S]
t	[t]	T	Shift [T]
u	[u]	U	Shift [U]
v	[v]	V	Shift [V]
w	[w]	W	Shift [W]
x	[x]	X	Shift [X]
y	[y]	Y	Shift [Y]
z	[z]	Z	Shift [Z]
0 - 9	[0] - [9]	Symbols	Symbol key, or Shift plus corresponding key.

CK31 SIP Function Keys

To Enter	3270/Native	5250	VT/ANSI
Back Tab	Shift B.T.	Shift B.T.	Shift B.T.
Ctrl	Not Applicable	Not Applicable	Ctrl
Delete	Shift Del	Shift Del	Shift Del
Forward Tab	Tab	Tab	Tab
Lock	Cap	Cap	Cap
Return	Enter	Enter	Enter
Shift	Shift	Shift	Shift
Space bar	Space	Space	Space

CK31 SIP Function Keys

To Enter	3270/Native	5250	VT/ANSI
Clear	[Clear]	[Clear]	Not Applicable
PA1	[PA1]	Not Applicable	Not Applicable
PA2	[PA2]	Not Applicable	Not Applicable
PA3	[PA3]	Not Applicable	Not Applicable

CK31 SIP Editing Keys

To Enter	3270/Native	5250	VT/ANSI
E-Inp	[E-Inp]	Not Applicable	Not Applicable
Clr	Not Applicable	[Clear]	Not Applicable
Del	[Del]	[Del]	Not Applicable
Enter	[Enter]	[Enter]	Not Applicable
EOF	[EOF]	Not Applicable	Not Applicable
Home	[Hm]	[Hm]	Not Applicable
Insert	[Ins]	[Ins]	Not Applicable
New Line (Return)	[NewLn]	[N.L.]	Not Applicable
Reset	[Reset]	[Reset]	Not Applicable
Find	Not Applicable	Not Applicable	[Find]
Insert here	Not Applicable	Not Applicable	[Insert]
Next screen	Not Applicable	Not Applicable	[NextSc]
Prev screen	Not Applicable	Not Applicable	[PrevSc]
Remove	Not Applicable	Not Applicable	[Remove]
Select	Not Applicable	Not Applicable	[Select]

Note: For 3270 and 5250 Computers, pressing [Esc] on the keypad also performs the Reset function.

CK31 SIP Auxiliary Keys

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
- (hyphen)	[-]	F9	⦣ F9
, (comma)	[,]	F10	⦣ F10
. (period)	[.]	F11	⦣ F11
_ (underscore)	Shift [-]	F12	⦣ F12
Enter	Enter	F13	⦣ F13
F1 or PF1	⦣ F1	F14	⦣ F14
F2 or PF2	⦣ F2	F15	⦣ F15
F3 or PF3	⦣ F3	F16	⦣ F16
F4 or PF4	⦣ F4	F17	⦣ F17
F5	⦣ F5	F18	⦣ F18
F6	⦣ F6	F19	⦣ F19
F7	⦣ F7	F20	⦣ F20
F8	⦣ F8		
For 3270, 5250, Native computers:			
F21	⦣ F21	F23	⦣ F23
F22	⦣ F22	F24	⦣ F24

CK31 SIP Auto-Login Restart

To enter Auto-Login Restart, tap the SIP ⦣ Autolog keys or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CK31 SIP 5250 Field Exit Characters

To Enter	Tap the SIP Keys
Field Exit	F→

CK31 SIP 5250 Signal Keys

To Enter	Tap the SIP Keys
Attn	⦣ Att
Help *(from error state)*	⦣ Hlp

CK31 SIP 5250 Special Control Keys

To Enter	Tap the SIP Keys
Del	Shift Del
Erase Input	λ Erl
Error Reset	Res
Hex	Hex
Home	λ Hm
Insert	λ Ins
Shift Lock	Cap

CK31 SIP 5250 Special Host Keys

To Enter	Tap the SIP Keys
Sys Req	λ SR

CK31 SIP 5250 Additional Functions

To Enter	Tap the SIP Keys
↵ (New Line)	N.L.
¬ (Not symbol)	Shift [^]
Dup *(duplicate enabled fields only)*	λ Dup
Field-	λ F-
Field+	λ F+
Field Mark	Not supported.

CK31 SIP VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press λ Mode from the SIPs.

CK31 SIP VT/ANSI Control Key

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
SOH	[A]	DC1, X-ON	® [Q]
STX	® [B]	DC2	® [R]
ETX	® [C]	DC3, X-OFF	® [S]
EOT	® [D]	DC4	® [T]
ENQ	® [E]	NAK	® [U]
ACK	® [F]	SYN	® [V]
BEL	® [G]	ETB	® [W]
BS	®[H]	CAN	® [X]
HT	® [I]	EM	® [Y]
LF	® [J]	SUB	® [Z]

CK31 SIP VT/ANSI Control Key (continued)

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
VT	Ctrl [K]	ESC	Esc
FF	Ctrl [L]	FS	Ctrl [1]
CR	Ctrl [M]	GS	Ctrl [2]
SO	Ctrl [N]	RS	Ctrl [3]
SI	Ctrl [O]	US	Ctrl [4]
DLE	Ctrl [P]	DEL	S |

CK31 SIP VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	Shift Mn on the SIP, [K] on the keypads, or double-tap the upper-right corner of the display.

CK32 Handheld Computers

This information pertains to CK32 Handheld Computers.

Characters on the CK32 Keypads

The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys.

CK32 Shift Keys

Shift Key	Function
[Orange]	Press plus a key to type a character or do an operation printed in orange on the overlay.
[Green]	Press plus a key to type a character or do an operation printed in green on the overlay.

CK32 Keypads

Your CK32 has a 56-key or a 42-key keypad.

56-Key keypad

42-Key keypad

3270, 5250, and Native Terminal Emulation keypads for the CK32

56-Key keypad

42-Key keypad

VT/ANSI Terminal Emulation keypads for the CK32

CK32 Cursor Keys

	56-Key			42-Key		
To Enter	**3270/ Native**	**5250**	**VT/ANSI**	**3270/ Native**	**5250**	**VT/ANSI**
Window/viewport up	[▲]	[▲]	◻◼ [S]	[▲]	[▲]	◻◼ [F11]
Window/viewport down	[▼]	[▼]	◻◼ [T]	[▼]	[▼]	◻◼ [F12]
Window/viewport right	[Func] [▲]	◼◻ [▲]	◻◼ [Q]	[▶]	[▶]	◻◼ [F10]
Window/viewport left	[Func] [▼]	◼◻ [▼]	◻◼ [P]	[◀]	[◀]	◻◼ [F9]

CK32 Paging Keys

	56-Key			42-Key		
To Enter	3270/Native	5250	VT/ANSI I	3270/ Native	5250	VT/ANSI
Page up	[W]	[W]	[W]	[U]	[U]	[U]
Page down	[Y]	[Y]	[Y]	[D]	[D]	[D]
Page right	[V]	[V]	[V]	[R]	[R]	[R]
Page left	[U]	[U]	[U]	[L]	[L]	[L]

CK32 VT/ANSI Host Cursor Keys

	Press the Computer Keys		
To Enter	56-Key	42-Key	Tap the SIP Keys
Host cursor up key	[▲]	[▲]	↑
Host cursor down key	[▼]	[▼]	↓
Host cursor right key	[Shift] [▼]	[▶]	→
Host cursor left key	[Shift] [▲]	[◀]	←

CK32 Scan Button

The scan button on the CK32 is the blue center button on the keypad. See the user’s manual for more information.

Note: See the *CK32 Handheld Computer User's Manual* for information how to enter alpha characters using the CK32 keypads. Keep in mind the B [Alt] and B [A] keys are toggle keys for the 56-key keypad and the [Shift] key and the B [Shift] keys are toggle keys for the 42-key keypads – these remain on until pressed again to turn them off.

CK32 Standard Keys

To Enter	56-Key Keypad	42-Key Keypad
a	[A]	[7]
b	[B]	[8]
c	[C]	[9]

CK32 Standard Keys (continued)

To Enter	56-Key Keypad	42-Key Keypad
d	[D]	[4]
e	[E]	[5]
f	[F]	[6]
g	[G]	[1]
h	[H]	[2]
i	[I]	[3]
j	[J]	[0]
k	[K]	[F1]
l	[L]	[F2]
m	[M]	[F3]
n	[N]	[F4]
o	[O]	[F5]
p	[P]	[F6]
q	[Q]	[F7]
r	[R]	[F8]
s	[S]	[F9]
t	[T]	[F10]
u	[U]	[F11]
v	[V]	[F12]
w	[W]	[Ctl]
x	[W]	[Alt]
y	[Y]	[Alpha]
z	[Y]	[Sp]
A	[Alt] [A]	[Shift] [7]
B	[Alt] [B]	[Shift] [8]
C	[Alt] [C]	[Shift] [9]
D	[Alt] [D]	[Shift] [4]
E	[Alt] [E]	[Shift] [5]
F	[Alt] [F]	[Shift] [6]
G	[Alt] [G]	[Shift] [1]
H	[Alt] [H]	[Shift] [2]
I	[Alt] [I]	[Shift] [3]
J	[Alt] [J]	[Shift] [0]
K	[Alt] [K]	[Shift] [F1]
L	[Alt] [L]	[Shift] [F2]
M	[Alt] [M]	[Shift] [F3]
N	[Alt] [N]	[Shift] [F4]
O	[Alt] [O]	[Shift] [F5]
P	[Alt] [P]	[Shift] [F6]
Q	[Alt] [Q]	[Shift] [F7]

CK32 Standard Keys (continued)

To Enter	56-Key Keypad	42-Key Keypad
R	◻■ [Alt] [R]	[Shift] ■◻ [F8]
S	◻■ [Alt] [S]	[Shift] ■◻ [F9]
T	◻■ [Alt] [T]	[Shift] ■◻ [F10]
U	◻■ [Alt] [U]	[Shift] ■◻ [F11]
V	◻■ [Alt] [V]	[Shift] ■◻ [F12]
W	◻■ [Alt] [W]	[Shift] ■◻ [Ctl]
X	◻■ [Alt] ■◻ [W]	[Shift] ■◻ [Alt]
Y	[Alt] [Y]	[Shift] ■◻ [Alpha]
Z	Ctrl [A] ■◻ [Y]	[Shift] ■◻ [Sp]
0-9	[0] - [9]	[0] - [9]
Symbols	[Shift] plus corresponding key *(3270, Native)* ■◻ plus corresponding key *(VT/ANSI)*	[Shift] plus corresponding key *(3270, Native, VT/ANSI*)

CK32 Function Keys – 3270, Native, VT/ANSI

To Enter	56-Key Keypad	42-Key Keypad
Caps Lock	■◻ [C]	■◻ [Shift]
Ctrl	◻■ [Alt]	[Ctl]
Delete	◻■ ←	◻■ ←
Forward Tab	[Tab]	[Tab]
Return	[Enter]	[Enter]
Shift	■◻ [Alt]	[Shift]
Space bar	■◻ ←	[Sp]

CK32 Additional Functions

To Enter	56-Key Keypad	42-Key Keypad
Access TE configuration menus	[Alt] [M] or ◻■ [K]	[Alt] ■◻ [F3] or ◻■ [K]
Toggle between Application mode and Numeric Keypad mode (*VT/ANSI)*	Use the TE 2000 configuration menus or set from the host.	

CK32 Special Characters

Note: Some of these characters are not shown on the keypad; these are considered "hidden" but can still be entered via the following key presses:

CK32 Special Characters – 3270/5250/Native Keypads

To Enter	56-Key Keypad	42-Key Keypad
` *(grave accent)*	C [L]	Not Applicable
<	[U]	[L]
>	[V]	[R]
]	[C]	[Alpha]
[	[B]	[Tab]
` *(single quote)*	[G]	Not Applicable
"	[K]	Not Applicable
{	[N]	[U]
}	[O]	[D]
:	[P]	Not Applicable
;	[Q]	Not Applicable
?	[R]	[5]
~	[T]	Not Applicable
,	[D]	Right [Enter]
!	[Shift] [1]	[Shift] [1]
@	[Shift] [2]	[Shift] [2]
#	[Shift] [3]	[Shift] [3]
$	[Shift] [4]	[Shift] [4]
%	[Shift] [5]	[Shift] [5]
^	[Shift] [6]	[Shift] [6]
&	[Shift] [7]	[Shift] [7]
*	[Shift] [8]	[Shift] [8]
(	[Shift] [9]	[Shift] [9]
)	[Shift] [0]	[Shift] [0]

CK32 Special Characters – VT/ANSI Keypads

To Enter	56-Key Keypad	42-Key Keypad
,	[D]	Not Applicable
` *(grave accent)*	C [L]	Not Applicable
` *(single quote)*	C [G]	Not Applicable
"	C [K]	Not Applicable
{	C [N]	[U]
}	C [O]	[D]

CK32 Special Characters – VT/ANSI Keypads (continued)

To Enter	56-Key Keypad	42-Key Keypad
:	[P]	Not Applicable
;	[Q]	Not Applicable
?	[R]	[5]
]	[C]	[Alpha]
[	[B]	[Tab]
<	[U]	[L]
>	[V]	[R]
!	[Shift] [1]	[Shift] [1]
@	[Shift] [2]	[Shift] [2]
#	[Shift] [3]	[Shift] [3]
$	[Shift] [4]	[Shift] [4]
%	[Shift] [5]	[Shift] [5]
^	[Shift] [6]	[Shift] [6]
&	[Shift] [7]	[Shift] [7]
*	[Shift] [8]	[Shift] [8]
(	[Shift] [9]	[Shift] [9]
)	[Shift] [0]	[Shift] [0]

CK32 Auto-Login Restart

To enter Auto-Login Restart, press **[F4]** on the keypad, or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CK32 AID-Generating Function Keys *(N/A = Not Applicable)*

	56-Key Keypad		42-Key Keypad	
To Enter	3270	Native	3270	Native
F1	[F1]	[F1]	[F1]	[F1]
F2	[F2]	[F2]	[F2]	[F2]
F3	[F3]	[F3]	[F3]	[F3]
F4	[F4]	[F4]	[F4]	[F4]
F5	[F1]	[F1]	[F5]	[F5]
F6	[F2]	[F2]	[F6]	[F6]
F7	[F3]	[F3]	[F7]	[F7]
F8	[F4]	[F4]	[F8]	[F8]
F9	[E]	[E]	[F9]	[F9]
F10	[F]	[F]	[F10]	[F10]
F11	[G]	[G]	[F11]	[F11]

CK32 AID-Generating Function Keys *(N/A = Not Applicable)*

	56-Key Keypad		42-Key Keypad	
To Enter	**3270**	**Native**	**3270**	**Native**
F12	■□ [H]	■□ [H]	[F12]	[F12]
F13	■□ [I]	■□ [I]	N/A	N/A
F14	■□ [J]	■□ [J]	N/A	N/A
F15	■□ [K]	■□ [K]	N/A	N/A
F16	■□ [L]	■□ [L]	N/A	N/A
F17	■□ [M]	■□ [M]	N/A	N/A
F18	■□ [N]	■□ [N]	N/A	N/A
F19	■□ [O]	■□ [O]	N/A	N/A
F20	■□ [P]	■□ [P]	N/A	N/A
F21	■□ [Q]	■□ [Q]	N/A	N/A
F22	■□ [R]	■□ [R]	N/A	N/A
F23	■□ [S]	■□ [S]	N/A	N/A
F24	■□ [T]	■□ [T]	N/A	N/A
Clear	□■ [C]	□■ [C]	□■ [F6]	□■ [F6]
PA1	□■ [F1]	N/A	□■ [F1]	N/A
PA2	□■ [F2]	N/A	□■ [F2]	N/A
PA3	□■ [F3]	N/A	□■ [F3]	N/A

CK32 Top-Row Function Keys *(N/A = Not Applicable)*

	56-Key Keypad		42-Key Keypad	
To Enter	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**
F1	[F1]	[F1]	[F1]	[F1]
F2	[F2]	[F2]	[F2]	[F2]
F3	[F3]	[F3]	[F3]	[F3]
F4	[F4]	[F4]	[F4]	[F4]
F5	■□ [F1]	■□ [F1]	[F5]	[F5]
F6	■□ [F2]	■□ [F2]	[F6]	[F6]
F7	■□ [F3]	■□ [F3]	[F7]	[F7]
F8	■□ [F4]	■□ [F4]	[F8]	[F8]
F9	■□ [E]	■□ [E]	[F9]	[F9]
F10	■□ [F]	■□ [F]	[F10]	[F10]
F11	■□ [G]	■□ [G]	[F11]	[F11]
F12	■□ [H]	■□ [H]	[F12]	[F12]
F13	■□ [I]	■□ [I]	N/A	N/A
F14	■□ [J]	■□ [J]	N/A	N/A
F15	■□ [K]	■□ [K]	N/A	N/A
F16	■□ [L]	■□ [L]	N/A	N/A
F17	■□ [M]	■□ [M]	N/A	N/A

CK32 Top-Row Function Keys (N/A = Not Applicable) **(continued)**

	56-Key Keypad		42-Key Keypad	
To Enter	**5250**	**VT/ANSI**	**5250**	**VT/ANSI**
F18	[N]	[N]	N/A	N/A
F19	[O]	[O]	N/A	N/A
F20	[P]	[P]	N/A	N/A
F21	[Q]	N/A	[L]	N/A
F22	[R]	N/A	[U]	N/A
F23	[S]	N/A	[D]	N/A
F24	[T]	N/A	[R]	N/A
Clear	[C]	N/A	[F6]	N/A
Enter	[Enter]	[Enter]	[Enter]	[Enter]
Help *(non-error state)*	[5]	N/A	[7]	N/A
Print	[4]	N/A	[4]	N/A
Record Backspace *(Home)*	[8]	N/A	[8]	N/A
Roll Down	[D]	N/A	[Alt]	N/A
Roll Up	[U]	N/A	[Ctl]	N/A

CK32 3278 SNA Keys

To Enter	56-Key Keypad	42-Key Keypad
Clr (E-Inp)	[B]	[F8]
Del	←	←
Enter	[Enter]	[Enter]
EOF	[3]	[3]
Home	[8]	[8]
Insert	,	,
New Line (Return)	[9]	[9]
Reset	[0]	[0]

CK32 5250 Tab Keys

To Enter	56-Key Keypad	56-Key Keypad
Forward Tab	[Tab]	[Tab]
Back Tab	[Shift] [Tab]	[Shift] [Tab]

CK32 5250 Field Exit Key

To Enter	56-Key Keypad	56-Key Keypad
Field Exit	[Fld Exit]	[9]

CK32 5250 Signal Keys

To Enter	56-Key Keypad	56-Key Keypad
Attn	[A]	[F5]
Help *(from error state)*	[5]	[7]

CK32 5250 Special Control Keys

To Enter	56-Key Keypad	42-Key Keypad
Del	A	←
Erase Input	[3]	[F8]
Error Reset	[0]	[0]
Hex	[1]	[F10]
Home	[8]	[8]
Insert	,	,
Shift Lock	[A]	[Shift] [Shift]

CK32 5250 Additional Functions

To Enter	56-Key Keypad	42-Key Keypad
¢ (cent sign)	Not supported	Not supported
↵ (New Line)	[9]	[9]
¬ (Not symbol)	Not supported	Not supported
Dup *(duplicate enabled fields only)*	[D]	[F7]
Field-	[.] (period)	[.] (period)
Field+	[1]	[1]
Field Mark	[M]	[F11]

CK32 VT/ANSI Editing Keys

To Enter	56-Key Keypad	42-Key Keypad
Find	[4]	[4]
Insert here	, C	,
Next screen	[3]	[3]
Prev screen	[1]	[1]
Remove	[0]	[0]
Select	[.] (period)	[.] (period)

CK32 VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press ▭ [7] on the keypad.

Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CK32 Cursor SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	↑	↑	↑
Window/viewport down	↓	↓	↓
Window/viewport right	→	→	→
Window/viewport left	←	←	←

CK32 SIP Paging Keys

To Enter	3270/Native	5250	VT/ANSI
Page up	↑	↑	↑
Page down	↓	↓	↓
Page right	→	→	→
Page left	←	←	←

CK32 SIP Alphanumeric Characters

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
a	[a]	A	Shift [A]
b	[b]	B	Shift [B]
c	[c]	C	Shift [C]
d	[d]	D	Shift [D]
e	[e]	E	Shift [E]
f	[f]	F	Shift [F]
g	[g]	G	Shift [G]
h	[h]	H	Shift [H]
i	[i]	I	Shift [I]
j	[j]	J	Shift [J]
k	[k]	K	Shift [K]
l	[l]	L	Shift [L]
m	[m]	M	Shift [M]
n	[n]	N	Shift [N]

CK32 SIP Alphanumeric Characters (continued)

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
o	[o]	O	Shift [O]
p	[p]	P	Shift [P]
q	[q]	Q	Shift [Q]
r	[r]	R	Shift [R]
s	[s]	S	Shift [S]
t	[t]	T	Shift [T]
u	[u]	U	Shift [U]
v	[v]	V	Shift [V]
w	[w]	W	Shift [W]
x	[x]	X	Shift [X]
y	[y]	Y	Shift [Y]
z	[z]	Z	Shift [Z]
0 - 9	[0] - [9]	Symbols	Symbol key, or Shift plus corresponding key.

CK32 SIP Function Keys

To Enter	3270/Native	5250	VT/ANSI
Back Tab	Shift B.T.	Shift B.T.	Shift B.T.
Ctrl	Not Applicable	Not Applicable	Ctrl
Delete	Shift Del	Shift Del	Shift Del
Forward Tab	Tab	Tab	Tab
Lock	Cap	Cap	Cap
Return	Enter	Enter	Enter
Shift	Shift	Shift	Shift
Space bar	Space	Space	Space
Clear	Clear	Clear	Not Applicable
PA1	PA1	Not Applicable	Not Applicable
PA2	PA2	Not Applicable	Not Applicable
PA3	PA3	Not Applicable	Not Applicable

CK32 SIP Editing Keys

To Enter	3270/Native	5250	VT/ANSI
E-Inp	E-Inp	Not Applicable	Not Applicable
Clr	Not Applicable	Clear	Not Applicable
Del	Del	Del	Not Applicable
Enter	Enter	Enter	Not Applicable
EOF	EOF	Not Applicable	Not Applicable
Home	Hm	Hm	Not Applicable
Insert	Ins	Ins	Not Applicable

CK32 SIP Editing Keys (continued)

To Enter	3270/Native	5250	VT/ANSI
New Line (Return)	[NewLn]	[⌊] [N.L.]	Not Applicable
Reset	[Reset]	[Reset]	Not Applicable
Find	Not Applicable	Not Applicable	[⌊] [Find]
Insert here	Not Applicable	Not Applicable	[⌊] [Insert]
Next screen	Not Applicable	Not Applicable	[⌊] [NextSc]
Prev screen	Not Applicable	Not Applicable	[⌊] [PrevSc]
Remove	Not Applicable	Not Applicable	[⌊] [Remove]
Select	Not Applicable	Not Applicable	[⌊] [Select]

Note: *For 3270 and 5250 Computers*, pressing **[Esc]** on the keypad also performs the Reset function.

CK32 SIP Auxiliary Keys

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
- (hyphen)	[-]	F9	[⌊] [F9]
, (comma)	[,]	F10	[⌊] [F10]
. (period)	[.]	F11	[⌊] [F11]
_ (underscore)	[Shift] [F18 F+ F20]	F12	[⌊] [F12]
Enter	[Enter]	F13	[⌊] [F13]
F1 or PF1	[⌊] [F1]	F14	[⌊] [F14]
F2 or PF2	[⌊] [F2]	F15	[⌊] [F15]
F3 or PF3	[⌊] [F3]	F16	[⌊] [F16]
F4 or PF4	[⌊] [F4]	F17	[⌊] [F17]
F5	[⌊] [F5]	F18	[⌊] [F18]
F6	[⌊] [F6]	F19	[⌊] [F19]
F7	[⌊] [F7]	F20	[⌊] [F20]
F8	[⌊] [F8]		
For 3270, 5250, Native computers:			
F21	[⌊] [F21]	F23	[⌊] [F23]
F22	[⌊] [F22]	F24	[⌊] [F24]

CK32 SIP Auto-Login Restart

To enter Auto-Login Restart, tap the SIP [⊾] [Autolog] keys or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CK32 SIP 5250 Field Exit Characters

To Enter	Tap the SIP Keys
Field Exit	[F→]

CK32 SIP 5250 Signal Keys

To Enter	Tap the SIP Keys
Help *(from error state)*	[⊾] [Hlp]

CK32 SIP 5250 Special Control Keys

To Enter	Tap the SIP Keys
Del	[Shift] [Del]
Erase Input	[⊾] [Erl]
Error Reset	[Res]
Hex	[Hex]
Home	[⊾] [Hm]
Insert	[⊾] [Ins]
Shift Lock	[Cap]

CK32 SIP 5250 Special Host Keys

To Enter	Tap the SIP Keys
Sys Req	[⊾] [SR]

CK32 SIP 5250 Additional Functions

To Enter	Tap the SIP Keys
↵ (New Line)	[N.L.]
Dup *(duplicate enabled fields only)*	[⊾] n
Field-	[⊾] [F-]
Field+	[⊾] [F+]
Field Mark	Not supported.

CK32 SIP VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press [Mode] from the SIPs.

CK32 SIP VT/ANSI Control Keys

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
SOH	Ctrl [A]	DC1, X-ON	Ctrl [Q]
STX	Ctrl [B]	DC2	Ctrl [R]
ETX	Ctrl [C]	DC3, X-OFF	Ctrl [S]
EOT	Ctrl [D]	DC4	Ctrl [T]
ENQ	Ctrl [E]	NAK	Ctrl [U]
ACK	Ctrl [F]	SYN	Ctrl [V]
BEL	Ctrl [G]	ETB	Ctrl [W]
BS	Ctrl [H]	CAN	Ctrl [X]
HT	Ctrl [I]	EM	Ctrl [Y]
LF	Ctrl [J]	SUB	Ctrl [Z]
VT	Ctrl [K]	ESC	Esc
FF	Ctrl [L]	FS	Ctrl [1]
CR	Ctrl [M]	GS	Ctrl [2]
SO	Ctrl [N]	RS	Ctrl [3]
SI	Ctrl [O]	US	Ctrl [4]
DLE	Ctrl [P]	DEL	Shift Del

CK32 SIP VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	Shift Mn on the SIP, [K] on the keypads, or double-tap the upper-right corner of the display.

CK61 Mobile Computers

This information pertains to CK61 Mobile Computers.

Characters on the CK61 Keypads

The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys.

CK61 Shift Keys

Shift Key	Function
[Orange]	Press plus a key to type a character or do an operation printed in orange on the overlay.
[Green]	Press plus a key to type a character or do an operation printed in green on the overlay.

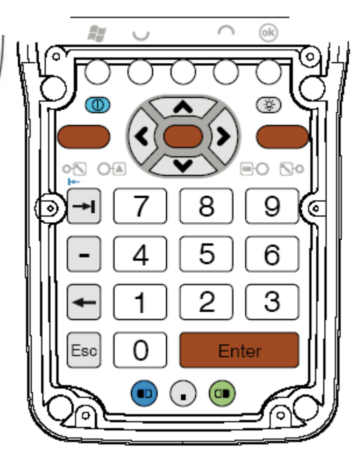
CK61 Keypads

Your CK61 Computer has a 58-key or a 32-key keypad.

58-Key keypad

58-Key keypad
VT/ANSI

32-Key keypad



Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CK61 Cursor Keys

	3270/5250/Native		VT/ANSI	
To Enter	**58-Key Keypad**	**Tap the SIP Keys**	**58-Key Keypad**	**Tap the SIP Keys**
Window/viewport up	[^]		[P]	
Window/viewport down	[v]		[Q]]	
Window/viewport right	[>]		[O]	
Window/viewport left	[<]		[N]	

CK61 Paging Keys

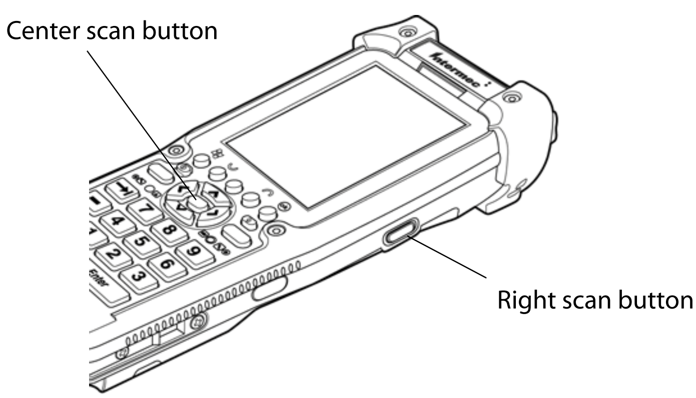
	3270/5250/Native		VT/ANSI	
To Enter	**58-Key Keypad**	**Tap the SIP Keys**	**58-Key Keypad**	**Tap the SIP Keys**
Page up	[^]		[^]	
Page down	[v]		[v]	
Page right	[>]		[>]	
Page left	[<]		[<]	

CK61 VT/ANSI Host Cursor Keys

To Enter	58-Key Keypad	Tap the SIP Keys
Host cursor up	[^]	
Host cursor down	[v]	
Host cursor right	[>]	
Host cursor left	[<]	

CK61 Scan Buttons

TE 2000 remaps the scan buttons on the CK61 to the blue button on the right side of the CK61 and the blue button in the center of the keypads.



CK61 Standard Keys

Note: See the *CK60 Mobile Computer User's Manual* for information on how to enter alpha characters using the CK61 keypads. Press [Space] to toggle to all CAPS – keep in mind these remain on until you press them again to turn them off.

CK61 Standard Keys

To Enter	58-Key Keypad	Tap the SIP Keys	To Enter	58-Key Keypad	Tap the SIP Keys
a	[A]	[a]	A	[Shift] [A]	Shift [A]
b	[B]	[b]	B	[Shift] [B]	Shift [B]
c	[C]	[c]	C	[Shift] [C]	Shift [C]
d	[D]	[d]	D	[Shift] [D]	Shift [D]
e	[E]	[e]	E	[Shift] [E]	Shift [E]
f	[F]	[f]	F	[Shift] [F]	Shift [F]
g	[G]	[g]	G	[Shift] [G]	Shift [G]
h	[H]	[h]	H	[Shift] [H]	Shift [H]
i	[I]	[i]	I	[Shift] [I]	Shift [I]
j	[J]	[j]	J	[Shift] [J]	Shift [J]
k	[K]	[k]	K	[Shift] [K]	Shift [K]
l	[L]	[l]	L	[Shift] [L]	Shift [L]
m	[M]	[m]	M	[Shift] [M]	Shift [M]
n	[N]	[n]	N	[Shift] [N]	Shift [N]
o	[O]	[o]	O	[Shift] [O]	Shift [O]
p	[P]	[p]	P	[Shift] [P]	Shift [P]
q	[Q]	[q]	Q	[Shift] [Q]	Shift [Q]
r	[R]	[r]	R	[Shift] [R]	Shift [R]
s	[S]	[s]	S	[Shift] [S]	Shift [S]
t	[T]	[t]	T	[Shift] [T]	Shift [T]
u	[U]	[u]	U	[Shift] [U]	Shift [U]
v	[V]	[v]	V	[Shift] [V]	Shift [V]
w	[W]	[w]	W	[Shift] [W]	Shift [W]
x	[X]	[x]	X	[Shift] [X]	Shift [X]
y	[Y]	[y]	Y	[Shift] [Y]	Shift [Y]
z	[Z]	[z]	Z	[Shift] [Z]	Shift [Z]
0 - 9	0 - 9	[0] - [9]	Symbols	Symbol key, or Shift plus corresponding key.	

CK61 Function Keys

To Enter	58-Key Keypad	Tap the SIP Keys 3270/Native	5250	VT/ANSI
Back Tab	[Shift] →|	Shift B.T.	Shift B.T.	Shift B.T.
Ctrl	[Ctrl]	Not Applicable	Not Applicable	Ctrl
Forward Tab	→|	Tab	Tab	Tab
Caps Lock	[Space]	Cap	Cap	Cap
Return	[M]	Enter	Enter	Enter
Shift	[Shift]	Shift	Shift	Shift
Space bar	[Space]	Space	Space	Space
Clear	[J]	Clear	Clear	Not Applicable
PA1	[N]	PA1	Not Applicable	Not Applicable
PA2	[O]	PA2	Not Applicable	Not Applicable
PA3	[P]	PA3	Not Applicable	Not Applicable

CK61 Editing Keys

To Enter	58-Key Keypad	Tap the SIP Keys 3270	5250	Native	VT/ANSI
E-Inp	[M]	E-Inp	Erl	E-Inp	N/A
Clr	[B]	N/A	[Clr]	N/A	N/A
Del	[CTRL]	Del	Del	Del	Del
Enter	[ENTER]	Enter	Enter	Enter	Enter
EOF	[X]	EOF	N/A	EOF	N/A
Home	[8]	Hm	Hm	Hm	N/A
Insert	[7]	Ins	Ins	Ins	N/A
New Line (Return)	[Q]	NewLn	N.L.	NewLn	N/A
Reset	[V]	Reset	Reset	Reset	N/A
Find	[E]	N/A	N/A	N/A	Find
Insert here	[7]	N/A	N/A	N/A	Insert
Next screen	[W]	N/A	N/A	N/A	NextSc
Prev screen	[V]	N/A	N/A	N/A	PrevSc
Remove	[L]	N/A	N/A	N/A	Remove
Select	[A]	N/A	N/A	N/A	Select

Note: *For 3270 and 5250 computers*, pressing **[Esc]** on the keypad also performs the Reset function.

CK61 Auxiliary Keys

To Enter	58-Key Keypad	Tap the SIP Keys	To Enter	58-Key Keypad	Tap the SIP Keys
- (hyphen)	[1]	[-]	F9	[A]	F9
, (comma)	[V]	[,]	F10	[B]	F10
. (period)	[←]	[.]	F11	[C]	F11
_ (underscore)	[4]	Shift [-]	F12	[D]	F12
F1	[F1]	F1	F13	[E]	F13
F2	[F2]	F2	F14	[F]	F14
F3	[F3]	F3	F15	[G]	F15
F4	[F4]	F4	F16	[H]	F16
F5	[F1]	F5	F17	[I]	F17
F6	[F2]	F6	F18	[J]	F18
F7	[F3]	F7	F19	[K]	F19
F8	[F4]	F8	F20	[L]	F20
For 3270, 5250, Native computers:					
F21	[M]	F21	F23	[O]	F23
F22	[N]	F22	F24	[P]	F24

CK61 Auto-Login Restart

To enter Auto-Login Restart, tap the SIP Autolog keys, press [T] on the 58-key keypad, or scan the following bar code. Note that Code 39 Full 7ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CK61 3270, 5250 Field Exit Characters

To Enter	58-Key Keypad	Tap the SIP Keys
Field Exit	Right blue button	F→

CK61 5250 Signal Keys

To Enter	58-Key Keypad	Tap the SIP Keys
Attn	[G]	Att
Help *(from error state)*	[E]	Hlp

CK61 5250 Special Host Keys

To Enter	58-Key Keypad	Tap the SIP Keys
Sys Req	[W]	SR

CK61 5250 Additional Functions

To Enter	58-Key Keypad	Tap the SIP Keys
¬ (Not symbol)	[Shift] [6]	Shift [^]
Dup *(duplicate enabled fields only)*	[Y]	Dup
Field-	[L]	F-
Field+	[K]	F+
Field Mark	Not supported	Not supported
Hex	[I]	Ctrl

CK61 VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press Mode from the SIPs.

CK61 VT/ANSI Control Keys

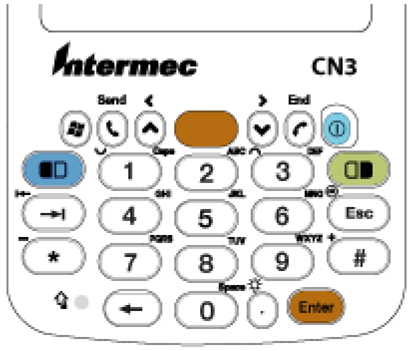
To Enter	58-Key Keypad	Tap the SIP Keys	To Enter	58-Key Keypad	Tap the SIP Keys
SOH	[CTRL] [A]	Ctrl [A]	DC1, X-ON	[CTRL] [Q]	Ctrl [Q]
STX	[CTRL] [B]	Ctrl [B]	DC2	[CTRL] [R]	Ctrl [R]
ETX	[CTRL] [C]	Ctrl [C]	DC3, X-OFF	[CTRL] [S]	Ctrl [S]
EOT	[CTRL] [D]	Ctrl [D]	DC4	[CTRL] [T]	Ctrl [T]
ENQ	[CTRL] [E]	Ctrl [E]	NAK	[CTRL] [U]	Ctrl [U]
ACK	[CTRL] [F]	Ctrl [F]	SYN	[CTRL] [V]	Ctrl [V]
BEL	[CTRL] [G]	Ctrl [G]	ETB	[CTRL] [W]	Ctrl [W]
BS	[CTRL] [H]	Ctrl [H]	CAN	[CTRL] [X]	Ctrl [X]
HT	[CTRL] [I]	Ctrl [I]	EM	[CTRL] [Y]	Ctrl [Y]
LF	[CTRL] [J]	Ctrl [J]	SUB	[CTRL] [Z]	Ctrl [Z]
VT	[CTRL] [K]	Ctrl [K]	ESC	Reset	Reset
FF	[CTRL] [L]	Ctrl [L]	FS	[CTRL] [1]	Ctrl [1]
CR	[CTRL] [M]	Ctrl [M]	GS	[CTRL] [2]	Ctrl [2]
SO	[CTRL] [N]	Ctrl [N]	RS	[CTRL] [3]	Ctrl [3]
SI	[CTRL] [O]	Ctrl [O]	US	[CTRL] [4]	Ctrl [4]
DLE	[CTRL] [P]	Ctrl [P]	DEL	Shift Del	Shift Del

CK61 Additional Functions

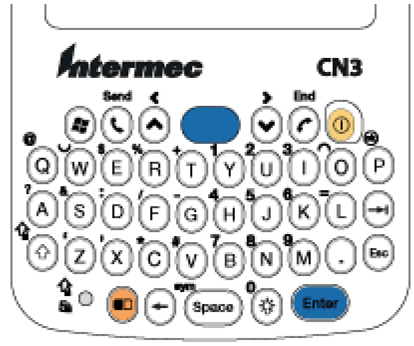
To Enter	Press the Applicable Keys
Access TE configuration menus	Shift Mn on the SIP or double-tap the upper-right corner of the display.

CN3 Mobile Computers

The following information pertains to CN3 Mobile Computers.



Numeric keypad



QWERTY keypad

The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys.

CN3 Shift Keys

Shift Key	Function
[Orange]	Press plus a key to type a character or do an operation printed in orange on the overlay.
[Green]	Press plus a key to type a character or do an operation printed in green on the overlay.

Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CN3 Cursor SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up			
Window/viewport down			
Window/viewport right			
Window/viewport left			

CN3 Paging SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Page up			
Page down			
Page right			
Page left			

CN3 VT/ANSI Host Cursor Keys

To Enter	Press the Computer Keys	Tap the SIP Keys
Host cursor up key	[▲]	↑
Host cursor down key	[◀]	↓
Host cursor right key	[▶]	→
Host cursor left key	[▼]	←

CN3 Standard Keys

Note: See the *CN3 Mobile Computer User's Manual* for information on how to enter alpha characters using the CN3 keypads. Keep in mind the [Orange] key sequence, the [Green] key sequence on the numeric keypad, and the [Shift] ⇑ key on the QWERTY keypad are toggles – these remain on until pressed again to turn them off.

CN3 Alphanumeric Characters

To Enter	Press the Numeric Keypad Keys	Press the QWERTY Keypad Keys	Tap the SIP Keys
a	[2]	[A]	[a]
b	[2] [2]	[B]	[b]
c	[2] [2] [2]	[C]	[c]
d	[3]	[D]	[d]
e	[3] [3]	[E]	[e]
f	[3] [3] [3]	[F]	[f]
g	[4]	[G]	[g]
h	[4] [4]	[H]	[h]
i	[4] [4] [4]	[I]	[i]
j	[5]	[J]	[j]
k	[5] [5]	[K]	[k]
l	[5] [5] [5]	[L]	[l]
m	[6]	[M]	[m]
n	[6] [6]	[N]	[n]
o	[6] [6] [6]	[O]	[o]
p	[7]	[P]	[p]

CN3 Alphanumeric Characters (continued)

To Enter	Press the Numeric Keypad Keys	Press the QWERTY Keypad Keys	Tap the SIP Keys
q	[7] [7]	[Q]	[q]
r	[7] [7] [7]	[R]	[r]
s	[7] [7] [7] [7]	[S]	[s]
t	[8]	[T]	[t]
u	[8] [8]	[U]	[u]
v	[8] [8] [8]	[V]	[v]
w	[9]	[W]	[w]
x	[9] [9]	[X]	[x]
y	[9] [9] [9]	[Y]	[y]
z	[9] [9] [9] [9]	[Z]	[z]
A	[1] [2]	[Shift] [A]	Shift [A]
B	[1] [2] [2]	[Shift] [B]	Shift [B]
C	[1] [2] [2] [2]	[Shift] [C]	Shift [C]
D	[1] [3]	[Shift] [D]	Shift [D]
E	[1] [3] [3]	[Shift] [E]	Shift [E]
F	[1] [3] [3] [3]	[Shift] [F]	Shift [F]
G	[1] [4]	[Shift] [G]	Shift [G]
H	[1] [4] [4]	[Shift] [H]	Shift [H]
I	[1] [4] [4] [4]	[Shift] [I]	Shift [I]
J	[1] [5]	[Shift] [J]	Shift [J]
K	[1] [5] [5]	[Shift] [K]	Shift [K]
L	[1] [5] [5] [5]	[Shift] [L]	Shift [L]
M	[1] [6]	[Shift] [M]	Shift [M]
N	[1] [6] [6]	[Shift] [N]	Shift [N]
O	[1] [6] [6] [6]	[Shift] [O]	Shift [O]
P	[1] [7]	[Shift] [P]	Shift [P]
Q	[1] [7] [7]	[Shift] [Q]	Shift [Q]
R	[1] [7] [7] [7]	[Shift] [R]	Shift [R]
S	[1] [7] [7] [7] [7]	[Shift] [S]	Shift [S]
T	[1] [8]	[Shift] [T]	Shift [T]
U	[1] [8] [8]	[Shift] [U]	Shift [U]
V	[1] [8] [8] [8]	[Shift] [V]	Shift [V]
W	[1] [9]	[Shift] [W]	Shift [W]
X	[1] [9] [9]	[Shift] [X]	Shift [X]
Y	[1] [9] [9] [9]	[Shift] [Y]	Shift [Y]
Z	[1] [9] [9] [9] [9]	[Shift] [Z]	Shift [Z]
0	[0]	B [backlight]	[0]
1	[1]	B [Y]	[1]
2	[2]	B [U]	[2]
3	[3[	B [I]	[3]

CN3 Alphanumeric Characters (continued)

To Enter	Press the Numeric Keypad Keys	Press the QWERTY Keypad Keys	Tap the SIP Keys
4	[4]	[H]	[4]
5	[5]	[J]	[5]
6	[6]	[K]	[6]
7	[7]	[B]	[7]
8	[8]	[N]	[8]
9	[9]	[M]	[9]
Symbols	Symbol key, or Shift plus corresponding key.		

CN3 Function SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Back Tab	Shift B.T.	Shift B.T.	Shift B.T.
Ctrl	Not Applicable	Not Applicable	Ctrl
Delete	Shift Del	Shift Del	Shift Del
Forward Tab	Tab	Tab	Tab
Lock	Cap	Cap	Cap
Return	Enter	Enter	Enter
Shift	Shift	Shift	Shift
Space bar	Space	Space	Space
Clear	Clear	Clear	Not Applicable
PA1	PA1	Not Applicable	Not Applicable
PA2	PA2	Not Applicable	Not Applicable
PA3	PA3	Not Applicable	Not Applicable

CN3 Editing SIP Keys

To Enter	3270/Native	5250	VT/ANSI
E-Inp	E-Inp	Not Applicable	Not Applicable
Clr	Not Applicable	Erl	Not Applicable
Del	Del	Del	Not Applicable
Enter	Enter	Enter	Enter
EOF	EOF	Not Applicable	Not Applicable
Home	Hm	Hm	Not Applicable
Insert	Ins	Ins	Not Applicable
New Line (Return)	NewLn	N.L.	Not Applicable
Reset	Reset	Reset	Not Applicable
Find	Not Applicable	Not Applicable	Find
Insert here	Not Applicable	Not Applicable	Insert
Next screen	Not Applicable	Not Applicable	NextSc

CN3 Editing SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Prev screen	Not Applicable	Not Applicable	⌞ PrevSc
Remove	Not Applicable	Not Applicable	⌞ Remove
Select	Not Applicable	Not Applicable	⌞ Select

Note: *For 3270 and 5250 computers*, pressing [Esc] on the CN3 keypad also performs the Reset function.

CN3 Auxiliary Keys and Special Characters

To Enter	Press the Numeric Keypad Keys	Press the QWERTY Keypad Keys	Tap the SIP Keys
^ (accent)	Not Applicable	Not Applicable	Shift [^]
@ (ampersand)	Not Applicable	◼◻ [Q]	Shift [@]
& (and)	Not Applicable	◼◻ [S]	Shift [&]
* (asterisk)	Not Applicable	◼◻ [C]	Shift [*]
: (colon)	Not Applicable	◼◻ [D]	Shift [:]
, (comma)	Not Applicable	◼◻ [X]	[,]
$ (dollar)	Not Applicable	◼◻ [E]	Shift [$]
" (double quote)	Not Applicable	Not Applicable	Shift [']
! (exclamation)	Not Applicable	Not Applicable	Shift [!]
` (grave accent)	Not Applicable	◼◻ [Z]	Shift [`]
> (greater than)	Not Applicable	Not Applicable	Shift [>]
- (hyphen)	◼◻ [*]	◼◻ [G]	[-]
[(left bracket)	Not Applicable	Not Applicable	[[ ]
{ (left curly bracket)	Not Applicable	Not Applicable	Shift [[ ]
((left parentheses)	Not Applicable	Not Applicable	Shift [ )]
< (less than)	Not Applicable	Not Applicable	Shift [<]
% (percent)	Not Applicable	◼◻ [R]	Shift [%]
. (period)	[.]	[.]	[.]
+ (plus)	◼◻ [#]	◼◻ [T]	Shift [+]
# (pound)	Not Applicable	◼◻ [V]	Shift [#]
? (question mark)	Not Applicable	◼◻ [A]	Shift [?]
] (right bracket)	Not Applicable	Not Applicable	[ ]]
} (right curly bracket)	Not Applicable	Not Applicable	Shift [ ]]
) (right parentheses)	Not Applicable	Not Applicable	Shift [( ]
; (semicolon)	Not Applicable	Not Applicable	[;]
` (single quote)	Not Applicable	◼◻ [Z]	[']
~ (tilde)	Not Applicable	Not Applicable	Shift [~]
_ (underscore)	Not Applicable	Not Applicable	Shift [-]
BackTab	◼◻ [→|]	◼◻ [→|]	⌞ B.T.
CapsLock	◻◼ [1]	◼◻ [Shift]	Cap
Enter	[Enter]	[Enter]	Enter

CN3 Auxiliary Keys and Special Characters (continued)

To Enter	Press the Numeric Keypad Keys	Press the QWERTY Keypad Keys	Tap the SIP Keys
frown	■□ [3]	■□ [O]	Not Applicable
ok	■□ [Esc]	■□ [P]	Not Applicable
Shift	□■	[Shift]	Shift
smiley	■□ [1]	■□ [W]	Not Applicable
Space	□■ [0]	[Space]	Space

Note: *3270, 5250, Native computers* – for the F11-F24 keys VT/ANSI computers – for the F11-F20 keys press either the uppercase sequence or the lowercase sequence (separated by the “or” conjunction), but not both. Keep in mind the [Orange] ■□ key sequence, the [Green] □■ key sequence on the numeric keypad, and the [Shift] ⇑ key on the QWERTY keypad are toggle keys – these remain on until pressed again to turn them off. Intermec Application Servers do not support the F5 (break) function.

CN3 Function Keys

To Enter	Tap the SIP Keys
F1 or PF1	F1
F2 or PF2	F2
F3 or PF3	F3
F4 or PF4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10
F11	F11
F12	F12
F13	F13
F14	F14
F15	F15
F16	F16
F17	F17
F18	F18
F19	F19
F20	F20
For 3270, 5250, Native computers:	
F21	F21
F22	F22
F23	F23
F24	F24

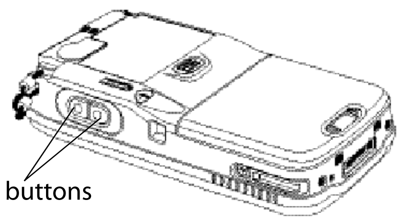
To enter Auto-Login Restart, tap the SIP [⅃] [Autolog] keys or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CN3 Scan Buttons

The scan buttons on the CN3 are the blue buttons on either side of the computer.



CN3 5250 Field Exit Characters

To Enter	Tap the SIP Keys
Field Exit	[F→]

CN3 5250 Signal Keys

To Enter	Tap the SIP Keys
Attn	[⅃] [Att]
Help *(from error state)*	[⅃] [Hlp]

CN3 5250 Special Control Keys

To Enter	Tap the SIP Keys
Del	[Shift] [Del]
Erase Input	[⅃] [Erl]

CN3 5250 Special Control Keys (continued)

To Enter	Tap the SIP Keys
Error Reset	Res
Hex	Hex
Home	Hm
Insert	Ins
Shift Lock	Cap

CN3 5250 Special Host Key

To Enter	Tap the SIP Keys
Sys Req	SR

CN3 5250 Additional Functions

To Enter	Tap the SIP Keys
↵ (New Line)	N.L.
¬ (Not symbol)	Shift [^]
Dup *(duplicate enabled fields only)*	Dup
Field-	F-
Field+	F+
Field Mark	Not supported.

CN3 VT/ANSI Transmission Mode

To toggle between Line Edit (block) mode and Character mode, press Mode from the SIPs.

CN3 VT/ANSI Control Keys

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
SOH	Ctrl [A]	DC1, X-ON	Ctrl [Q]
STX	Ctrl [B]	DC2	Ctrl [R]
ETX	Ctrl [C]	DC3, X-OFF	Ctrl [S]
EOT	Ctrl [D]	DC4	Ctrl [T]
ENQ	Ctrl [E]	NAK	Ctrl [U]
ACK	Ctrl [F]	SYN	Ctrl [V]
BEL	Ctrl [G]	ETB	Ctrl [W]
BS	Ctrl [H]	CAN	Ctrl [X]
HT	Ctrl [I]	EM	Ctrl [Y]
LF	Ctrl [J]	SUB	Ctrl [Z]
VT	Ctrl [K]	ESC	Esc
FF	® [L]	FS	Ctrl [1]

CN3 VT/ANSI Control Keys (continued)

To Enter	Tap the SIP Keys	To Enter	Tap the SIP Keys
CR	Ctrl [M]	GS	Ctrl [2]
SO	Ctrl [N]	RS	Ctrl [3]
SI	Ctrl [O]	US	Ctrl [4]
DLE	Ctrl [P]	DEL	Shift Del

CN3 VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	Shift Mn on the SIP, [Green] [K] on the keypads, or double-tap the upper-right corner of the display.

CV30 Fixed Mount Computers

This information pertains to CV30 Fixed Mount Computers.

Characters on the CV30 Keypads

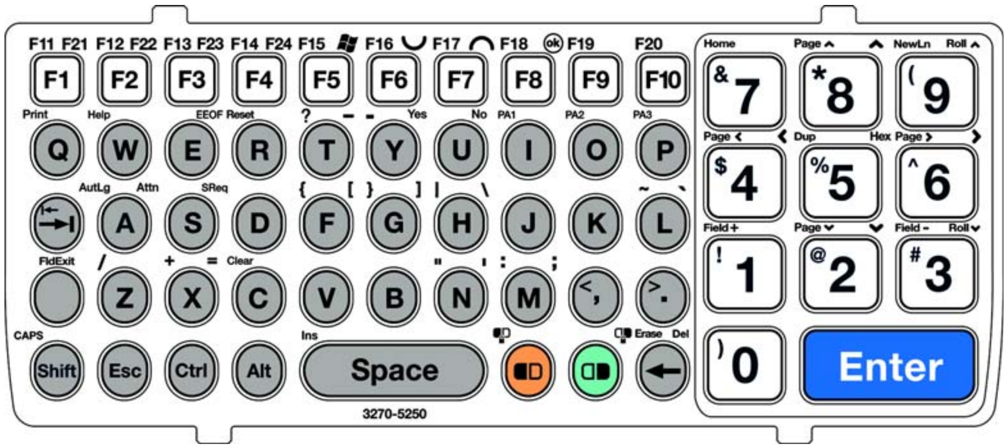
The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys.

CV30 Shift Keys

Shift Key	Function
[Orange]	Press [Orange] plus a key to type a character or do an operation printed in orange on the overlay.
[Green]	Press [Green] plus a key to type a character or do an operation printed in green on the overlay.

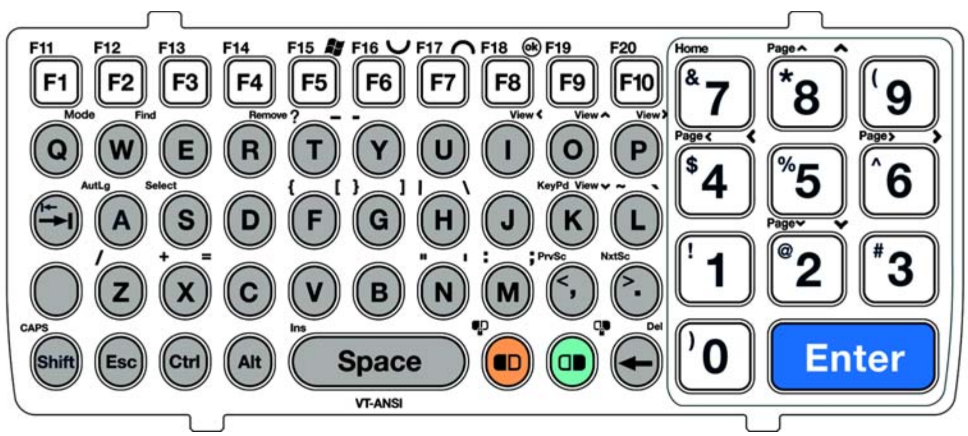
CV30 3270, 5250, and Native Compact Keypads

Use the following CV30 compact keypad (P/N VE011-8040-A0) with the TE 2000 3270, 5250, and Native applications.



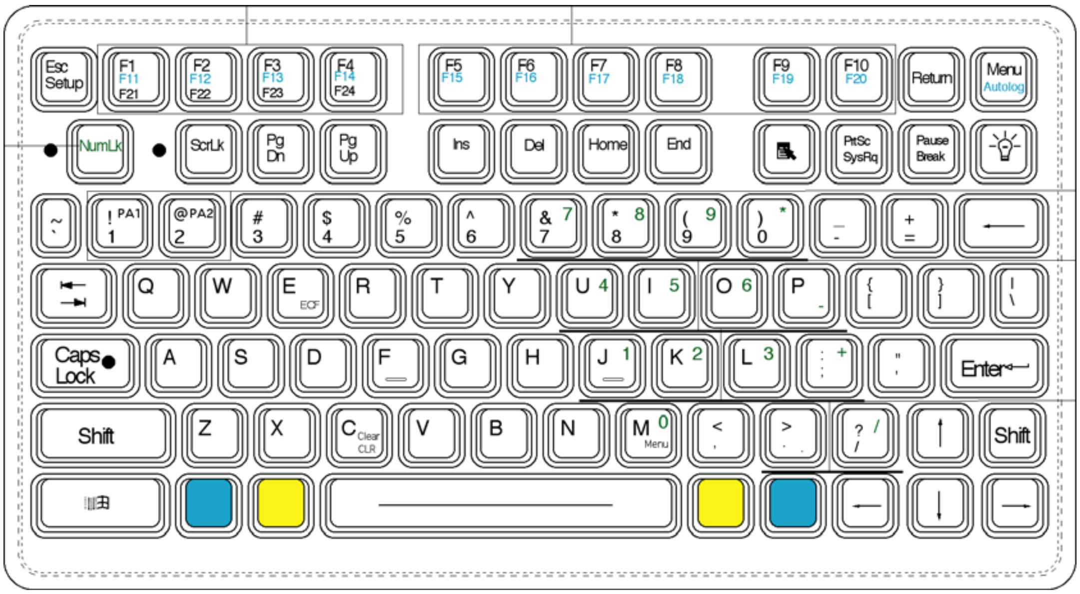
CV30 VT/ANSI Compact Keypad

Use the following CV30 compact keypad (P/N VE011-8041-A0) with the TE 2000 VT/ANSI application.



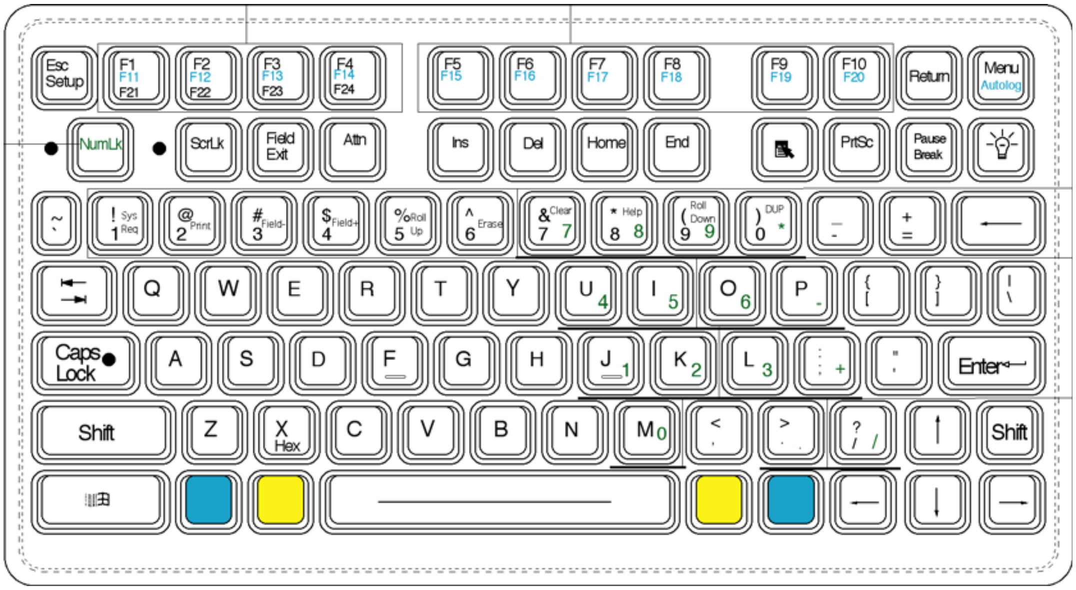
CV30 3270 and Native Rugged QWERTY Keypads

Use the following rugged QWERTY keypad (P/N 850-551-005) with the TE 2000 3270 and Native applications.



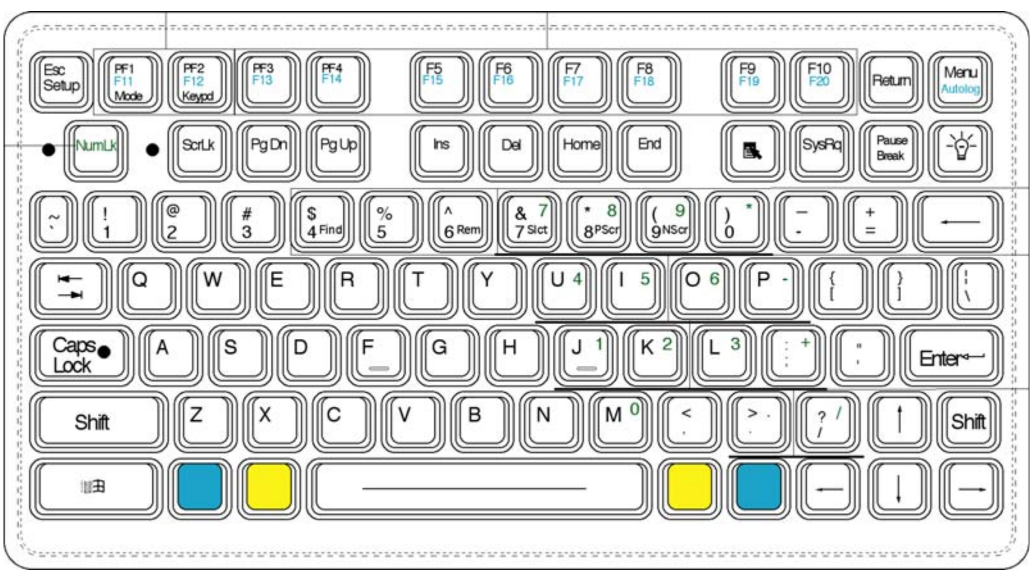
CV30 5250 Rugged QWERTY Keypad

Use the following rugged QWERTY keypad (P/N 850-551-004) with the TE 2000 5250 application.



CV30 VT/ANSI Rugged QWERTY Keypad

Use the following rugged QWERTY keypad (P/N 850-551-002) with the TE 2000 VT/ANSI application.



Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CV30 Cursor Alphanumeric Keys

To Enter	Press the Compact Keys 3270/5250/ Native	Press the Compact Keys VT/ANSI	Press the Ruggedized Keys 3270/ Native/5250	Press the Ruggedized Keys VT/ANSI
Window/viewport up	[8]	[O]	↑	[Blue] ↑
Window/viewport down	[2]	[K]	↓	[Blue] ↓
Window/viewport right	[6]	[P]	→	[Blue] →
Window/viewport left	[4]	[I]	←	[Blue] ←

CV30 Cursor SIP Keys

To Enter	3270/Native/5250	VT/ANSI
Window/viewport up		
Window/viewport down		
Window/viewport right		
Window/viewport left		

CV30 Paging Alphanumeric Keys

To Enter	Press the Compact Keys 3270/5250/ Native	Press the Compact Keys VT/ANSI	Press the Ruggedized Keys 3270/Native/ 5250	Press the Ruggedized Keys VT/ANSI
Page up	[8]	[8]	[Blue] ↑	[Gold] ↑
Page down	[2]	[2]	[Blue] ↓	[Gold] ↓
Page right	[6]	[6]	[Blue] →	[Gold] →
Page left	[4]	[4]	[Blue] ←	[Gold] ←

CV30 Paging SIP Keys

To Enter	3270/Native/5250	VT/ANSI
Page up		
Page down		
Page right		
Page left		

CV30 VT/ANSI Host Cursor Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Host cursor up key	[8]	↑	↑
Host cursor down key	[2]	↓	↓
Host cursor right key	[6]	→	→
Host cursor left key	[4]	←	←

CV30 AID-Generating Alphanumeric Keys

	Press the Compact Keys		Press the Ruggedized Keys		
To Enter	3270/5250/ Native	VT/ANSI	3270/Native	5250	VT/ANSI
Clear	[C]	Not Applicable	[Gold] [C]	[Gold] [7]	Not Applicable
F1	[F1]	[F1]	[F1]	[F1]	[F1]
F2	[F2]	[F2]	[F2]	[F2]	[F2]
F3	[F3]	[F3]	[F3]	[F3]	[F3]
F4	[F4]	[F4]	[F4]	[F4]	[F4]
F5	[F5]	[F5]	[F5]	[F5]	[F5]
F6	[F6]	[F6]	[F6]	[F6]	[F6]
F7	[F7]	[F7]	[F7]	[F7]	[F7]
F8	[F8]	[F8]	[F8]	[F8]	[F8]
F9	[F9]	[F9]	[F9]	[F9]	[F9]
F10	[F10]	[F10]	[F10]	[F10]	[F10]
F11	B [F1]	B [F1]	[Blue] [F1]	[Blue] [F1]	[Blue] [F1]
F12	B [F2]	B [F2]	[Blue] [F2]	[Blue] [F2]	[Blue] [F2]
F13	B [F3]	B [F3]	[Blue] [F3]	[Blue] [F3]	[Blue] [F3]
F14	[F4]	[F4]	[Blue] [F4]	[Blue] [F4]	[Blue] [F4]
F15	[F5]	[F5]	[Blue] [F5]	[Blue] [F5]	[Blue] [F5]
F16	[F6]	[F6]	[Blue] [F6]	[Blue] [F6]	[Blue] [F6]
F17	[F7]	[F7]	[Blue] [F7]	[Blue] [F7]	[Blue] [F7]
F18	[F8]	[F8]	[Blue] [F8]	[Blue] [F8]	[Blue] [F8]
F19	[F9]	[F9]	[Blue] [F9]	[Blue] [F9]	[Blue] [F9]
F20	[F10]	[F10]	[Blue] [F10]	[Blue] [F10]	[Blue] [F10]
F21	[F1]	Not Applicable	[Gold] [F1]	[Gold] [F1]	Not Applicable
F22	[F2]	Not Applicable	[Gold] [F2]	[Gold] [F2]	Not Applicable
F23	[F3]	Not Applicable	[Gold] [F3]	[Gold] [F3]	Not Applicable
F24	[F4]	Not Applicable	[Gold] [F4]	[Gold] [F4]	Not Applicable

CV30 AID-Generating Alphanumeric Keys (continued)

	Press the Compact Keys		Press the Ruggedized Keys		
To Enter	**3270/5250/ Native**	**VT/ANSI**	**3270/Native**	**5250**	**VT/ANSI**
PA1	■□ [I] *(3270)*	Not Applicable	[Gold] [1]	Not Applicable	Not Applicable
PA2	■□ [O *(3270)*]	Not Applicable	[Gold] [2]	Not Applicable	Not Applicable
PA3	■□ [P] *(3270)*	Not Applicable	[Gold] [3]	Not Applicable	Not Applicable
Help *(nonerror)*	■□ [W] *(5250)*	Not Applicable	Not Applicable	[Gold] [8]	Not Applicable
Print	■□ [Q] *(5250)*	Not Applicable	Not Applicable	[Gold] [2]	Not Applicable
Record Bksp *(Home)*	■□ [7] *(5250)*	■□ [7]	Not Applicable	[Home]	Not Applicable
Roll Down	□■ [3] *(5250)*	Not Applicable	Not Applicable	[Gold] [9]	Not Applicable
Roll Up	□■ [9] *(5250)*	Not Applicable	Not Applicable	[Gold] [5]	Not Applicable

CV30 AID-Generating SIP Keys

To Enter	3270	5250	Native	VT/ANSI
Clear	⦣ Clear	⦣ Clear	⦣ Clear	Not Applicable
F1	⦣ F1	⦣ F1	⦣ F1	⦣ F1
F2	⦣ F2	⦣ F2	⦣ F2	⦣ F2
F3	⦣ F3	⦣ F3	⦣ F3	⦣ F3
F4	⦣ F4	⦣ F4	⦣ F4	⦣ F4
F5	⦣ F5	⦣ F5	⦣ F5	⦣ F5
F6	⦣ F6	⦣ F6	⦣ F6	⦣ F6
F7	⦣ F7	⦣ F7	⦣ F7	⦣ F7
F8	⦣ F8	⦣ F8	⦣ F8	⦣ F8
F9	⦣ F9	⦣ F9	⦣ F9	⦣ F9
F10	⦣ F10	⦣ F10	⦣ F10	⦣ F10
F11	⦣ F11	⦣ F11	⦣ F11	⦣ F11
F12	⦣ F12	⦣ F12	⦣ F12	⦣ F12
F13	⦣ F13	⦣ F13	⦣ F13	⦣ F13
F14	⦣ F14	⦣ F14	⦣ F14	⦣ F14
F15	⦣ F15	⦣ F15	⦣ F15	⦣ F15
F16	⦣ F16	⦣ F16	⦣ F16	⦣ F16
F17	⦣ F17	⦣ F17	⦣ F17	⦣ F17

CV30 AID-Generating SIP Keys (continued)

To Enter	3270	5250	Native	VT/ANSI
F18	[⦣] [F18]	[⦣] [F18]	[⦣] [F18]	[⦣] [F18]
F19	[⦣] [F19]	[⦣] [F19]	[⦣] [F19]	[⦣] [F19]
F20	[⦣] [F20]	[⦣] [F20]	[⦣] [F20]	[⦣] [F20]
F21	[⦣] [F21]	[⦣] [F21]	[⦣] [F21]	Not Applicable
F22	[⦣] [F22]	[⦣] [F22]	[⦣] [F22]	Not Applicable
F23	[⦣] [F23]	[⦣] [F23]	[⦣] [F23]	Not Applicable
F24	[⦣] [F24]	[⦣] [F24]	[⦣] [F24]	Not Applicable
PA1	[⦣] [PA1]	Not Applicable	Not Applicable	Not Applicable
PA2	[⦣] [PA2]	Not Applicable	Not Applicable	Not Applicable
PA3	[⦣] [PA3]	Not Applicable	Not Applicable	Not Applicable
Enter/Rec Adv	[Enter]	[Enter]	[Enter]	[Enter]
Help *(nonerror)*	Not Applicable	[⦣] [Hlp]	Not Applicable	Not Applicable
Print	Not Applicable	[⦣] [Prt]	Not Applicable	Not Applicable
Record Backspace *(Home)*	Not Applicable	[⦣] [Hm]	Not Applicable	Not Applicable
Roll Down	Not Applicable	[⦣] [R↓]	Not Applicable	Not Applicable
Roll Up	Not Applicable	[⦣] [R↑]	Not Applicable	Not Applicable

CV30 Auto-Login Restart

To enter Auto-Login Restart, press B [A] on the 3270/5250/Native compact keypad, [Blue], [Menu] on the ruggedized keypad, tap the } @ keys on the SIPs, or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CV30 Tab Keys - 3270/5250

		Press the Ruggedized Keys		
Enter	**Press the Compact Keys**	**3270**	**5250**	**Tap the SIP Keys**
Back Tab	[Shift] [Tab]	[Shift] →	|←	[Shift] [B.T.]
Forward Tab	[Tab]	→	→|	[Tab]

CV30 3278 SNA Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Clr (E-Inp)	←	[Blue] [C]	E-Inp
Del	←	[Del]	Del
Enter	[Enter]	[Enter]	Enter
EEOF	[E]	[Blue] [E]	EOF
Home	[7]	[Home]	Hm
Insert	[Space]	[Ins]	Ins
New Line (Return)	[9]	[Return]	NewLn
Reset	[R]	[Esc]	Reset

CV30 5250 Field Exit Key

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Field Exit	[FldExit]	[Field Exit]	F→

CV30 5250 Signal Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Attn	[A]	[Attn]	Att
Help *(from error state)*	[W]	[Gold] [8]	Hlp

CV30 5250 Special Control Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Del	[BkSp]	[Del]	Shift Del
Erase	[BkSp]	[Gold] [6]	Erl
Reset	[R]	[Esc]	Res
Hex	[5]	[Gold] [X]	Hex
Home	[7]	[Home]	Hm
Ins	[Space]	[Insert]	Ins
Shift Lock	[Shift]	[Caps Lock]	Cap

CV30 5250 Special Host Key

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Sys Req	[S]	[Gold] [1]	SR

CV30 5250 Additional Functions

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
¢ (cent sign)	Not Applicable	Not Applicable	Not Applicable
↵ (New Line)	[9]	[Return]	N.L.
¬ (Not symbol)	[Shift] [6]	[Shift] [6]	Shift [^]
Dup *(duplicate enabled fields only)*	[5]	[Gold] [0]	Dup
Field-	[3]	[Gold] {3]	F-
Field+	[1]	[Gold] [4]	F+
Field Mark	Not Applicable	Not Applicable	Not Applicable

CV30 VT/ANSI Standard Keys

To Enter	Press the Compact Keys or Press the Ruggedized Keys	Tap the SIP Keys
0-9	[0] through [9]	[0] through [9]
Symbols	[Shift] plus corresponding key	Shift plus corresponding key

CV30 VT/ANSI Function Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Backspace	[Shift] [Tab]	|←	[← BS]
Caps Lock	[CAPS]	[Caps Lock]	Cap
Ctrl	[Ctrl]	[Blue]	Ctrl
Delete	[BkSp]	[Del]	Del
Forward Tab	[Tab]	→|	Tab
Return	[Enter]	[Enter]	Enter
Shift	[Shift]	[Shift]	Shift
Space	[Space]	[Space bar]	Space

CV30 VT/ANSI Editing Keys

To Enter	Press the Compact Keys	Press the Ruggedized Keys	Tap the SIP Keys
Find	[W]	[Gold] [4]	Find
Ins	[Space]	[Ins]	Insert
NxtSn	[.] (period)	[Gold] [9]	NextSc
PrvSc	[,] (comma)	[Gold] [8]	PrevSc
Remove	[R]	[Gold] [6]	Remove
Select	[S]	[Gold] [7]	Select

CV30 VT/ANSI Transmission Mode

To toggle between Line Edit (block) and Character modes, press **[F1]** on the compact keypads, **[Gold] [F1]** on the ruggedized keypads, or press Mode from the SIPs.

CV30 VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	[Alt] [M] on compact keypads, [Gold] [M] or [Menu] on the ruggedized keypads, Shift Mn on the SIPs, or double-tap the upper-right corner of the display.

CV60 Vehicle Mount Computers

This information pertains to CV60 Vehicle Mount Computers.

Characters on the CV60 Keypads

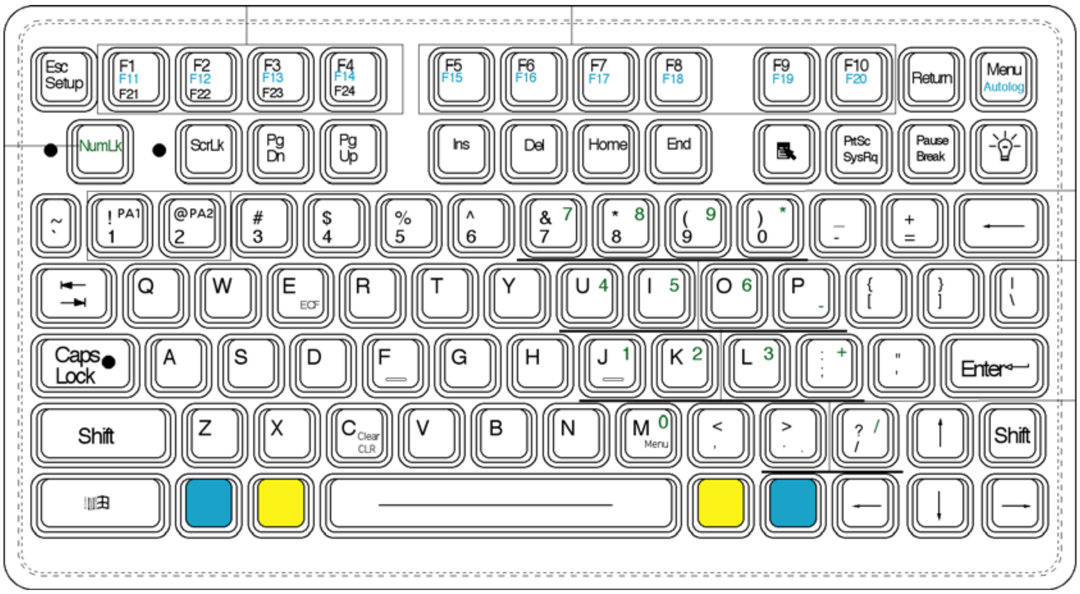
The special characters and functions printed above the keys are color-coded to correspond with the matching shift keys.

CV60 Shift Keys

Shift Key	Function
[Blue]	Press [Blue] plus a key to type a character or do an operation printed in blue on the overlay.
[Gold]	Press [Gold] plus a key to type a character or do an operation printed in gold on the overlay.

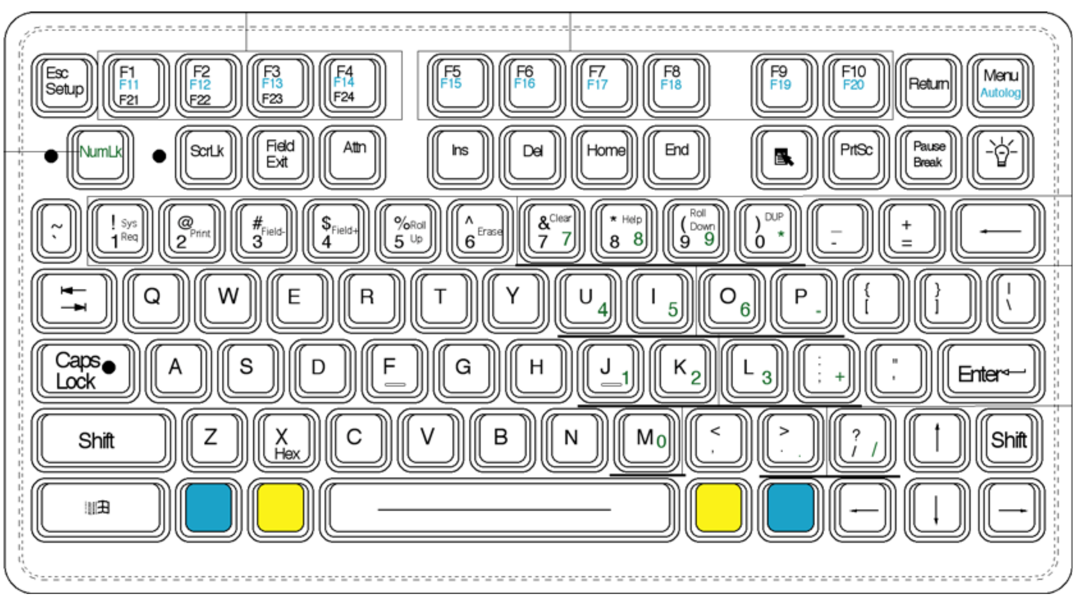
CV60 3270 and Native Physical Keypads

Use the following CV60 keypad (P/N 850-551-005) with the TE 2000 3270 and Native applications.



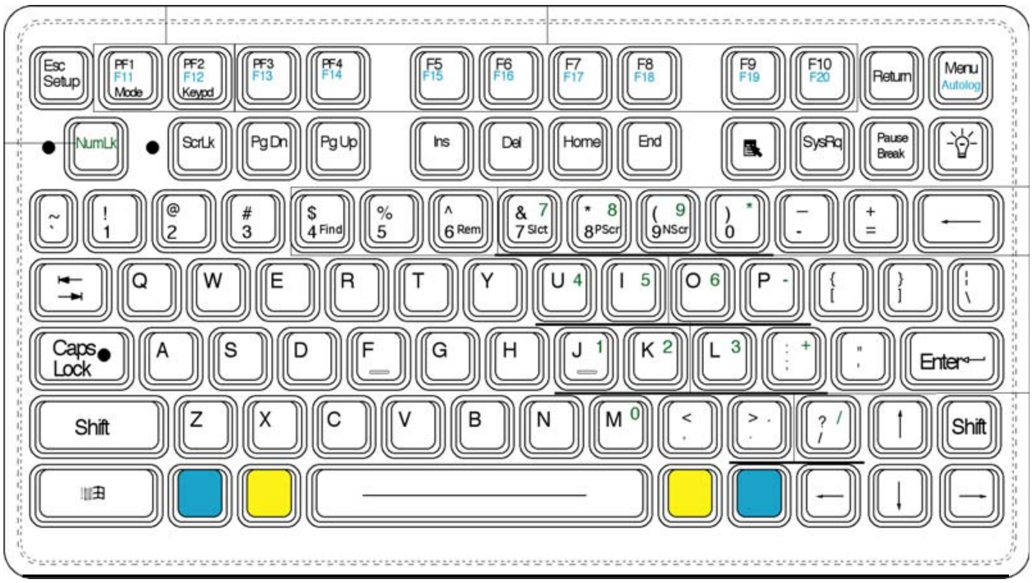
CV60 5250 Physical Keypad

Use the following CV60 keypad (P/N 850-551-004) with the TE 2000 5250 application.



CV60 VT/ANSI Physical Keypad

Use the following CV60 keypad (P/N 850-551-002) with the TE 2000 VT/ANSI application.



Note: The following key sequences are based on the SIPs unless otherwise noted. Sequences provided are based on the assumption that you are starting with the Main keypad.

CV60 Cursor Alphanumeric Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	↑	↑	[Gold] ↑
Window/viewport down	↓	↓	[Gold] ↓
Window/viewport right	→	→	[Gold] →
Window/viewport left	←	←	[Gold] ←

CV60 Cursor SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Window/viewport up	↑	↑	
Window/viewport down	↓	↓	
Window/viewport right	→	→	
Window/viewport left	←	←	

CV60 Paging Alphanumeric Keys

To Enter	3270/Native	5250	VT/ANSI
Page up	[Blue] ↑	[Blue] ↑	[Blue] ↑
Page down	[Blue] ↓	[Blue] ↓	[Blue] ↓
Page right	[Blue] →	[Blue] →	[Blue] →
Page left	[Blue] ←	[Blue] ←	[Blue] ←

CV60 Paging SIP Keys

To Enter	3270/Native	5250	VT/ANSI
Page up			
Page down			
Page right			
Page left			

CV60 VT/ANSI Host Cursor Keys

To Enter	3270/Native	5250
Host cursor up key	Hex	↑
Host cursor down key	↓	↓
Host cursor right key	→	→
Host cursor left key	F21	←

CV60 AID-Generating Keys

	Press the Keypad Keys		Tap the SIP Keys	
To Enter	3270/Native	5250	3270/Native	5250
Clear	[Gold] [C]	[Gold] [7]	Clear	Clear
F1	[F1]	[F1]	F1	F1
F2	[F2]	[F2]	F2	F2
F3	[F3]	[F3]	F3	F3
F4	[F4]	[F4]	F4	F4
F5	[F5]	[F5]	F5	F5
F6	[F6]	[F6]	F6	F6
F7	[F7]	[F7]	F7	F7
F8	[F8]	[F8]	F8	F8
F9	[F9]	[F9]	F9	F9
F10	[F10]	[F10]	F10	F10
F11	[Blue] [F1]	[Blue] [F1]	F11	F11

CV60 AID-Generating Keys

	Press the Keypad Keys		Tap the SIP Keys	
To Enter	**3270/Native**	**5250**	**3270/Native**	**5250**
F12	[Blue] [F2]	[Blue] [F2]	F12	F12
F13	[Blue] [F3]	[Blue] [F3]	F13	F13
F14	[Blue] [F4]	[Blue] [F4]	F14	F14
F15	[Blue] [F5]	[Blue] [F5]	F15	F15
F16	[Blue] [F6]	[Blue] [F6]	F16	F16
F17	[Blue] [F7]	[Blue] [F7]	F17	F17
F18	[Blue] [F8]	[Blue] [F8]	F18	F18
F19	[Blue] [F9]	[Blue] [F9]	F19	F19
F20	[Blue] [F10]	[Blue] [F10]	F20	F20
F21	[Gold] [F1]	[Gold] [F1]	F21	F21
F22	[Gold] [F2]	[Gold] [F2]	F22	F22
F23	[Gold] [F3]	[Gold] [F3]	F23	F23
F24	[Gold] [F4]	[Gold] [F4]	F24	F24
PA1	[Gold] [1]	Not Applicable	PA1	Not Applicable
PA2	[Gold] [2]	Not Applicable	PA2	Not Applicable
PA3	[Gold] [3]	Not Applicable	PA3	Not Applicable
Enter/Rec Adv	Not Applicable	[Enter]	Not Applicable	Enter
Help *(nonerror)*	Not Applicable	[Gold] [8]	Not Applicable	Hlp
Print	Not Applicable	[Gold] [2]	Not Applicable	[Prt]
Record Backspace *(Home)*	Not Applicable	[Home]	Not Applicable	Hm
Roll Down	Not Applicable	[Gold] [9]	Not Applicable	[R↓]
Roll Up	Not Applicable	[Gold] [5]	Not Applicable	[R↑]

CV60 Top-Row Function Keys - VT/ANSI

To Enter	Press the Keypad Key	Tap the SIP Keys
F1	[F1]	F1
F2	[F2]	F2
F3	[F3]	F3
F4	[F4]	F4
F5	[F5]	F5
F6	[F6]	F6
F7	[F7]	F7

CV60 Top-Row Function Keys - VT/ANSI (continued)

To Enter	Press the Keypad Key	Tap the SIP Keys
F8	[F8]	[⦣] [F8]
F9	[F9]	[⦣] [F9]
F10	[F10]	[⦣] [F10]
F11	[Blue] [F1]	[⦣] [F11]
F12	[Blue] [F2]	[⦣] [F12]
F13	[Blue] [F3]	[⦣] [F13]
F14	[Blue] [F4]	[⦣] [F14]
F15	[Blue] [F5]	[⦣] [F15]
F16	[Blue] [F6]	[⦣] [F16]
F17	[Blue] [F7]	[⦣] [F17]
F18	[Blue] [F8]	[⦣] [F18]
F19	[Blue] [F9]	[⦣] [F19]
F20	[Blue] [F10]	[⦣] [F20]

CV60 Auto-Login Restart

To enter Auto-Login Restart, press [Blue], [Menu] on the alphanumeric keypad, tap the [⦣] [Autolog] keys on the SIPs, or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

CV60 Tab Keys – 3270, 5250

To Enter	Press the Alphanumeric Keys: 3270	Press the Alphanumeric Keys: 5250	Tap the SIP Keys: 3270	Tap the SIP Keys: 5250
Back Tab	[Shift] →	|←	[Shift] [B.T.]	[Shift] [B.T.]
Forward Tab	→	→|	[Tab]	[Tab]

CV60 3278 SNA Keys

To Enter	Press the Keypad Key	Tap the SIP Keys
Clr (E-Inp)	[Blue] [C]	[⦣] [E-Inp]
Del	[Del]	[⦣] [Del]
Enter	[Enter]	[Enter]
EOF	[Blue] [E]	[⦣] [EOF]

CV60 3278 SNA Keys (continued)

To Enter	Press the Keypad Key	Tap the SIP Keys
Home	[Home]	Hm
Insert	[Ins]	Ins
New Line (Return)	[Return]	NewLn
Reset	→	Reset

CV60 5250 Field Exit Key

To Enter	Press the Keypad Key	Tap the SIP Key
Field Exit	[Field Exit]	F→

CV60 5250 Signal Keys

To Enter	Press the Keypad Key	Tap the SIP Key
Attn	[Attn]	Att
Help *(from error state)*	[Gold] [8]	Hlp

CV60 5250 Special Control Keys

To Enter	Press the Keypad Key	Tap the SIP Key
Del	[Del]	Shift Del
Erase Input	[Gold] [6]	Erl
Error Reset	→	Res
Hex	[Gold] [X]	Hex
Home	[Home]	Hm
Insert	[Insert]	Ins
Shift Lock	[Shift] [Caps Lock]	Cap

CV60 5250 Special Host Key

To Enter	Press the Keypad Key	Tap the SIP Key
Sys Req	[Gold] [1]	SR

CV60 5250 Additional Functions

To Enter	Press the Keypad Key	Tap the SIP Key
¢ (cent sign)	Not supported	Not supported
↲ (New Line)	[Return]	N.L.
¬ (Not symbol)	[Shift] [6]	Shift [^]
Dup *(duplicate enabled fields only)*	[Gold] [0]	Dup

CV60 5250 Additional Functions (continued)

To Enter	Press the Keypad Key	Tap the SIP Key
Field-	[Gold] [3]	⦣ F-
Field+	[Gold] [4]	⦣ F+
Field Mark	Not supported	Not supported

CV60 VT/ANSI Standard Keys

To Enter	Press the Keypad Key	Tap the SIP Key
0-9	[0] through [9]	[0] through [9]
Symbols	[Shift] plus corresponding key	Shift plus corresponding key

CV60 VT/ANSI Function Keys

To Enter	Press the Keypad Key	Tap the SIP Key
Backspace	|←	[←BS]
Caps Lock	[Caps Lock]	Cap
Ctrl	[Gold]	Ctrl
Delete	[Del]	Del
Forward Tab	→|	Tab
Return	[Enter]	Enter
Shift	[Shift]	Shift
Space bar	[Space bar]	Space

CV60 VT/ANSI Editing Keys

To Enter	Press the Keypad Key	Tap the SIP Key
Find	[Gold] [4]	⦣ Find
Insert here	[Ins]	⦣ Insert
Next screen	[Gold] [9]	⦣ NextSc
Prev screen	[Gold] [8]	⦣ PrevSc
Remove	[Gold] [6]	⦣ Remove
Select	[Gold] [7]	⦣ Select

CV60 VT/ANSI Transmission Mode

To toggle between Line Edit (block) and Character modes, press [Gold] [F1] on the alphanumeric keypad or press } = on the SIPs.

CV60 VT/ANSI Additional Functions

To Enter	Press the Applicable Keys
Access TE configuration menus	[Shift] [M], ▭■ [K], or [Menu] on the alphanumeric keypad, Shift Mn on the SIP, or double-tap the upper-right corner of the display.

4 Using the Terminal Emulation Menus

This chapter lists all TE 2000 parameters. If a certain parameter does not apply to your computer, the parameter does not appear in its TE 2000 configuration menus.

This chapter contains these secttions:

- Function Keys
- Configuring TE Parameters

Menu Navigation Keys

These paragraphs describe how to navigate through the TE 2000 configuration menus.

Y ("Yes") Key

Several displays provide a warning that a certain action can cause your computer to lose data stored in memory. Press the Y ("yes") key to proceed as instructed. Press another key to exit the menu without executing your original choice.

Number Keys [0] through [9]

Many menus have numbered options. Press the corresponding numeric key to make a selection.

Note: For the 700 Colors, CK31, CK32, CK61, CN3, CV30, and CV60, you may tap an option to make a selection.

Once you make a selection and you remain at that menu screen, the option is highlighted, but you can select additional options. To deselect an enabled option, press the number corresponding to that option.

Pressing a number may bring up a submenu. Use the submenu to further modify the choice you made in the parent menu. After the modifications, you may return to the parent menu to make additional selections. (This depends on the menu and function.) Also, you may press a numeric key to exit a menu or cold start the computer. These situations are covered later in this chapter.

Various menus require entering a number, but do not necessarily have simple choices such as 1, 2, 3, 4, etc. Instead, you may have to enter a number from a range of numbers, such as 0-32.

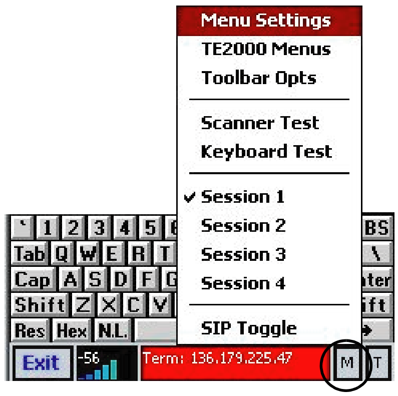
These instances are detailed in the text that applies to those menus, or in the menu displays.

Note: There are situations where pressing [6] causes the computer to exit from a submenu or to reboot.

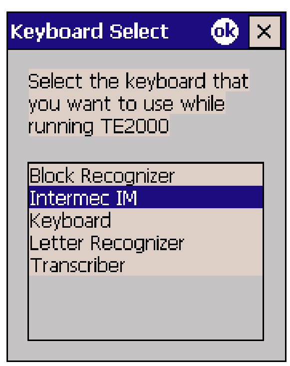
Configuration Menus

This chapter contains the TE configuration menus for the 700 Color, CK31, CK32, CK61, CN3, CV30, and CV60 unless otherwise noted.

When a user taps the **Menu Settings** button *("M" circled in this illustration)* on the toolbar, the following menu appears:



- Select **Menu Settings** to configure the contents of this menu. Enter a password, the default being *cr52401*.
- Select **TE2000 Menus** to access the TE 2000 Main Menu. Information about this is described later in this chapter.
- Select **Toolbar Opts** to configure the options that appear in the toolbar on the computer.
- Select **SIP Toggle** to enable or disable the SIP keyboard.
- Select **SIP Settings** to select which Soft Input Panel (SIP) you want to use with the TE 2000 application:



- Select **Session Menu** to change the current session.
- Select any of **Session 1** through **Session 4** to go to that session.

Configuring TE Parameters

This chapter contains the TE configuration parameters for the 700 Colors, CK30, CK31, CK32, CK61, CN3, CV30, and CV60 unless otherwise noted.

Note: Screens are shown with all of the available options.

- If an option is **not** assigned to a specific computer, then that option is available for all of the computers addressed in this publication.
- If an option is specific towards a computer, this option does not appear on the display, or is blank, for all other computers.

Key Sequence to Open Main Menu

To configure TE 2000 parameters on your respective computer, at the initialization screens or anywhere in a TE 2000 session, access the Main Menu pressing the appropriate key sequence:

Computer	Key Sequence
700 Color, CN3	Tap twice on the upper-right corner of the display; or via the Soft Input Panel (SIP) keypad, tap [Shift] [Mn]
CK30	[Alt] [M] *(52-key keypad);* ■□ [Ctl] [M] *(50-key keypad);* [Alt] □■ [F3] *(42-key keypad)*
CK31	Tap twice on the upper-right corner of the display. Via the external keypad, press: [Alt] [M] *(52-key keypad);* ■□ [Ctl] [M] *(50-key keypad);* [Alt] □■ [F3] *(42-key keypad)* Via the SIP keypad, tap [Shift] [Mn]
CK32	Tap twice in the upper-right corner of the display. [Alt] [M] or ■□ [K] (56-key keypad) [ALT] □■ [F] or ■□ [K] (42-key keypad)
CK61	Tap twice on the upper-right corner of the display, or via the SIP keypad, tap [Shift] [Mn]
CV30, CV60	Tap twice on the upper-right corner of the display; via the external keypad, press [Alt] [M], or via the SIP keypad, tap [Shift] [Mn]

Note: You can also scan this bar code label to access the Main Menu. Code 39 Full ASCII must be enabled in the firmware (default is disabled).

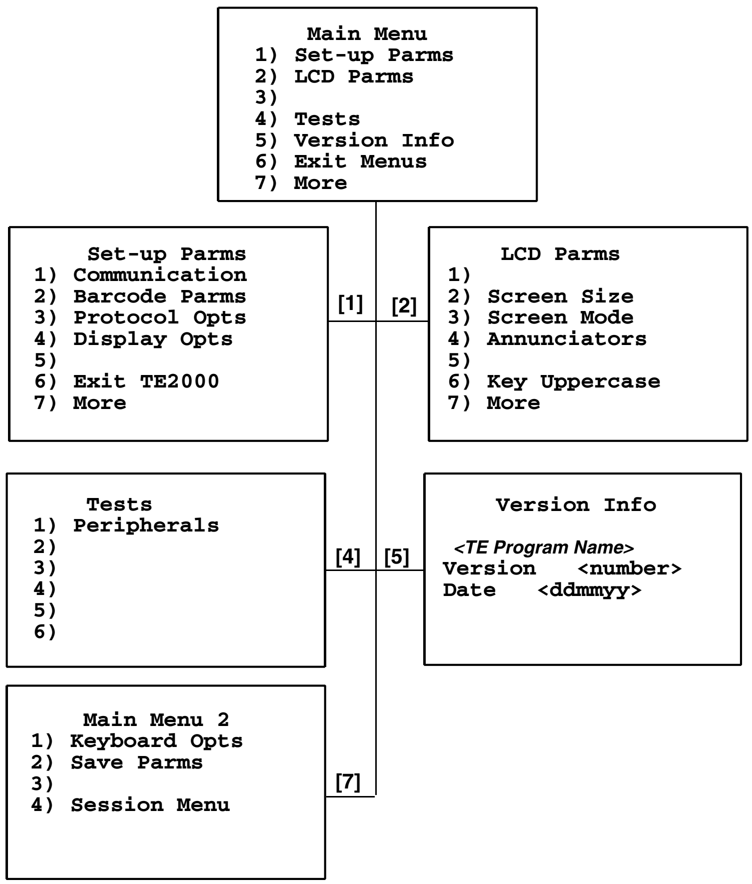
TE configuration menus

%TECFG

Opening the Main Menu

The Main Menu is the first screen displayed when you open the computer menus. All other menus are accessed from the Main Menu. At the initialization screens or anywhere in a TE 2000 session, press the appropriate keys to access the Main Menu (*defined within each computer's specific information*). After the Main Menu appears, enter a number (1-7) to make a selection.

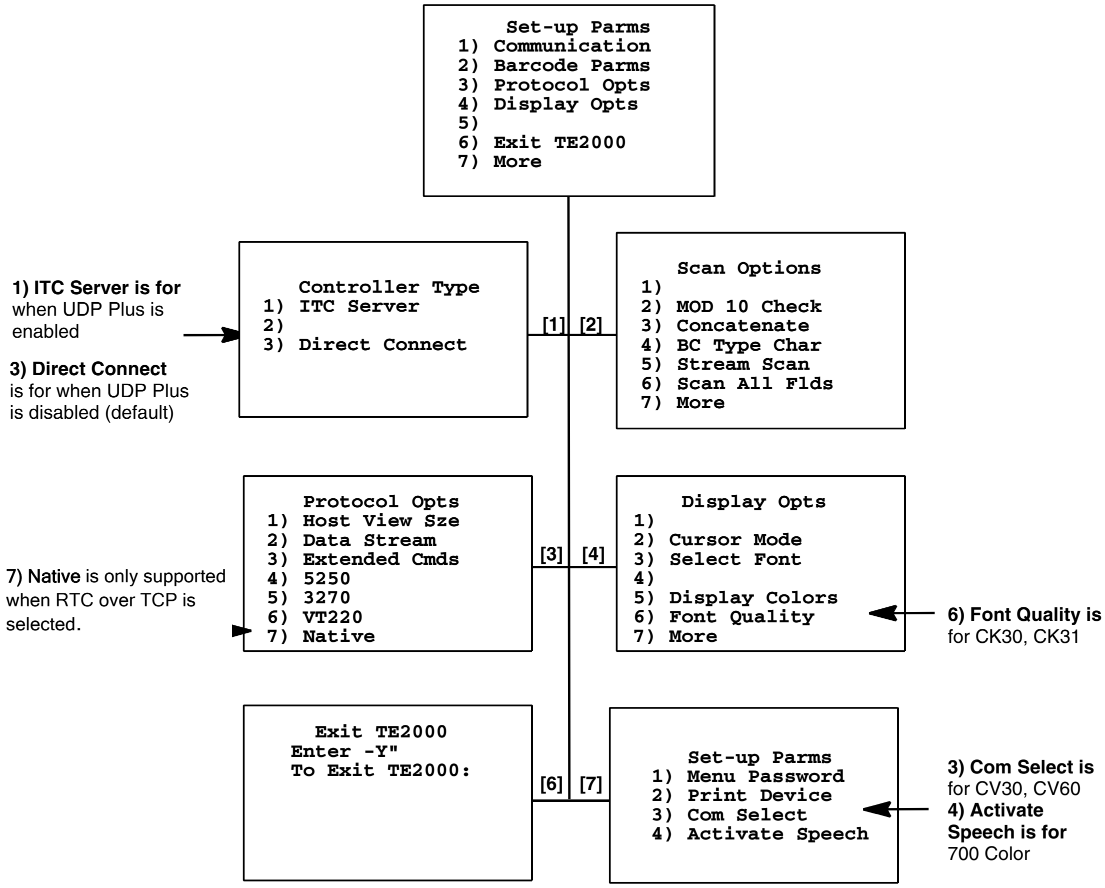
To return to the Main Menu, press [Enter] several times, then select **6) Exit Menus**, to return to the operating system.



1) Set-up Parms

This is password-protected to prevent unauthorized users from changing parameters. However, you can customize the parameter in the TE_Settings.ini setup file. For more information, see "Customizing Your Configuration" on page 153.

Changed parameters apply to the current session. If more than one session is available, use the **7) More** > **2) Save Parms** > **4) Session Menu** to verify or change the current session before going to the next session.To open the **1) Set-up Parms** menu, press [1], [Enter], then type "cr52401" at the prompt to access the menus on the next page.


Set-up Parms
1) Communication
2) Barcode Parms
3) Protocol Opts
4) Display Opts
5)
6) Exit TE2000
7) More
1) ITC Server is for when UDP Plus is enabled
3) Direct Connect is for when UDP Plus is disabled (default)
Controller Type
1) ITC Server
2)
3) Direct Connect
[1]
[2]
Scan Options
1)
2) MOD 10 Check
3) Concatenate
4) BC Type Char
5) Stream Scan
6) Scan All Flds
7) More
7) Native is only supported when RTC over TCP is selected.
Protocol Opts
1) Host View Sze
2) Data Stream
3) Extended Cmds
4) 5250
5) 3270
6) VT220
7) Native
[3]
[4]
Display Opts
1)
2) Cursor Mode
3) Select Font
4)
5) Display Colors
6) Font Quality
7) More
6) Font Quality is for CK30, CK31
Exit TE2000
Enter -Y"
To Exit TE2000:
[6]
[7]
Set-up Parms
1) Menu Password
2) Print Device
3) Com Select
4) Activate Speech
3) Com Select is for CV30, CV60
4) Activate Speech is for 700 Color

1) Setup Parameters

The following setup parameter information pertains to all computers unless otherwise noted.

1) Communication

The communication options for the computers are broken down as follows:

1) ITC Server (UDP Plus Enabled)

ITC Server options are as follows. To set the options, select **1) ITC Server**, **2) Host/Srvr**, then the **Server Setup** option (Server A, B, or C).

```
   Controller Type
1) ITC Server
2)
3)

        [1]

    ITC Server
1)
2) Host/Srvr

        [2]

   Server Setup
1) Server A
2) Server B
3) Server C

 Server <A, B, or C>
Server IP
 <IP address>

 Host Setup<A, B, or C>
1) Host Name
2) Upline Prot
3) Port Number
4) Emulation
```

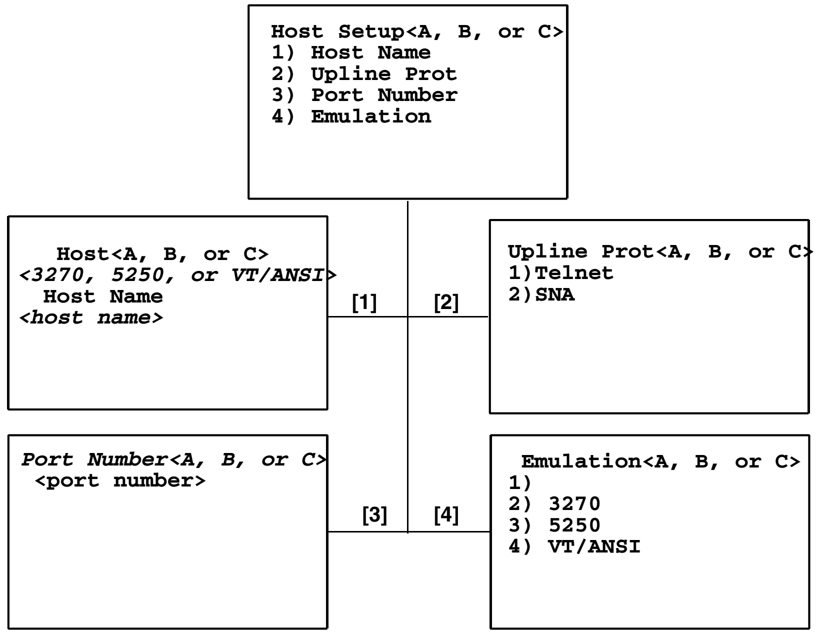
Server IP <A, B, or C>

The Server <A> IP address is read from the TE_Settings.ini file (if one is saved) or the TE 2000 application on the computer. When an IP address is entered, it is written to the computer's copy of the TE 2000 application for the next boot.

If using the Server <B> or Server <C> option as a fallback Intermec® Application Server, enter the server's IP address or DNS name. The boot sequence continues to try Server <A>, <B>, then <C> in this order by filling the firmware controller slot with the IP values entered for each server.

Host Setup <A, B, or C>

Menu options are as follows:



1) Host Name

The host name can be 64 or fewer characters in length. Enter spaces or punctuation where needed. *Note that the TE 2000 application performs an auto-enter when the 64th character is entered*. It is case-sensitive and must match a host name in the list of available hosts defined on the Intermec Application Server, or remain blank. The name can also be the IP address of the host to which you want to connect. Enter the IP address as four decimal numbers separated by periods.

- If a computer was linked with a host name on the Intermec Application Server, or a default host was configured on this Intermec Application Server, you do not need to enter a host name on this screen.
- The computer displays a list of available hosts if these conditions are met. The user can then select the host from the list.
- The Intermec Application Server contains multiple hosts.
- The computer is not linked with a host name on the Intermec Application Server.
- The host name is blank.

2) Upline Prot

When enabled, upline protocol options behave as follows:

- **Telnet** *(Default)*
  Forces the Intermec Application Server to create a Telnet connection to the host.
- **SNA**
  Forces the Intermec Application Server to create an SNA connection to the host.

3) Port Number

This option overrides the port number set for the upline if you have entered a host name and selected Telnet as your upline protocol. *Default port number is 23.* The maximum port number is 65535.

4) Emulation

Use this option to tell the computer each host computer type.
Default is VT/ANSI.

Note: You can also set the type through the **2) Data Stream** option on the **3) Protocol Opts** menu.

3) Direct Connect (UDP Plus Disabled - *default*)

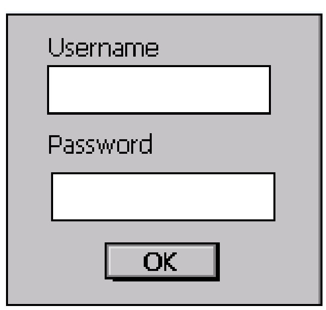
To set these options, select **3) Direct Connect**, 2) **Host Setup**, then **1) Host A**. Use the **Emulation (A)** option to tell the computer the type of each host computer. *Default is VT/ANSI.*

Note: You can also set the emulation type through the **2) Data Stream** option on the **3) Protocol Opts** menu.

Note: The SSH feature is only available on the CK32, CK61, CN3, and CV60 running TE 2000 v8.25 or later.

To nominate an SSH session (VT/ANSI)

1. At the **2) Host Setup** screen, select **1) Host A**.
2. At the *Port Number* screen, change “23” to “22” for the port number.
3. Select **4) VT/ANSI** for its emulation.
4. Enter subsequent settings as appropriate.
5. After you exit menus, the following login screen prompts you for your **Username** and **Password** on the remote site. Enter the appropriate information, then click **OK** to continue.



To terminate an SSH session

- If you are at a command prompt, type “exit.”
- Enter the menus and repeat steps 1-4 under “To nominate an SSH session” above. If you do not want an SSH connection, do not nominate port 23.

SSH Server Application Programming Hints

In general, an SSH server application is simpler than a TCP/IP server because there is no TCP/IP or server code.

The SSH server provides all the services to connect clients to the host and control the TCP/IP ports. All a SSH server application does is position the cursor and read standard input (stdin).

Note: The following procedure applies if you already built a TCP/IP server application.

To convert a TCP/IP server application to a SSH server

1. Remove all multiprocessing/spawn code and all TCP/IP send() calls that echo the data.
2. Replace your TCP/IP recv() calls with the following:

```
while((char ret = getc()) strncat(string, &ret, 1);
```

3. Replace all send calls that paint the screen with equivalent `putc()` or `puts()` calls.

Lock down SSH Applications

Preventing a user from getting to the command prompt is required in many instances. A telnet server can accomplish this by bringing itself up during startup (as part of the standard services). SSH, as a server, is already started and logging into it as a shell brings up the command prompt.

On a Linux system, you can prevent this by creating a login script that executes your host application and then posts “exit” as its last command. Specifically, in the user account .pro file, append the following lines:

```
trap 2
./ pgmName
exit
```

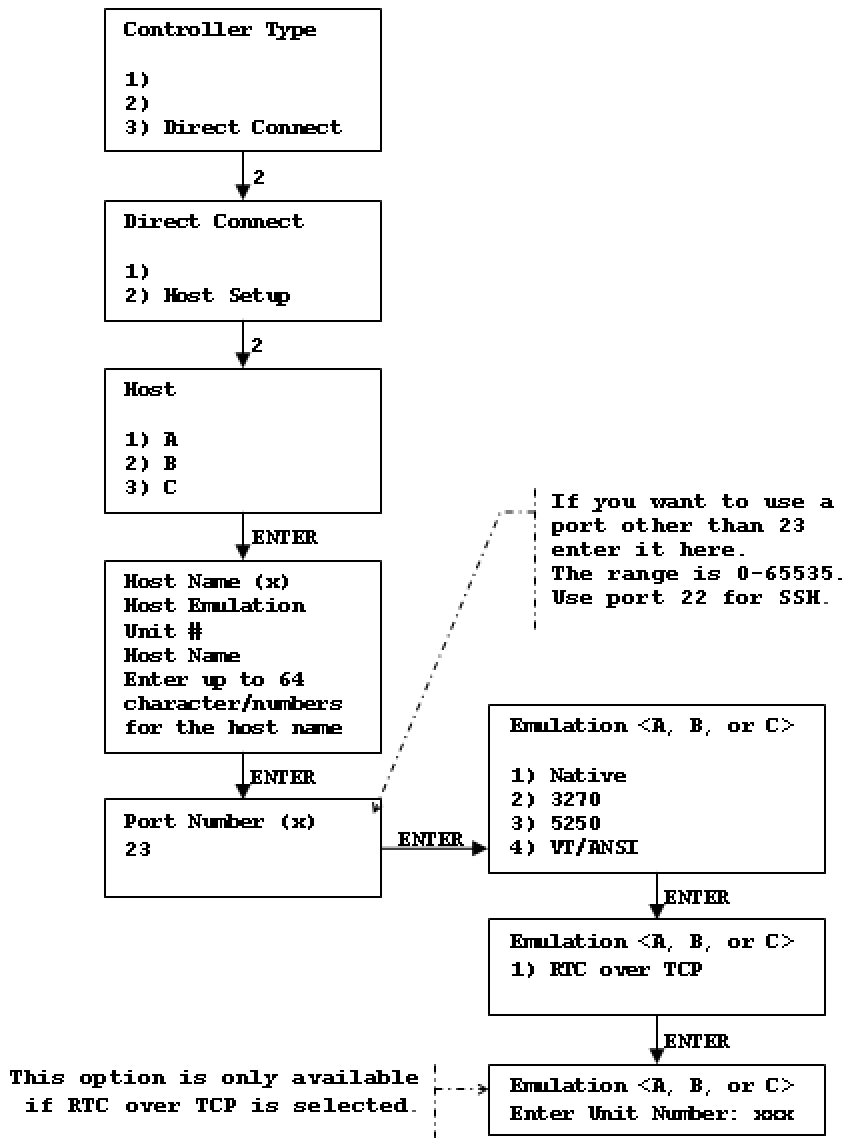
RTC over TCP

This is a protocol enhancement that along with a Telnet Gateway Appliance (TGAP) ensures connection persistence in a roaming mobile environment. **1) RTC over TCP** only works when the specified host is a TGAP. The actual end-host connection is made by the TGAP based on configuration settings.

RTC over TCP and the TGAP support IBM 3270, 5250, VT/ANSI, and Norand Native data streams. For more information on how session persistence is achieved and system requirements for using a TGAP, refer to the TGAP section of the manuals for the MobileLAN Access WA2X family of access points.

Unit # should be left set to the default of 127 for non-Native data stream emulations and Native emulations where the host application does not expect a powerup message.

Unit # is only used when the data stream emulation is Native. Set the **Unit #** to a unique value for each of up to 127 computers when the Native data stream host application expects a powerup message containing the Unit Number.



When **RTC over TCP** is enabled, the **Unit #** is not 127, and **Native** data stream is selected, then the Telnet Gateway Appliance sends a powerup record to the host immediately upon connecting. This emulates behavior available in the Intermec DCS300 and G4000 gateway products. The powerup message derived from the **Unit #** is intended to provide a migration path for users that have existing, Native data stream host applications that expect the powerup message.

Note: Native only applies when RTC over TCP is selected.

2) Barcode Parms

Use the **Scan Options** menu to assign scan options, enable bar code symbologies to use, and set options for each enabled bar code symbology, such as the bar code lengths. Press [Enter] to advance to other **Scan Options** menus.

```
  Scan Options
1)
2) MOD 10 Check
3) Concatenate
4) BC Type Char
5) Stream Scan
6) Scan All Flds
7) More
```

Note: The computer can only decode the enabled symbologies when **ParmsBySession** is enabled.

2) Mod 10 Check

When you select this option, a check digit is added at the end of the bar code after a good read for transmission to the host. The host can then validate the transmitted data using the check. This feature is not needed with modem transmission protocols.

The bar code number is divided by 10, until the number (or modulus) is less than 10. If the modulus subtracted from 10 (remainder) is equal to zero, then the bar code number is valid.

3) Concatenate

Each bar code read is added to the end of the previous bar code read until the computer meets a condition forcing transmission to the host.

When this option is OFF, each bar code read is placed at the beginning of the current input field. After a bar code is placed in a field, any subsequent read replaces the first read.

4) BC Type Char

This adds a character associated with the bar code type at the beginning of the scanned bar code.

5) Stream Scan

If the scanned bar code is too big for the input field, the overflow information appears in the next field and continues until the entire bar code is entered. When this option is OFF, and the scanned bar code is too big for the input field, the overflow information is dropped.

6) Scan All Flds

When this option is selected, the scanner is enabled when the cursor is in an input field. When this option is OFF, the host computer must enable the scanner for each input field that requires scanned data.

7) More

This brings you additional scanning options that adjust scanning timeout and characters sent before and after the scanned data.

- **2) Scan PreChar** and **3) Scan PostChar**

  Enter a hexadecimal value from 00-FF for the character to be sent preceding the scanned data or after the scanned data. The value of 20h means that pre- or post-characters are not sent.

- **4) ParmsBySession**

  When enabled, the symbology parameters under **1) Set-up Parms** > **2) Barcode Parms** configure Windows CE computer scanning devices on a per-session basis.

  When disabled (default), the **2) Barcode Parms** option is ignored. The TE 2000 application does not configure symbology parameters for Windows CE computer scanning devices. The settings selected via the Windows CE System Data Collection applet are used for all TE 2000 sessions. You can configure the symbology parameters under **1) Set-up Parms > 2) Barcode Parms**, however, these are not used.

- **5) Use Wedge Mode**

  This causes the TE 2000 application to use the system Scanners and Symbologies settings. All scanned data will come into the TE 2000 application from the system Virtual Wedge per the Virtual Wedge settings as if the scanned bar code characters were typed into the keyboard.

  When Use Wedge Mode is enabled, none of the TE 2000 menu options for scanning will function, including the following: Auto Tab Scan, Aut Entr Scn, BRT Auto Enter, ANY Auto Enter, Scan PreChar, Scan PostChar, symbology enable, disable, mix-max or fixed lengths, drop leading, or trailing characters.

  Scanner data will not be inhibited when in a non-scannable field in TE 2000 3270 or TE 2000 5250.

  When Use Wedge Mode is enabled, all of the system Scanners and Symbologies capabilities will be available. This means that all of the symbology options and grid options for special edits can be used.

  Use Wedge Mode can be enabled in TE_Settings.ini by session but the Session 1 setting will be propagated across all sessions. If you enable Use Wedge Mode in TE 2000 menus and then save the parameters, you will see that the TE_Settings.ini contains the same Use Wedge Mode value for all sessions.

- **6) No ScanAhead**

  Enable this to disable only SF51 Cordless Scanners from scanning ahead until the host sends a scanner enable command. Default is disabled.

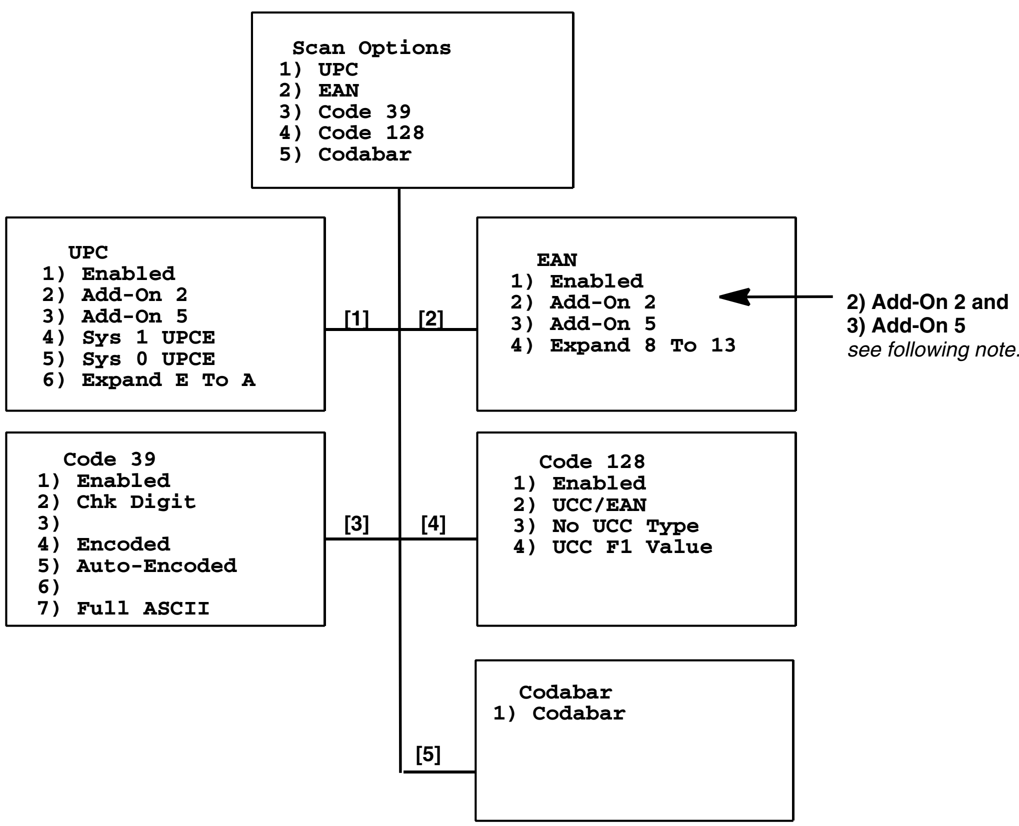
Scan Options to Enable Bar Codes

```
 Scan Options
1)
2) Scan PreChar
3) Scan PostChar
4) ParmsBySession
5) Use Wedge Mode
6) No ScanAhead
```

There are menus of options to enable (or disable) various bar code symbologies. To enable a bar code, press the numeric key corresponding to the number of the bar code symbology (shown on menu), then press [Enter].

After enabling a particular bar code, options may have to be selected for the bar code. After selecting the desired options, set the length requirements (such as minimum and maximum length, fixed lengths, leading and trailing character options, etc.).

Once the length requirements are set, the computer returns to the same Scan Options menu. Choose another bar code symbology from that menu or press [Enter] to advance to the next Scan Options menu.



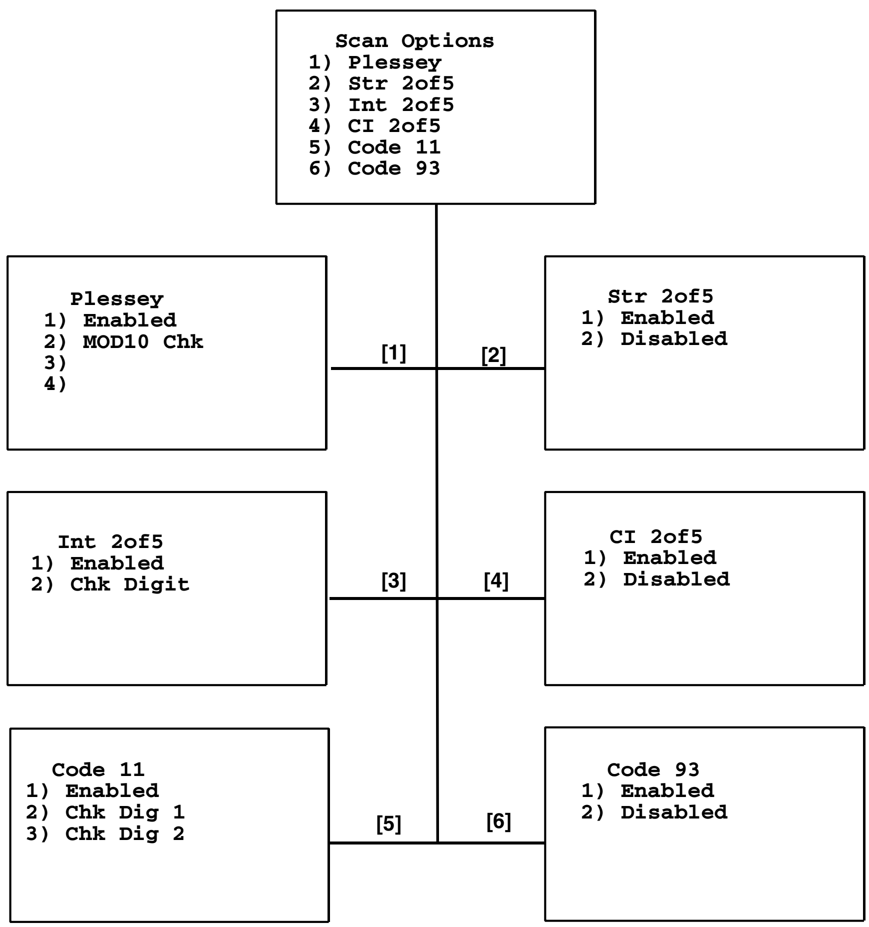
Note: This implementation of the TE 2000 application processes EAN 8 and UPC E bar codes in the same manner. The type character for EAN 8 with Add-On 2 or Add-On 5 decode as UPC E with Add-On 2 or Add-On 5, respectively. The bar code type character also indicates UPC E with Add-On 2 or Add-On 5, respectively. Consequently, the drop leading and trailing UPC options are then used when an EAN 8 with Add-On 2 or Add-On 5 is decoded.

Scan Options

This menu offers additional bar code selections. After enabling a bar code symbology, various options, including length can be set.

When all options are set, the display returns to the Scan Options menu. This menu and the bar code symbologies to be enabled are shown below.

When all needed bar code symbologies are enabled, press [Enter] to return to the Scan Options menu. Press [Enter] again to back out and return to the **1) Set-up Parms** menu.



Lengths Options

The Lengths Options menus determine the maximum and minimum length for a specific bar code symbology. Setting the length of enabled bar codes helps the computer determine if a scanned bar code is valid and improves response time. The length options must be set for each enabled bar code. *Enter this information in the order listed.*

1. Key in the maximum length (0-99), then press [Enter]. Set this as high as you intend to set the minimum length.
2. Key in the minimum length (0-99), then press [Enter].
3. Fixed-length entries override the maximum and minimum length entries (minimum and maximum are used for chosen codes). If fixed lengths are not needed for the enabled bar code, enter zero (0).

   Enter the fixed length (0-99), then press [Enter]. Up to four prompts to set another fixed length will appear. Enter zero (0) for the unused fixed lengths.

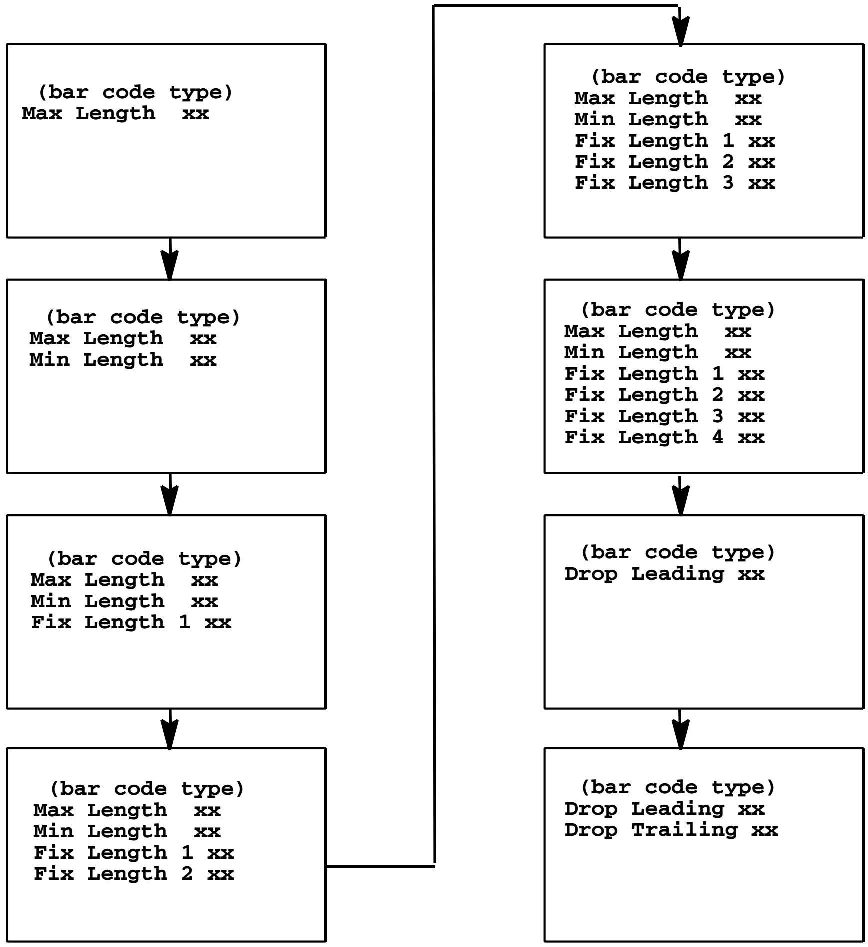
   Note: I 2 of 5 only supports fixed lengths 1 and 2. Entries in 3 and 4 are ignored.

4. Enter the number of characters (0-15) to drop from the front (leading) edge of the scanned bar code, then press [Enter].
5. Enter the number of characters (0-15) to drop from the rear (trailing) edge of the scanned bar code, then press [Enter].

Exceptions

- Codabar, Str 2of5, Int 2of5, and CI 2of5 bar code symbologies set the length, if fixed-length 1 is non-zero, to three fixed-length entries equal to the first three fixed-lengths (fourth fixed-length is ignored). Otherwise, the length is set to greater than or equal to the minimum length.
- Code 39, Code 128, Code 93, and Plessey set the length to any length greater than or equal to the minimum length.
- UPC/EAN and Code 11 lengths are not used – ANY is allowed.
- Codabar options will not decode less than two data characters (four characters including Start and Stop).
- CI 2of5 options override Str 2of5 options, if both CI 2of5 and Str 2of5 are selected.
- EAN Add-on settings override UPC add-on settings, if both UPC and EAN options are selected.
- If “Sys 1 UPCE” or “Sys 0 UPCE” is selected, then both UPCE number system 0 and 1 are enabled.
- Code 11 does not support “No check digits.” Select either 1 check digit (Chk Dig 1) or 2 check digits (Chk Dig 2) according to your bar codes. If neither check digit option is selected, then 2 check digits will be used as a default.

After all of the length options are set for the enabled bar code, the display returns to either the Scan Options menu or the Scan Options 2 menu (depending on from which menu the bar code was enabled).



3) Protocol Opts

Use the **3) Protocol Opts** menu to designate the type of terminal emulation the host supports. You can also use it to designate additional commands and command sets.

```
  Protocol Opts
1) Host View Sze
2) Data Stream
3) Extended Cmds
4) 5250
5) 3270
6) VT/ANSI
7) Native
```

7) Native is blank if UDP Plus

1) Host View Sze

The **1) Host View Sze** option tells the computer the format of display information sent from the host. The display viewing size equals 1920 (the number of rows times the number of columns). The exception is with 5250 Terminal Emulation which has an additional row for error messages and therefore has 2000 characters.

Note: The host view size is only settable for Host A. If the host view size in Host A is not supported by the host defined in B or C, TE 2000 can lock up. This happens on all terminal platforms.

The default setting for the respective terminal emulation is as follows. Remember that the total characters must equal 1920 (number of rows by the number of columns) unless using 5250 Terminal Emulation with error messages.

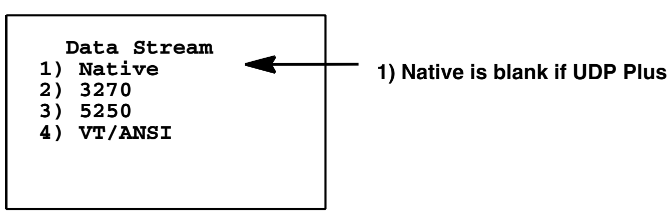
- Native: 120 columns by 16 rows (equals 1920)
- VT/ANSI: 80 columns by 24 rows (equals 1920)
- 5250: 80 columns by 24 rows (25 with error messages)
- 3270: depends on what the original setting was before changing to the 3270 Terminal Emulation.

Enter any of the following numbers to change the width, then press [Enter] to return to the **3) Protocol Opts** menu. “xx” is the width.

- 3270, Native: between 1-80
- 5250: 80 or 132
- VT/ANSI: 80

2) Data Stream

Enter the number corresponding to the type of host computer to specify the data stream used. Press [Enter] to return to the **3) Protocol Opts** menu.



Note: You can also set the emulation type through the **1) ITC Server** or **3) Direct Connect** options.

3) Direct Connect options.

3) Extended Cmds (Commands)

Use **3) Extended Cmds** to enable or disable extended commands. By default, extended commands are disabled. For more information about extended commands, see "Extended Commands" on page 343.

When this option is *enabled*, the host computer can change or use RS-232 communications (such as printers), bar code options, display screen and font size, and error tone features of the unit. Press [1] to enable or press [2] to disable the extended command list. Press [Enter] to return to the **3) Protocol Opts** menu.

```
   Extended Cmds
1) Enabled
2) Disabled
```

4) 5250 (Options)

The **4) 5250** menu enables the following features *if the 5250 communication protocol is used.* Press the appropriate number to enable an option, then press [Enter] to return to the **3) Protocol Opts** menu.

```
  5250
1) Beep On Error
2) Auto Tab Scan
3)
4) Device Name
5) Allow NAWS
6) Skip FldExit
7) More
```

[7]

```
  5250
1)
2) Destructive BS
3) Lock Error Msg
4) Use Color
5) 3477-Fx Mode
6) Scan AutoEnter
7) More
```

1) Beep On Error

Causes the buzzer to emit when an error condition occurs; does not lock up the keypad.

2) Auto Tab Scan

Causes the cursor to automatically tab forward to the next input field after a good scan.

4) Device Name

Enters a physical name for a device. This is 1-30 characters with allowable values of A-Z, a-z, 0-9, wildcards, pound symbols (#), dollar signs ($), ampersands (@), and underscores (_).

Once a device name is generated, only the first ten characters of the name are transmitted back to the host. *5250E RFC 1572 Support.*

Note: The first character of **4) Device Name** cannot start with a number (0-9).

Use	To Get
%I1, %I2, %I3, %I4	1st through 4th octet of IP address
%M1. %M2. %M3. %M4. %M5	1st through 5th part of MAC address
%S0, %S1, %S2, %S3, %S4, %S5, %S6, %S7, %S8, %S9, %SA	0 through 10th position of serial number device.

- **1) 5250 Allow Alias**

  If a device name is entered, then you are presented with this screen. If your current device name returns an error in use, then the computer attempts to append a “$” (dollar sign) to the end of your device name to initiate a session to your host. When disabled, the device name is resent to the host for the host to send a FIN packet to the computer, causing the computer to reboot. This continues until the device name is no longer in use.

```
   Allow Alias
1) 5250 Allow Alias
```

5) Allow NAWS (Negotiate About Window Size)

When enabled, the computer supports Telnet option 31 if prompted. When disabled, this option returns a “won't do” message.

6) Skip FldExit

When enabled, a field that would generate an error code of “0020” (field exit required) when [Enter] was pressed now sends the field data.

The legacy TE 2000 application would send the entire field so a “7” entered into a 3-character blank-filled field would transmit as “7ƀƀ”, a “12” would transmit as “12ƀ”. The TE 2000 application strips the field fill characters so “7” transmits as a “7”, “12” transmits as a “12,” and so forth.

Note: A "ƀ" indicates a one-byte space.

7) More

Select **7) More** for additional functions.

- **2) Destructive BS** (backspace)

  Press [2] to use the backspace key to remove (delete) any previously-keyed data characters. When disabled, the backspace key goes back one character, but does not delete that character.

- **3) Lock Error Msg**

  The TE 2000 cursor modes ensure that the cursor location of the screen and its surrounding data is visible. The 5250 error message line is typically not visible because the error message line 24 is too far away from the cursor location. Enabling this option causes a 5250 Write Error Code command error message to be visible by placing the cursor on the error message line. When the RESET key is pressed, the cursor is restored to where it would have been if this option was not enabled (the location specified by a 5250 Insert Cursor command) and the cursor mode option then causes that area of the screen to be visible.

  Disabling this option causes normal behavior for the 5250 Write Error Code command error message. The Write Error Code command error message is displayed on the error message line of the screen and the cursor is placed at the location specified by the host Insert Cursor command. The cursor mode renders the cursor area of the screen visible.

- **4) Use Color**

  When enabled, the TE 2000 5250 application emulates single byte IBM-5292-2 and double byte IBM-5555-C01 computers. Monochrome was and still emulates single byte IBM-5291-1 and double byte IBM-5555-B01 computers.

  The 5250 Column Separator attribute is emulated using the ExtraLight font attribute which may not be noticeable on some computers.

- **5) 3477-Fx Mode**

  Enabling this option tells the 5250 host that the computer is a 132-Column display. The host then sends screens formatted for 132 column displays.

  DBCS mode ignores this option as the Telnet RFC for DBCS only supports an 80-column computer type.

  When this option is enabled, 5250 Telnet negotiation of computer type reports "IBM-3477-FG" when the **4) Use Color** option is disabled and reports "IBM-3477-FC" when the **4) Use Color** option is enabled.

- **6) Scan AutoEnter**

  If this option is enabled and the field scanned into does *not* have the **AutoEnter** attribute, then the TE 2000 application emulates the pressing of the **Enter AID** key.

- **7) More**

Select 7) **More** for additional functions.

1) Allow RTL

Enable this option to honor all 5250 commands that support right to left (Hebrew). When disabled, all 5250 commands for doing right to left are ignored. Default is enabled.

```
      5250
1) Allow RTL
```

5) 3270 (Options)

The **5) 3270** menu enables the following features *if the 3270 communication protocol is used*. Press the appropriate number to enable an option, then press [Enter] to return to the **3) Protocol Opts** menu.

```
  3270
1) Keybrd Unlock
2) Auto Tab Scan
3) Auto Entr Scn
4)
5)
6) Origin Set
7) More
```

[7]

```
  3270
1) Device Name
2) Allow NAWS
3) BRT Auto Enter
4) Any Auto Enter
```

1) Keybrd Unlock

Unlocks the keyboard after the [PA1], [PA2], or [CLEAR] keys are pressed. (Normally the keyboard remains locked, until unlocked or the computer is reset by the host, after any of these keys are pressed.) When using Token Ring on your host, the tab key also unlocks the keyboard.

2) Auto Tab Scan

Causes the cursor to tab forward to the next input field after a good scan.

Note: 2) Auto Tab Scan cannot be enabled at the same time as **3) Auto Entr Scan**.

3) Auto Entr Scan

Actuates the **<Enter>** function when a good scan is obtained.

Note: 3) Auto Entr Scan cannot be enabled at the same time as **2) Auto Tab Scan**.

6) Origin Set

When enabled, resets the screen origin when an exclamation mark is found in the data stream.

7) More

Select **7) More** for additional features.

- **1) Device Name**

  Enters a physical name for a device. This is 1-30 characters with allowable values of A-Z, a-z, 0-9, wildcards, pound symbols (#), dollar signs ($), ampersands (@), and underscores (_).

  Once a device name is generated, only the first ten characters of the name are transmitted back to the host.

Note: The first character of **1) Device Name** cannot start with a number (0-9).

Use	To Get
%I1, %I2, %I3, %I4	1st through 4th octet of IP address
%M1. %M2. %M3. %M4. %M5	1st through 5th part of MAC address
%S0, %S1, %S2, %S3, %S4, %S5, %S6, %S7, %S8, %S9, %SA	0 through 10th position of serial number device.

- **2) Allow NAWS** (Negotiate About Window Size)

  When enabled, the computer supports Telnet option 31 if prompted. When disabled, this option returns a “won't do” message.

- **3) BRT** (bright) **Auto Enter**

  When enabled, if the last field on a screen has the reverse video attribute set, then when that field is exactly filled, the computer automatically sends the data for this screen back to the host with an “Enter AID” code. If disabled, the screen waits for the user to hit the [AID] key prior to sending data back to the host. *Default is disabled.*

- **4) Any Auto Ente**r

  When enabled, an automatic "enter" occurs when a reverse video attribute field is filled by keying or scanning data. Any extra scanned data is discarded. If disabled, the screen waits for the user to hit the [AID] key prior to sending data back to the host. **Default is disabled**.

6) VT/ANSI (Options)

The **6) VT/ANSI** menu enables the following features *if the VT/ANSI communication protocol is used.*

```
   VT/ANSI
1) DEL To BS
2) CR To CRLF
3) Auto Entr Scn
4) Auto Tab Scan
5) Local Echo
6) AnswerBack
7) More
```

1) DEL to BS (Delete to Backspace)

Causes the [CLEAR] key to delete the character to the left of the cursor position, and moves the cursor back one space. When this option is not enabled, the [CLEAR] key deletes the character at the cursors position. *Default is disabled.*

2) CR to CRLF (Carriage Return to Carriage Return/Line Feed)

Causes [Enter] to perform a carriage return and a line feed. When not enabled, [Enter] performs a carriage return only. *Default is disabled.*

3) Auto Entr Scn (Auto Enter Scan)

Actuates the <**Enter**> function after a good scan. *Default is enabled.*

Note: 3) Auto Entr Scn cannot be enabled at the same time as **4) Auto Tab Scan**.

4) Auto Tab Scan

Causes the computer to automatically tab to the next input field after a good scan. *Default is disabled.*

Note: 4) Auto Tab Scan cannot be enabled at the same time as **3) Auto Entr Scan**.

5) Local Echo

Displays characters from computer memory but not from host memory. *Default is disabled.*

6) AnswerBack

Enables you to enter a character string that is sent to the host in response to an inquiry (hexadecimal 05). The range is 0-50.

When ENQ is sent, only the first thirty characters of the name are transmitted back to the host. *Default is the computer's serial number.*

```
 AnswerBack
  Message
```

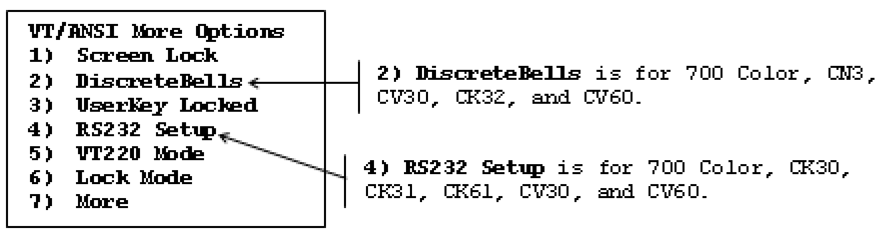
If you enter any of the following control characters, it is sent out. *Note that control strings count as one character.*

<ACK>	<DC2>	<EOT>	<GS>	<SI>	<US>
<BEL>	<DC3>	<ESC>	<HT>	<SO>	<VT>
<BS>	<DC4>	<ETB>	<LF>	<SOH>	
<CAN>	<DLE>	<ETC>	<NAK>	<STX>	
<CR>	<EM>	<FF>	<NUL>	<SUB>	
<DC1>	<ENQ>	<FS>	<RS>	<SYN>	

Use	To Get
%I1, %I2, %I3, %I4	1st through 4th octet of IP address
%M1. %M2. %M3. %M4. %M5	1st through 5th part of MAC address
%S0, %S1, %S2, %S3, %S4, %S5, %S6, %S7, %S8, %S9, %SA	0 through 10th position of serial number device.

7) More

Calls up the following VT/ANSI More Options screen.



- **1) Screen Lock**

  Sets and locks the screen to a specified size. Any characters outside this screen size are ignored by the computer. Default is disabled.

- **2) DiscreteBells** *(700 Color, CN3, CV30, CV60, CK32)*

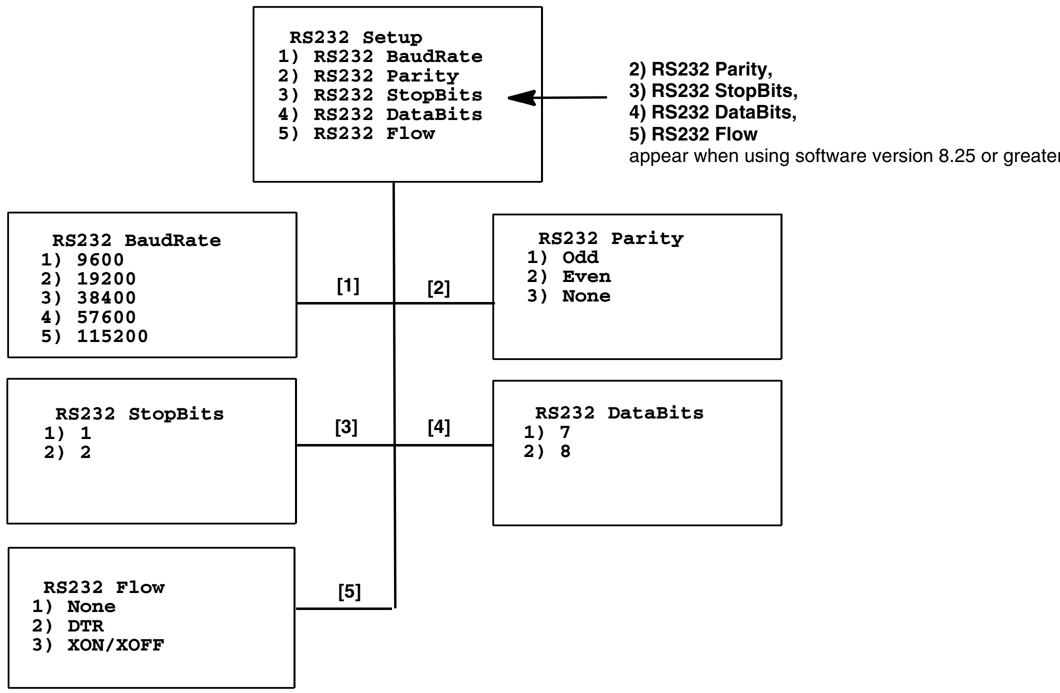
  If enabled, each bell character is played one after the other. When disabled, the bell characters are concatenated into one. Default is disabled.

- **3) UserKey Locked**

  If set, the host ignores a host command defining the Function keys. *Default is disabled.*

- **4) RS232 Setup** *(700 Color, CK30, CK31, CK61, CV30, CV60)*

  When using the media copy command, use this option to set up the communications port.



- **1) RS232 BaudRate**

  Select 9600, 19200, 38400, 57600, or 115200. *Default is 9600.*

- **2) RS232 Parity** *(software version 8.25 or greater)*

  Select from Odd, Even, or None. *Default is None.*

- **3) RS232 StopBits** *(software version 8.25 or greater)*

  Select either 1- or 2-bits. *Default is 1.*

- **4) RS232 DataBits** *(software version 8.25 or greater)*

  Select either 7- or 8-bit. *Default is 8.*

- **5) RS232 Flow** *(software version 8.25 or greater)*

  Select from None, DTR, or XON/XOFF. *Default is None.*

- **5) VT220 Mode**

  Selects operating in character or block mode. Select a choice, then press [Enter] to return to the **3) Protocol Opts** menu.

  - **1) Char** *(Default)*

    Sets the mode to Character mode. The computer sends each character as it is pressed.

  - **2) Block**

    Sets the mode to Line Edit (block) mode. The computer sends a block of characters when a terminating key is pressed.

  - **3) Transmit BS**

When enabled, pressing the backspace key sends a backspace to the host for the host to echo back to the computer. When disabled, the backspace key is handled locally on the computer by doing a destructive backspace to the printed data characters on the display.

```
  VT220 Mode
1) Char
2) Block
3) Transmit BS
```

- **6) Lock Mode**

  Use the Mode key on the computer's keyboard or keypad to toggle between Line Edit (block) mode and Character mode. Use the Lock Mode option to disable the Mode key in the VT/ANSI data stream. *By default, Lock mode is disabled.*

- 7) **More**

  Selects the telnet option for the VT/ANSI data stream.

```
  VT/ANSI
1)
2) Term Setup
3) Send XON
4) Keypad Mode
5) VT Cursor Mode
6) Terminal Mode
7) More
```

  - **2) Term Setup**

    Selects the compliance level of the emulated computer. *Default is VT340 to support all commands.* Make a selection, then press [Enter] to return to the **3) Protocol Opts** menu.

```
  Term Setup
1) ANSI
2) VT100
3) VT220
4) VT320
5) VT340
6) IBM 330X
```

  - **3) Send XON**

    Default is enabled which indicates that when an RIS is received from the host, the XON character is returned after compliance of this command. *Default is enabled.*

  - **4) Keypad Mode**

    Determines what is returned to the host when keys are pressed. *Default is 2) Numeric.*

- **1) Application**

  Generates application ESC sequences for the key code. For help, refer to your VT manual.

- **2) Numeric**

  Generates ANSI cursor control ESC sequences that correspond to what appears on the face of the keys.

```
  Keypad Mode
1) Application
2) Numeric
```

- **5) VT Cursor Mode**

  Determines what is returned to the host when cursor keys are hit. *Default is **2) Cursor**.*

- **1) Application**

  Generates application ESC sequences for the key code. For help, refer to your VT manual.

- **2) Cursor**

  Generates ANSI cursor control ESC sequences that correspond to what appears on the face of the cursor key.

```
  VT Cursor Mode
1) Application
2) Cursor
```

- **6) Terminal Mode** (*VT220/320 only*)

  Sets the terminal mode to 7-bit or 8-bit. This option sets the mode VT-series computers used to exchange escape sequences, control commands, and status reports with an application. *Default is 1) **7-Bit**.*

```
  Terminal Mode
1) 7-Bit
2) 8-Bit
```

- 7) **More**

  Below are the additional options:

```
  VT/ANSI
1) Use PC Char Set
2) Allow NAWS
3) Terminal ID
4) Auto Wrap
5) Allow LineMode
6) Do Gold Key
7) More
```

- **1) Use PC Char Set**

  Select this option to default the font character to the computer's character set instead of a DEC computer character set.

- **2) Allow NAWS** (Negotiate About Window Size)

  When enabled, the computer supports Telnet option 31 if prompted. When disabled, this option returns a “won't do” message.

- **3) Terminal ID**

  This enables the entry of a character string sent back to the host in response to IAC SB computer type SE. The range is 0 through 30 characters, with null string as the default. If set to null, then ANSI, VT100, VT220, VT320, or VT340 is returned as selected.

```
  Terminal ID
 Terminal Type
```

- **4) Auto Wrap**

  If enabled, graphic display characters received when the cursor is at the right margin appear on the next line. The screen scrolls up if the cursor is at the end of the scrolling region. If disabled (default), graphic characters received when the cursor is at the right margin replace previously displayed characters.

- **5) Allow LineMode**

  When enabled, the computer prompts to negotiate to the default LineMode RFC 1184. When disabled, the computer does not negotiate to LineMode RFC 1184.

- **6) Do Gold Key**

  When enabled, the [F1] key acts as the gold key on a VT/ANSI computer. This sends the [F1] key value, which then “hides” the next key pressed, but puts its value into a buffer to send to the host (default). Note that the gold key action is only invoked when the computer is executing in block mode. When disabled, the [F1] key acts as a function key.

- 7) **More**

  Select this for more menu options:

  1) Do UTF8

  Select this option to enable UTF-8 encoding as defined by RFC3629. If the character is between 0 and 0x7f, then nothing changes. If the character is between 0xc0 and 0xfd, then convert the character to a displayable character using the following formula where *x* is the first character in the string and *y* is the second character.

  (x - 0xc0) *2**6 + (y - 0x80)

  Note: If the character is between 0x8000-0xffff, then the UTF8 translation is done using the following formula.

  0xef 0xbb 0xbf is looked for saying that the following characters are encoded using ISO/IEC 10646 Universal Multiple-Octet Code Character Set with the UTF8 signature.

  Then each character is comprised of three characters *x*, *y*, and *z* which are converted using the following formula:

  (x-0xe0) * 0x1000 + (y-0x80) * 0x40 + (z-0x80)

  If disabled (default), then characters are translated regularly as defined by the current gl, gr character sets selected.

4) Display Opts

These adjust a number of display features.

```
 Display Opts
1)
2) Cursor Mode
3) Select Font
4)
5) Display Colors
6) Font Quality
7) More
```

6) Font Quality is for CK30, CK31

2) Cursor Mode

This selects the cursor style. After making a selection, press [Enter] to return to the **4) Display Opts** menu. The cursor shape is shown between “>” and “<”.

1) Underline Blink

This causes the underline character to appear and disappear in its current location.

2) Block Blink

This causes the block character cursor to appear and disappear in its current location, using the color of the foreground RGB value.

3) Underline

This option provides a solid underline character to display the cursor's current location.

4) Block

This provides a reverse or highlighted block character displaying the cursor's current location, using the color of the foreground RGB value.

```
 Cursor Mode
1) Underline Blink
2) Block Blink
3) Underline
4) Block
  > <
```

3) Select Font

Selects a font type to appear on the display.

1) Lucida Console

This displays the font `Lucida Console`:
`ABCDEFGHIJKLMNOPQRSTUVWXYZ`

2) Courier New

This displays the font `Courier New`:
`ABCDEFGHIJKLMNOPQRSTUVWXYZ`

3) Courier New Bold

This displays the font **`Courier New Bold`**:
`ABCDEFGHIJKLMNOPQRSTUVWXYZ`

4) Cour. N.B. Italic

This displays the font ***Courier New Bold Italic***:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```
         Select Font
1) Lucida Console
ABCDEFGHIJKLMNOPQRXTUVWXYZ
2) Courier New
ABCDEFGHIJKLMNOPQRSTUVWXYZ
3) Courier New Bold
ABCDEFGHIJKLMNOPQRSTUVWXYZ
4) Cour. N.B.Italic
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

5) Display Colors

This adjusts the foreground (text) or the background colors of the display.

```
   Display Colors
1) Foreground RGB
2) Background RGB
3) Swap Fore/Back
```

1) Foreground RGB

This assigns a color setting for the foreground (text). Select a color in the first screen, which is then displayed in the second screen. In the second screen, enter a numeric value from 0-255.

```
   Foreground RGB
1) Red
2) Blue
3) Green
```

```
   Foreground RGB
        Red
        XXX
```

(or Blue or Green)

2) Background RGB

This assigns a color setting for the background. Select a color in the first screen, which then is displayed in the second screen. In the second screen, enter a numeric value from 0-255.

```
   Background RGB
1) Red
2) Blue
3) Green
```

```
   Background RGB
        Red
        XXX
```

(or Blue or Green)

3) Swap Fore/Back

This renders the original foreground color as the new background color and the original background color as the new foreground color.

6) Font Quality (CK30, CK31)

This selects either a standard font quality or a clear type font quality to appear on the display.

1) Standard

This displays the font in standard quality.

2) Clear Type

This displays the font in the clear type quality.

```
 Font Quality
1) Standard
ABCDEFGHIJKLMNOPQRXTUVWXYZ
2) Clear Type
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

7) More

Select this for more menu options:

```
 Display Opts
1) Code Page
2) Ignore Bold
3) Ignore CnrTaps
4) Menu Hotspot
5) Func Hotspot
6) No Lockdown
7) More
```

1) Code Page

This selects the ASCII code page to use. If you are running the 3270 or 5250 emulations which display characters in EBCDIC, see **"Customizing 5250 EBCDIC to ASCII Translation" on page 192** to find the additional requirements necessary to correctly display the fonts in your selected language.

```
   Code Page
1) English
2) Cyrillic
3) Greek
4) Hebrew
5) Central Europe
6) Latin 2
7) More
```

- **1) English**

  This displays text in the English language, – code page 37.

- **2) Cyrillic** *(Russian)*

  This displays text in the Cyrillic (Russian) language – code page 855.

- **3) Greek**

  This displays text in the Greek language – code page 1253.
- **4) Hebrew**

  This displays text in the Hebrew language – code page 862.
- **5) Central Europe**

  This displays text in the Polish language – code page 1250.
- **6) Latin 2**

  This displays text in the Latin language – code page 8859-2.
- 7) **More**

  Select this for more menu options:

```
   Code Page
1) Turkish
2) Cyrillic Win
```

  - **1) Turkish**

    This displays text in the Turkish language – code page 1254.
  - **2) Cyrillic Win** *(Russian)*

    This displays text in the Cyrillic Windows (Russian) language – code page 1251.

2) Ignore Bold

When this option is enabled, the bold attribute is ignored and text is displayed using the normal attribute. When disabled, the characters with the bold applied is displayed in bold *(default is disabled)*.

3) Ignore CnrTaps

When this option is disabled, then double-taps to the screen are ignored, including double-taps in the upper right hand corner to access the TE 2000 menus and double-taps in the upper left hand corner to bring up switch session dialog boxes.

When this option is enabled, then double-taps in both the left- and right-hand corners are honored and the user can access the TE 2000 menus or the session switching dialog boxes. *Default value is to allow double-taps.*

4) Menu Hotspot

When this option is enabled, it recognizes a numeric menu option on the screen such as “90. Sign off”. Specifically ((<line begin> | <space(s)>) <digit(s)> `.' <space> <non-space>). This sends the <numeric string> to the keypad followed by the [Enter] key as if those keys were pressed.

On a double-click, if this is enabled but is not recognized, then an error beep is emitted.

5) Func Hotspot

When this option is enabled, recognizes Function key descriptions on the screen for [F1] through [F24] such as "F3=Exit". Specifically ((<line begin> | <space(s)>) 'F' <digit(s)> '=' <non-space>). This sends the Function key represented by <numeric string> to the keypad as if that key was pressed.

On a double-click, if this is enabled but is not recognized, then an error beep is emitted.

6) No Lockdown

Enabling this option switches TE 2000 from a locked-down application display with no system menu bars to a normal Windows CE display with an accessible system taskbar. This option is not by-session but applies to all sessions.

The 700 Colors, CK31, CK32, CK61, CN3, CV30, and CV60 default to **No Lockdown** disabled (locked down).

CK30s default to **No Lockdown** enabled (not locked down). Disabling this option removes the taskbar showing signal strength and disables the [green] C, [orange] B access to the System Main Menu.

7) More

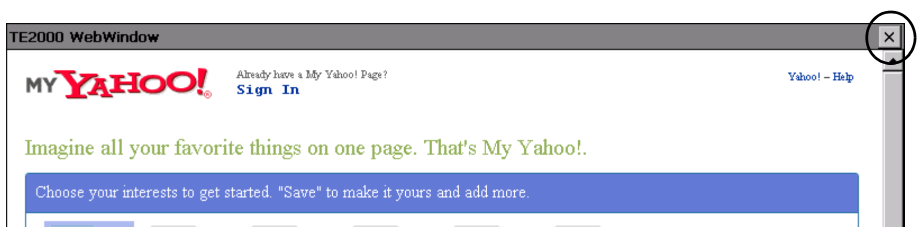
Select this for more menu options:

```
 Display Opts
1) URL Hotspots
```

- **1) URL Hotspots** *(software version 8.25 or greater)*

  Enabling this option allows you to double-tap a displayed http://url address or file://address to open that location in the default web browser.

Tap the exit button in the upper right-hand corner to close.



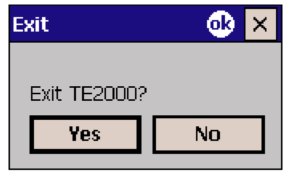
These sample screens are from the CV60.

6) Exit TE 2000

This returns control to the Windows operating system. 700 Colors, CK31s, CK32s, CK61s, CN3s, CV30s, and CV60s prompt you for a password before displaying the Exit dialog. Either press [Y] or tap **Yes** to exit the TE 2000 application and return to the Windows operating system.

```
 Exit TE2000
Enter `Y'
to Exit TE2000
```

CK30



700 Color, CK31 CK32, CK61, CN3, CV30, CV60

7) More

7) **More** provides the following additional setup parameters:

```
 Set-UP Parms
1) Menu Password
2) Print Device
3) Com Select
4) Activate Speech
5) RFID Setup
```

3) Com Select is for CV30, CV60
4) Activate Speech is for 700 Color
5) RFID Setup appears in software version 8.25 or greater

1) Menu Password

This parameter enables menu password protection for the TE 2000 configuration menus. This provides added protection because once enabled, the password must be entered before any further activity can occur within the terminal emulation menu screens. The default is to disable this option.

To enable the password, press [1], then type “3193693” for the fixed password set by Intermec. *The “1)” will change to reverse video to indicate it is enabled.* With this enabled, enter this password when you access the terminal emulation menu screens. Press [Enter] until you reach the **Main Menu**, then press **6) Exit Menus** to return to the main computer screen.

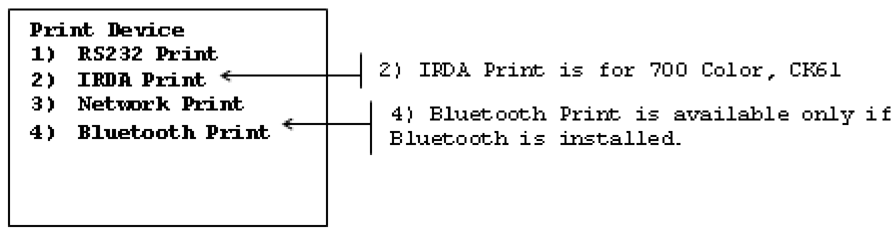
To disable the password, press [1] to display in normal viewing text. You do not need to enter a password to disable this function.

Note: The menu password can be set via a custom configuration. For help on creating a custom configuration file, see “Customizing Your Configuration” on page 153.

2) Print Device

Establishes extended commands for the RS-232, IrDA, network, or Bluetooth printer drivers.

- Press [1] for the RS-232 driver
  This is the default for the 700 series, CV60, CK30, CK31, CK60, and CV30.
- Press [2] for the IrDA driver.
  This is the appropriate setting for the 700 Color, and CK61.
- Press [3] for the Network printer.
- Press [4] for Bluetooth Prt.
  This is the default for the CN3 and CK32.



3) Network Print

Enter the IP address of the RF Printer to which the computer is to communicate. If you want to use a serial port other than port “23”, you may select from the range of 0-65535.

4) Bluetooth Print

For Bluetooth-enabled computers, when this option is selected, the print data is sent to the Bluetooth device specified in the registry. The registry is updated by using the Bluetooth Device Manager application to discover other Bluetooth devices.

Note: The TE 2000 application only supports Bluetooth devices attached to COM6.

3) Com Select

(CV30, CV60 only) Use this to select which COM port to use while doing extended commands #F, #G, or #P; or to direct output to the media copy command in the VT/ANSI data stream. For more information on the #F, #G, and #P extended commands, see “Extended Commands” on page 343.

- For CV60s, press [1] for COM1 (default) or press [2] for COM2.
- For CV30s, press [1] for COM1 (default) or press [2] for COM3.

```
  Com Select
1) Com1
2) Com2
```

CV60s

```
  Com Select
1) Com1
2) Com3
```

CV30s

4) Activate Speech (700 Color)

Use these options to support the speech feature, if installed. For more information about speech, see "Extended Commands" on page 343.

```
 Activate Speech
1) Enter Lic Key
2) Listen Timeout
```

1)Enter Lic Key

Enter a license key from 1 to 16 characters. The license key accepts numbers 0-9, the letters A-F, and hex characters. Below is a sample:

```
 Activate Speech
   Enter Lic Key
  0123456789ABCDEF
```

Note: The speech license key is case-senstive. The input for this license by way of the TE XML does not parse the data to make it all uppercase.

2)Listen Timeout

Enter a time value for however long the speech engine listens for voice input. The valid range is from 0 to 60000 seconds, with a default of 120.

```
    Listen Time
      Seconds
        120
```

5) RFID Setup

Use this option to enter a 16-character IP address of a RFID reader to connect to that reader. Select **2) Remote** to enter the IP address of the RFID reader in the second screen. For the CV30 and Color 700, the default connection is **1) Local**; otherwise, the default connection is **2) Remote**.

```
   RFID Setup
1) Local
2) Remote
```

1) Local is for CV30, 700 series

[2]

```
      Remote
  Reader Address
0123456789ABCDEF
```

2) LCD Parms (Parameters)

2) LCD Parms adjusts these Liquid Crystal Display (LCD) features:

- The screen size (number of rows displayed, and the number of characters displayed on each row).
- The screen mode (how the cursor positions itself on the display).
- Making all alphabetic character keystrokes uppercase characters.
- Scrolling window parameters.

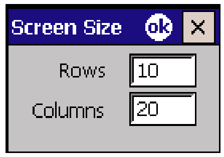
```
  LCD Parms
1)
2) Screen Size
3) Screen Mode
4) Annunciators
5)
6) Key Uppercase
7) More
```

2) Screen Size

2) Screen Size selects the number of lines and characters in each line to be viewed on the display. To change the screen size, enter a value between the minimum and maximum.

Choices for the computers include the following. Only one option can be selected at a time. (Default is 10 x 20 for 700 Colors, CK30s, CK31s, CK61s, CN3s, and CV30s – 24 x 80 for CV60s)

- For 700 Colors, CK30s, CK31s, CK61s, and CN3s, enter a value between 8-21 for rows and between 10-32 for columns.
- For CV30s and CV60s, enter a value between 8-27 for rows and a value between 16-132 for columns.



3) Screen Mode

3) Screen Mode defines the cursor position and movement as you scroll through data in the display buffer, which stores data in a standard CRT format as sent from the host computer. Since the screen is so small, use **3) Screen Mode** options to optimize your view. Default is **2) Corner Mode**. Use arrow keys to scroll (or move) the screen's view port on the display.

```
 Screen Mode
1) Center Cursor
2) Corner Mode
3) Page Mode
4) Lazy Mode
5) Locked Mode
6) Define Origin
```

1) Center Cursor

In this mode, the cursor begins near the center of the display and attempts to remain there as you scroll through the data. Upon reaching an outside boundary of the full CRT screen, the display stops advancing while the cursor moves beyond the center of the screen. When the cursor reaches the boundary of the CRT screen, an error tone sounds, such as a “beep.” The cursor does not wrap to the next line in the display.

2) Corner Mode

Starts the display at the upper-left corner of the full CRT screen and keeps the cursor in the lower-right corner of the display. As you scroll, the cursor remains there as data advances in the scrolled direction (up, down, right, or left) until an outside boundary of the full CRT screen is reached. Then the screen stops moving in relation to the display and the cursor moves in the scrolled direction (the cursor no longer remains in the corner of the display). When the cursor reaches the CRT screen boundary it stops moving. The cursor does not wrap to the next line in the display. An error tone sounds if you try to move the cursor beyond the boundary.

This option is recommended when your application uses only the upper-left corner of the full CRT screen.

3) Page Mode

Divides the full CRT screen into predefined “pages,” and starts the computer display on page 1. The cursor first appears in the upper-right corner of the display. As you scroll, only the cursor moves (the data on the screen does not appear to move). When you scroll off the edge of the displayed page, the display snaps to the next (or previous) page. Upon reaching the boundary of the CRT screen, the cursor stops moving and an error tone sounds each time you attempt to move beyond the boundary.

Some "pages" in **3) Page Mode** overlap each other (the same information is shown on both pages). This overlap occurs because the 24-row by 80-column CRT screen cannot be divided equally.

4) Lazy Mode

Starts the cursor in the upper left corner of the computer display. The cursor moves across the display in the scrolled direction. When the cursor goes beyond the edge of the display, the data begins to move in the opposite direction that the cursor is moving in and the cursor remains at the edge of the display. When you reach an outside boundary of the full CRT display, an error tone sounds each time you try to move beyond the boundary.

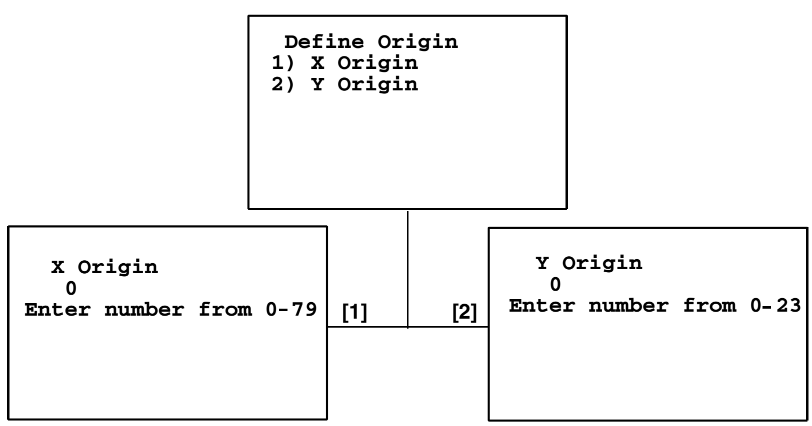
5) Locked Mode

When this is selected the screen view is locked to the upper left-hand corner of the display. Any characters selected outside of the display window size are written to the screen but are not visible. The windowing keys do not move the visible window.

Note: In 5250 emulation, the err_row is mapped to the last row of the screen size selected.

6) Define Origin

Use this option to specify the "X,Y" origin of the computer display. Enter a number from the range of 0-79 for the "X" origin and a number from the range of 0-23 for the "Y" origin. *Default is 0 for both "X" and "Y" origins.*



4) Annunciators

4) Annunciators selects and changes the location of the annunciators on the display screen. Press the up or down arrows to position the annunciators around each of the four display corners, once as a line and once as a box. You can also select “Stealth Mode” which displays the annunciators only when you make a change and then hides it when you press a key. Default is “Lower Right” for 700 Colors, CK30, CK31, CK32, CK61, and CN3 and "Stealth Mode" for CV30 and CV60.

```
  Annunciators
  Use cursor up
 and down keys to
   Adjust.
```

6) Key Uppercase

When enabled, this causes the alphabetic keys (A-Z) to display as uppercase characters regardless of the shift or caps lock mode settings.

7) More

Select **7) More** to access the following screen:

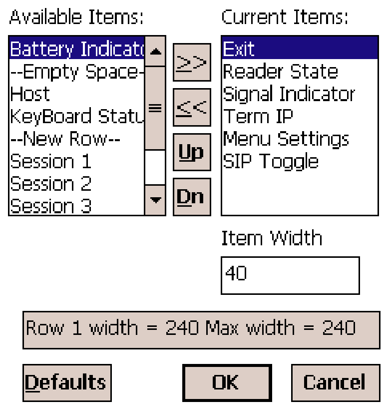
```
     LCD Parms
 1) Scroll Window
 2) Menu Settings
 3) Toolbar Opts
```

1) Scroll Window

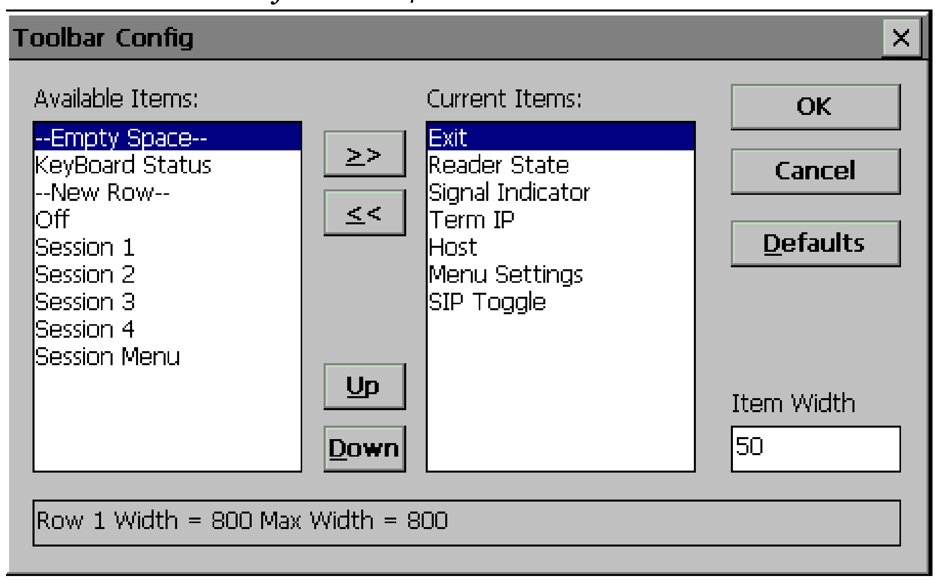
1) Scroll Window defines the cursor movement, just how far it moves with each press of the arrow keys. *Default is **1) Tab Size**.*

2) Menu Settings

When you select this option, you are prompted for a password. The default password is *cr52401*.



This sample toolbar configuration screen applies to the 700 Colors, CK31, CK32, CK61, and CN3. Note the ***Battery Indicator*** *option.*

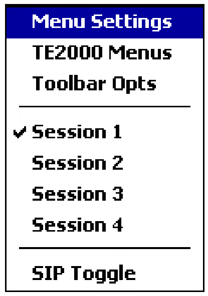


This sample toolbar configuration screen applies to CV30s and CV60s.

The current menu settings are listed on the right in the **Current Items** box. The available items that you can add to the current menu settings are listed on the left in the **Available Items** box. To select an item, tap that item, then tap either **>>** or **<<** button to move the item.

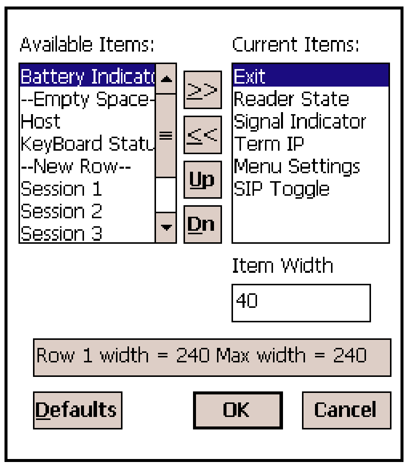
To rearrange the order the items are displayed in the **Current Items** box, select an item from the list, then use the **Up** or **Dn** buttons to move the selected item up or down the list.

This configures the way menu options are listed when the user taps the **Menu Settings** button created by the toolbar settings. When this button is pressed, this menu appears.

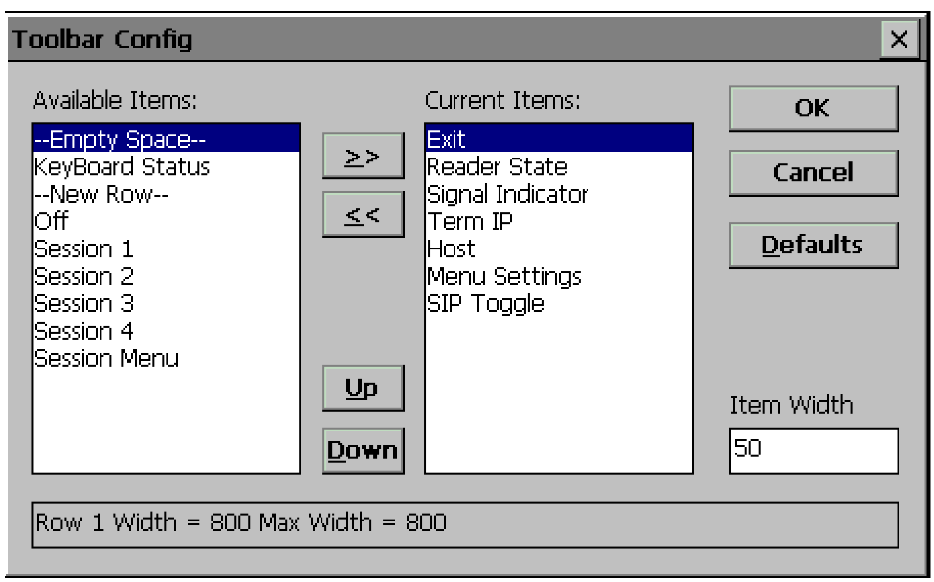


3) Toolbar Opts

(700 Color, CK31, CK32, CK61, CN3, CV30, CV60 only) When you select this option, you are prompted for a password. The default password is cr52401.



This sample toolbar configuration screen applies to the 700 Colors, CK31, CK32, CK61, and CN3. Note the ***Battery Indicator*** *option.*



This sample toolbar configuration screen applies to CV30s and CV60s.

In the **Current Items** box on the right, are the set toolbar values. The items you can include in the toolbar are listed under **Available Items** on the left.left hand side is the available items that you can add to the toolbar. To select an item, tap that item, then tap the appropriate >> or << buttons to move the item to the other box.

To rearrange the order that items are displayed in the toolbar, select an item from the **Current Items** list, then use the **Up** or **Dn** buttons to move the selected item up or down the list of items shown in the toolbar.

As you select an item, **Item Width** shows the default width value and the current row width is recalculated. On 700 Colors, CK30, CK31, CK32, CK61, and CN3, the maximum toolbar width is 240 pixels. On the CV30 and CV60, the maximum toolbar width is 800 pixels.

Note: We do not test for the maximum width of computer displays. If the toolbar exceeds the maximum width of your display, then the items in the toolbar that go past the maximum width are not shown and the data that is not visible is not checked for errors.

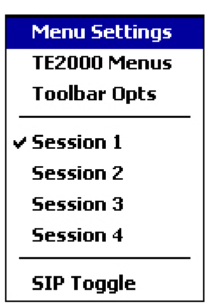
Under **Available Items**:

- Select **Host** to display the current session name if defined. If a session name is not defined then the current IP address of the host or DNS name for the host is displayed in the toolbar.
- For 700 Colors, CK31, CK32, CK61, and CN3:

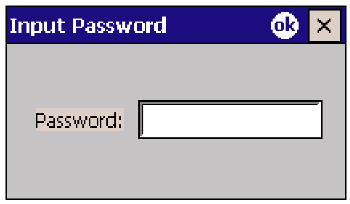
  Select **Battery Indicator** to display (in a graphical form) the percentage of the battery life still available in the toolbar. Below is a color chart:

Color	Percentage of Available Battery Life
Green	Above 66%
Yellow	Between 33% and 66%
Red	Below 33%
Blinking Red	Below 10%
Brown	Unknown

- Select **Menu Settings** to add an "M" button to your toolbar. Tap this to access the menu settings selected from the menu settings screen in option two under LCD parms. Default settings bring up this screen:



- Select **Switch Session** to add an "S" button to your toolbar. Tap this to access Session Switching. If tapped while in the TE 2000 menus, no action occurs.
- Select **Exit** to add an "EXIT" button to your toolbar. Tap this button to exit the TE 2000 application. Enter the password (default is cr52401), then tap **Yes** to continue.
- For CV60s only, select **Off** to add an "OFF" button to your toolbar. Tap this button to exit out of the TE 2000 application, save the registry, and turn off the CV60.



- Select **Signal Indicator** to show the RSSI (Radio or Ready Signal Strength Indicator) which displays the RSSI frequency retrieved from the radio module. The RSSI indicator is updated every 200 milliseconds. The color scheme is set up as follows.

Color	Status of RSSI
Dark Green	100% Excellent
Light Green	80% Favorable
Yellow	60% Good
Orange	40% Poor
Red	20% Bad

- Select **Term IP** to display the IP address of the computer. If the IP address is not known because the DCHP address is not assigned to the computer, then the literal "unknown" is displayed in this area.
- Select **SIP Toggle** to add a "T" button to your toolbar. Tap the "T" button to toggle the SIP to its new state without regard to the SIP Toggle setting in the TE 2000 menus.
- For CK30s and CK31s with lockdown mode

  Select **Orange/Green Keys** to add an icon to the toolbar to indicate the status of the orange and green keys on the keypad. For more information about the lockdown mode, see "3) Toolbar Opts" on page 145. In lockdown mode, the status of **Shift**, **Alt**, and **CAPS** are not displayed.
- Select **KeyBoard Status** in the Toolbar Config menu to add an icon to the toolbar that indicates the status of the CTL, ALT, SHF key modifiers.

If either the Shift key or the CAPS key is enabled, then the status is "SHF." If the CTRL key is pressed, the status is "CTL." If the ALT key is pressed, the status is "ALT".

- Select **Empty** to include a set number of blank pixels on the toolbar.
- Select **New Row** to divide your toolbar into two rows. All items after the new row then shows the new width of the toolbar in the selected row. Use the **Up** and **Dn** arrows to position where in your current toolbar menus you want to split the toolbar.
- Select **Session 1**, **Session 2**, **Session 3**, or **Session 4** to reserve space on the toolbar to display the session name first if entered or the IP address or DNS name of the host that to which that session is connected. Then, tap on this toolbar section to switch to that session. If no session name, IP address or DNS name is entered, then space is reserved on the toolbar and is left empty.

Note: In software version 8.25 or greater, the reader state can be defined for all computer types. CK31s, CK61s, and CN3s can show the connection state to the remote RFID reader. 700 Colors also show the state of the IP4.

- Select **Reader State** to include the state of the associated reader in the toolbar. The button displays the following:

  On 700 Colors with IP4 readers, selecting this does toggle the IP4 trigger state between RFID and SCAN:

  - "NO" will appear if a socket connection was not made to the data collection engine. It also appears any time there is a communication error between the computer and the RFID reader
  - "RF" appears when a valid socket connection is made to the data collection engine.
  - "SC" appears when the reader trigger state is set to scan.

  On CV60s with IF4 and IV7 readers:

  - "ERROR" is displayed if a socket connection was not made to the data collection engine. It also appears anytime there is a communication error between the computer and the RFID reader.
  - "READ" is displayed when a valid socket connection is made to the data collection engine and the TE 2000 application is in the read tag mode.
  - "WRITE" is displayed when a valid connection is made to the data collection engine and the TE 2000 application is executing a write extended command.

  On CV30s with IF4 and IV7 readers:

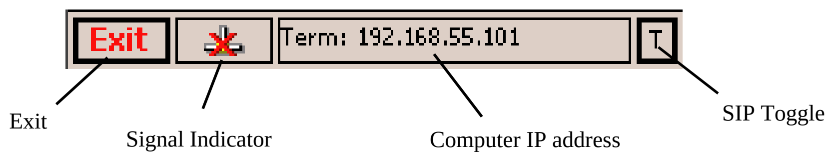
  - "RF" appears when a valid socket connection is made to the data collection engine.

- "NO" appears if a socket connection was not made to the data collection engine. It also appears anytime there is a communication error between the computer and the RFID reader.

Note: For **Host**, **Term IP**, **Session 1**, **Session 2**, **Session 3**, and **Session 4** options, when your computer is connected to the host for TCP/IP and when it is connected to the controller running UDP Plus, these options are displayed with white text in a blue background. When the computer is not connected, these appear in black text on a white background.

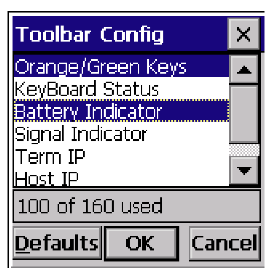
Example

In this example, the toolbar is set with an "Exit" option, the RSSI status, the IP address "192.168.55.101" for "Session 1," and includes the SIP Toggle option (**T**). Press **Defaults** to reset the toolbar to its default values.



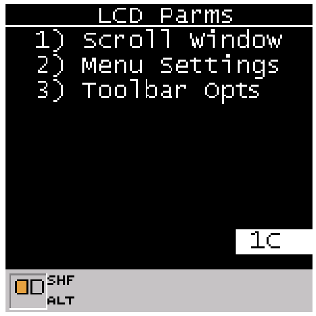
3) Toolbar Opts

(CK30 only) Unlike other computers, CK30 screen limitations prevent the use of a dual pane window to select items on the toolbar. Instead, the CK30 presents a single pane multiple selection window. Items are selected by tabbing into the pane, then using the up and down arrow keys on the keypad. Once highlighted, select the item by pressing B A on the keypad. You can select more than one item, as shown below.



In lockdown mode, the CK30 command bar is not visible, so the shift planes are not displayed. A control was added to the CK30 toolbar to indicate the shift planes.

Likewise, in lockdown mode, the status of **Shift**, **Alt**, and **CAPS** are not displayed. Selecting "KeyBoard Status" in the Toolbar Config menu presents an equivalent control, as shown in the bottom, left corner of the following illustration.



You can select "Keyboard Status" to show the current keyboard state. If either the Shift key or the CAPS key is enabled, then this area displays "SHF." If the CTRL key is pressed, then this area shows "CTL." If the ALT key is pressed, then "ALT" is shown.

4) Tests

Use the **4) Tests** menus to verify the computer scanner and keyboard.

```
  Tests
1) Peripherals
2)
3)
4)
5)
6)
```

1) Peripherals

1) Peripherals tests the keyboard and scanner. Detailed descriptions of each peripheral test follows.

```
  Peripherals
1)
2)
3)
4) Keyboard Test
5) Scanner Test
```

4) Keyboard Test

This test echoes the key value until you press [Enter] to return to the previous menu. Press any key on the keypad except [Enter], and a character corresponding to the pressed key should appear on the display.

```
   Keyboard Test
   Press a key.
  ENTER To exit
      Key =
   hex(0033) = 3
```

5) Scanner Test

This option tests the operation of an attached bar code scanner. From this menu, scan an enabled bar code to display its line under **Scan Code** and its length under **Length**. The bar code and the bar code length should appear on the display to pass. Press [Enter] to exit the text.

```
    Scanner Test
Scan Code>
TEST-SHEET
Length>
10
```

Note: If you are running in wedge mode, then no data is displayed from the scanner like that of a keyboard data entry. All keystrokes cause the computer to emit a beep except the [Enter] key, which takes you back to the previous menu.

5) Version Info (Information)

5) **Version Info** shows the name, version, and release date of the program you are using, and the version of the radio driver. For a list of TE 2000 program names, see “Program Names” on page 5.

```
  Version Info

 <TE Program Name>
Version   <number>
Date   <ddmmyy>
```

6) Exit Menus

Note: If direct connect is used, the computer may reboot upon exiting the TE configuration menu.

Use **6) Exit Menus** to exit the TE 2000 configuration menus. If you changed any parameter settings, the computer displays the following when you exit the menus.

```
 Save Parms
Enter 'Y'
to save parms
```

If you press [Y] (“yes”), you are prompted for a password. “cr52401” is the default password.

```
 Save Parms
Enter
Password
> .......
```

After you enter the correct password, your settings are saved to Flash. Some changes may cause the current session to your host to close and a new session to open. If you press a key other than [Y], you exit the menus and the new settings are *not* saved. In this case, the new settings are lost when you reboot your computer.

When you exit the TE 2000 configuration menus, the following information appears:

```
<program name> <version>
Session: <current session number>
<session name>
<emulation type>
Host: <host name>
<device name>
```

7) More (Main Menu 2)

Select 7) **More** to open the **Main Menu 2** screen.

```
  Main Menu 2
1) Keyboard Opts
2) Save Parms
3)
4) Session Menu
```

1) Keyboard Opts

Use **1) Keyboard Opts** to configure your keyboard options.

```
 Keyboard Opts
1) Type*Ahead
2) SIP Settings
3) Shift F13-F24
```

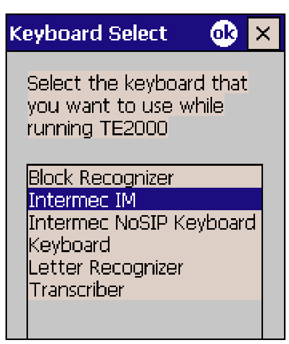
1) Type-Ahead

Use this option to key in information when the computer cannot immediately send data to the host computer enters information when the computer cannot immediately send data to the host. This stores keystrokes after the Input Inhibited annunciator (below) appears on the status line, and then saves them for th‘e next input field. Type-ahead is enabled by default.

X *Input inhibited annunciator*

2) SIP Settings

This option appears if the Enable SIP option is set to TRUE. Use this to select which SIP keyboard to use with the TE 2000 application on your computer.



3) Shift F13-F24

When this option is enabled, then pressing [Shift] [F1] through [Shift] [F12] simulate pressing F13 through F24 respectively.

2) Save Parms (Parameters)

Use 2) **Save Parms** to retain the changes made to TE configuration settings. When saved, changes become the default settings for the computer.

Note: Use this option sparingly. Each time it is used, additional memory space is occupied because previously saved changes are not erased.

Ensure that the parameters are correct before choosing **2) Save Parms**. When selected, you are prompted to enter a seven-character password. The default password is “cr52401;” which you can alter with a custom configuration to be a string of 1 to 10 characters long. After you enter the correct password, your changes are written to Flash.

If an error is made or the incorrect password is entered, the computer will return to **2) Save Parms** screen without saving your entries.

```
 Save Parms
Enter
Password:
> . . . . . . .
```

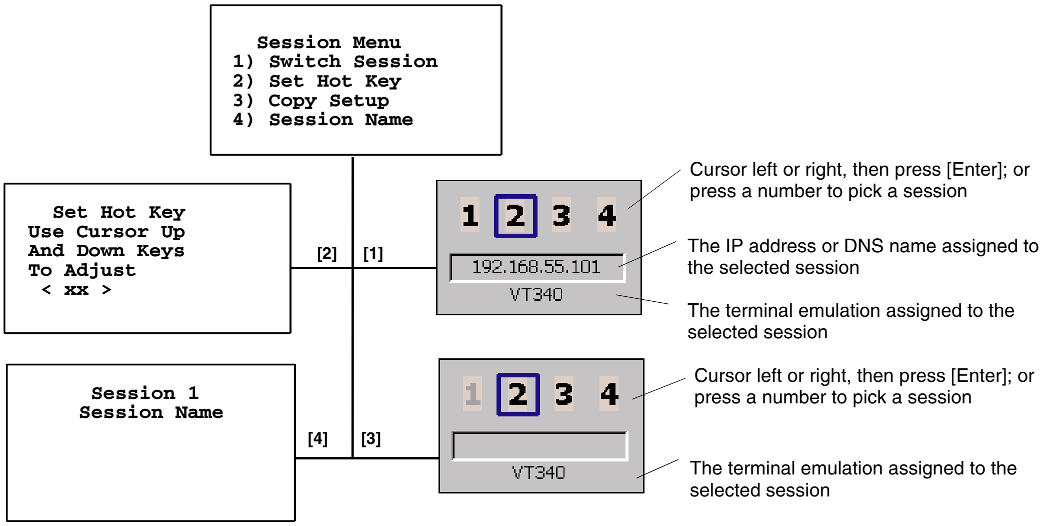
```
 Save Parms
Working
Please Wait
........
```

Note: *For 700 Colors, CK31s, CK61s, CN3s, CV30s, and CV60s*, the next menu appears after double-tapping the upper-left corner of your display.

4) Session Menu (700 Color, CK30, CK31, CN3, CV30, CV60 with TCP/IP)

Note: Computers with UDP Plus do not allow multiple sessions.

4) Session Menu defines different host communication sessions, designates a “hot key” that switches quickly between sessions, and assigns sessions customizable host names (or, friendly names).



1) Switch Session

Use this option to change the current session. Be sure to identify (or designate) the current (or intended) session before setting parameters.

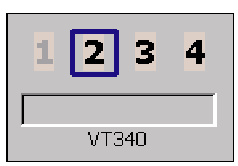
2) Set Hot Key

Displays the current hot key. Use the up and down keys to view the available choices. Press [Enter] to designate a key for the hot key.

3) Copy Setup *(password -protected)*

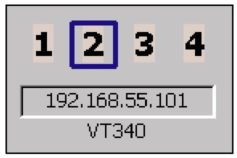
This copies parameters from the current session to the session chosen. The copied session data is grayed out and the chosen session is outlined with a box, like in this example.

To change the session to copy to use right and left arrows. To change the session that you want to copy from, you must exit this menu, then go to that session.



4) Session Name

Use this to assign customizable host names (or, friendly names) to each session. Enter a name for Session 1, then press [Enter] to return to the Session Menu. You may use up to 16 characters for 700 Colors, CK30s, CK31s, CK61s, and CN3s and up to 64 characters for CV30s and CV60s for each name.



To assign a name to another session, press [1] to switch to that session, return to the Session Menu, then press [4] to assign a name to that session.

5 Customizing Your Configuration

This chapter describes the procedures you can use to customize the standard TE 2000 program by Intermec. You customize the TE 2000 program by creating or modifying configuration files and then downloading them to your computer.

- Use the auto-login feature to send the same login information each time you log in to the host.
- Display double-byte characters.
- Create a custom parameter set-up file to download a customized file to all computers so they have the same setup information.
- Change the text of TE 2000 configuration menus or system messages.
- Preinitialize the TE 2000 program.
- Remap the computer's keys.
- Remap characters.
- Implement ITCColor.dat attribute colors.
- Customize 5250 EBCDIC to ASCII translation.

Note: If a config.dat file is present on your computer, its settings are backed up in a config.old file and then written into a TE_Settings.ini file. If there is no config.dat file on your computer, all settings are written to the TE_Settings.ini file.

Using the Auto-Login Feature

Use the auto-login feature to send the same login information each time you log in to the host. When you start the TE 2000 application, the computer checks for an auto-login script file. If a script file exists, the computer runs the login commands from the auto-login script file before the TE 2000 program starts.

To use the auto-login feature, develop an auto-login script file and load it on the computer. Instructions are listed in the following pages with a list of control characters and the procedure for disabling the auto-login feature.

Developing Auto-Login Script Files

A typical auto-login script file consists of Input and InputHidden commands followed by a HostName command, followed by a series of WaitFor and Send commands. A very simple script file may not have any input commands if all of the computers are using the same account.

Note: The auto-login script must be an ASCII text file with autolog.scr as the required file name or the file is not processed.

Commands

You can use several commands to create auto-login script files. All commands are case-sensitive. For example, **WaitFor** is a command, but **Waitfor** is not a valid command. For examples of script files, see "Sample Auto-Login Script Files" on page 160.

- **Display**

  This new autologin command is only available in TE 2000 version 8.28 and above. The user decides if the data from the host application appears on the computer screen. For additional security, the display can be turned off from updating messages from the host during the autologin process.

  Display has three parameters: OFF, ON, and HREF. Display ON enables displaying data received from the host. Display OFF disables displaying data received from the host. HREF is used to specify a bitmap to display instead of the information received from the host, if any. Use the following syntax to specify the file.

  ```
  Display "<Link HREF=file://\te2000\ball.bmp></Link>"
  ```

  The bitmap appears on the screen using its actual dimensions and will not be adjusted to fit the screen size. No further screen updates occur until the associated

  ```
  Display "ON"
  ```

  command is executed in the auto-login script file to re-enable display of information received from the host.

- **Input**

  This is called with two parameters. The first one is a character string enclosed in quotes used as a prompt to the user. The second one is a string variable name indicating where the text string is stored.

- **InputHidden**

  Same as the **Input** script command except that user input is echoed as a string of asterisks.

- **HostName**

  This command is case-sensitive and must be presented as mixed-case letters. **HostName** is followed by a character string enclosed in quotes. The character string can be a host name or an asterisk. The **HostName** command acts as an IF clause. If the host name matches, the following section of the script file is executed up to the next **HostName** command. If an asterisk is used, it matches any host name.

Note: If a session (friendly) name is entered, then this is used in place of the host name or server IP name to section off the auto-login script.

- **WaitFor**

  Wait for a list of up to ten strings. The strings must be enclosed in quotes and must be separated by a comma. The strings cannot exceed 20 characters in length.

- **Send**

  This command sends a character string enclosed in quotes or a string variable to the host. The character string enclosed in quotes can have an embedded control key in the TE 2000 application.

- **Pause "xxxxx"**

  Delays the computer for x milliseconds, halts computer operation from receiving and processing for the duration specified.

- **PromptSessionStart**

  This command is a predefined variable. If this variable is defined and set to any value other than 0, the application prompts the user to press [Enter] before starting a Telnet session with the host. Do not put quotes around the variable.

- **Restart "x"**

  Restarts the autologin script file. The "x" is a dummy argument.

- **Keyboard "0"**

  Disables the keyboard. Key presses are ignored.

- **Keyboard "1"**

Enables the keyboard. Key presses are processed. The keyboard is enabled by default.

Note: Input the **Keyboard "0"** or **Keyboard "1"** command into the autologin script file after the **PromptSessionStart** command (if present) and after the **HostName** command (if present) but before another **HostName** command is found in the file.

- **# (pound symbol)**

  Documents the script file. Text following a # (pound) symbol is considered a comment unless the # symbol is in a quoted string.

Search Strings

Some auto-login search string limitations are the searches are case-sensitive, the maximum search string length is 20 characters, and each WaitFor command searches the entire screen from the top. You can use line wrapping to look for unique strings.

If a screen from the host has multiples of the word you are looking for, you can use the preceding spaces to identify a unique string.

Example

If the screen sent to the computer is:

```
Linux rlogin 2.4.6
login
```

The autologin script would be:

```
PromptSessionStart=1
HostName "*"
#wait for host login screen and send login and password
WaitFor "login"
Send "billy<ENTER>"
WaitFor "password"
Send "letmein<ENTER>"
```

In this example, you can search for the three leading spaces from the end of the previous line to make a unique search string.

Control Characters

You can include control characters in your auto-login script file. The control character must be enclosed by < > (angle brackets) in AutoLog.scr.

Note: Some control characters may be represented by their hexadecimal values. For a full description of control characters and hexadecimal equivalents, see "Tables" on page 407.

3270 Control Characters

This table lists control characters for the TE 2000 3270 application.

3270 Control Characters for Auto-Login Script File

Control Character	Definition	Control Character	Definition
<CLEAR>	Clear	<INS>	Insert
<CUR_DN>	Cursor Down	<LTAB>	Left Tab
<CUR_LF>	Cursor Left	<NEWLN>	New Line
<CUR_RT>	Cursor Right	<PA1>	PA1
<CUR_UP>	Cursor Up	<PA2>	PA2
<DEL>	Delete	<PA3>	PA3
<ENTER>	Enter	<RESET>	Error reset
<ERS_EOF>	Erase EOF	<RTAB>	Right Tab
<F1> - <F24>	Function keys	<SPACE>	Space
<HOME>	Home		

5250 Control Characters

This table lists control characters for the TE 2000 5250 application.

5250 Control Characters for Auto-Login Script File

Control Character	Definition	Control Character	Definition
<ATTN>	Attention	<HOME>	Home
<CLEAR>	Clear	<INS>	Insert
<CUR_DN>	Cursor Down	<LTAB>	Left Tab
<CUR_LF>	Cursor Left	<NEWLN>	New Line
<CUR_RT>	Cursor Right	<RESET>	Error Reset
<CUR_UP>	Cursor Up	<ROLL_DOWN>	Roll Down
<DEL>	Delete	<ROLL_UP>	Roll Up
<ENTER>	Enter	<RTAB>	Right Tab
<ERS_EOF>	Erase Input	<SPACE>	Space
<F1> - <F24>	Function keys		

Native Control Characters

This lists control characters for the Native terminal emulation application.

Native Control Characters for Auto-Login Script File

Control Character	Definition	Control Character	Definition
<CLEAR>	Clear	<F1> - <F4>	Function keys
<CR>	Carriage return	<HOME>	Home
<CUR_DN>	Cursor down	<INS>	Insert
<CUR_LF>	Cursor Left	<LTAB>	Left Tab
<CUR_RT>	Cursor Right	<NEWLN>	New Line
<CUR_UP>	Cursor Up	<RESET>	Error reset
<DEL>	Delete	<RTAB>	Right Tab

Native Control Characters for Auto-Login Script File

Control Character	Definition	Control Character	Definition
<ERS_EOF>	Erase EOF	<SPACE>	Space

VT/ANSI Control Characters

This table lists control characters for the TE 2000 VT/ANSI application.

VT/ANSI Control Characters for Auto-Login Script File

Control Character	Definition	Control Character	Definition
<ACK>	Acknowledgment	<F1> - <F20>	Function keys
<BEL>	Bell	<F21>	Toggles from Character mode/Line Edit (block) mode
<BS>	Backspace	<FF>	Form Feed
<CAN>	Cancel	<FS>	File Separator
<CR>	Carriage Return	<GS>	Group Separator
<CUR_DN>	Cursor Down	<HT>	Horizontal Tab
<CUR_LF>	Cursor Left	<INS>	Insert
<CUR_RT>	Cursor Right	<LF>	Line Feed
<CUR_UP>	Cursor Up	<LTAB>	Left Tab
<DC1>	Device Control 1 (XON)	<NAK>	Negative Acknowledge
<DC2>	Device Control 2	<NUL>	Null, or all zeros
<DC3>	Device Control 3 (XOFF)	<RS>	Record Separator
<DC4>	Device Control	<RTAB>	Right Tab
<DEL>	Delete	<SI>	Shift In
<DLE>	Data Link Escape	<SO>	Shift Out
<EM>	End of Medium	<SOH>	Start of Heading
<ENQ>	Enquiry	<SPACE>	Space
<ENTER>	Enter	<STX>	Start of Text
<EOT>	End of Transmission	<SUB>	Substitute
<ESC>	Escape	<SYN>	Synchronous Idle
<ETB>	End Transmission Block	<US>	Unit Separator
<ETX>	End of Text	<VT>	Vertical Tab

Loading the Auto-Login Script File

Follow these procedures to download an auto-login script file to your computer. The method depends on the type of computer you are using.

This procedure requires that you have Microsoft ActiveSync installed on your desktop and that you have established a remote connection to your computer.

To load the auto-login script file

1 On the computer, tap **Start > ActiveSync > Sync**.

2 From your desktop, start Microsoft ActiveSync, then click **Explore** to access the contents of the computer via its “Mobile Device” directory or folder. Browse to the appropriate location where to install the files.

3 Also from your desktop, select **Start > Windows Explorer**, then browse for the files to transfer to your computer.

4 Drag those files from the desktop Windows Explorer to the ActiveSync “Mobile Device” window.

5 From the computer, tap **Start > Programs > File Explorer**, then browse to where to place the files. If missing, then ensure all connections and locations are accurate before attempting this procedure again.

Note: Always store customized files such as the AutoLog.scr, remap.cfg, TE_Settings.ini, cfglit.dat, or other such files in the directory where the TE 2000 executable is stored to ensure that it is executed. If the same file exists in more than one location, only the file stored in the directory with the highest precedence will be executed. The other files will be ignored.

The order of precedence is:

1 In the directory with the executable

2 In the root of the Secure Digital card, if present.

3 In the Flash File Store, if present.

4 In the computer’s root directory.

Directory where the TE 2000 executable is stored

Computer model	Additional Conditions	Executable Directory
CV30, CV60, CK32, CK61		\TE2000
CK30, CK31	TE 2000 versions prior to v8.20	\CK_FFS
CK30, CK31	TE 2000 version v8.20 and later	\CK_FFS\TE2000
700 Colors	If a Secure Digital card is present If there is no Secure Digital card	\SDMMC Disk\TE2000 \Flash file store\TE2000
CN3		\te2000

Disabling the Auto-Login Feature

To disable auto-login, you may rename or delete the AutoLog.scr file. Renaming the file ensures that you can use the same auto-login script file later by changing the name back to AutoLog.scr. If you want to enable a new script file, you can use the instructions in this section to rename a different script file to AutoLog.scr.

Follow these procedures to disable (or delete) the auto-login script file. The method depends on the type of computer you are using.

To disable (delete) the auto-login file

1. Using your web browser, access the IP address at *http://<IP address>.*
2. Double-click the **File Manager** desktop icon.
3. Select **File**.
4. Press the [Del] key to remove the auto-login feature.

For the computer, using Microsoft ActiveSync

1. Browse for the auto-login file.
2. Right-click the file for a pop-up menu.
3. Select **Delete** to remove the file from the computer.

Sample Auto-Login Script Files

You can use these sample script files as they are or as the starting point for creating your own auto-login script files.

Example 1

Auto-Login With All Computers Using the Same Account

```
HostName "*"                                  #Use this to log into any host
WaitFor "login:"                              #Wait for the login prompt
Send "username<NEWLN>"                        #Send the user name
WaitFor "Password:"                           #Wait for the password prompt
Send "letmein<ENTER>"                         #Send the password
```

- The **HostName** command matches the host the user accesses.
- The **WaitFor** command waits for a string to be displayed by the host. **WaitFor** takes up to 10 strings, 20 characters long. The strings must be enclosed in quotes and separated by a comma.
- The first **Send** command sends a fixed user name, the second **Send** sends a fixed password.
- Angle brackets < and > can enclose uppercase mnemonics or hexadecimal values.

Example 2

```
Auto-Login With Different User Names and Passwords
Input "Enter user name", username            #Prompt for user name
InputHidden "Enter password", password       #Prompt for password
HostName "*"
WaitFor "login:"                             #Wait for login prompt
Send username                                #Send the user name
Send "<NEWLN>"                               #Send a carriage return
WaitFor "Password:"                          #Wait for password prompt
Send password                                #Send the users password
Send "<ENTER>"                               #Send a carriage return
```

- The **Input** and **Send** commands use input variables. **Input** commands require a prompt string followed by a comma and a variable name in which to store the string.
- The **InputHidden** command displays "*" in place of any characters the user types. Place all **Input** commands before the first **HostName** command.
- The **Send** command only accepts a single argument, so you need two **Send** commands to send the user name and a carriage return.

Example 3

```
Auto-Login to an Application
Input "Enter user name", username            #Prompt for user name
InputHidden "Enter Password", password       #Prompt for password
HostName "*"
WaitFor "login:"                             #Wait for login prompt
Send username                                #Send the user name
Send "<NEWLN>"                               #Send a carriage return
WaitFor "Password:"                          #Wait for password prompt
Send password                                #Send the users password
Send "<ENTER>"                               #Send a carriage return
WaitFor "Main Menu"                          #Wait for the main menu
Send "3"
Send "<ENTER>"                               #Pick option 3 from menu
WaitFor "Wip Menu"                           #Await work-in-process menu
Send "1"
Send "<ENTER>"                               #Pick option 1 from menu
```

- Example 3 modifies the script file in Example 2. The additional modification (which starts with WaitFor "Main Menu") allows you to move automatically to an application after logging in.

Example 4

```
Auto-Login With Variable Processing
Input "Enter user name", username            #Prompt for user name
InputHidden "Enter Password", password       #Prompt for password
HostName "BigHost"                           #Use script portion for BigHost
WaitFor User:"                               #Wait for the user prompt
Send username                                #Send the user name
Send "<NEWLN>"                               #Send a carriage return
WaitFor "Password:"                          #Wait for password prompt
Send password                                #Send the users password
Send "<ENTER>"                               #Send a carriage return
HostName "*"                                 #Match any host name
WaitFor "login:"                             #Wait for login prompt
Send username                                #Send the user name
Send "<NEWLN>"                               #Send a carriage return
WaitFor "Password:"                          #Wait for password prompt
Send password                                #Send the users password
Send "<ENTER>"                               #Send a carriage return
WaitFor "Main Menu"                          #Wait for the main menu
Send "3"
Send "<ENTER>"                               #Pick option 3 from menu
WaitFor "Wip Menu"                           #Await work-in-process menu
Send "1"
Send "<ENTER>"                               #Pick option 1 from menu
```

- A section for the host name BigHost is added to the beginning of the script file. If you log into any host other than BigHost, the script file starts at the **HostName** "*" line. This allows for different processing on each host.

Auto-Login Restart

Restarts the auto-login script file from the correct **HostName** statement in the script file when a host session is broken. For this command to work, the **WaitFor** string must match the last data sent from the host. For example, if the **WaitFor** string is the login prompt "login:" with a space after the colon, the **WaitFor** string must include a space for the auto-login restart to work.

To use the **Auto-Login Restart** command, press the keys listed in "Using the Computer Keypad" on page 19 or scan the following bar code. Note that Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Auto-Login Restart

%ALRS

Displaying Double-Byte Characters (5250, VT/ANSI)

To order a Double-Byte Character Set, contact your Intermec representative and then follow instructions in the computer user manual to install it.

Loading Double-Byte Characters (700 Color)

The following files are downloaded for 700 Colors devices. Be sure to set the computer's screen size to 8x16.

To load cab files that support double-byte characters

1 From your desktop, select **Start > Windows Explorer**, then browse the "C:\\Program Files\Intermec\TE2000" path for the following .cab files:

Language	Pocket PC 2003 or Windows Mobile 2003
Traditional Chinese	BIG730.arm.cab
Japanese	JP730.arm.cab
Thai	THAI730.arm.cab
Korea	DIopen1.cab

2 Select the appropriate .cab file, do a right-click for a pop-up menu, then select **Copy**.

3 Within the **Mobile Device** directory, select **My Pocket PC > SDMMC_Disk > cabfiles**.

4 Do a right-click for a pop-up menu, then select **Paste**.

5 If the Secure Digital storage card is not installed in 700 Colors, paste the .cab file into the root of the object store.

6 After extracting the .cab files, perform a warm-boot on the computer. When rebooted, the newly copied .cab file automatically activates the double-byte character fonts for usage.

Loading Double-Byte Characters (CK30, CK61)

The CK30, CK61s need these .cab files for double-byte character support.

To load cab files that support double-byte characters

1 From your desktop, select **Start** > **Windows Explorer**, then browse the "C:Program Files\Intermec\TE2000" path for the following .cab files:

Language	Windows CE
Simplified Chinese	GBCK30.arm.cab
Traditional Chinese	BIGCK30.arm.cab
Japanese	JPCK30.arm.cab
Thai	THAICK30.arm.cab
Korea	KRCK30.arm.cab

2 Select the appropriate .cab file, do a right-click for a pop-up menu, then select **Copy**.

3 Within the **Mobile Device** directory on your SecureDigital storage card, select **My Pocket PC > SDMMC_Disk > cabfiles**.

4 Do a right-click for a pop-up menu, then select **Paste**.

5 After extracting the .cab files, wait three minutes before performing a warm-boot on the CK30 or CK61. When rebooted, the newly copied .cab file automatically activates the double-byte character fonts for usage.

Creating a Custom Parameter Set-Up File

You can create a custom TE 2000 set-up file to preset almost any parameter you can set from the TE 2000 configuration menus. These parameter settings become the default (cold start) configuration for the computer.

Note: Currently, Intermec does not support the imager via the "Barcode Parms" section described in Chapter 4, "Using the Terminal Emulation Menus" on page 97. There are also some limitations to the laser implementation. The following message appears when you access the TE 2000 application on your Intermec computer. Tap **OK** to close this message.

```
TE2000 Image Scanner
Barcode Parms menu
options NOT supported!
```

Upgrading from Config.dat Configuration

Note that the settings from the Config.dat file are the same as the settings for the TE_Settings.ini file. For example, the config.dat setting:

```
[Host A].Host
```

would appear as

```
session_1|host_a|host = 136.179.84.76
```

in the TE_Ssettings.ini file.

All values follow the [iccu values] section and use the pipe character to separate the names. The settings and literals are the same as the old config.dat with the exception of the use of the different naming convention using the pipe ("|"). Below are a few values from the config.dat file as they appear in the TE_SETTINGS.INI file.

```
[iccu values]
   session_1|host_a|host = 136.179.84.76
   session_1|host_a|port_number = 23
   session_1|destructive_bs = 0
```

TE_Settings.ini Configuration

```
"enable_sip" = 0 or 1
```

This enables or disables the interaction betweenTE 2000 and the onscreen keyboard. If the value is 1, TE 2000 displays the SIP when it starts and when it gains focus. If the value is 0, TE 2000 does not display the SIP and it does not toggle the SIP when you click the SIP toggle button.

```
"sip_settings" = {674EC110-EFF0-47D3-B828-CDB2A6CCD3EB}
```

This is a GUID (globally unique identifier) identifying the SIP that the TE 2000 application is to use by default. This can be the GUID of any registered SIP in the system.

These are for debugging purposes only. They inform which version of TE 2000 created the TE_Settings.ini file.

- program_name = FWP700H0
- program_version = 7.53

TE_Settings.ini Parameter Formats

Each parameter in the setup file is followed by one of three different formats that indicates the type of parameter and the values it can contain. Formats are as follows:

- **Literal strings.** List of fixed values to choose from. Session 1 or 2 may qualify Screen Mode. It may take the value Center Cursor, Corner Mode, Page Mode, Lazy Mode, or Locked Mode. Session 1 is the default qualifier. These configuration lines are valid:

```
screen_mode = Lazy Mode;
session_1|screen_mode = Lazy Mode;
session_2|screen_mode = Lazy Mode;
```

- **Numeric parameters**: Numeric parameters have minimum and maximum values. Parameters can be either decimal or hexadecimal:
  - Decimal parameters consist of digits 0 through 9.
  - Hexadecimal parameters consist of 0x or 0X, followed by 1 to 4 digits of 0 through 9, a through f, or A through F. These are equivalent: 160, 0xA0, and 0Xa0.

  Port Number is a variable with a minimum value of 0 and a maximum of 65535. These lines are valid:

  ```
  port_number = 1;
  session_1|host_a|port_number = 1;
  session_2|host_a|port_number = 1;
  ```

- **String parameters:** String parameters are variables with minimum and maximum lengths.

  Program Name is unqualified. It must have eight characters. These configuration lines are valid:

  ```
  program_name = ABCDEFGH;
  program_name = FWP248H0;
  ```

TE_Settings.ini Parameters and Qualifiers

Note: If parameters are duplicated, an error is not generated. Instead, the computer processes the entire file and uses the last entered value of the duplicated parameter.

Parameters in the set-up file apply to all model numbers unless otherwise noted here. The following pages list each parameter and its qualifiers. Some qualifiers have default values. If you want to use the default value, you can omit the qualifier from the parameter set-up file.

cfglit.dat specifies the text of setup menus or system messages. Parameters and qualifier strings in the setup file assume the default cfglit.dat is used. For instructions on how to change this file, see “Changing Text” on page 178.

Set-Up Parameters Options

Note that brackets indicate default settings and values.

Set-Up Parameters Options

Options	Session [1], 2, 3 4	Values
Data Stream	host_a|data_stream *(default)* host_b|data_stream host_c|data_stream	Native *(if Real Time Control (RTC) over TCP is selected, then blank if UDP Plus or TCP/IP), 3270, 5250, [VT/ANSI]*
Host Name	host_a|host *(default)* host_b|host host_c|host	String, minimum = 0, maximum = 16 [null string]
Port Number	host_a|port_number *(default)* host_b|port_number host_c|port_number	Numeric, minimum = 0, maximum = 65535 [23]
RTC over TCP	host_a|rtc_over_tcp *(default)* host_b|rtc_over_tcp host_c|rtc_over_tcp	Enabled, [Disabled]
Server IP	host_a|server_ip *(default)* host_b|server_ip host_c|server_ip	Numeric, minimum = 1, maximum = 16 [16]
SNA	sna	Enabled, [Disabled]
Telnet	telnet	Enabled, [Disabled}
Terminal Type	host_a|terminal_type *(default)* host_b|terminal_type host_c|terminal_type	Numeric, minimum = 0, maximum = 255 [83]
Unit Number (same as radio number)	host_a|unit_# *(default)* host_b|unit_# host_c|unit_#	Numeric, minimum = 0, maximum = 127 [127]

Bar Code Parameters

Note that brackets indicate default settings and values.

Bar Code Parameters

Parameters	Session [1], 2, 3, 4	Values
BC Type Character	bc_type_char	Enabled, [Disabled]
Code 39 Encoded	code_39_encoded	Enabled, [Disabled]
Concatenate	concatenate	Enabled, [Disabled]
MOD 10 Check	mod_10_check	Enabled, [Disabled]
No ScanAhead	no_scanahead	Enabled, [Disabled]
ParmsBySession	parmsbysession	Enabled, [Disabled]
Scan All Fields	scan_all_flds	Enabled, [Disabled]
Scan Postamble Character	scan_postchar	Numeric, minimum = 0, maximum = 255 [32]
Scan Preamble Character	scan_prechar	Numeric, minimum = 0, maximum = 255 [32]
Scanner Type	scanner_type	[No Scanner], Wand, Laser, Wand Emulate, Auto Detect
Stream Scan	stream_scan	Enabled, [Disabled]
Use Wedge Mode	use_wedge_mode	[Enabled], Disabled

Bar Code Symbologies

Note that brackets indicate default settings and values.

Bar Code Symbologies

Symbologies	Session [1], 2, 3, 4	Values
ABC Codabar	abc_codabar	When enabled, scanner_type is 33 [1]
Codabar	codabar	When enabled, scanner_type is 32 [1]
Code 11	code_11	When enabled, scanner_type is 56 [1]
Code 39	code_39	When enabled, scanner_type is 40 [1]
Code 93	code_93	When enabled, scanner_type is 72 [1]
Code 128	code_128	When enabled, scanner_type is 64 [1]
Computer Identics 2 of 5	ci_2of5	When enabled, scanner_type is 51 [1]
EAN	ean	When enabled, scanner_type is 17 [1]
Interleaved 2 of 5	int_2of5	When enabled, scanner_type is 50 [1]
Plessey	plessey	When enabled, scanner_type is 8 [1]
Straight 2 of 5	str_2of5	When enabled, scanner_type is 49 [1]
UPC	upc	When enabled, scanner_type is 16 [1]

Generic Bar Code Options

Note that brackets indicate default settings and values.

Note: Generic bar code options must be qualified by one of the bar code symbology strings. For example: "session_1|upc|max_length = 13;"

Generic Bar Code Options

<table>
<tr><th>Options</th><th>Session [1], 2, 3, 4</th><th>Values</th></tr>
<tr><td>Drop Leading</td><td>upc|drop_leading
ean|drop_leading
code_128|drop_leading
code_39|drop_leading
codabar|drop_leading
abc_codabar|drop_leading
str_2of5|drop_leading
int_2of5|drop_leading
ci_2of5|drop_leading
code_11|drop_leading
code_93|drop_leading
plessey|prop_leading</td><td>Numeric, minimum = 0,
maximum = 15 [0]</td></tr>
<tr><td>Drop Trailing</td><td>upc|drop_trailing
ean|drop_trailing
code_128|drop_trailing
code_39|drop_trailing
codabar|drop_trailing
abc_codabar|drop_trailing
str_2of5|drop_trailing
int_2of5|drop_trailing
ci_2of5|drop_trailing
code_11|drop_trailing
code_93|drop_trailing
plessey|drop_trailing</td><td>Numeric, minimum = 0,
maximum = 15 [0]</td></tr>
<tr><td>Fixed Length 1</td><td>upc|fix_length_1
ean|fix_length_1
code_128|fix_length_1
code_39|fix_length_1
codabar|fix_length_1
abc_codabar|fix_length_1
str_2of5|fix_length_1
int_2of5|fix_length_1
ci_2of5|fix_length_1
code_11|fix_length_1
code_93|fix_length_1
plessey|fix_length_1</td><td>Numeric, minimum = 0,
maximum = 99 [0]</td></tr>
</table>

Generic Bar Code Options (continued)

Options	Session [1], 2, 3, 4	Values
Fixed Length 2	upc|fix_length_2 ean|fix_length_2 code_128|fix_length_2 code_39|fix_length_2 codabar|fix_length_2 abc_codabar|fix_length_2 str_2of5|fix_length_2 int_2of5|fix_length_2 ci_2of5|fix_length_2 code_11|fix_length_2 code_93|fix_length_2 plessey|fix_length_2	Numeric, minimum = 0, maximum = 99 [0]
Fixed Length 3	upc|fix_length_3 ean|fix_length_3 code_128|fix_length_3 code_39|fix_length_3 codabar|fix_length_3 abc_codabar|fix_length_3 str_2of5|fix_length_3 int_2of5|fix_length_3 ci_2of5|fix_length_3 code_11|fix_length_3 code_93|fix_length_3 plessey|fix_length_3	Numeric, minimum = 0, maximum = 99 [0]
Fixed Length 4	upc|fix_length_4 ean|fix_length_4 code_128|fix_length_4 code_39|fix_length_4 codabar|fix_length_4 abc_codabar|fix_length_4 str_2of5|fix_length_4 int_2of5|fix_length_4 ci_2of5|fix_length_4 code_11|fix_length_4 code_93|fix_length_4 plessey|fix_length_4	Numeric, minimum = 0, maximum = 99 [0]
Maximum Length	upc|max_length ean|max_length code_128|max_length code_39|max_length codabar|max_length abc_codabar|max_length str_2of5|max_length int_2of5|max_length ci_2of5|max_length code_11|max_length code_93|max_length plessey|max_length	Numeric, minimum = 0, maximum = 99 [0]

Generic Bar Code Options (continued)

Options	Session [1], 2, 3, 4	Values
Minimum Length	upc|min_length ean|min_length code_128|min_length code_39|min_length codabar|min_length abc_codabar|min_length str_2of5|min_length int_2of5|min_length ci_2of5|min_length code_11|min_length code_93|min_length plessey|min_length	Numeric, minimum = 0, maximum = 99 [0]

UPC Options

Note that brackets indicate default settings and values.

UPC Options

Options	Session [1], 2, 3, 4	Values
Add-On 2	upc|add-on_2	Enabled, [Disabled]
Add-On 5	upc|add-on_5	Enabled, [Disabled]
Expand E to A	upc|expand_e_to_a	Enabled, [Disabled]
System 0 UPCE	upc|sys_0_upce	Enabled, [Disabled]
System 1 UPCE	upc|sys_1_upce	Enabled, [Disabled]

EAN Options

Note that brackets indicate default settings and values.

EAN Options

Options	Session [1], 2, 3, 4	Values
Add-On 2	ean|add-on_2	Enabled, [Disabled]
Add-On 5	ean|add-on_5	Enabled, [Disabled]
Expand 8 to 13	ean|expand_8to13	Enabled, [Disabled]

Code 39 Options

Note that brackets indicate default settings and values

Code 39 Options

Options	Session [1], 2, 3, 4	Values
Auto-Encoded	code_39|auto-encoded	Enabled, [Disabled]
Check Digit	code_39|chk_digit	Enabled, [Disabled]
Encoded	code_39|encoded	Enabled, [Disabled]
Full ASCII	code_39|full_ascii	Enabled, [Disabled]

Interleaved 2 of 5 Option

Note that brackets indicate default settings and values.

Interleaved 2 of 5 Option

Option	Session [1], 2, 3, 4	Values
Interleaved 2 of 5 Chk Digit	interleaved_2_of_5_chk_digit	Enabled, [Disabled]

Code 11 Options

Note that brackets indicate default settings and values.

Code 11 Options

Options	Session [1], 2, 3, 4	Values
Check Digit 1	code_11|chk_dig_1	Enabled, [Disabled]
Check Digit 2	code_11|chk_dig_2	Enabled, [Disabled]

Plessey Option

Note that brackets indicate default settings and values.

Plessey Option

Options	Session [1], 2, 3, 4	Values
MOD 10 Check	plessey|mod10_chk	Enabled, [Disabled]

Code 128 Options

Note that brackets indicate default settings and values.

Code 128 Options

Options	Session [1], 2, 3, 4	Values
No UCC Type	code_128|no_ucc_type	Enabled, [Disabled]
UCC/EAN	code_128|ucc/ean	Enabled, [Disabled]
UCC F1 Value	code_128|ucc_f1_value	Numeric, minimum = 0, maximum = 255 [0]

3270 Protocol Options

Note that brackets indicate default settings and values.

3270 Protocol Options

Options	Session [1], 2, 3, 4	Values
Allow Negotiate About Window Size (NAWS)	allow_naws	Enabled, [Disabled]
Any Auto Enter	any_auto_enter	Enabled, [Disabled]
Auto Enter Scan	auto_entr_scn	[Enabled], Disabled

3270 Protocol Options

Options	Session [1], 2, 3, 4	Values
Auto Tab Scan	auto_tab_scan	Enabled, [Disabled]
Bright (BRT) Auto Enter	brt_auto_enter	Enabled, [Disabled]
Device Name	devicename	String, minimum length=0, maximum length=30 [null string]
Keyboard Unlock	keybrd_unlock	Enabled, [Disabled]
Origin Set	origin_set	Enabled, [Disabled]
Skip Field Exit	skip_fldexit	Enabled, [Disabled]

5250 Protocol Options

Note that brackets indicate default settings and values.

5250 Protocol Options

Options	Session [1], 2, 3, 4	Values
3477-Fx Mode	3477-fx_mode	Enabled, [Disabled]
Allow Alias	allow_alias	Enabled, [Disabled]
Allow RTL	allow_rtl	[Enabled], Disabled
Allow Negotiate About Window Size (NAWS)	allow_naws	Enabled, [Disabled]
Auto Tab Scan	auto_tab_scan	Enabled, [Disabled]
Beep On Error	beep_on_error	Enabled, [Disabled]
Device Name	devicename	String, minimum length = 0, maximum length = 30 [null string]
Destructive BS	destructivebs	Enabled, [Disabled]
Lock Error Message	lock_error_msg	Enabled, [Disabled]
Scan AutoEnter	scan_autoenter	Enabled, [Disabled]
Skip Field Exit	skip_fldexit	Enabled, [Disabled]
Telnet	telnet	Enabled, [Disabled]
Use Color	use_color	Enabled, [Disabled]

Native

Note that brackets indicate default settings and values.

Native Protocol Option

Options	Session [1], 2	Values
F1 is Function -0	f1_is_func-0	Enabled, [Disabled]

VT/ANSI Protocol Options

Note that brackets indicate default settings and values.

VT/ANSI Protocol Options

Options	Session [1], 2, 3, 4	Values
Allow LineMode	allow_linemode	[Enabled], Disabled
Allow Negotiate About Window Size (NAWS)	allow_naws	Enabled, [Disabled]
AnswerBack	answerback	String, minimum length=0, maximum length = 50 [Default string default is computer serial number.]
Any Auto Enter	any_auto_enter	Enabled, [Disabled]
Auto Enter Scan	auto_entr_scn	[Enabled], Disabled
Auto Tab Scan	auto_tab_scan	Enabled, [Disabled]
Auto Wrap	auto_wrap	Enabled, [Disabled]
CR to CRLF	cr_to_crlf	Enabled, [Disabled]
DEL to BS	del_to_bs	Enabled, [Disabled]
Discrete Bells	discrete_bells	Enabled, [Disabled]
Do Gold Key	do_gold_key	[Enabled], Disabled
Do UTF-8	do_utf8	Enabled, [Disabled]
Keypad Mode	keypad_mode	[Numeric], Application
Local Echo	local_echo	Enabled, [Disabled]
Lock Mode	lock_mode	Enabled, [Disabled]
RS-232 BaudRate	rs232_baudrate	[9600], 19200, 38400, 57600, 115200 *(700 Color, CK30, CK31, CK61, CV30, CV60)*
RS-232 DataBits	rs232_databits	7, [8] *(700 Color, CK30, CK31, CK61, CV30, CV60 - software version 8.25 or greater)*
RS-232 Flow	rs232_flow	[NONE], DTR, XON/XOFF *(700 Color, CK30, CK31, CK61, CV30, CV60 - software version 8.25 or greater)*
RS-232 Parity	rs232_parity	[NONE], EVEN, ODD *(700 Color, CK30, CK31, CK61, CV30, CV60 - software version 8.25 or greater)*
RS-232 StopBits	rs232_stopbits	[1], 2 *(700 Color, CK30, CK31, CK61, CV30, CV60 - software version 8.25 or greater)*
Screen Lock	screen_lock	Enabled, [Disabled]
Send XON	send_xon	[Enabled], Disabled

VT/ANSI Protocol Options (continued)

Options	Session [1], 2, 3, 4	Values
Telnet	telnet	Enabled, [Disabled]
Terminal Mode	terminal_mode	[7-Bit], 8-Bit
Terminal Type	terminal_type	String, min = 0, max = 30, [Null string, not defined]
Terminal Setup	term_setup	ANSI, VT100, VT220, VT320, [VT340], IBM 330x
Transmit BS	transmit_bs	Enabled, [Disabled]
Use PC Character Set	use_pc_char_set	Enabled, [Disabled]
User Key Locked	userkey_locked	Enabled, [Disabled]
VT Cursor Mode	vt_cursor_mode	[Cursor], Application
VT220 Mode	vt220_mode	[Char], Block

Generic Protocol Options

Note that brackets indicate default settings and values.

Generic Protocol Options

Options	Session [1], 2, 3, 4	Values
Extended Commands	extended_cmds	Enabled, [Disabled]
Host View Sze	host_view_sze	1-80 *3270, Native* 80,132 *5250* 80 *VT/ANSI*

Display Options

Note that brackets indicate default settings and values.

Display Options

Options	Session [1], 2, 3, 4	Values
Code Page	code_page	[1] = English, 2 = Cyrillic *(Russian)*, 3 = Greek, 4 = Hebrew, 5 = Central Europe, 6 = Latin 2, 7 = Turkish, 8 = Cyrillic Win *(Russian)*
Cursor Mode	cursor_mode	Underln Blink, Block Blink, Underline, [Block]
Font Quality *(CK30, CK31)*	font_quality	[Standard], Clear Type
Func Hotspot	func_hotspot	[Enabled], Disabled
Ignore Bold	ignore_bold	Enabled, [Disabled]
Ignore CnrTaps	ignore_cnrtaps	[Enabled], Disabled
Menu Hotspot	menu_hotsport	[Enabled], Disabled
No Lockdown	no_lockdown	[Enabled]*, [Disabled]**

Display Options (continued)

Options	Session [1], 2, 3, 4	Values
* Enabled is the default for CK30. ** [Disabled] is the default for 700 Color, CK31, CK32, CK61, CN3, CV30, CV60.		
Select Font	select_font	[Lucida Console]*, [Courier New]**, [Courier New Bold]***, Cour. N.B. Italic
* [Lucida Console] is the default for CK30, CK31, CK32, CK61, and CV60. ** [Courier New] is the default for CV30. *** [Courier New Bold] is the default for 700 Color and CN3.		
URL Hotspots	url_hotspots	Enabled, [Disabled] *(software version 8.25 or greater)*

More Options

Note that brackets indicate default settings and values.

More Options

Options	Session [1], 2, 3, 4	Values
Change Menu Password	change_menu_password	String, minimum length = 1, maximum length = 10 [3193693] You must enable and set the password before changing it. Use "Main Menu Password" to enable the password.
Com Select	com_select	COM1 *(default, CV60)*, COM2, COM1 *(default, CV30)*, COM3
Enter Lic Key	enter_lic_key	String, minimum length = 0, maximum length = 16 [null string] *(700 Color)*
Listen Timeout	listen_timeout	Numeric, minimum = 0, maximum = 60000 [120] *(700 Color)*
Main Menu Password	menu_password	Enabled, [Disabled] You must enable and set the password to access the Main Menu. Use "Change menu password" to set the password.
Print Device	print_device	[RS232 Print] (CN3, 700 Series, CV60, CK60, CK30, CK31, CV30) [Bluetooth] (CN3, CK32) IRDA Print (CK61), Network Print
Printer Address	printer_address	String, minimum length=0, maximum length=16 [null string]
Printer Port	printer_port	Numeric, minimum = 0, maximum = 65535 [23]

More Options (continued)

Options	Session [1], 2, 3, 4	Values
Reader Address	reader_address	String, minimum length=0, maximum length=16 [null string] *(700 Color, CK30, CK31, CK61, CV30, CV60)*
RFID Setup	rfid_setup	0 *(default: 700 Color, CV30)*, [1] *(default: CK30, CK31, CK61, CN3, CK32, CV60) software version 8.25 or greater*

LCD Options

Note that brackets indicate default settings and values.

LCD Options

Options	Session [1], 2, 3, 4	Values
Annunciator Format	annun_format	[Vertical]*, [Horizontal]**
* [Vertical] is the default for 700 Color, CK30, CK31, CK61, and CN3. ** [Horizontal] is the default for CV30 and CV60, and CK32.		
Annunciator Position	annun_position	Upper Right, Upper Left, [Lower Right]*, Lower Left, [Stealth]**
* [Lower Right] is the default for 700 Color, CK30, CK31, CK32, CK61, CN3. ** [Stealth] is the default for CV30 and CV60.		
Define Height (Scroll Window)	define_height	Numeric, minimum = 1, maximum = 24 [8]
Define Width (Scroll Window)	define_width	Numeric, minimum = 1, maximum = 80 [8]
Key Uppercase	key_uppercase	[Enabled], Disabled
Screen Columns	screen_cols	Numeric, minimum = 16, maximum = 80 *(700 Color, CK30, CK31, CK32, CK61, and CN3)* or 132 *(CV30 and CV60)* [80]*, [20]**
* [80] is the default for CV60. ** [20] is default for 700 Color, CK30, CK31, CK32, CK61, CN3, and CV30.		
Screen Mode	screen_mode	Center Cursor, [Corner Mode], Page Mode, Lazy Mode, Locked Mode
Screen Rows	screen_rows	Numeric, minimum = 8, maximum = 25 *(700 Color, CK30, CK31, CK32, CK61, and CN3)* or 27 *(CV30 and CV60)* [10]*, [25]**
* [10] is the default for 700 Color, CK30, CK31, CK32, CK61, CN3, and CV30. ** [25] is the default for CV60.		
Scroll Window	scroll_window	Screen Size, Scroll Setting, [Tab Size]

LCD Options (continued)

Options	Session [1], 2, 3, 4	Values
XOrigin	xorigin	Numeric minimum = 0, maximum = 79 [0]
YOrigin	yorigin	Numeric minimum = 0, maximum = 23 [0]

More (Main Menu 2) **Options**

Note that brackets indicate default settings and values.

More *(Main Menu 2)* ***Options***

Options	Session [1], 2, 3, 4	Values
Enable SIP	enable_sip	[0] *for CK30, CK31, CK32, CK61, and CV30*, [1] *for 700 Color, CN3, and CV60*
Note: This enables or disables the interaction between the TE 2000 application and the onscreen keyboard. If this value is 1, the TE 2000 application displays the SIP when it starts and when it gains focus. If this value is 0, the TE 2000 application does not display the SIP.		
Foreground Session	foreground_sess	None
Session Name	session_name	String, minimum length = 0, maximum length = 64 [null string]
Set Hot Key	set_hot_key	F1 *through* F24, [Disabled]
Shift F13-F24	shift_f13_f24	Enabled, [Disabled]
Type-Ahead	type-ahead	[Enabled], Disabled

Additional Parameters

These parameters do not have equivalent TE 2000 configuration menu options. Note that brackets indicate default settings and values.

Additional Parameters

Options	Session [1], 2, 3, 4	Values
Program Name	program_name	String, minimum length = 1, maximum length = 8
(appears on initialization and version screens)		
Program Version	program_version	String, minimum length = 1, maximum length = 4
(appears on initialization and version screens)		
Return Result	return_result	[Enabled], Disabled
Test feature for terminal emulation; if disabled, extended commands (#K only) do not return results to the host application		
Set-Up Parms Menu Password	password	String, minimum length = 1, maximum length = 10 [CR52401]

Changing Text

Modify cfglit.dat strings to change the text in the TE 2000 configuration menus, configuration parameters, or system messages.

An identification (ID) number identifies each literal string. To create your own literal file, you create a text file that associates these numbers with the actual literal strings. You then use makelit.exe to convert the text file to a format the configuration program can use.

Each line in the literal text file begins with the literal ID number. After the ID number, you type the quoted string that is used when that ID number is referenced. If you omit an ID number, its string appears as "Bad Literal File" when you run the program in the computer.

Literal ID numbers are available upon request from Intermec. Contact your Intermec representative for more information about ID numbers.

You can create a sample file containing the default literal strings using makelit.exe to "reverse engineer" the standard cfglit.dat file. To do this, type the following command line to unpack cfglit.dat into a cfglit.txt file:

```
makelit -r cfglit.dat cfglit.txt
```

The cfglit.txt file this command creates contains all the default strings the configuration program uses. One line in cfglit.txt looks like this:

```
0x2f10 "RS232 PORTIN USEPLEASE WAIT!"
```

"0x2f10" is the literal ID number for the RS-232 port-in-use message that appears when a personal computer sends an RS-232 command to the TE program. The message text follows the ID number in a quoted string. The embedded "\n" sequence within the quoted string indicates a "new line" character and outputs a carriage return/line feed. To change the text of the message that appears, change the quoted string.

For example, change the previous line to look like this:

```
0x2f10 "PrintingPlease Wait!"
```

When you have a text file with one line for every ID number, use makelit.exe to convert the file to an indexed literal file. If your text file is named cfglit.txt, you would type the following command which creates the new literal cfglit.dat file. For instructions on how to download the file, see "Downloading Files" on page 191.

```
makelit cfglit.txt cfglit.dat
```

Preinitializing the TE 2000 Program

You can preinitialize the TE 2000 program. You must name the 3270 initialization file as 3270.ini; the 5250 initialization file as 5250.ini; or the VT/ANSI initialization file as VT220.ini. The file is processed when you reset or warm start the computer. The file is processed as if the radio had received the data, and must be in the “on-air” format. For instructions on how to download the file, see “Downloading Files” on page 191.

Data is encoded in binary format. To create the .ini file, you may need a HEX editor or other special program.

Preinitializing the 3270 Program

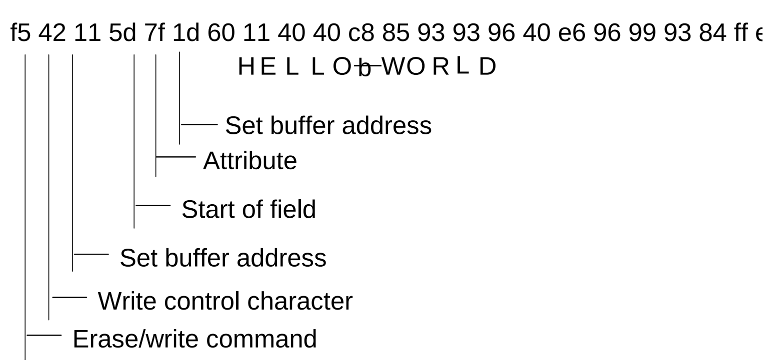
3270.ini contains 3270 data stream commands and orders. Below is the format for the 3270 data stream. The information assumes you have a working knowledge of the data stream command formats or escape sequences, or both. The following 3270 commands are supported:

0xf1	Write	0x7e	Erase write alternate
0xf2	Read buffer	0x6e	Read modified all
0xf5	Erase write	0x6f	Erase all unprotected
0xf6	Read modified		

The following 3270 orders are supported:

0x07	Beep (Intermec extension)	0x13	Insert cursor
0x09	Program tab	0x14	Repeat to address
0x11	Set buffer address	0x1d	Start of field
0x12	Erase unsupported to address	0x1f	Scanner (Intermec extension)

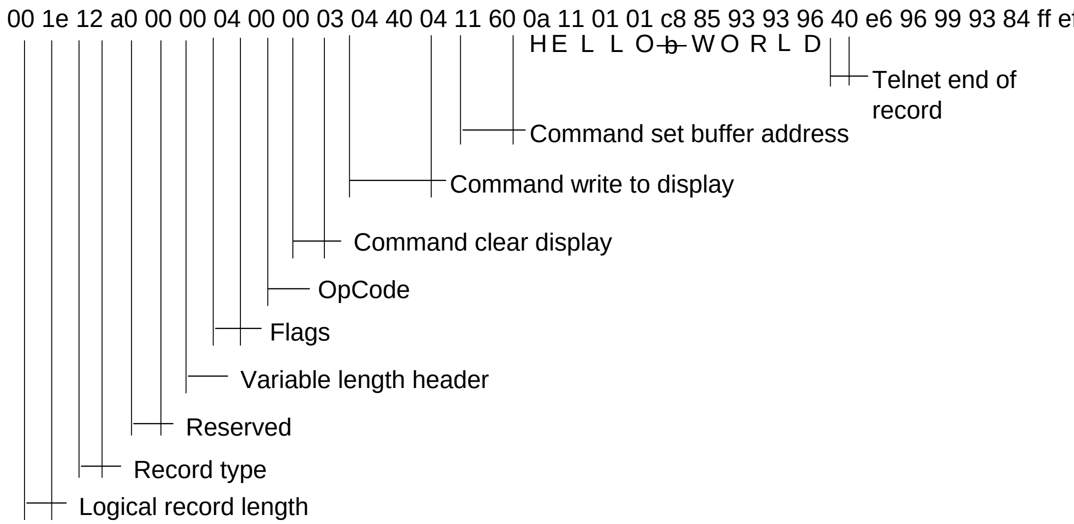
The following example shows how to display “HELLO WORLD” from within a data stream initialization file. The line of hexadecimal digits represent the binary values that must be stored in the initialization files.



Preinitializing the 5250 Program

5250.ini files are composed of the following:

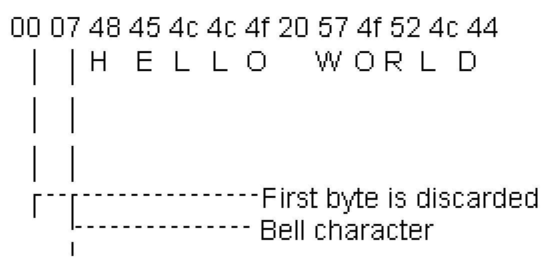
The following example shows how to display “HELLO WORLD” and beep the beeper from within a data stream initialization file. The line of hexadecimal digits represent the binary values that must be stored in the initialization files.



Preinitializing the VT/ANSI Program

The VT220.ini file starts with a single byte that the computer ignores. This byte should always be 0 (zero). The remainder of the file contains standard computer escape sequences.

The following example shows how to display “HELLO WORLD” and beep the beeper from within a data stream initialization file. The line of hexadecimal digits represent the binary values that must be stored in the initialization files.



Remapping the Computer's Keys

Note: These instructions assume the remap.cfg file is downloaded onto your computer. For instructions on how to download the file to the computer, see "Downloading Files" on page 191.

You may need to remap the computer's keys to press a key in TE 2000 that is not on a standard 101-key keyboard. You can also remap a computer key to transmit a text string or message to the personal computer.

To remap the computer keys, you create the remap.cfg file and add a Remap command to remap a computer key. You can remap a single key or a two-key sequence. You can add a Remap command or create a macro in the remap.cfg file that remaps a single key or a two-key sequence.

You can remap any computer key or two-key sequence that does not perform a specific function on the computer. For example, you can remap the [B] key because it only types the lowercase letter B.

Each computer key or two-key sequence generates a 4-digit hexadecimal remap code as listed in the "Key Code Table" on page 183. The key code tables list the 4-digit hexadecimal codes for ASCII characters for the computers. These codes identify the key or keys pressed. For example:

Key	Action	4-Digit Hex Key Code
[B]	Types a lowercase B	0062
[SHIFT] [B]	Types an uppercase B	0042
None	0002	

Remapping a Key or Two-Key Sequence

1. Choose the key or two-key sequence to remap and determine the current 4-digit hexadecimal code of the keys and the code you will enter to remap the keys. For help, see the applicable "Key Code" table.
2. Connect the computer to your personal computer.
3. Using any text editor, enter the keys you want remapped on individual lines in this format: `remap=<key>="string"` or `remap=<key>=<key>`
   - *remap* is the command you enter in remap.cfg.
   - *key* is the 4-digit hexadecimal key or keys to which or from which you are remapping.
   - *string* is the new function for the key or keys, which can be a text string, ASCII mnemonic, or another 2-byte hexadecimal code. Enclose the entire string in quotation marks.
4. Save the new file as remap.cfg.
5. Download remap.cfg to drive C: on your computer.

Example 1

Suppose you want to remap “+” on your computer to send a message and then enter a carriage return. In the remap.cfg file, add this command:

```
remap=<002b>="My battery is low.<CR>"
```

Example 2

To remap the function of the function keys to another key, replace *string* with the transmitted code for the function and replace *key* with the 4-digit hexadecimal key that will do the function. For example, to remap the [F6] function to the “B” key, add this command to the remap.cfg file:

```
remap=<0042>="<ESC>[17~"
```

Example 3

This is an example of multiple hexadecimal codes in the right-most argument which makes a key into a text sequence with embedded EHLLAPI values for the [F4] and [F5] keys.

```
remap=<xxxx>="EHLLAPI value"<1034>"EHLLAPI value
2"<1035>"END
```

Creating a Macro

1 Using any text editor, add the `macro=<key>="string"` Macro command to the end of the remap.cfg file, where:
   - *macro* is the command you enter in remap.cfg.
   - *key* is the 4-digit hexadecimal key or keys you are remapping.
   - *string* is the new action for the key or keys. The string can be a text string, ASCII mnemonic, or another 2-byte hexadecimal code. Enclose the entire string in quotation marks.

2 At the end of the macro, type **`runmacro=<key>`** where *key* is the 4-digit hexadecimal code that identifies the keys that activate the macro.

3 Save the file name as remap.cfg for the macros to work.

4 Append the new remap to the original hex file.

5 Download remap.cfg to the “C:\” drive on your computer.

Example

You can assign “+” to activate a macro that remaps [B] to send the message, “Change the battery pack now.” In the remap.cfg file, add this command:

```
macro=<0042>="Change the battery pack now.<CR>"
runmacro=<002b>
```

Nesting

Macros do not nest. The right-most argument is processed as key strokes and not scanned for macro values. For example:

```
// swap "3" and "5" keys
remap=<0033>="5"      // map "3" key to a "5" key
remap=<0035>="3"      // map "5" key to a "3" key
```

The “3” key produces a “5” key. If nesting was allowed, the “5” key is recognized as a macro that produces the “3” key and the “3” key is recognized as a “5” key that produces the “7” key, and so forth.

Remapping Keys for Each Session

Use the following syntax to remap keys for each session. The session number can be 1, 2, 3, or 4.

```
remap=Session1<keyval>="string"
Macro=Session1<keyval>="string"
Runmacro=Session1<keyval>
```

Note: The string comparison for the “Session” string is case-sensitive.

Key Code Table

Note: Values not listed here may work but are not supported.

```
remap=<0020>="string"   /* SPACE key */
remap=<0021>="string"   /* ! key */
remap=<0022>="string"   /* " key */
remap=<0023>="string"   /* # key */
remap=<0024>="string"   /* $ key */
remap=<0025>="string"   /* % key */
remap=<0026>="string"   /* & key */
remap=<0027>="string"   /* ' key */
remap=<0028>="string"   /* ( key */
remap=<0029>="string"   /* ) key */
remap=<002a>="string"   /* * key */
remap=<002b>="string"   /* + key */
remap=<002c>="string"   /* , key */
remap=<002d>="string"   /* - key */
remap=<002e>="string"   /* . key */
remap=<002f>="string"   /* / key */

remap=<0030>="string"   /* 0 key*/
remap=<0031>="string"   /* 1 key*/
remap=<0032>="string"   /* 2 key*/
remap=<0033>="string"   /* 3 key*/
remap=<0034>="string"   /* 4 key*/
remap=<0035>="string"   /* 5 key*/
remap=<0036>="string"   /* 6 key*/
remap=<0037>="string"   /* 7 key*/
remap=<0038>="string"   /* 8 key*/
remap=<0039>="string"   /* 9 key*/

remap=<003a>="string"   /* : key */
remap=<003b>="string"   /* ; key */
remap=<003c>="string"   /* < key */
remap=<003d>="string"   /* = key */
remap=<003e>="string"   /* > key */
```

```
remap=<003f>="string"    /* ? key */
remap=<0040>="string"    /* @ key */

remap=<0041>="string"    /* A key*/
remap=<0042>="string"    /* B key*/
remap=<0043>="string"    /* C key*/
remap=<0044>="string"    /* D key*/
remap=<0045>="string"    /* E key*/
remap=<0046>="string"    /* F key*/
remap=<0047>="string"    /* G key*/
remap=<0048>="string"    /* H key*/
remap=<0049>="string"    /* I key*/
remap=<004a>="string"    /* J key*/
remap=<004b>="string"    /* K key*/
remap=<004c>="string"    /* L key*/
remap=<004d>="string"    /* M key*/
remap=<004e>="string"    /* N key*/
remap=<004f>="string"    /* O key*/
remap=<0050>="string"    /* P key*/
remap=<0051>="string"    /* Q key*/
remap=<0052>="string"    /* R key*/
remap=<0053>="string"    /* S key*/
remap=<0054>="string"    /* T key*/
remap=<0055>="string"    /* U key*/
remap=<0056>="string"    /* V key*/
remap=<0057>="string"    /* W key*/
remap=<0058>="string"    /* X key*/
remap=<0059>="string"    /* Y key*/
remap=<005a>="string"    /* Z key*/

remap=<005b>="string"    /* [ key */
remap=<005c>="string"    /* key */
remap=<005d>="string"    /* ] key */
remap=<005e>="string"    /* ^ key */
remap=<005f>="string"    /* _ key */
remap=<0060>="string"    /* ` key */

remap=<0061>="string"    /* a key*/
remap=<0062>="string"    /* b key*/
remap=<0063>="string"    /* c key*/
remap=<0064>="string"    /* d key*/
remap=<0065>="string"    /* e key*/
remap=<0066>="string"    /* f key*/
remap=<0067>="string"    /* g key*/
remap=<0068>="string"    /* h key*/
remap=<0069>="string"    /* i key*/
remap=<006a>="string"    /* j key*/
remap=<006b>="string"    /* k key*/
remap=<006c>="string"    /* l key*/
remap=<006d>="string"    /* m key*/
remap=<006e>="string"    /* n key*/
remap=<006f>="string"    /* o key*/
remap=<0070>="string"    /* p key*/
remap=<0071>="string"    /* q key*/
remap=<0072>="string"    /* r key*/
remap=<0073>="string"    /* s key*/
remap=<0074>="string"    /* t key*/
remap=<0075>="string"    /* u key*/
remap=<0076>="string"    /* v key*/
remap=<0077>="string"    /* w key*/
```

```
remap=<0078>="string"   /* x key*/
remap=<0079>="string"   /* y key*/
remap=<007a>="string"   /* z key*/

remap=<007b>="string"   /* { key */
remap=<007c>="string"   /* | key */
remap=<007d>="string"   /* } key */
remap=<007e>="string"   /* ~ key */
remap=<007f>="string"   /* Del key */

remap=<1030>="string"   /* Home key */
remap=<1031>="string"   /* F1 key */
remap=<1032>="string"   /* F2 key */
remap=<1033>="string"   /* F3 key */
remap=<1034>="string"   /* F4 key */
remap=<1035>="string"   /* F5 key */
remap=<1036>="string"   /* F6 key */
remap=<1037>="string"   /* F7 key */
remap=<1038>="string"   /* F8 key */
remap=<1039>="string"   /* F9 key */
remap=<103c>="string"   /* Backspace key */

remap=<1061>="string"   /* F10 key */
remap=<1062>="string"   /* F11 key */
remap=<1063>="string"   /* F12 key */
remap=<1064>="string"   /* F13 key */
remap=<1065>="string"   /* F14 key */
remap=<1066>="string"   /* F15 key */
remap=<1067>="string"   /* F16 key */
remap=<1068>="string"   /* F17 key */
remap=<1069>="string"   /* F18 key */
remap=<106a>="string"   /* F19 key */
remap=<106b>="string"   /* F20 key */
remap=<106c>="string"   /* F21 key */
remap=<106d>="string"   /* F22 key */
remap=<106e>="string"   /* F23 key */
remap=<106f>="string"   /* F24 key */
```

3270 Key Codes

```
remap=<1042>="string"   /* Back Tab key */
remap=<1043>="string"   /* Clear key */
remap=<1044>="string"   /* Del key */
remap=<1045>="string"   /* Enter key */
remap=<1046>="string"   /* Erase End of Field (EOF) */
remap=<1049>="string"   /* Insert key */

remap=<104c>="string"   /* Window/viewport left key */
remap=<104e>="string"   /* New line */
remap=<1052>="string"   /* Reset key */
remap=<1054>="string"   /* Tab key */
remap=<1055>="string"   /* Window/viewport up key */
remap=<1056>="string"   /* Window/viewport down key */
remap=<105a>="string"   /* Window/viewport right key */

remap=<1078>="string"   /* PA1 */
remap=<1079>="string"   /* PA2 */
remap=<107a>="string"   /* PA3 */

remap=<2041>="string"   /* Auto-Login Restart key */
remap=<206c>="string"   /* Menu key */
```

```
remap=<304c>="string"   /* Page left key */
remap=<3055>="string"   /* Page up key */
remap=<3056>="string"   /* Page down key */
remap=<305a>="string"   /* Page right key */
```

5250 Key Codes

```
remap=<1042>="string"   /* Back Tab key */
remap=<1043>="string"   /* Clear key */
remap=<1044>="string"   /* Del key */
remap=<1045>="string"   /* Enter key */
remap=<1048>= "string"  /* Help key */
remap=<1049>="string"   /* Insert key */
remap=<104c>="string"   /* Window/viewport left key */
remap=<104e>="string"   /* New Line key */

remap=<1050>="string"   /* Print key */
remap=<1052>="string"   /* Reset key */
remap=<1054>="string"   /* Tab key */
remap=<1055>="string"   /* Window/viewport up key */
remap=<1056>="string"   /* Window/viewport down key */
remap=<105a>="string"   /* Window/viewport right key */

remap=<1075>="string"   /* Roll Up key */
remap=<1076>="string"   /* Roll Down key */

remap=<2041>="string"   /* Auto-Login Restart key */
remap=<206c>="string"   /* Menu key */
remap=<302d>="string"   /* Field minus key */
remap=<302b>="string"   /* Field plus key */

remap=<3045>="string"   /* Field Exit key */
remap=<3046>="string"   /* Erase Input key */
remap=<3048>="string"   /* System request key */
remap=<304c>="string"   /* Page left key */

remap=<3051>="string"   /* Attention key */
remap=<3055>="string"   /* Page up key */
remap=<3056>="string"   /* Page down key */
remap=<3057>="string"   /* Field mark key */
remap=<3058>="string"   /* HEX key */
remap=<305a>="string"   /* Page right key */

remap=<4044>="string"   /*Duplicate key */
```

Native Key Codes

```
remap=<1042>="string"   /* Back Tab key */
remap=<1043>="string"   /* Clear key */
remap=<1044>="string"   /* Del key */
remap=<1045>="string"   /* Enter key */
remap=<1046>="string"   /* Erase End of Field (EOF) */
remap=<1049>="string"   /* Insert key */

remap=<104e>="string"   /* New line */
remap=<1052>="string"   /* Reset key */
remap=<105a>="string"   /* Tab key */

remap=<1061>="string"   /* F10 key */
remap=<1062>="string"   /* F11 key */
remap=<1063>="string"   /* F12 key */
remap=<1064>="string"   /* F13 key */
remap=<1065>="string"   /* F14 key */
```

```
remap=<1066>="string"   /* F15 key */
remap=<1067>="string"   /* F16 key */
remap=<1068>="string"   /* F17 key */
remap=<1069>="string"   /* F18 key */
remap=<106a>="string"   /* F19 key */
remap=<106b>="string"   /* F20 key */
remap=<106c>="string"   /* F21 key */
remap=<106d>="string"   /* F22 key */
remap=<106e>="string"   /* F23 key */
remap=<106f>="string"   /* F24 key */

remap=<2041>="string"   /* Auto-Login Restart key */
remap=<206c>="string"   /* Menu key */
```

VT/ANSI Key Codes

```
remap=<1054>="string"   /* Tab key */
remap=<0009>="string"   /* Tab key */
remap=<1042>="string"   /* Back Tab key */
remap=<1044>="string"   /* Del key */
remap=<1045>="string"   /* Enter key */
remap=<1049>="string"   /* Insert key */
remap=<104c>="string"   /* Left key */
remap=<1055>="string"   /* Up key */
remap=<1056>="string"   /* Down key */
remap=<105a>="string"   /* Right key */

remap=<1075>="string"   /* Page up key */
remap=<1076>="string"   /* Page down key */

remap=<2041>="string"   /* Auto-Login Restart key */
remap=<206c>="string"   /* Menu key */

remap=<304c>="string"   /* Window/viewport left key */
remap=<3055>="string"   /* Window/viewport up key */
remap=<3056>="string"   /* Window/viewport down key */
remap=<305a>="string"   /* Window/viewport right key */

remap=<3061>="string"   /* Find key */
remap=<3062>="string"   /* Insert here key */
remap=<3063>="string"   /* Remove key */
remap=<3064>="string"   /* Select key */
remap=<3065>="string"   /* Previous screen key */
remap=<3066>="string"   /* Next screen key */

remap=<3067>="string"   /* Keypad key */
remap=<3068>="string"   /* Keypad Enter key */
remap=<3069>="string"   /* Keypad 0 key */
remap=<306a>="string"   /* Keypad 1 key */
remap=<306b>="string"   /* Keypad 2 key*/
remap=<306c>="string"   /* Keypad 3 key*/
remap=<306d>="string"   /* Keypad 4 key*/
remap=<306e>="string"   /* Keypad 5 key*/
remap=<306f>="string"   /* Keypad 6 key*/
remap=<3070>="string"   /* Keypad 7 key*/
remap=<3071>="string"   /* Keypad 8 key*/
remap=<3072>="string"   /* Keypad 9 key*/
remap=<3073>="string"   /* Keypad . key*/

remap=<3075>="string"   /* Page left key */
remap=<3076>="string"   /* Page right key */
```

```
remap=<3077>="string"    /* Keypad - key*/
remap=<3078>="string"    /* Keypad + key*/
```

Remapping Characters

You can use display character translation files to remap characters as they are written to the display. The translation file name for TE 2000 terminal emulation must be 3270.xlt for 3270, 5250.xlt for 5250, VT220.xlt for VT/ANSI, or Native.xlt for Native. For instructions on how to download the file to the computer, see "Downloading Files" on page 191.

Display character translation files are binary files consisting of ordered pairs of eight bit values. Each pair of values remaps a displayable character to a different displayable character.

- The first byte of a pair is the ASCII value of the character to replace.
- The second byte of a pair is the ASCII value that replaces the first.

These translations are made when a character is written to a display device. If the character is sent to the host (keystroke, scan data) or sent to an external device (printer), it is sent as the original, untranslated value.

Suppose you want a computer running terminal emulation to replace the uppercase B with the Greek letter beta, and replace the uppercase Z with the Greek letter omega. Create an .xlt file that is four bytes long (two ordered pairs of two bytes each). The file should contain the 0x42, 0xE1, 0x5A, and 0xEA bytes in this order. These represent the ASCII display character set values for B, beta, Z, and omega, respectively.

ASCII Equivalents for EBCDIC Values *(3270, 5250)*

	4A	4F	5A	5B	5F	6A	79	7B	7C	7F	A1	C0	D0	E0
English (US)	9B	0E	21	24	AA	7C	60	23	40	22	7E	7B	7D	5C
German	8E	21	9A	24	5E	94	60	23	15	22	E1	84	81	99
Danish/ Norwegian	23	21	0F	8F	5E	ED	60	92	05	22	81	91	86	5C
Finnish/ Swedish	15	21	0F	8F	5E	7C	82	8E	99	22	81	84	86	90
Italian	F8	21	82	24	5E	95	97	9C	15	22	8D	85	8A	87
Spanish	5B	OE	5D	–	AA	A4	60	A5	40	22	06	7B	7D	5C
French	F8	21	15	24	5E	97	60	9C	85	22	06	82	8A	87
Belgian	5B	21	50	24	5E	97	60	23	85	22	06	82	8A	87
English (UK)	24	OE	21	9C	AA	7C	60	23	40	22	5F	7B	7D	5C

IBM Character Sets

Number	Language	IBM Set
1	English (U.S.)	037-850
2	German	273-850

IBM Character Sets

Number	Language	IBM Set
3	Danish/Norwegian	277-850
4	Swedish/Finnish	278-850
5	Italian	280-850
6	Spanish	284-850
7	French	297-850
8	Belgian	500-850

Example

This example remaps 14 characters appropriate to U.S. English to 14 characters more appropriate to Austrian/German. ASCII hexadecimal file 5250.XLT remaps for German.

0x9B 0x8E
0x0E 0x21
0x21 0x9A
0x24 0x24
0xAA 0x5E
0x7C 0x94
0x60 0x60
0x23 0x23
0x40 0x15
0x22 0x22
0x7E 0xE1
0x7B 0x84
0x7D 0x81
0x5C 0x99

EBCDIC Hex Value	Original to be Replaced	Replacement for Display
4A	¢	Ä
4F	|	!
5A	!	Ü
5B	$	$
5F	¬	^
6A	¦	ö
79	`	`
7B	#	#
7C	@	§
7F	"	"
A1	~	β
C0	{	ä
D0	}	ü
E0	\	Ö

Implementing ITCColor.dat Attribute Colors

You must name the file ITCColor.dat and you may place this where the other TE 2000 configuration files are placed for discovery by the TE 2000 application.

The two registry keys for the default foreground (Text) and background colors are still used and are set *before* the color file is read from these keys:

```
HKEY_LOCAL_MACHINEOFTWAREntermecE2000ackGroundColor
```

```
HKEY_LOCAL_MACHINEOFTWAREntermecE2000oreGroundColor
```

Both keys must exist or the default colors of black background and white text is used.

The color file contains lines defining the color Index and the Color for eight normal foreground (Text) and background (Back) color pairs and eight inverse color pairs for a total of 32 colors. Each line is a maximum of 80 characters. A line can be empty, have leading spaces, have a comment indicated by a semicolon character, have a pair of values (color Index and Color). All characters from a semicolon to the end of the line are ignored. A line is terminated by a carriage return, line feed character, or both. Invalid lines are ignored. You may define all, none, or any of the colors in the file in any order. The file may exist or not. The normal Text colors and the Inverse Text color are defaulted to black. If an Index-Color is not defined or the file does not exist, default colors are used.

The color Index is a decimal value of 0 through 31. It specifies the character attributes associated with Color. The index values are documented in the sample ITCColor.dat file on the next page.

Color is a 32-bit hex value used to specify an RGB color (0x00bbggrr). In RGB format, the low-order (rr) byte contains a value for the relative intensity of red; the second byte (gg) contains a value for green; and the third byte (bb) contains a value for blue. The high-order byte must be zero. The maximum value for a single byte is 0xFF.

Sample Color File

```
Column   1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890
======================================================================
  ; comment
; Normal
00 0x00000000   ;Text
01 0x007f7f7f   ;Back
02 0x000000ff   ;Text   Bold
03 0x007f7f00   ;Back   Bold
04 0x0000ff00   ;Text   Blink
05 0x007f007f   ;Back   Blink
06 0x0000ffff   ;Text   Bold    Blink
07 0x007f0000   ;Back   Bold    Blink
08 0x00ff0000   ;Text   Underline
09 0x00007f7f   ;Back   Underline
10 0x00ff00ff   ;Text   Bold    Underline
11 0x00007f00   ;Back   Bold    Underline
12 0x00ffff00   ;Text   Blink   Underline
13 0x0000007f   ;Back   Blink   Underline
14 0x00ffffff   ;Text   Bold    Blink   Underline
15 0x00000000   ;Back   Bold    Blink   Underline

; Inverse
16 0x00000000   ;Text
17 0x00ffffff   ;Back
18 0x0000007f   ;Text   Bold
19 0x00ffff00   ;Back   Bold
20 0x00007f00   ;Text   Blink
21 0x00ff00ff   ;Back   Blink
22 0x00007f7f   ;Text   Bold    Blink
23 0x00ff0000   ;Back   Bold    Blink
24 0x007f0000   ;Text   Underline
25 0x0000ffff   ;Back   Underline
26 0x007f007f   ;Text   Bold    Underline
27 0x0000ff00   ;Back   Bold    Underline
28 0x007f7f00   ;Text   Blink   Underline
29 0x000000ff   ;Back   Blink   Underline
30 0x007f7f7f   ;Text   Bold    Blink   Underline
31 0x007ffff7   ;Back   Bold    Blink   Underline
```

Downloading Files

From your web browser, use the *ftp://<IP address>* URL to download files. For the 700 Color and CN3, use the Microsoft ActiveSync application as described in “Loading the Auto-Login Script File” on page 158.

Customizing 5250 EBCDIC to ASCII Translation

The 5250 data stream translates all data from the host from 8-bit EBCDIC to 8-bit ASCII for processing in the computer. Before the data is sent back to the host, it is again translated from ASCII to EBCDIC. The default translation is shown on the next page.

You can customize the operation of the 5250 data stream by changing the default EBCDIC to ASCII translation table. You can replace the default table with one that is combined with the HEX file that you download to the computer. You can use ASEBTBLD.exe to create the file. You must name it ASCEBD.tbl.

Type **`asebtbld`** to display this information:

```
ASEBTBLD ASCII-EBCDIC Translation Table Creation.
$Revision: 1.0  $
$Date: 03 Apr 1998 13:46:14 $
Copyright 1995, Norand Corporation.
Usage: ASEBTBLD [<options>] <commands> <fname>
  <options>:
    -r<file>     Input file containing replacement table type,
                 0x00-0xff table index, 0x00-0xff value.
    -v           Verbose * display processing steps.
  <fname>        Output file name, extension ignored.
```

ASEBTBLD creates <*fname*>.TBL from the default ASCII and EBCDIC tables using replacement values specified in -r<*file*>. Type the following command to create the ASCEBD.tbl file, with the replacement values specified in changes.my.

`asebtbld -rchanges.my ascebd`

The replacement file is an ASCII text file formatted as follows:

```
-----------------top of replacement file--------------------
/* Any line beginning with `/*' in column 1 is a comment.
/*   A=ASCII=>
/*       EBCDIC
/* E=EBCDIC=>     0-based       hex
/*     ASCII       index value   Anything after value is a comment
/*   -------       -------       ---------
     A     0x30     0xf0     ASCII   `0' returns EBCDIC `0'
     E     0xf0     0x30     EBCDIC   `0' returns ASCII `0'
     E     0xc9     0x3f     Unknown     EBCDIC 0xc9 returns ASCII `?'
/* Any number of blanks, tabs allowed before, between and
/* after values.
/* blank lines allowed
-----------------bottom of replacement file-------------------
```

ASEBTBLD creates the default tables starting on the next page if there is no replacement file, or an empty replacement file, is specified.

Displaying EBCDIC Non-English Code Pages

If you have selected a code page other than the default English, you need to perform these steps to ensure the associated EBCDIC is displayed correctly.

To display non-English code pages

1. Find the appropriate EBCDIC code and the ASCII code page on the web in places like the Microsoft web site at: www.microsoft.com.
2. Check the appropriate tables to cross-check whether there is a default translation in the EBCDIC to ASCII table.
3. If there is a translation, verify whether you have the correct ASCII character for the code page you had selected.
4. If there is no translation, add an entry to the changes.my file.
5. Do this process for every character that needs translation.
6. When all of the characters are translated, save the changes.my file.
7. Run the asebtbld application as described on the previous page.

Example

If you had selected the Turkish code page (ASCII code page 1026) and you want to display the EBCDIC code page 1254 for Turkey, look up character 0x42. This character shows that the default EBCDIC to ASCII translation is 0x00 (no default translation). Searching the ASCII code page 1026 reveals the 0xe2 character, which you add to the changes.my file as follows.

```
E 0x42 0xe2
```

Note: The tables used in this example are on the next page.

Code Page 01026 HEX Digits

1st > 2nd v	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP)	&	-	ø	Ø	°		¢	ç		ü	0
-1	(RSP)	é	/	É	a	j	ö	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	§	E	N	U	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8	{	ì	[	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	ß	Ñ		i	r	z	¾	I	R	Z	9
-A	Ç			:	«	ä	¡	¬	-	1	2	3
-B	.	|	,	Ö	»	ö	¿	|	ô	û	Ô	Û
-C	<	*	%		}	æ	]	-	~	\	#	¨
-D	(	)	_	·	`	‚	$	¨	ò	ù	Ò	Ù
-E	+	;	>	=	|	Æ	@	'	ó	ú	Ó	Ú
-F	!	^	?	Ü	±	¤	®	×	õ	ÿ	Ô	(EO)

Code Page 01254 Windows Turkish HEX Digits

1st> 2nd v	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0			(SP)	0	@	P	`	p	_		(RSP)	°	À		à	
-1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
-2			¨	2	B	R	b	r	‚	'	¢	2	Â	Ò	â	ô
-3			#	3	C	S	c	s		«	£	3	Ã	Ó	ã	ó
-4			$	4	D	T	d	t	"	"	¤	'	Ä	Ô	ä	ô
-5			%	5	E	U	e	u	…	²	¥		Å	Õ	å	õ
-6			&	6	F	V	f	v		-	|	¶	Æ	Ö	æ	ö
-7				7	G	W	g	w		–	§	·	Ç	×	ç	÷
-8			(	8	H	X	h	x	^	~	¨	,	È	Ø	è	ø
-9			)	9	I	Y	i	y		t	E	1	É	Ù	é	ù
-A			*	:	J	Z	j	z					Ê	Ú	ê	ú
-B			+	;	K	[	k	{	<	>	«	»	Ë	Û	ë	û
-C			,	<	L		1	|			¬	¼	Ì	Û	ì	û
-D			-	=	M	]	m	}			-	½	Í	Ü	í	ü
-E			.	>	N	^	n	~			®	¾	Î		î	
-F			/	?	O	_	o				-	¿	Ï	ß	ï	ÿ

ASCII to EBCDIC

0x00

0x00	0x01	0x02	0x03	0x37	0x2d	0x2e	0x2f
0x16	0x05	0x25	0x0b	0x0c	0x0d	0x4f	0x0f
0x10	0x11	0x12	0x13	0x3c	0x00	0x32	0x1c
0x18	0x19	0x3f	0x27	0x22	0x00	0x35	0x00

0x20

0x40	0x5a	0x7f	0x7b	0x5b	0x6c	0x50	0x7d
0x4d	0x5d	0x5c	0x4e	0x6b	0x60	0x4b	0x61
0xf0	0xf1	0xf2	0xf3	0xf4	0xf5	0xf6	0xf7
0xf8	0xf9	0x7a	0x5e	0x4c	0x7e	0x6e	0x6f

0x40

0x7c	0xc1	0xc2	0xc3	0xc4	0xc5	0xc6	0xc7
0xc8	0xc9	0xd1	0xd2	0xd3	0xd4	0xd5	0xd6
0xd7	0xd8	0xd9	0xe2	0xe3	0xe4	0xe5	0xe6
0xe7	0xe8	0xe9	0xad	0xe0	0xbd	0x6a	0x6d

0x60

0x79	0x81	0x82	0x83	0x84	0x85	0x86	0x87
0x88	0x89	0x91	0x92	0x93	0x94	0x95	0x96
0x97	0x98	0x99	0xa2	0xa3	0xa4	0xa5	0xa6
0xa7	0xa8	0xa9	0xc0	0x6a	0xd0	0xa1	0x07

0x80

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x4a	0x00	0x00	0x00	0x00

0xa0

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x5f	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0xad	0x9d	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0xbc

0xc0

0xab	0x8c	0x8b	0x8d	0xbf	0x8e	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0xbb	0xac	0x00	0x00	0x00	0x00	0x00

0xe0

0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

EBCDIC to ASCII

0x00

0x00	0x01	0x02	0x03	0x00	0x09	0x00	0x7F
0x00	0x00	0x00	0x0B	0x0C	0x0D	0x0E	0x0F
0x10	0x11	0x12	0x13	0x00	0x00	0x08	0x00
0x18	0x19	0x00	0x00	A_DUP	0x00	0x00	0x00

0x20

0x00	0x00	0x1C	0x00	0x00	0x0A	0x17	0x1B
0x00	0x00	0x00	0x00	0x00	0x05	0x06	0x07
0x00	0x00	0x16	0x00	0x00	0x1E	0x00	0x04
0x00	0x00	0x00	0x00	0x14	0x16	0x00	0x1A

0x40

0x20	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x9b	0x2E	0x3C	0x28	0x2B	0x0e
0x26	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x21	0x24	0x2A	0x29	0x3B	0xaa

0x60

0x2D	0x2F	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x7C	0x2C	0x25	0x5F	0x3E	0x3F
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x60	0x3A	0x23	0x40	0x27	0x3D	0x22

0x80

0x00	0x61	0x62	0x63	0x64	0x65	0x66	0x67
0x68	0x69	0x00	0xc2	0xc1	0xc3	0xc5	0x00
0x00	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F	0x70
0x71	0x72	0x00	0x00	0x00	0xb4	0x00	0x00

0xa0

0x00	0x7E	0x73	0x74	0x75	0x76	0x77	0x78
0x79	0x7A	0x00	0xc0	0xda	0xb3	0x00	0x00
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0x00	0x00	0x00	0xd9	0xbf	0x00	0x00	0xc4

0xc0

0x7B	0x41	0x42	0x43	0x44	0x45	0x46	0x47
0x48	0x49	0x00	0x00	0x00	0x00	0x00	0x00
0x7D	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F	0x50
0x51	0x52	0x00	0x00	0x00	0x00	0x00	0x00

0xe0

0x5c	0x00	0x53	0x54	0x55	0x56	0x57	0x58
0x59	0x5A	0x00	0x00	0x00	0x00	0x00	0x00
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
0x38	0x39	0x00	0x00	x00	0x00	0x00	0x00

6 Programming

This chapter contains programming information for 3270, 5250, Native, and VT/ANSI computers.

- 3270 data stream commands
- 5250 display data stream commands
- Native programming
- VT/ANSI programming

3270 Data Stream Commands and Orders

This chapter describes 3270 SNA/SDLC and asynchronous display data stream commands and orders that the host application can send to the computers. Orders govern format and data expectations within a data stream command. Also included in this chapter is information on formatting data to be displayed on the computers.

This chapter is not intended as a specific guide for programming, but a comparison with your existing system. The information helps identify differences between your screen-generating utilities and features supported by the computers.

Screen Design Aid

This chapter, along with your Screen Design Aid (SDA) documentation, can help you resolve occasional questions concerning discrepancies that may arise regarding attribute selection and the capabilities of the computers. This chapter includes information you may not need, because you can use your SDA to do much, if not all, of the programming to handle commands and orders for the network.

This information is provided to assure you have the most accurate and detailed information available and to describe variations from the 3270 data stream where they occur.

If you are comfortable using the SDA on your host computer, you can flip through this chapter to find out which commands and orders the computers support. You can then read Chapter 7, **"Extended Commands" on page 343**, which describes the extended commands you can use to perform operations beyond the typical capability of the 3278 Model 2 Computer.

Applications

If you need to write applications specifically for the computer, it is recommended that you use the corner mode window onto the 3278 Model 2 Computer and write programs using the portion of the display available without moving the window. This is the most efficient method for developing custom applications for the computer.

3270 Data Stream Commands

The 3270 data stream consists of application data, commands and structured field functions, and orders which are transmitted between the controller and the host system. The computers support a subset of the IBM 3270 command structure to display and accept keyboard or scanner input.

The following chart lists the SNA/SDLC and asynchronous commands that the computers support.

Command	SNA/SDLC (Hex) Syntax	Asynchronous Syntax
Erase All Unprotected	6F	?
Erase/Write	F5	5
Erase/Write Alternate	7E	=
Read Buffer	F2	2
Read Modified	F6	6
Read Modified All	6E	>
Reset Computer	Not applicable	G
System Services Control Point		4
Write	F1	1

Erase All Unprotected Command

The **Erase All Unprotected** command performs the following functions at the selected computers:

- Clears unprotected buffer locations to nulls.
- Resets to 0 the MDT bit for each unprotected field.
- Positions cursor address to the first location of the buffer.

Erase/Write Command

The **Erase/Write** command performs both an erase operation and a write operation, as follows:

1. The erase operation clears the device buffer to nulls, positions the cursor address at the first location in the buffer, and resets the buffer address to its first location.
2. The **Erase/Write** command performs the write and write control character (WCC) operations in the same manner as the **Write** command. If no WCC is sent, the **Erase/Write** command will not erase the buffer.

Erase/Write Alternate Command

The **Erase/Write Alternate** command performs the same function in the computer as the **Erase/Write** command, and allows both the erase and write operations to take place.

Read Buffer (RB) Command

This command causes buffer data from the computer to transmit to the main storage in the central processing unit. All buffer data and nulls from the beginning to the end of the buffer location are included in the transfer. The default setting for the beginning of buffer data transfer is 0. If the **RB** command is chained from the **Write** or **Erase/Write** command, data transfer begins from the current buffer address with all nulls suppressed.

The **RB** response starts with a 3-byte heading that includes the AID character and a 2-character cursor address. The controller inserts a **Start Field** order to identify the beginning of each field. Alphanumeric data for each field then follows the **RB** command header.

The **RB** command data stream uses the following format:

```
<AID code> <cursor address> <SF order> <attribute byte> <first data field>
<second data field>
```

Note: Response times increase for the **Read Buffer** command because of the large quantity of data processed during the read buffer operation.

Read Modified Command

A major feature of the **Read Modified** command is null suppression. During read modified command operations, null codes are not sent.

During a **Read Modified** command, all modified fields are transferred to the program. All nulls are suppressed during data transfer and thus are not included in the read data stream. As a field is modified by the operator, the MDT bit is set in the attribute byte for that field. Then, when a read-modified operation is performed, successive attribute bytes are examined for a set MDT bit. When the bit is found, the data in the associated field is read (with nulls suppressed) before the next attribute byte is examined. If the screen is unformatted the entire screen (nulls suppressed) is sent.

Read Modified All Command

Read Modified All is the same as the **Read Modified** command.

Reset Computer

This command is an asynchronous command that returns the computer to its first "state," a known condition. All computers are reset when "t" is omitted. Computers respond to a poll with their power-up message.

Function	Syntax	Controller	R Response
Resets computer(s)	Gt<cr> [t only] G<cr> [all]	OK<cr>	tP,x<cr>

Variables

- t = Computer number (network address)
- x= Computer type code (071 for 3270)

Example

Command: G002<cr>
Response: Controller resets computer 002

System Services Control Point (SSCP) Command

The **SSCP** command reads in new data from the host and places it at the current cursor position. If a 0x15 new line character is in the data, the computer performs a newline function. The **SSCP** command has no orders or WCC character associated with it and generates an unformatted screen.

Write Command

A **Write** command from the host computer consists of a command code, a WCC, and any orders or new buffer data (or both) required to modify the existing buffer contents. Execution of the **Write** command is in two steps: 1) The WCC is processed and 2) Buffer data and orders are processed.

Data may be any ASCII character in the range of 0x20 to 0x7E, and must define the character content of a field.

Buffer data characters may be written into any buffer location without erasing or modifying data in other locations. Data is stored in successive buffer locations until orders alter buffer addresses, or all data is entered.

During the write operation, the buffer address advances one location as each character is stored. If the SBA order does not immediately follow the WCC command, computers will write data from the first buffer location.

Write Control Character

Bits 0, 1, and 2 of the WCC cause action in the computer. The following describes the format of the WCC byte.

WCC Byte Format

Bit	Format	Description
0	Reset MDT bits	When set to 1, all MDT bits in the selected buffer are reset before any data is written or orders are executed.
1	Keyboard restore bit	When set to 1, the computer allows input. Stations lock out all data when sending data to the host and does not accept input until receiving a command with this bit set.
2	Sound alarm bit	When set to 1, selected computers emit three short beeps after operation.
3-7		Unused (“do not care”) bits.

Note: Bit numbers are in the opposite order of those described in the *IBM 3270 Information Display System Component Description* manual.

The computer ignores all WCC bits defined as “unused.” Only the three least significant bits of the WCC byte cause action in the computer. These are converted to ASCII characters. Appendix B, “Tables,” contains a binary to ASCII conversion table.

Orders and buffer data follow the WCC. Data characters are any of the displayable ASCII characters in the range hexadecimal 20 through 7E. They define the character content of a protected or unprotected field.

Buffer data characters may be written into any specific location of the buffer without erasing or modifying the data in other buffer locations. Data characters are stored in successive buffer locations until an order alters the buffer address, or until all data is entered.

During the write operation, the buffer address is advanced one location as each character is stored. If the Set Buffer Address order does not immediately follow WCC, the computer writes data from the first buffer location.

3270 Data Stream Format

The 3270 data stream computer command format (WtDxyz) is an asynchronous format that an asynchronous host uses to send data to the computer.

Syntax	Controller Response	R Response
WtDxyz<cr>	None	tKas{Atu.../Atu}<c r>

Variables

- t = Computer number
- x = 3270 data stream command:

  ? Erase All Unprotected
  5 Erase/Write
  = Erase Write Alternate
  2 Read Buffer
  6 Read Modified
  > Read modified all
  G Reset computer
  4 SSCP
  1 Write

- y =3270 write control character (WCC)

  Bit 0 Reset MDT bits
  Bit 1 Keyboard restore
  Bit 2 Sound alarm

- z = Computer orders or ≤1024 characters
- a = AID key code
- s = 4-byte right-justified, zero-filled address of the current cursor position when the AID key was pressed (values 0-1919)
- /At = Set Buffer order followed by a 4-byte right-justified, zero-filled address of the field from where the data is returned
- u = Data entered or modified for this field
- {} = Optional

Example

Command:
```
W000D5G/A0000/F'ID:/F@%/A0017/F'/A0081/
DESCRIPTION:/A0160/F@%/A0177/F'/A0241/
QUANTITY:/F@&/A0253/F'<cr>
```

This command is for a computer in the window and 16x9 display mode. The command displays the following on the computer:

```
ID:%
DESCRIPTION:
%
QUANTITY:&
```

If the user enters “210458236” for the ID field, “PEACH CRATES” for the description field, and “85” for the quantity field, this screen displays:

```
ID:210458236
DESCRIPTION:
PEACH CRATES
QUANTITY: 85
```

When the computer operator presses [ENTER], the host receives the following response:

```
000K'0252/A0004210458236/A0161PEACH CRATES/A025085
```

Orders

Computer orders may be included in each 3270 data stream command, either alone or intermixed with display data. Orders are executed sequentially only after the complete data stream has been received by the computer and checked for errors. Orders are not stored in the display buffer. For the asynchronous data stream, they are added to the data stream following a “/” character.

Buffer control orders position, define, and format data being written into the buffer. They also erase selected unprotected data in the buffer and reposition the cursor address. The following table lists the orders.

3170 Data Stream Orders

	Code				
Order	**SNA/ SDLC**	**Async.***	**Byte 2**	**Byte 3**	**Byte 4**
EUA	12	/Exxxx	1st address byte	2nd address byte	Not used
IC	13	/C	Not used	Not used	Not used
PT	05	/P	Not used	Not used	Not used
RA	3C	/Rxxxxy	1st address byte	2nd address byte	Character to be repeated
SCN	1F	N/A	Bar code control byte	1st option byte	2nd option byte
SBA	11	/Axxxx	1st address byte	2nd address byte	Not used
SF	1D	/Fa	Attribute character	Not used	Not used
* xxxx = a 4-byte buffer address, 0 to 1919, right-adjusted, zero-filled; y = character to be repeated; ttt = scanner order designator; a = an attribute byte.					

The following topics describe buffer control orders executed sequentially at the selected computer after the station received the complete data stream and checked data integrity. These orders are not stored in the display buffer.

Most of the buffer control orders position, define, and format data written into the buffer, erase selected unprotected data in the buffer, and reposition the cursor address. The Scanner order is an extension of the 3270 data stream and is not supported by the 3278 Model 2.

EUA Order

The Erase Unprotected to Address (EUA) order inserts nulls in all unprotected buffer locations, starting at the current buffer address and ending at (but not including) the specified stop address. For the asynchronous data stream, the stop address is specified by address bytes "xxxx" which immediately follow the EUA order in the write data stream. For SNA/SDLC (non-Telnet), the stop address is specified by the two address bytes which immediately follow the EUA in the write data stream.

If an invalid stop address is specified (that is, an address outside the range of the computer's display buffer) the following occur in random order:

- The write operation is terminated at this point.
- The character is not stored.
- The computer recognizes this as an error condition and performs a clear memory operation.
- The display buffer is set to nulls.
- A clear AID character is transmitted to the computer control unit for forwarding to the host computer.

When the stop address is lower than the current buffer address, the EUA order wraps from the end of the buffer to the beginning of the buffer and continues. When the stop address equals the current buffer address, all unprotected character locations in the buffer are erased (set to nulls).

Insert Cursor (IC) Order

The IC order changes the stored cursor address to the location specified by the current buffer address. The current buffer address does not change.

For example, if the IC order is issued when the current buffer address is 320 and the present stored cursor address is 0 (zero), the cursor address changes from 0 to 320. The current buffer address at the end of this operation remains at address 320. If multiple IC orders appear in the write data stream, the last one encountered is used in subsequent operations.

If the IC order is placed at a location past the end of display on the computer, the current buffer address will be updated so that the cursor address will be located on the display.

At the end of write data stream, after all orders have been executed by the computer and all required data has been entered into the buffer, the stored cursor address determines which unprotected field should be displayed to the user first.

The following algorithm is used:

1. The computer searches the buffer backward to find the first protected field attribute before the stored cursor address. Unprotected attributes are not considered.
2. The computer does a forward tab (or equivalent) to an unprotected field.
3. If a protected field attribute is not found before the first location of the display buffer is found (location 0), then an unconditional forward tab equivalent is executed to display the first unprotected field in the buffer.

Program Tab (PT) Order

The PT order advances the current buffer address of the first character position to the next unprotected field. Special conditions are as follows:

- If the PT order is issued when the current buffer address is the location of an attribute byte of an unprotected field, the buffer address is set to the next location of that field (location 1).
- If a PT order follows data, the rest of the field is null-filled.
- If a PT order in the write data stream does not follow a control command, order (or order sequence such as WCC), IC order, or RA order (4-character sequence), nulls are inserted from the current buffer address to the end of the field.
- When the PT order follows a control command, order, or sequence order, the buffer content is not modified for that field.

The PT order stops its search at the last location in the buffer. If an attribute character for an unprotected field is not found by this point, the buffer address is set to 0 (first location of the buffer). If the PT order finds an unprotected attribute in the last location of the buffer, the buffer address is still set to location 0.

To continue the search for an unprotected attribute, a second PT order must be issued immediately following the first one. Since the current buffer address was set to 0 by the first PT order, the second PT order begins its search starting at location 0. If the previous PT order was inserting nulls in each character location when it was terminated at the last buffer location, the new PT order will continue to insert nulls from buffer location 0 to the end of the current field.

Repeat Address (RA) Order

The RA order stores a character “y” in all buffer locations starting at the current buffer address and ending at (but not including) the specified stop address. For the asynchronous protocol, the stop address is “xxxx.” The stop address and the character to be repeated are identified by the five bytes immediately following the RA order in the write data stream.

For the SNA/SDLC protocol, the third character following the RA order is always interpreted as the character to be repeated. If an invalid stop address is specified such as an address outside the range of the computer's display buffer, the following events occur (not necessarily in the order listed):

- The write operation is terminated at that point.
- The character is not stored.
- The computer recognizes this as an error condition.
- The computer automatically performs a clear memory operation.
- The entire display buffer is set to nulls.
- A clear AID character is transmitted to the computer control unit for forwarding to the host.

When the stop address is lower than the current buffer address, the RA operation wraps from the end of the buffer to the beginning of the buffer and continues. When the stop address equals the current buffer address, the specified character is stored in all buffer locations.

Attribute characters will be overwritten by the RA order if they occur before the RA order stop address.

Set Buffer Address (SBA) Order

The 3-byte SBA order specifies a new buffer address which operations are to start or continue. Operations start at address 0 by default.

Use the SBA order to write data into various areas of the buffer. An SBA order may also:

- Precede another order in the data stream to specify the starting address for a PT, RA, or EUA order.
- Specify the address at which an attribute byte is stored by an SF order.
- Specify the cursor address for an IC order.

If the SBA order specifies an invalid address (that is, an address outside the range of the computer's display buffer) the following events occur:

1. The write operation is terminated at that point.
2. The computer recognizes this as an error condition.
3. The computer automatically performs a clear memory operation.
4. The entire display buffer is set to nulls.
5. A clear AID character is transmitted to the computer control unit for forwarding to the host.

The result of this action is the same as if the user had selected Function 7 (clear memory).

Start Field (SF) Order

This order notifies the computer that the next byte in the write data stream is an attribute character. The computer stores the next byte and attribute character at the current buffer address. As the attribute character is stored, the computer sets a control bit at the address, identifying that byte as an attribute character during subsequent program or device operations with the buffer data.

Note: The byte immediately following the SF order in the data stream is always stored as an attribute character even when the byte is intended as an order or an alphanumeric data character.

The attribute character defines the characteristics or “attributes” of the data that follows. Each attribute character plus all the data following it, up to the next attribute character or the last location of the buffer, is called a field.

In addition to defining the start of a field, attribute characters define these characteristics for all character locations contained within the field:

- Protected from modification by the user, or unprotected (available for the operator to modify or enter data). If unprotected, the field is an input field.
- Alphanumeric (an input field that accepts alphabetic, numeric, or symbol characters) or numeric (an input field that accepts only the period, dash, and numbers 0-9).
- Field displayed or not displayed to the user.

Each character occupies one of the character locations in the buffer, but cannot display. During display operations, attribute characters appear as blanks.

3270 Attribute Character Bit Assignments

Byte #	Description
7	Not used
6	Not used
5*	0 = Unprotected 1 = Protected
4*	0 = Alphanumeric 1 = Numeric only
3, 2	00 = Field is displayable, normal video mode 01 = Field is displayable, normal video mode 10 = Field is displayable, reverse video mode 11 = Field is non-displayable
1	1 = Keyboard only data 0 = Keyboard or bar code reader input allowed
0	MDT bit identifies modified fields which must be sent to the controller when transmission begins: 0 = Field was not modified. 1 = Field was modified; may also be set by the application program in the data stream to force transmission of a particular field.
* Bits 5 and 4 equal to 11 cause an automatic skip.	

Attribute characters are treated as characters that are protected from user intervention. They cannot be replaced by alphanumeric characters entered from the keyboard or scanner. However, the MDT of the attribute can be changed by the operator by using the previous table. Also, attribute characters are not protected from being overwritten by alphanumeric data that is included in a **Write**, **Erase/Write**, or **Erase All Unprotected** command data stream.

Additional **SF** features include **Automatic Skip**, **Auto-Forward Tab**, and **Auto Enter**.

Automatic Skip

This automatically forwards tabs to the next input field when the current input field is full. This is done by setting the attribute byte of the next field to be protected and numeric (bits 5 and 4 of the attribute character are equal to 11). This indicates to the computer that when the last byte in the input field is filled, the computer should advance to the next input field.

Auto-Forward Tab

The **Auto-Forward Tab** on a good bar code scan is used for unprotected (input) fields to allow emulation of keystroke operations when a good scan is detected. The **Auto-Forward Tab** automatically advances to the next input field after processing bar code data. For asynchronous communications, initiate this feature by inserting a "%" character in the first byte of an unprotected (input) field after the attribute byte of a **SF** order (/Fa%).

Auto Enter

The **Auto Enter** feature is also used for unprotected input fields to allow emulation of keystroke operations when a good scan is detected. **Auto Enter** on a good bar code scan automatically emulates the pressing of the [Enter] key after processing bar code data, sending data to the host system.

For asynchronous communications, initiate this feature by inserting an "&" character in the first byte of an unprotected input field after the attribute byte of a **SF** order (/Fa&).

Sending Data to the Host

Data is returned to the host computer when the operator presses the [Enter] key on the computer's keyboard. When this happens, the station transmits all modified data fields back to the host computer. Data is returned to the host computer in the following format:

```
<AID> <cursor address> <SBA> <attribute> <address +1>
<data> <SBA> <attribute> <address +1> <alphanumeric data>
```

3270 AID Keys

The 3270 AID keys request an application program from the host computer. Press [Enter], [F1]-[F24], [PA1], [PA2], [PA3], or the [CLEAR] key for the desired program. The AID key sends the AID code (an ASCII value) and any modified data. The [CLEAR], [PA1], [PA2], and [PA3] keys send only the AID code. This table lists AID keys and values.

3270 AID-Generating Key Conversions

AID Key	Hexadecimal	ASCII
F1	41	A
F2	42	B
F3	43	C
F4	44	D
F5	45	E
F6	46	F
F7	47	G
F8	48	H
F9	49	I
F10	4A	J
F11	4B	K
F12	4C	L
F13	4D	M
F14	4E	N
F15	4F	O
F16	50	P
F17	51	Q
F18	52	R
F19	53	S
F20	54	T
Enter	27	'
Clear	37	7
PA1	59	Y
PA2	5A	Z
PA3	2C	,

5250 Display Data Stream

This section describes the following:

- Host 5250 display data stream commands the host sends to computers.
- Host 5250 data stream orders the host sends to computers. Orders govern format and data expectations within the Write to Display command.
- Asynchronous 5250 prefixes.
- System messages and parameter errors that show in the computer display.

This chapter is not intended as a programming guide, but as a comparison with your existing system. It helps identify differences between your screen-generating utilities and the features the computers support.

Screen Design Aid

This chapter, along with your Screen Design Aid (SDA) documentation, can help you resolve occasional questions regarding attribute selection and the capabilities of the computers. You can use your SDA to do much, if not all, of the programming to handle commands and orders for the network. This chapter is provided to ensure you have the most accurate and detailed information available and to describe variations from the 5250 data stream where they occur.

If you are comfortable using the SDA on your host computer, this chapter will tell you which commands and orders the computers support. Then read Chapter 7, "Extended Commands" on page 343, which describes the extended commands you can use to perform operations beyond the typical capability of the IBM Display Station.

Applications

If you need to write applications specifically for the computer, it is recommended that you use the corner mode window onto the IBM Display Station and write programs using the portion of the display available without moving the window. This is the most efficient method for developing custom applications for the computer.

Data Stream Command Structure

The 5250 command structure enables the computer to display prompts and accept keyboard or scanner input. A second kind of command, called an order, works with the Write-To-Display command to define display characteristics, buffer sizes, and acceptable data types.

Display data stream commands form two categories: the input commands and the output commands. The *IBM 5250 Information Display System Functions Reference Manual* (IBM part number SA21-9247-6 provides more detailed descriptions than those presented in this guide).

These are the input and output commands and hex codes with additional reference information about each command.

Input and output commands and hex codes

Input Commands	Hex Code	Reference
Read Immediate	72	See“Read Immediate” on page 211
Read Modified Immediate Alternate	83	See “Read Modified Immediate Alternate” on page 212
Read Input Fields	42	See “Read Input Fields” on page 213
Read MDT Fields	52	See “Read MDT Fields” on page 214
Read MDT Alternate	82	See “Read MDT Alternats” on page 215
Read Screen	62	See “Read Screen (Immediate)” on page 216
Save Screen	02	See “Save Screen (Immediate)” on page 216
Write Structured Field	f3	See “Write Structured Field” on page 217
Clear Format Table	50	See “Clear Format Table” on page 219
Clear Unit	40	See “Clear Unit” on page 220
Clear Unit Alternate	20	See “Clear Unit Alternate” on page 220
Restore Screen	12	See “Restore Screen” on page 221
Roll	23	See “Roll” on page 222
Write Error Code	21	See “Write Error Code” on page 223
Write to Display (WTD)	11	See “Write to Display (WTD)” on page 224

Input Commands

Input commands consist of immediate and AID-associated Read commands. Immediate commands are executed when the controller receives the command. AID-associated commands are queued until the operator presses an AID-generating key such as Enter/Rec Adv. Other AID-generating keys are listed in the chapter for each computer.

Read Immediate

The **Read Immediate** command is similar to the **Read Screen** command, except the data comes from the format table rather than the display. The command sends back the contents of all the input fields on the display.

Restrictions

This command must be the last command in the chain and the controller must have change of direction (CD). The command is rejected if the computer is in an error, system request, or SS message state.

Note: If the display supports a separate message line, and the message line is selected, the command is rejected only for the system request and SS message states. The command is processed with the display in error state.

Format

The command takes the following form.

ESC	Read Immediate Command
Hex 04	Hex 72

Results

Information associated with this command returns in the LU-LU nonexpedited flow. What the user receives when they issue this command depends on the condition of the master modified data tag (MDT) bit:

- If the MDT bit is not set, the user receives:
  Cursor Address AID Code
- If the master MDT bit is set, the user receives:
  Cursor Address AID Code Field Data

The field data consists of the contents of all input fields as they appear on the display, unless resequencing is specified. Any attributes contained in a field are treated as data and returned as such. Field boundary attributes are not considered part of the field. All nulls are converted to blanks. If the specified field is a signed numeric field, the last character is not sent.

If that same field is negative, the zone position of the next-to-the-last character is changed to hex D. In each case, the returned cursor address indicates the current location of the cursor and the AID code is hex 00.

Note: Queued Read commands and pending AID codes are not cleared. The format table, display annunciators, insert mode keying history, and display contents are not affected.

Read Modified Immediate Alternate

The **Read Modified Immediate Alternate** command enables the host to read data from modified input fields without depending on the operator to press an AID request key.

Format

The command takes the following form:

ESC	Read Modified Immediate Alternate Command
Hex 04	Hex 72

Results

This command is processed the same as the **Read Immediate** command, except for these differences in the format of data sent to host:

- Controller only returns data from those fields in the format table that have the MDT bit on.
- Data field is delimited by Set Buffer Address (SBA) orders in the same way as the response data field for the **Read MDT Fields** command.
- Leading/imbedded nulls in each field's data are not converted to blanks.
- Trailing nulls within each field's data are suppressed.

Read Input Fields

The **Read Input Fields** command causes the computer to wait for the operator to press an active AID-generating key, then sends the contents of all fields defined in the format table to the host.

Restrictions

The operator must press an AID-generating key to execute this command. Also, CD in the Request/Response Header (RH) must be on before the AID byte can be serviced. This command is cleared if:

- The host system issues a session control request (such as Unbind).
- The host system issues a **Clear Unit** command and the controller executes it.
- The host system sends another **Read** command to the same LU and overlays this command.
- The **Read** command is executed.

Format

- The command takes the following form:

ESC	**Read Input Fields Command**	**CC Byte 1**	CC Byte 2
Hex 04	Hex 42	Hex 00	Hex 13

Results

Information associated with this command returns to the user in the LU-LU non-expedited flow. The host system cannot receive this information until the operator presses an AID-generating key. What is received in response to this command depends on the master MDT bit.

- If the master MDT bit is not set, the user receives a cursor address (position of the cursor when the AID-generating key was pressed) or an AID code (the code for the AID-generating key the operator used).
- The user receives the cursor address and AID code if any of the Clear, Help, Print, or Record Backspace in home position keys are pressed.

- If the master MDT bit is on, the user receives the cursor address (position of the cursor when the AID-generating key was pressed), the AID code (code for the AID-generating key the operator used), and field data (returned only when any of the Roll Up, Roll Down, Enter/ Rec Adv, or unmasked function AID-generating keys is used --- when returned, the field data consists of the contents of all input fields as they appear on the display unless resequencing was specified).

Any attributes in a field are treated as data and returned as such. Field attributes are not considered part of the field. All nulls are converted to blanks. All pending AID request bytes are cleared. If the specified field is signed numeric, the first character is not sent; if that same field is negative, the zone position of the next-to-the-last character is changed to hex D. Though data is not sent, the CC bytes are processed.

To determine the appropriate codes for the control character (CC) bytes, see **“Write to Display (WTD)” on page 224**.

Format

The command takes the following form:

Bytes 1 and 2	Byte 3	Byte 4	Byte 5
Cursor address	AID code	Field data	Field data

Byte 6 and all that follow contain the remaining field data from the format table. The computer returns complete contents of each field, with nulls converted to blanks.

Read MDT Fields

The **Read MDT Fields** command causes the computer to wait for the operator to press an active AID-generating key, then sends all fields that have their MDT bit set. The bit could have been set by the operator entering data in the field or by the WTD command.

Restrictions

The operator must press an AID-generating key to execute this command. In addition, the CD in the RH for the requested LU must be on before any information can be sent back to the host system in response to the command. The command is cleared if:

- A session control request (such as Unbind) is issued by the host.
- A **Clear Unit** command is issued.
- The host system sends another **Read** command to the same LU and overlays this command.
- The **Read** command is serviced.

Format

The command takes the following form.

ESC	Read MDT Command	CC Byte 1	CC Byte 2
Hex 04	Hex 52	Hex 00	Hex 13

The format of the returned data is the following. The {} brackets around the last three fields indicate these fields may repeat as a unit.

Bytes 1 and 2	Byte 3	Byte 4	Byte 5	Byte 6
Cursor address	AID code	{SBA	Field address	Field data}

The cursor address gives the cursor location on the display. The SBA fields are formatted as **Set Buffer Address** orders. The field address comes from the address portion of the order and gives the address of the modified field (excluding the attribute).

Results

The contents of each field that has an MDT bit on are returned to the host system in the order that the fields appear in the format table, if one of the Roll Up/Page Down, Roll Down/Page Up, Enter/Rec Adv, or unmasked function AID-generating keys is used.

If no MDT bits are on, or if the operator does not use one of the acceptable AID-generating keys (Clear, Help, Print, or Record Backspace), only the cursor and AID code are returned to the host system.

The host system can use field control words (FCWs) to rearrange the sequence in which the fields are returned. If data is returned, the following formatting is done. If the field is not a transparent data field, this occurs:

- Trailing nulls are stripped. If the field consists of all nulls, only the Set Buffer Address, row, and column are returned, and leading and embedded nulls are converted to blanks.

If signed numeric, the last character is not sent. If that same field is negative, the zone position of the next-to-the-last character is changed to hex D. Hex 10 and hex 11 are control data. Avoid writing hex 10 and hex 11 to the display as data unless written in transparent data fields.

Read MDT Alternats

Read MDT Alternate is similar to the **Read MDT Fields** command with these exceptions:

- Leading and embedded nulls within the fields remain nulls. Trailing nulls are stripped.
- For fields that are nulls but have their MDT bit on, the controller returns an SBA order followed by the field's address.

Read Screen (Immediate)

This command sends the contents of the display to the host in the same order information appears on the screen. For example, row 1 goes first.

Restrictions

This command must be the last command in the SNA chain, and CD must be on. The computer rejects the command when it is in the prehelp error, posthelp error, system request, or SS message state. During transmission the keyboard is locked, but pending AID requests or Read commands are retained.

Note: If the display supports a separate message line, and the message line is selected, the command is rejected only for the system request and SS message states. The command is processed with the display in error state.

Format

The command takes the following form:

ESC	Read Screen Command
Hex 04	Hex 62

Results

The contents of the entire display, including the attributes, is sent to the host system just as it appears in the regeneration buffer (no formatting or conversion is done). Cursor address and AID are not returned. This command does not clear either pending Read commands or AID requests. The keyboard is temporarily locked. The following are unaltered:

- Annunciators
- Cursor location
- Keying history
- Display contents
- Modes
- Format table

Save Screen (Immediate)

The **Save Screen** command sends the state of the computer to the host. Information sent to the host includes:

- Format table
- Display buffer
- Keyboard state
- Cursor location
- Code returned in response to the Help key
- Any outstanding AID requests
- Any outstanding Read commands

Restrictions

This command must be the last command in the SNA chain and CD must be on. The command is rejected if the addressed LU is in either the system request or SS message state. Byte 5 and all remaining bytes contain the Save Screen information.

Format

The command takes the following form:

ESC	Save Screen Command
Hex 04	Hex 62

The format of the returned data is:

Byte 1	Byte 2	Byte 3	Byte 4
Escape	Restore Screen	<computer state>	<computer state>

Results

All data required for restoring the display are sent to the host system. It must not be modified by the host system if the result of the **Restore Screen** command is to have integrity.

Write Structured Field

The **Write Structured Field** command introduces a data field into the SNA flow. The host system uses this type of data structure to support the following operations

- Generation of a pass-thru data stream
- Receipt of 5250 device functional information
- Control over display stations during text support operation

Format

The command takes the following form:

ESC	Write Structured Field Command	Length	Class	Query	Flag
Hex 04	Hex f3	xx xx	xx	xx	xx

Note: Only command classes 0xd8 and 0xd9 are supported. For command class 0xd8, only type codes of 0x20 and 0x21 for query are supported. For command class 0xd9, only type code of 0x70 for query is supported.

Results

Write Structured Field Command Results

Bytes	Command Class	Definition
0-1	0x00 0x00	Cursor row/column
2	0x08	Inbound Write Structured Field aid
3-4		Length of query reply
5		Command class
6		Command type – query
7		Flag byte

Write Structured Field Command Results (continued)

Bytes	Command Class	Definition
8-9	0x02 0x00	Controller Hardware Class PC DOS non DBCS WSF
10-12		Controller code level
13-28	0x00	Reserved
29	0x01	5250 Emulation
30-33	0xf5 0xf2 0xf9 0xf1	Device type 5291
34-36	0xf0 0xf0 0xf0	Model 000
37	0x01	Standard keyboard
38	0x00	Extended keyboard ID
39	0x00	Reserved
40-43		Display serial number
44-45	0x01 0x00	Maximum number of input fields 256
46-48	0x00	Reserved
49-53		Controller display capability Byte 49 definition: Bits 0-1 01 Row 1/Col 1 support Bit 2 1 Read MDT alternate support Bit 3 0 Display does not have PA1/PA2 support Bit 4 0 Display does not have PA3 support Bit 5 0 Display does not have cursor select support Bit 6 0 Display does not have Move cursor Order support Bit 7 1 Read MDT Immediate Alt command support Byte 50 definition: Bit 0-3 0001 24x80 screen size Bit 4 0 No light pen support Bit 5 0 No magnetic stripe support Bit 6-7 00 Monochrome display Byte 51 definition 0x00 Reserved Byte 52 definition: Bit 0-2 000 No Double-Byte Character support Bit 3-7 Reserved Byte 53 definition: Bit 0-2 000 No graphics capability
54-60		Reserved

Note: For command class 0xd8 with a type code of 0x20 only data through byte 7 is returned to the host.

Output Commands

Some output commands have associated data and control information (write characters and orders). **Write** commands are executed immediately.

Clear Format Table

The **Clear Format Table** command clears the format table without erasing data in the display buffer.

Restrictions

The computer rejects the command when in an error, system request, or SS message state.

Note: If the display supports a separate message line, and the message line is selected, the command is rejected only for the system request and SS message states. The command is processed with the display in error state.

Format

The command takes the following form:

ESC	Clear Format Table Command
Hex 04	Hex 50

Results

The following list describes what happens when this command is executed.

Note: Operator-selected reverse image and the Message Waiting annunciator are not affected by this command.

- The keyboard is locked.
  The keyboard clicker is turned off.
  The Input Inhibited annunciator is turned on.
  The insert mode is cleared (as is the annunciator).
- The format table is cleared. The format table header formats as follows:

Format ID	Hex 00
First field transmitted to host system	0 *(Resequencing is disabled)*
Error line	Bottom line of display

- System insert cursor address set to row 1, column 1 to clear previous **Insert Cursor** orders.
- Pending AID requests, all keying history, and the master MDT bit is cleared.
- A cursor blinking from waiting for a required Field Exit key is reset.

Clear Unit

This command clears the display and format table.

Restrictions

The computer rejects the command when it is in the SS message state.

Format

The command takes the following form:

ESC	Clear Unit Command
Hex 04	Hex 40

Results

When the computer receives the **Clear Unit** command, the following occur and the presentation screen size is set to 24 rows by 80 columns.

- The keyboard locks.
  The Input Inhibited annunciator is turned on.
  The error state (or system request state) is cleared.
  The shift, alternate, function, and insert modes are cleared.
- Format table cleared, default header assumed as this is not format level 0.
- The MDT bit is cleared.
- All function keys are set up to return data.
- The display is cleared by writing nulls to the display buffer.
- A normal attribute is written byte to row 1, column 1 of the display.
- Cursor is placed at row 1, column 2; making the insert cursor address.
- AID requests, pending **Read Input Fields**, **Read MDT Fields** cleared.

Note: Operated-selected reverse image and the Message Waiting annunciator are not affected by this command.

Clear Unit Alternate

The host system issues this command to clear the display screen and the format table. This command sets the presentation screen size to 27 rows by 132 columns.

Restrictions

If the host system issues this command to a computer without 132-column support, the computer rejects the command with a negative response of “10030101.”

Format

The command takes the following form.

ESC	Clear Unit Alternate Command	Parameter
Hex 04	Hex 20	Hex 00

Note: The parameter byte must contain Hex 00. Any other value causes the computer to return a negative response of “10030105.”

Restore Screen

This command restores all data stored by the last **Save Screen** command.

Restrictions

Computer rejects this command when it is in a system request or SS message state.

Format

The command takes the following form:

ESC	Restore Screen Command	Data From Last Save Command
Hex 04	Hex 12	<saved data>

Results

These events happen when this command is executed and are restored:

- Contents of the display and format table.
- State of the keyboard, including the insert mode with the annunciators.
- Location of the cursor and the way it was displayed.
- System insert cursor address.
- State of the master MDT bit.
- Error code and explanation provided by the host system in the **Write Error Code** command, if the computer was in an error state. Error code and information are returned via the operator's use of the Help key.
- Requirements to send LU-LU Lustat when error line is available.
- Any **Read** command that was pending at the time.
- Any outstanding AID requests at the time of the **Save Screen** command.
- These conditions from the previous save are not restored:
- Condition of the Shift key and the associated annunciator.
- Status of the Message Waiting annunciator.

- The current status of the hex mode is reset.
- The following conditions cause parameter errors: invalid data is detected and the required amount of data is not received.

Note: A **Clear Unit** command is executed if an error is detected.

Roll

This causes the image presented in the computer display to roll up or down across the larger image area available on the IBM Display Station.

Restrictions

The command is rejected if the display is in the prehelp error, posthelp error, system request, or SS message state.

Note: If the display supports a separate message line, and the message line is selected, the command is rejected only for system request and SS message states. The command is processed with the display in error state.

Format

The command consists of five bytes in the following form:

			Row Number	
ESC	**Command**	**Direction**	**Top**	**Bottom**
Hex 04	Hex 23	Hex 08	Hex 01	Hex 23

Bits in the direction byte define the direction and number of lines to roll. Bit 7 is the most significant bit.

Bit	Description
7	0 = Roll up; 1 = Roll down
6-0	Number of lines to roll

The top row number tells the first line for inclusion in the roll. Bottom row number tells the last row number to include. Top/bottom numbers together define the number of lines presented on the display for the roll.

Results

These conditions govern the **Roll** command.

- Lines vacated due to the **Roll** are not cleared to nulls.
- Format table is not changed (could cause bizarre results).

Note: If display does not conform to the format table, roll is not done. A scroll distance of zero has no effect on the display.

- Rolled over data and lines rolled out of the area are lost (cannot be rolled back onto screen).
- The state of the keyboard and pending AID bytes are not affected.
- The following conditions cause parameter errors:
  - A top line of zero.
  - A top line greater than or equal to the display length.
  - A bottom line of zero.
  - A bottom line greater than the display length.
  - A top line greater than or equal to the bottom line.
  - A roll area greater than the bottom line minus the top line.

Write Error Code

The **Write Error Code** command forces the computer into the prehelp state (the same state entered when an operator makes a keying error).

Restrictions

This clears any outstanding AID requests. The computer rejects this command if it is in the prehelp, system request, or SS messages state.

Format

The command may take any of the following forms. The **Insert Cursor** (IC) order may also be embedded in the error message.

Note: You must provide either an **Insert Cursor** order or an error message or both to avoid an error.

ESC	Write Error Code Command	Insert Cursor Order
Hex 04	Hex 21	Position cursor

ESC	Write Error Code Command	Error Message
Hex 04	Hex 21	<message text>

ESC	Write Error Code Command	Insert Cursor Order	Error Message
Hex 04	Hex 21	Position cursor	<message text>

You can use the **IC order** to place the cursor at the beginning of the field where the error occurred. This helps the operator find the source of the error. The **IC order** does not affect where the error message is written.

The computer writes only the first 78 bytes of the error message to the error line. If the error line is used for error presentation, the computer saves data on the error line and restores it after the operator presses the Reset key.

When the Help key is pressed (prehelp error state only) in response to the error condition, characters from error line columns 2, 3, 4, 5 are returned to the host system in a packed form and sent as a Signal command.

These characters form an index code that elicits a user-generated description of the error for the operator.

- Prehelp error state is selected for the keyboard.
- Input Inhibited annunciator is on.
- Insert mode, and Insert annunciator and cleared.
- Command, dead key diacritic, and hex modes are cleared.
- Cursor is blinking.
- The line in the format table header defined as an error line is saved.
- Cursor moves to the location specified by the **IC order**. If no **IC order** is given, the cursor does not move.
- All characters (except **IC order**) found between the command byte and the end of the chain or next ESC are written on the error line. If the data exceeds 80 characters, an error occurs.
- All outstanding AID bytes are cleared.
- When the operator presses [HELP], the controller places a nonblink high intensity attribute in column 1 of the error line, replacing anything that was previously there.
- The locked state of the keyboard is cleared if the keyboard was locked and the computer was not in a posthelp error state. This allows the operator to release the keyboard by pressing the Reset key.

The following conditions cause parameter errors:

- Neither an **IC order** nor data follows the command.
- Invalid **IC order**.
- More data than 80 bytes is specified.

Write to Display (WTD)

The **WTD** command does the following:

- Modifies the format table and display buffer contents.
- Governs MDT flags and Message Waiting annunciator.
- Nulls nonbypass fields.
- Controls keyboard lock and cursor blink.

Restrictions

This command is rejected if the display is in a prehelp error, posthelp error, system request, or SS message state.

If the display supports a separate message line, and the message line is selected, then only system request or SS message state result in a contention state error. If the error line is in use, however, and the WTD attempts to redefine the error line (using Start of Header byte 4), a negative response is generated.

Format

The WTD command takes either of the following forms.

ESC	WTD Command	CC Byte 1	CC Byte 2	Orders or Data
Hex 04	Hex 11	Hex 00	Hex 13	<orders>
Hex 04	Hex 11	Hex 00	Hex 13	<data>

Note: CC Byte 1 and CC Byte 2 are the write control characters.
CC Byte 1 may:
  Clear the master MDT flag and reset the MDT flags.
  Null all appropriate nonbypass fields.
CC Byte 2 sets:
  Cursor blink
  Keyboard lock
  Alarm
  Message Waiting annunciator

Any character that is not an order and not associated with an order is considered data and is written on the screen at the current display address. The address is then incremented by 1 for each character written. These characters should be hex 00, hex 1C, or above hex 1F so there is no conflict with the codes reserved for orders.

Results

If the WTD command changes the format table, keyboard remains locked until specifically unlocked by CC Byte 1 or subsequent WTD command.

See the following table to determine the appropriate commands. To select a command, check the body of the table for the row showing the desired combination of conditions. Then refer to the left-most column to find the corresponding hex code.

Write to Display Control Byte 1

		Clear Master MDT and Reset MDT Flags		Null All Appropriate Nonbypass Fields	
Hex Code	Reset Pending AID, Lock Keyboard	Nonbypass Fields	All	Field with MDT On	All
00					
20	X				
40	X	X			

Write to Display Control Byte 1 (continued)

		Clear Master MDT and Reset MDT Flags		Null All Appropriate Nonbypass Fields	
Hex Code	**Reset Pending AID, Lock Keyboard**	**Nonbypass Fields**	**All**	**Field with MDT On**	**All**
60	X		X		
80	X			X	
A0	X	X			X
C0	X	X		X	
E0	X		X		X

Below are the bit positions for CC Byte 2. The most significant bit is 7; the least significant bit is 0. Invalid orders or no data, orders, or CC following the command byte cause parameter errors:

CC Byte 2 Bit Positions

Bit	Description
7	Always 00
6	0 = Cursor moves to default or **IC order** position when keyboard unlocks. * 1 = Cursor does not move when keyboard unlocks.
5	0 = No action; 1 = Reset blinking cursor
4	0 = No action; 1 = Set blinking cursor (if reset bit also = 1)
3	0 = No action; 1 = Unlock keyboard, reset pending AID bytes
2	0 = No action; 1 = Sound alarm
1	0 = No action; 1 = Reset Message Waiting annunciator
0	0 = No action; 1 = Set Message Waiting annunciator (even if bit 5 is set to 1)
** Exception: When a **WTD** is received with the keyboard unlocked and the **WTD** does not modify the keyboard state.*	

Orders

The following pages describe the host 5250 data stream orders the host can send to computers on the network. Orders govern format and data expectations within the WTD command. For more information about the WTD command, see “Write to Display (WTD)” on page 224.

The host application can include computer orders in the WTD command, alone or intermixed with display data. Orders execute sequentially and do not store in the display buffer with other data in the data stream.

Five buffer control orders position, define, and format data written into the buffer, erase selected unprotected data in the buffer, and reposition the cursor. The **TD** order writes transparent data to the display.

Below is a list of available orders and hex codes, including page references for additional information about each order.

Output Commands	Hex Code	Page Reference
Insert Cursor (IC)	13	this page
Move Cursor (MC)	14	"Move Cursor (MC)" on page 228
Repeat to Address (RA)	02	"Repeat to Address (RA)" on page 228
Set Buffer Address (SBA)	11	"Set Buffer Address (SBA)" on page 229
Start of Field (SF)	1D	"Start of Field (SF)" on page 229
Start of Header (SOH)	01	"Start of Header (SOH)" on page 235
Transparent Data (TD)	10	"Transparent Data (TD)" on page 237

Insert Cursor (IC)

This order either sets the system insert cursor address to the location specified by the two bytes that follow the order when it is included in the WTD command, or moves the cursor to the specified address without affecting the system IC address when it is included in the Write Error Code command. Byte 1 gives the row address, Byte 2 the cursor address.

Note: If multiple **IC** orders appear in the Write data stream, the last one encountered is used in subsequent operations.

Restrictions

A parameter error is posted when there are fewer than two bytes following the order, the row address equals 0 or is greater than 24, and the column address equals 0 or is greater than 80.

Format

The command takes in the following form.

Insert Cursor Order	Row Address	Column Address
Hex 13	Hex 01	Hex 01

Results

When the order is used in the **WTD** command, the cursor is not immediately moved; the address is saved for later use. The cursor is moved when the entire **WTD** is completed.

When the order is used in the **Write Error Code** command, the cursor is moved to the address given in the **IC** order and does not affect the system IC address. The cursor exits the field regardless of the type and does not perform any field checks. For example, it does not check for a filled field for a field specified as mandatory fill.

You can use the **IC** order in the **Write Error Code** command to tell the computer operator where an error has occurred. Maximum row and column addresses are determined by the character size selected through the computer's TE 2000 configuration menus or firmware.

Move Cursor (MC)

This order enables the host to move the cursor to a specified position without modifying the home address, and without regard to keyboard state.

Format

The command takes in the following form.

MC Order	Row Address	Column Address
Hex 14	1 byte	1 byte

If the WTD data stream contains multiple IC or MC orders, the last IC or MC order determines the cursor position. The IC order negates any previously coded MC orders and the MC order negates any previously coded IC orders, with the exception that the last IC order establishes home position. To set the home position and then move the cursor, code the IC order, and then the MC order. WTD control character values do not affect the MC order, including the move cursor flag.

Repeat to Address (RA)

The **RA** order stores a specified alphanumeric or null character in all buffer locations starting at the current buffer address and ending on the specified stop address. This stop address and the character to repeat are identified by the three bytes immediately following the RA order in the Write data stream. You can use RA orders to fill an area of the display.

Restrictions

A parameter error is posted when:

- Fewer than three bytes follow the order.
- A row address equals 0 or is greater than 24.
- The specified ending address is less than the current display address.

Note: Although any character can be repeated, avoid using hex 11 (**SBA** orders), because this value is the delimiter between the fields sent in response to the **Read MDT Fields** command.

Results

The character is repeated from the current display address through the ending display address specified. The current display address is then updated to the value of the last position +1.

Set Buffer Address (SBA)

This order specifies a new buffer address from which operations are to start or continue. You can use the **SBA** order to:

- Write data into various areas of the buffer.
- Precede another order in the data stream to specify the starting address for an **RA** order.
- Specify the address at which an attribute byte is stored by an **SF** order.

Restrictions

A parameter error is posted when there are fewer than two bytes following the order, the row address equals 0 or is greater than 24, and the column address equals 0 or is greater than 80.

When the **SBA** is not specified in the **WTD** command, the data starts at row 1, column 1. This is where the **WTD** command initialized it.

Format

The command takes in the following form.

SBA Order	Row Address	Column Address
Hex 11	Hex 01	Hex 01

Row addresses begin with 1, as do column addresses. However, the computer accepts column address hex 00 and translates it to column 1.

Start of Field (SF)

The **SF** order defines input and output fields. If an input field is being defined, it also resets any pending AID byte and locks the keyboard.

Note: Although this order can be used for output fields, it is not recommended because it degrades performance. Use the **SBA** order instead.

Format

SF order takes the following form with minimal information.

SF Order	Attribute	Length
Hex 1D	Hex 01	Hex 0012

If you include format and control information in the **SF** order, it takes one of the following two forms.

SF Order	[Format]	Attribute	Length
Hex 1D	Hex 4000	Hex 01	Hex 0012

SF Order	[Format]	[Control]	Attribute	Length
Hex 1D	Hex 4000	Hex 8102	Hex 01	Hex 0012

Only include the control word after you have provided a format word. You can also insert several control words between the format and attribute portions of the SF order, but these control words are not required.

The following procedure describes the format portion. The attribute portion of the SF order is disussed next.

To Format Portions of SF Order

In the **SF** order format portion, the two most significant bits (15 and 14) must always be hex 01. See below for descriptions of bits 13 through 0.

Format Portions of SF Order

Bit	Description
14-15	Always 01 (reserved) *Bit 15 is the most significant bit.*
13	0 = Nonbypass field; 1 = Bypass field.
12	0 = Duplication not allowed; 1 = Duplication allowed
11	0 = Field was not modified (**MDT**) bit; 1 = Field was modified
10-8	000 = Alpha shift; 001 = Alpha only; 010 = Numeric shift; 011 = Numeric only; 100 = Katakana shift *(not supported)*; 101 = Digits only; 110 = I/O *(feature input field)*; 111 = Signed numeric
7	0 = Auto enter disabled; 1 = Auto enter when field is exited
6	0 = Field Exit key is not required; 1 = Field Exit key is required
5	0 = Accept lowercase letters; 1 = Translate lowercase to uppercase
4	Reserved
3	0 = No mandatory entry; 1 = Mandatory entry
2-0	000 = No adjustment; 001 = Reserved; 010 = Reserved; 011 = Reserved; 100 = Reserved; 101 = Right adjust, zero fill; 110 = Right adjust, blank fill; 111 = Mandatory fill

FCWs (two bytes each) are optional. The user program can use the **WTD** command to send FCWs to the controller. When FCWs are used, they should follow the FFW of the **SF** order. FCWs found during modification of existing format table entries are ignored. LU does not accept an FCS of hex FFxx.

Computer supported-valid FCWs and functions are:

Supported FCWs and Functions

Value (Hex)	**Description**
80nn	Entry field resequencing. The "nn" specifies the next entry field in the sequence (hex 00 to 80).
8101	Magnetic stripe reader (MSR) entry field.
8103	MSR and selector light pen (SLP) entry field.
84nn	Transparency entry field. The "nn" is any two digits.
B140	Self-check modulus 11 entry field.
B1A0	Self-check module 10 entry field.

The first **FCW** of any type is used; subsequent **FCW**s of the same type are ignored. The controller does not check to see if the **FCW**s are formatted correctly or if the requested function is installed. During subsequent command and keystroke processing, the controller detects and reports these errors to the host if the **FCW** is required. **FCW**s types are as follows.

- **Resequencing**

  Resequencing enables the controller to send the input fields to the host in any specified order. Resequencing is done by chaining input fields with **FCW**s that specify the desired order of transmission. The resequencing **FCW** takes the following format:

Bits	Description
0-1	B 10
2-7	B 000000
8-15	The normal sequence position of the next field to be returned to the host. (The first field on the screen is number 1. Field numbers progress sequentially, left to right and top to bottom.)

Bit 3 of the **SOH** order contains the number of the field to be sent. If the first field identifier in the **SOH** is zero, resequencing does not occur. That is, all resequencing **FCW**s are ignored. Fields are sent to the host in the order defined in the format table. If resequencing occurs, the last field to be sent to the host must contain this **FCW**:

Bits	Description
0-1	B 10
2-7	B 000000
8-15	B11111111

Note: If **FCW**s create a closed loop, the controller detects the error and stops the transmission. An **FCW** for each field is not required, but one pointing to the next sequential field is assumed if no resequencing **FCW** is specified. The last field in the table must have a resequencing **FCW**.

- **MSR**

  MSR enables the scanner for an input field

- **SLP**

  SLP is accepted, but ignored

- **Transparency**

  Transparency defines a field that can contain data of any value.

- **Self-Check**

  Self-check on the controller provides additional integrity for data entry. All field types can be specified for self-checking. The following requirements must be met when specifying a field for self-checking:

  - An **FCW** must be defined for the field. Hex B1A0 selects Modulus 10 checking, and hex B140 selects Modulus 11.
  - Field lengths for checking are restricted to 33 positions. For signed numeric fields, only 32 positions can contain digits and the sign is not checked. If more than 33 characters are given a LUSTAT parameter error results.

  Self-check resolves fields and conditions as follows:

  - The function converts nonnumeric characters (including nulls and blanks) by using the four low-order bits from their EBCDIC representation when the low-order bits are in the range 0-9. For example, A in EBCDIC is C1, thus A = 1. R in EBCDIC is D9, thus R = 9.

    A “0” replaces all other characters with the four low-order bits in the range of A through F. For example, % is EBCDIC 6C, thus % = 0.

    Null and blank characters also convert to 0. All high-order nulls, zeros, and blanks in a field are converted to 0 and do not affect the value of the check number.

  - An all-null field checks correctly. This field can result when the operator unsuccessfully tried to enter digits into a field and checking fails. The controller enables the operator to exit the field from the first position by using the Field Exit key.

Field Types

Field Type	Description
Alpha only	Accepts only the characters A-Z (uppercase and lowercase) and the , . - and blank space.
Alpha shift	Accepts all data keys. The shift keys are acknowledged.
Auto enter	When the operator uses [FIELD EXIT] key to leave this field or puts the last character in the field, the computer treats the action as if the operator pressed [ENTER].
Bypass	Operator cannot put entries in this field. Attempts to enter data cause a keying error.
Digits only	Operator can only put the characters 0-9 in this field. [DUP] key is allowed if enabled.
Duplication	Allows operator to use the [DUP] key. When the operator presses [DUP], the controller fills the field from the cursor to the end of the field with "1C." (Typically the host program uses this to place data from a previous record into the field when it receives the data.) An overstruck asterisk is displayed for the "1C" characters.
Field Exit key required	Operator can only exit the field by pressing a non-data key such as [FIELD EXIT] or other cursor-moving key.
I/O	Only the scanner can put data in this field.
Mandatory entry	The operator must put data in this field before pressing [ENTER] or other AID-generating key.
Mandatory fill	When the operator puts data into this field, they must completely fill it before exiting.
Modified	Marks this field as modified.
Numeric only	Accepts only the characters 0-9 and the symbols + , . - and blank. [FIELD+] exits the field with a positive number; [FIELD-] with a negative number. [FIELD EXIT] exits the field as entered with the sign specified by the host. The controller marks a minus field by changing the high nibble of the last byte to "B" unless a + , . - or blank is in that byte. These characters cause an error to occur.
Numeric shift	Accepts any data characters.
Right-adjust	When the operator exits the field, the controller shifts the contents to the right and fills on the left with zeros or blank spaces, as specified. Bits 13-15 can be 0. The operator can make entries into any part of the field without any subsequent position adjustment. When the controller sends the field to the host, it also sends whatever is in the field (nulls if a **Clear Unit** command was received).

Field Types (continued)

Field Type	Description
Self-check	Modulus 10 self-check fields perform a MOD 10 Check on the input. Modulus 11 self-check fields perform a MOD 11 Check on the input. The controller performs these checks when the operator exits the field. If the checks fail, an input error is indicated and the Input Inhibited annunciator is turned on. The operator must press the [RESET] key to clear the condition. The network checks data by doing the appropriate modus on the low four bits of the characters in the field. If the low four bits are within the range “A” through “F,” zero is used in the modulus calculation.
Signed numeric	Only the characters 0-9 are allowed. The last byte of the field is reserved for the sign character. The operator cannot place data in this location. If the operator presses [FIELD-] to exit the field, the last character is set to “-” and the field is right adjusted. The high nibble of the last digit input is set to “B.” If the operator uses [FIELD+] or [FIELD EXIT] to exit the field, the last character is set to blank and the field is right-adjusted. The “-” or “blank” is sent in response to **Read Input Fields or Read MDT Fields** commands.

Control Portions of SF Order

Below describes the control portions of the **SF** order.

Control Portions of SF Order

Hex Code	Description
80XX	Resequencing, “xx” defines the next field number
80FF	Resequencing terminator
8101	Enable scanner (extended 5250); this is IBM's MSR option
8102	Reserved (extended 5250)
B1A0	Modulus 10 self-check
B140	Modulus 11 self-check

Attribute Portions of SF Order

Below describes the attribute portions of the **SF** order.

Attribute Portions of SF Order

Hex Code	Description
20	Normal (dark characters on light background)
21	Reverse image (light characters on dark background)
24	Underscore
2F	Nondisplay
30	Column separator

Results

The **SF** order does the following:

- The display address is set to the end-of-field address (as specified by the last SF order + 1). This does not happen if this is the first **SF** order or if an **SBA** order precedes it (the **SBA** points at the field starting attribute).
- The screen attribute in the **SF** order is written into the location defined by the display address.
- The start-of-field address is set to the display address +1.
- The end-of-field address is set to the display address + the field length specified by the **SF** order. The ending screen attribute is written to the end-of-field +1. This is a "20."
- The display address is incremented by 1.
- If this is an input field (one in which a field format word has been specified), a format table entry consisting of the field format and the FCWs is generated. In addition, if the **SF** order is rejected, the keyboard is locked and any outstanding AID byte is cleared.
- The format table is modified if the display address +1 is equal to the starting address of an input field that was previously defined. This happens as follows:
  - The field format word of the previously defined field is overlaid with the new one.
  - All FCWs and length parameters that were specified are ignored. Two bytes, however, are still required for the length even though no value check is performed on them.
  - Field ending address is set equal to the field's original ending address.
  - The screen ending address is not rewritten.
  - The defined field is not null filled. If any data characters follow the length field, they are written into the defined field.

Start of Header (SOH)

This order notifies the computer that the next byte in the Write data steam contains a header order specifying header information for the format table. When the controller receives this order, it clears the format table and then inserts the contents of the **SOH** order.

Restrictions

A parameter error is posted when the output data stream ends before the bytes needed are sent or when the first order byte is not between 1-54.

Format

The **SOH** order takes the following form.

Function	Hex Code
Order	01
Length	07
Reserved	00
Reserved	00
Resequence	00
Error row	00
F keys	000000

Results

Because the length of this order varies, the first byte after the order code contains the number of bytes included. Length may range from 0-7 bytes.

A length of 7 bytes enables or disables selected function keys designated by the bits in bytes 5 through 7. When bits are enabled, function keys return AID codes with input data. When bits are inhibited, or length is less than 7, function keys return AID codes but without input data.

The following table shows the data-included switches for the function keys: 0 = enable, 1 = inhibit, and bit 0 is the most significant bit.

Function Key Bit Switches *(Bytes 5, 6, 7)*

Bit #	Function Key	Bit #	Function Key	Bit #	Function Key
0	F24	0	F16	0	F8
1	F23	1	F15	1	F7
2	F22	2	F14	2	F6
3	F21	3	F13	3	F5
4	F20	4	F12	4	F4
5	F19	5	F11	5	F3
6	F18	6	F10	6	F2
7	F17	7	F9	7	F1

You can use *resequencing* to point to a field other than the first, as the field to send to the host in response to a **Read Input Fields** or **Read MDT Fields** command. If not zero, the field FCW is checked as it is sent for resequencing. If enabled, the FCW points to the next field to send. If resequencing is disabled, or no FCW exists, the next defined field is sent.

Bits	Description
0000 0000	Disable resequencing
XXXX XXXX	Number of first field to be transmitted to the host, in response to a **Read Input Fields** or **Read MDT Fields** command.

The error row may vary from 0 to 24. Row "00" signifies the default row, which is the last line for most 5250 devices. You can use other values to override the default row.

Transparent Data (TD)

The **TD** order lets you send data with any value to the computer display. The 2-byte length field after the order tells how many bytes of data follow.

Format

The order takes the following form:

TD Order	Length (2 Bytes)	Data
Hex 10	Hex 0020	<data>

Asynchronous 5250 Prefixes

Each asynchronous 5250 data stream command is preceded by a variable-length prefix. The first byte of the prefix contains the number of bytes in the prefix, not including the length byte. The length may be zero, which indicates an empty prefix. The second byte of the prefix (if the length is nonzero) must be either hex 40 or C0.

- Hex 40 indicates that the data in the rest of the message is LU-LU (normal 5250 commands).
- C0 indicates that the rest of the message is SS-LU data (an SS message). SS messages cause the computer to enter the SS message state, where Reset is the only allowable keystroke. The SS message is a one-line message displayed on the computer's status line until the Reset key is pressed.

After hex 40 or C0, the 5250 prefix may contain one 5250 signal. A signal is a sequence of 5 bytes that tells the computer to perform a special operation. The computer supports the Signal command formats in this chart.

Hex Code	Description
C9 00 00 00 01	Signal operator (turns on the Message Waiting annunciator and sounds the audible alarm)
C9 00 00 00 05	Resets the Message Waiting annunciator

A parameter error is posted when less data is in the message than indicated by the length byte. This causes a parameter error of "21." If the computer encounters data it does not understand, it ignores the rest of the prefix.

System Messages and Parameter Errors

The computer's display reserves a line for status information, which can display system (nonlocal) information such as a message waiting from the host computer, help messages in response to the Help key, or the system request state of the computer. You can program the location of the status line through the Start of Header order.

System codes can also appear on the status line. Parameter errors displayed when the stated condition occurs are listed below. Unless otherwise noted in the table, the computer's Reset key restores the status line and returns to the keying mode in effect when the problem occurred.

System Messages and Parameter Errors

Code	Description
0005	Attempt to enter data when not in a field.
0006	A key other than a data key or [ENTER] pressed in the system request state.
0007	A mandatory entry field has not had data put into it.
0008	The field is alpha only, and an invalid key was pressed.
0009	The field is numeric only, and an invalid key was pressed.
0010	The field is signed numeric, and an invalid key was pressed.
0011	Data cannot be entered into the last byte of a signed numeric field.
0012	No room in field for insert. The field is full or on the last byte.
0013	Cannot leave field while in insert mode.
0014	Field is mandatory fill and cannot be exited until filled.
0015	MOD 10 or MOD 11 Check has failed on field.
0016	Field error: Cannot be used in field that is not signed numeric.
0017	Attempt to use [FIELD EXIT] key in unfilled, mandatory fill field.
0018	Only a nondata key can be used to exit this field; for example, [FIELD EXIT].
0019	[DUP] key not allowed in this field.
0020	Cannot use AID-generating key to exit signed numeric or right-adjusted field. Must use [FIELD EXIT].
0021	Cannot exit a mandatory enter field without entering data.
0023	Invalid hex value entered.
0026	The [FIELD-] key was pressed, but the last character of the field was not 0-9.
0099	Invalid key pressed before computer was in session. Only [SYS REQ] is allowed.
C00	Invalid 5250 command received, byte sequence sent to host: 01 10 03 01 01.
PXX	System code where “XX” is the parameter error.

When a 5250 command contains invalid parameters, system code “PXX” appears on the bottom line of the display. The “XX” in the system code indicates the type of parameter error.

The message sent to the host is: 01 10 05 01 XX, where “XX” is the same parameter error that was put in the computer's display. The parameter errors are a subset of the 5250 data stream command set. Errors supported by the computers are listed in the following table.

Parameter Error	Description
P21	Premature end of data stream.
P22	A row or column specification is out of range.
P23	Ending address of the RA order is less than the start address.
P25	A signed input field had a length of 1. It must be at least 2.
P26	An input field start address was less than a previously-defined input field start address.
P27	An invalid **Restore Screen** command was received.
P28	An attempt was made to define an input field past the end of the display.
P29	Format table overflow (too many input fields were defined).
P2A	An attempt was made to write data past the end of the display.
P2C	Invalid roll up or roll down parameters were received.
P30	An invalid attribute was received as part of an **SF** order.
P87	MOD 10 or 11 input field defined with a length greater than 33.

Native Programming

Intermec Application Server/Gateway Commands

An Intermec Application Server or gateway command is either a “parameter” or a “diagnostic.” Detailed information about each command is listed.

CMT (Return Parameter Values)

This command returns the current value of all CMT parameters.

For compatibility, the Intermec Application Server or gateway response contains place-holders for commands supported on Intermec legacy controllers and multiplexers, but *not* supported on Enterprise Wireless LAN$^{\tau}$ servers and gateways.

Host Command

```
CMT<cr>
```

Server or Gateway Response

```
<parameter values><cr>
```

Example

This example shows that the CMT parameters are set to default values:
Command: CMT<cr>
Response: 002,255,000,000,000,000,000,006,000,
000,000,000,000,000,000,000,000,047,000<cr>

CMT, (Set All Parameters)

The CMT, command sets all CMT parameters at once through one command line. Variables and related are as follows.

Host Command

```
CMT,<a,b,c,...r><cr>
```

Server or Gateway Response

```
OK<cr>
```

Variables and Related Parameters:

- **a** CMT0
- **b** CMT1
- **c** CMT2
- **d** CMT3 *(placeholder for legacy system)*
- **e** CMT4
- **f** CMT5
- **g** CMT6 *(placeholder for legacy system)*
- **h** CMT7 *(placeholder for legacy system)*

- **i** CMT8
- **j** CMT10
- **k** CMT11 *(placeholder for legacy system)*
- **l** CMT12 *(placeholder for legacy system)*
- **m** CMT13
- **n** CMT14 *(placeholder for legacy system)*
- **o** CMT15 *(placeholder for legacy system)*
- **p** CMT16 *(placeholder for legacy system)*
- **q** CMT17 *(placeholder for legacy system)*
- **r** CMT18 *(placeholder for legacy system)*

Maintain the default settings by omitting a parameter and including the comma, as in Example 1. Parameters following an incomplete series remain unchanged, like in Example 2.

Example 1

Command: `CMT,2,2,1,,,,,<cr>`
Response: `OK<cr>`

This example does the following. The remaining values apply to legacy systems and do not change.

Sets CMT0 to 2
Sets CMT1 to 2
Sets CMT2 to 1
Sets CMT4, CMT5, CMT8, CMT10, and CMT13 to default values.

Example 2

Command: `CMT,1,1,<cr>`
Response: `OK<cr>`

The remaining values apply to legacy systems and do not change.

Sets CMT0 to 1
Sets CMT1 to 1
Leaves the parameters for CMT4, CMT5, CMT8, CMT10, and CMT13 set at the values in Example 1 (assuming that Example 1 precedes Example 2).

CMT0 (Set Response Delay)

CMT0 sets the time delay for the server or gateway to respond to a command from the host. This delay may be required after transmitting for the host to prepare to receive data.

The response delay is *x*S10 milliseconds. Values for *x* are 0-255 *(default is 2)*. A “0” indicates no response delay.

CMT0 without the *x* variable returns the previously set and stored value to the host. CMTI (Set Default Parameters) resets all parameters to defaults.

Host Command

`CMT0,`*`x`*`<cr>`

Server or Gateway Response

`OK<cr>`

Example 1

Command: `CMT0,5<cr>`
Action: Server or gateway sets the response delay to 50 milliseconds (5-10).

Example 2

Command: `CMT0<cr>`
Response: `5<cr>`

This response assumes the setting from Example 1.

CMT1 (Set Data Time-Out)

CMT1 sets the number of seconds the server or gateway waits for data from a computer after it receives an R (Read) command from the host. Data received within this time passes immediately to the host. The server or gateway returns an "empty" response (OK<cr>) to the host after the specified time-out if no data is available. Data the user enters at a computer after the time limit is returned after the next R (Read) command.

Host Command

`CMT1,`*`x`*`<cr>`

Server or Gateway Response

`OK<cr>`

Variables

- **0-254** (one-second increments)
  If the value is "0," the server or gateway responds immediately with computer data (if any) or with an OK.
- **255** (indefinite)
  Disables the data time-out. If an "R" Read command is issued, the host cannot send another command until a computer reports *(default)*.

Example 1

Command: `CMT1,1<cr>`
Action: Data time-out is 1 second.

Example 2

Command: `CMT1<cr>`
Response: `1<cr>`
This response assumes the setting from Example 1.

Example 3

Command: `CMT1,0<cr>`
Action: The server or gateway responds immediately with computer data, if it has any, or with an OK.

CMT2 (Interactive Mode)

Use CMT2 to enable or disable interactive mode.

Host Command

`CMT2,x<cr>`

Server or Gateway Response

`OK<cr>`

X Variable Values

- **0**
  Disables interactive mode. The server or gateway never sends an OK. The host sends commands to the server or gateway, which does not respond unless the command has an error. The server or gateway sends computer responses as it receives them *(default)*.
- **1**
  Enables interactive mode. The server or gateway responds to every host command whether or not data is requested. If it has no computer responses queued up, it responds with "OK<cr>."

Example 1

Command: `CMT2,1<cr>`
Response: `OK<cr>`

Example 2

Command: `CMT2<cr>`

Response: `1<cr>`

This response assumes the setting from Example 1.

CMT4 (Add Line Feed to Responses)

Use CMT4 to enable (or disable) the automatic addition of an ASCII line feed character to the end of each data string from the server or gateway to the host.

Host Command

`CMT4,x<cr>`

Server or Gateway Response

`OK<cr>`

X Variable Values

- **0**
  Disables automatic line feed character *(default)*.
- **1**
  Enables automatic line feed character.

Example 1

Command: `CMT4,1<cr>`
Response: `OK<cr><lf>`
The server or gateway adds an ASCII line feed character to each data string sent to the host.

Example 2

Command: `CMT4<cr>`

Response: `1<cr><lf>`

This response assumes the setting from Example 1.

CMT5 (Set Gap Time-Out)

CMT5 sets the time the server or gateway waits between characters before terminating a message to *x*•30 msec. You can only use CMT5 in interactive mode (CMT2,1).

Note: Intercharacter gap timeout for the Intermec Application Server is 10 seconds, regardless of the CMT5 setting.

Values for variable *x* are 0-255 (default: 0). “255” disables gap time-out (the server or gateway waits indefinitely for a carriage return).

When you enable CMT5, the server or gateway expects a carriage return to terminate messages from the host. Without the carriage return, it automatically terminates a message after a time interval equal to the gap time-out setting.

Host Command

`CMT5,x<cr>`

Server or Gateway Response

`OK<cr>`

Example 1

Command: `CMT5,3<cr>`
Action: Server or gateway sets gap time-out to 90 milliseconds (3•30).

Example 2

Command: `CMT5<cr>`
Response: `3<cr>`
This response assumes the setting from Example 1.

CMT8 (Set Data Error Checking)

CMT8 controls the following data checking mechanisms. These data checks perform separate from parity checks or other error checking devices in the network.

- **LRC** (Longitudinal Redundancy Checking)
  This 7-bit mechanism uses all character bits to form a check character at both ends of the communication link. These are compared.
- **CRC-16** (Cyclic Redundancy Checking)
  This uses an algorithm to divide a constant into the numeric binary value of all character bits in a block of data. Quotients are discarded and the remainders are compared.

Host Command

`CMT8,x<cr>`

Server or Gateway Response

`OK<cr>`

X Variable Values

- **0**
  Disables CRC-16 and LRC *(default)*.
- **1**
  Enables CRC-16 error checking, disables LRC.
- **2**
  Enables LRC error checking, disables CRC-16.

Example 1

Command: `CMT8,1<cr>`
Response: `OK<cr><0xC6><0xE2>`
The server or gateway checks host commands for a two-digit CRC-16 value following <cr>.

Example 2

Command: `R<cr><0xFC><0xA5>`

Response: `OK<cr><0xC6><0xE2>`

This response assumes the setting from Example 1.

CMT10 (Enable Extended Response Mode)

CMT10 is a data checking mechanism that adds a character count to all responses from the server or gateway to the host.

Character counts are one greater for CMT10,2 than for CMT10,1 because the CMT10,2 count includes the carriage return at the end of the string. CMT10,1 does not.

Host Command

`CMT10,x<cr>`

Server or Gateway Response

`OK<cr>`

X Variable Values

- **0**
  Disables the extended response *(default)*.
- **1**
  Enables the extended response suffix. A comma and a decimal character count of the string follow each string sent to the host.
- **2**
  Enables the extended response prefix. A decimal character and a comma precede each string.

Example 1

Command: `CMT10,1<cr>`

Response: `OK,2<cr>`

Example 2

Command: `CMT10,2<cr>`

Response: `0003,OK<cr>`

Example 3

Command: `CMT10<cr>`

Response: `0002,2<cr>`

This response assumes the setting from Example 2.

CMT13 (Automatically Enable Computer)

Use CMT13 to automatically enable computers. When the server or gateway receives a power-up message from a computer, the computer is automatically included in the active computer array.

Host Command

`CMT13,x<cr>`

Server or Gateway Response

`OK<cr>`

X Variable Values

- **0**
  Disables the computers. The E (Enable) command must explicitly enable the computer. If the E command does *not* enable a computer, and the server or gateway receives a power-up message from that computer, the power-up is ignored and the power-up message (*t*P) is not sent to the host *(default)*.
- **1**
  Automatically enables the computers.

Example

The server or gateway enables all responding computers:
Command: CMT13,1<cr>
Response: OK<cr>

CMT21 (One-Character Address Mode)

If the server or gateway is configured for one-byte addressing (also "RM2216" or "old" mode), it powers up and behaves like an RM2216 Multiplexer. Computer power-up messages have the form "1P", where "1" is the one-character address.

If the *x* variable for CMT21 is set to "1," *and* the server or gateway is configured for one-byte addressing, computer power-up messages have the form "1P,060" where "1" is the one-byte address and ",060" is the computer power-up type. The computer power-up type resembles the standard three-byte power-up because of the computer type, but the one-byte address appears.

The default setting is "0," which disables one-byte addressing. If the server or gateway is configured for three-character ("normal") addressing, the CMT21 command does nothing.

Host Command

CMT21,*x*<cr>

Server or Gateway Response

OK<cr>

Example 1

Command CMT21,1<cr>
Response: OK<cr>
Sets one-character addressing, Real Time Control (RTC) mode if host mode is RM2216.

Example 2

Command CMT21<cr>
Response: 1,<cr>
This response reports the setting from Example 1.

CMT22 (Multiple Buffering Mode)

Note: Multiple buffering is enabled on the Intermec Application Server.

Multiple Buffering Mode enables the server or gateway to buffer commands the host sends. Up to 25 commands per computer may be buffered before the server or gateway rejects the host command with a "?10" (out-of-buffers) response. This may happen if the host program issues commands to a computer that is not powered up or is out of range. The computer must act on the commands before the server or gateway allows the host to send more commands.

If CMT22 is disabled, the server or gateway handles commands in a transaction-based order. The device expects the host to wait until the response to the command is sent back to the host before the host sends the next command. If the host program violates this protocol by sending several commands to the server or gateway in succession without the device fully acting on the command, the commands may be overwritten and lost.

Host Command

```
CMT22,x<cr>
```

Server or Gateway Response

```
OK<cr>
```

X Variable Values

- **0**
  Disables the multiple buffering mode *(default)*.
- **1**
  Enables the multiple buffering mode.

Example 1: Multiple buffering disabled

Suppose the host program issues the time to a specific computer every second. For a Native computer, the command syntax may be W000DB/08:38:00<cr>, followed one second later by W000DB/08:38:01<cr>, W000DB/08:38:02<cr>, etc. If multiple buffering is disabled, the transaction may occur as follows.

If multiple buffering is disabled, the server or gateway does not guarantee that the data goes to the computer. Results are unpredictable, and depend on factors such as transaction throughput and host overhead. The example above shows that the fifth host command overwrote the second, third, and fourth commands in the server or gateway. In this case, the second, third, and fourth commands were not sent to the computer.

Host Commands

Host Command	Server or Gateway Response	Computer Display
W000DB/08:38:00	OK<cr>	08:38:00
W000DB/08:38:01	OK<cr>	08:38:00
W000DB/08:38:02	OK<cr>	08:38:00
W000DB/08:38:03	OK<cr>	08:38:00
W000DB/08:38:04	OK<cr>	08:38:04

Example 2: Multiple buffering enabled

If multiple buffering is enabled, host commands are not overwritten. Instead, they are buffered in the server or gateway and sent to the computer sequentially. (However, there is no guarantee that these commands will arrive at the computer 1 second apart.) Following is an example.

Host Command	Server or Gateway Response	Computer Display
W000DB/08:38:00	OK<cr>	08:38:00
W000DB/08:38:00	OK<cr>	08:38:00
W000DB/08:38:01	OK<cr>	08:38:00
W000DB/08:38:02	OK<cr>	08:38:00
W000DB/08:38:03	OK<cr>	08:38:01
W000DB/08:38:04	OK<cr>	08:38:02

CMT27 (6910 IGAP Operating Mode)

CMT27 is a mode-switching command that changes the operating mode on the 6910 Integrated Gateway/Access Point (IGAP). You must change the operating mode before you can configure the device parameters.

To configure the 6910 IGAP, you establish a local session with it through a VT100 computer emulation program (such as PROCOMM PLUS) installed on a personal computer. When the program is running, you issue command “CMT27,2” to switch the 6910 IGAP DIAG port from host mode to configuration mode.

When entering the command, follow these guidelines:

- Type CMT in uppercase and do not enter spaces between characters.
- There is a 1-second timeout period between characters. If you wait more than 1 second to type the next character, the command is ignored.
- If you type the wrong character, you cannot use the backspace key to correct it. You must wait 1 second or press [Enter] before trying again.

When you exit the 6910 IGAP configuration menus, the port automatically switches to host mode.

For complete information about configuring the 6910 IGAP, refer to the *6910 Integrated and 6910 Telnet Gateway/Access Points User's Guide* (P/N 961-047-122).

Host Command

CMT27,2<cr>

6910 IGAP Response

OK<cr>

Example

Command: CMT27,2<cr>
Response: OK<cr>
The 6910 IGAP DIAG port is in configuration mode.

CMTI (Restore Default Parameters)

CMTI restores all parameter commands to their default values.

Host Command

CMTI<cr>

Server or Gateway Response

Not applicable

Parameter Commands and Default Values

- **CMT0** 2
- **CMT1** 255
- **CMT2** 0
- **CMT4** 0
- **CMT5** 0
- **CMT8** 0
- **CMT9** 0
- **CMT10** 0
- **CMT13** 0
- **CMT21** 0
- **CMT22** 0
- **CMT27** Host mode

? (Get Transmit Message Status)

Use the ? command to request the status of the server or gateway transmit buffer. The device responds by returning a status code for each computer in the network (up to 127 codes).

Host Command

?<cr>

Server or Gateway Response

xxx...x<cr>

? (Get Transmit Message Status) Codes

- **A** Active computer
- **N** Disabled computer
- **C** Computer has a pending message

- **P** Inactive computer has a pending message
- **I** Inactive computer

Example

Command: `?<cr>`
Response: `CCCCAAAAPPPPNNN0N<cr>`
This response has 127 characters and shows the following:

Computers 000-003 have pending messages.
Computers 004-007 are active.
Computers 008-011 have pending messages, but are inactive.
Computers 012-126 are disabled.

E (Enable Computers)

The E command enables the computers (communication channels) in a network. You can either enable all or a specific computer with one command.

Host Command

`Es0s<cr>` (enables computers 000-126)
`Ets` (enables a specific computer)

Server or Gateway Response

`OK<cr>`

Variables

- *s*
  N = No *(disables computer)*
  Y = Yes *(enables computer)*
  X = "Do not care" *(retains current status)*
- *t*
  Computer number *(000-126)*

The first Y, N, or X setting applies to computer 000, the second applies to computer 001, the third to computer 002, etc. You do not need to list all the computers.

Before a computer can be polled or sent a message, it must be enabled by the host or allowed to enable and identify itself to the server or gateway. Enabling a computer opens a channel to that computer.

Example 1

Command: `EYYYYYXNNNNNNYYYYYYYYYY<cr>`
Action: Enables computers 000-004 and 012-021. Does not change computer 005. Disables computers 006-011. Does not change computers 022-126.

Example 2

Command: `E001Y<cr>`
Action: Enables computer 001.

R (Read)

The R command is the idle, polling command. Use it to enable the server or gateway to transmit computer data to the host. This is the only command where the device can send the computer responses. If the server or gateway has a computer response queued for transmission to the host, it cannot send it in response to a Write Display (WtD) command, but only in response to an R command.

Host Command

```
R<cr>
```

Server or Gateway Response

```
tK<cr>
tS<cr>
tP,x<cr>
tD
OK<cr>
```

Variables

- ***t*** Computer number *(000-126)*
- **P** Computer reset/turned-on
- **K** Keyboard data
- ***x*** Computer power-up type
- **S** Scanned data
- **D** Echo data

In general, an R command must follow the commands the host sends to the server or gateway. This enables the server or gateway to respond to or forward any data it received from a computer and stored.

The server or gateway sends an OK<cr> to the host when interactive mode is enabled, and the CMT1 time-out value was exceeded and the device has no data to send to the host.

The following table lists the computer power-up types. The logical screen size for all computers is 80x24, of which the computer displays a portion. Power-up type “000” means the computer has not powered up.

Computer Power-Up Types

Data Stream	Host View Size	Power-Up Type
Native	16	032
	21	033
	Any size other than 16 or 21	060
Non-Telnet 3270	16	034
	21	035
	80	037
	Any size other than 16, 21, or 80	061
Telnet 3270	Any size	071

Computer Power-Up Types (continued)

Data Stream	Host View Size	Power-Up Type
Non-Telnet 5250	Any size	062
Telnet 5250	Any size	072
Non-Telnet VT220	Any size	063
Telnet VT220	Any size	073

Example

Host: `R<cr>`
Response: `000P,032<cr>`
Native computer number 000 with host view size set to 16 has reset or powered up. (The power-up type is 032.)

T (Repeat Response)

The T command requires the server or gateway to retransmit its last message to the host. Use it in case of a partially missed or corrupted message as indicated by a parity, framing, or length error the host received.

Host Command

`T<cr>`

Server or Gateway Response

`<cr>`

Example

Command: `T<cr>`
Action: The server or gateway repeats the last message sent to the host, and adds a carriage return.

F (Return Computer Power-Up Type)

The F command lists all computers in a system. The F command returns a power-up type for each channel in the network (127 numbers). The response comes back in a series of codes, one for each *channel,* followed by a <cr>. For more information, see "Computer Power-Up Types" on page 252.

Host Command

`F<cr>`

Server or Gateway Response

`x,x,x0x,x,x,x<cr>`

Example

Command: `F<cr>`
Response:

```
032,032,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000<cr>
```

This example indicates that:

- Channels 0 and 1 are Native computers with host view size set to 16. (The power-up type is 032.)
- Channels 2-126 have not powered up. (The power-up type is 000.)

DME (Echo-Back Diagnostics)

Use DME to check the communication link between the host and the server or gateway. The server or gateway receives data from the host and sends an identical data string back to the host for comparison. The command's *x* variable is a maximum of 1024 characters of host data.

Host Command

```
DME/x<cr>
```

Server or Gateway Response

```
x<cr>
```

Example

Command: `DME/Host-to-gateway loopback test<cr>`
Response: `Host-to-gateway loopback test<cr>`

DMP (Reset)

DMP resets the server or gateway. The power-up message returns to the host when the self-tests are complete. Tests take several seconds.

Power-up restores the default parameters. Redefine the operational parameters that differ from the default parameters after this command.

Host Command

```
DMP<cr>
```

Server or Gateway Response

`,0<cr>`
`POWER UP QXSASYNC V<x.xx>,127<cr>` *(three-byte addressing)*

or

`,0<cr>`
`POWER UPMUX V<x.x>` *(one-byte "old" addressing)*

Example 1

The following sample response appears when the server or gateway is configured for three-byte ("normal") addressing:

Command: `DMP<cr>`
Response: `,0<cr>`
`POWER UP QXSASYNC V2.10,127<cr>`

Example 2

The following sample response appears when the server or gateway configured for one-byte ("old") addressing:

Command: `DMP<cr>`
Response: `,0<cr>`
`POWER UPMUX V2.2<cr>`

DMV (Send Software Version)

DMV sends the server or gateway's software version number to the host. The value for variable *x.xx* is the version number.

Host Command

`DMV<cr>`

Server or Gateway Response

`QXSASYNC V<x.xx><cr>`

Example 1

The following sample response appears when the server or gateway is configured for three-byte ("normal") addressing:

Command: `DMV<cr>`
Response: `QXSASYNC V2.10<cr>`

Example 2

The following sample response appears when the server or gateway is configured for one-byte ("old") addressing:

Command: `DMV<cr>`
Response: `FWMUXB8 V2.2<cr>`

Normal Communication Flow

The two basic types of commands are:

- Commands destined for the server or gateway
- Commands destined for the computer

Assume the system is set to interactive mode (CMT2,1). In general, the server or gateway can respond immediately to parameter or diagnostic commands. Following are two examples:

Example 1

Host command: `DMV<cr>`
Response: `QXSASYNC V2.10`

Example 2

Host command: `?<cr>`
Response: `IIIIIIIINNNNNNNN`

The server or gateway accepts the computer commands (with an OK<cr>) and then sends them to the computer. The server or gateway receives the computer response. After the host issues an R Read command (page), the server or gateway can send the computer response.

In general, a computer response cannot be sent in response to a non-Read command. An exception to this is the T (Repeat Response) command (page), where the last server or gateway response is retransmitted to the host. Following are two examples:

Example 1

Host command: `W000D/Hello World<cr>`
Response: `OK<cr>`

Example 2

Host command: `R<cr>`
Response: `000KXYZ`

Write Display Errors

Error responses from the server or gateway take the form "?*x*"<cr>, where *x* is the numeric error code listed in the following table.

Write Display Errors

Error Code	Name	Description
0	Communications Error	Low level, asynchronous driver detected a parity, stop, or other data error.
1	Syntax Error	Error in the parameter, diagnostic, or computer command. For the WtD (Write Display) computer command, syntax checking stops at the `/'.

Write Display Errors (continued)

Error Code	Name	Description
4	Length Error	Host message is too long. If the server or gateway is configured for three-byte addressing, the maximum length is 1000 bytes. For one-byte addressing, the maximum length is 128 bytes.
5	Disabled Error	Host is trying to send data to a disabled or inactive computer (has not received a power-up message from it).
6	Gap Error	Intercharacter gap time exceeded (timeout between start of message and carriage return). CMT5 Gap Time-Out (page 220) sets the time.
8	Check Error	Occurs when CMT8 Set Data Error Checking (page 220) is set for CRC checking, and the CRC did not compute on a received host frame.
10	Buffer Error	Multiple buffering is enabled, and a computer exceeded its allocation of 25 buffers. Corrective action is to wait for the buffers to be freed.
11	Sequence Error	R (Read) commands were issued before CMT commands were sent to initialize the server or gateway.

Native Computer Commands

This chapter describes each computer command with detailed information.

G (Reset Computer)

The G command returns the computer to its first “state,” a known condition. All computers reset when variable *t* (computer number) is omitted. Computers respond to a poll with their power-up message.

In the command, the *t* range is 000-126. If the server or gateway is configured for three-byte addressing, *t* is three digits. For one-byte addressing used by some Intermec® legacy systems, *t* is a character from 0-9 or A-F.

Command

`Gt<cr>` *(reset specific computer)*
`G<cr>` *(reset all computers)*

Server or Gateway Response

`OK<cr>`

R Response

`tP,x<cr>`

R Response Variables

- *t* Computer number *(000-126)*
- P Computer has reset or powered-up
- *x* Computer power-up type
  For more information, see **"Computer Power-Up Types" on page 252**.

Example

Command: `G002<cr>`
Action: Server or gateway resets computer 002.
R Response: `002P,032<cr>`

WtB (Audio Annunciator)

W*t*B turns on the computer audio tone for a specified time, in increments of 1/4 second up to 2.25 seconds.

Command

`WtBd<cr>`

Server or Gateway Response

`OK<cr>`

R Response

`tC0<cr>`

Command and R Response Variables

- *t* Computer number *(000-126)*
- *d* Audio tone duration times 250 milliseconds *(1-9 seconds)*

Example

Command: `W010B8<cr>`
Action: Computer 010 sounds tone for 2 seconds.

WtD/ (Write Display)

W*t*D/ sends data to a computer, enables its response, downloads the prompt which appears on the computer, and sets up input parameters. The single input field is returned to the host.

Command

`WtDo/x<cr>`

Server or Gateway Response

`OK<cr>`

R Response

```
tKrd<cr>
tSrbcm<cr>
```

Data returns to the host when:

1. You press the [ENTER] key.
2. The field is filled and the [ENTER] option is disabled.
3. The display scrolls to the top or bottom with the wrap disabled.
4. The scanner reads a valid bar code.

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** Options *(see the "Write Display Options" table below)*
- ***x*** Data

|
(0x7C) Allows all command extensions. For more information, see "Command Extensions" on page 260.

Write Display Options

o	Description	Default
.	The "." (decimal point) key is always allowed if the N option is not set.	Not allowed if the N option is set.
-	The "-" (minus sign) key is always allowed if the N option is not set.	Not allowed if the N option is set.
B	Sounds tone (S*t*D/ sets the duration)	Tone off
C	Displays cursor	Not displayed
D	Displays keyed-in data	Not displayed
E	[ENTER] required to transmit data	Not required
F	Current row returns as first character. All information is sent as a single block, not 112-byte blocks.	Not returned
H*x*	*x*=maximum length of keyboard input data. Values are 0-109.	3
L*x*	*x*=minimum length of keyboard input data. Values are 0-109.	0
M	MOD 10 check if option N is set, and options "." and "-" are not set	No check
N	Numeric input data only	Alphanumeric
O	When the O option is enabled, the line with the cursor is in reverse video.	All lines shown in standard video
P*x*	*x*=starting position of keyboard input. Values are 0-127.	0
Q	When the Q option is used and the [SKIP] key is pressed, the computer automatically transmits the [SKIP] key to the host.	If Q is not used, the [SKIP] key is not auto-transmitted.
S	Enable scanning	Not enabled
U	Sets keyboard to SHIFT mode	Not shifted
W	Enables scroll wrap (up or down)	Not enabled

R Response Variables

- *t* Computer number *(000-126)*
- K Keyboard data
- *r* Optional ASCII display row number
- *d* Keyboard input data
- S Scanner data. If “F” option is enabled, response is always “S.”
- *b* Bar code type
- *c* Bar code data
- *m* MOD 10 check digit, if enabled

Example

`W002DDCBSH6L3P16/SCAN BARCODE<cr>`

where:

- D=display key data in the display
- C=cursor displayed
- B=beep when displayed
- S=enable scanning
- H6=maximum key data is 6
- L3=minimum key data is 3
- P16=cursor is in position 16

Response:

```
SCAN BARCODE
```

Command Extensions

The vertical bar character “|” allows the following extensions.

Bank

Data to substitute for |D*xx* allows for a macro extension defined by the W*t*D/ (Set Bank) command. Following are Bank limitations:

- Banks cannot be nested.
- Banks cannot recursively call other banks inside the extension.
- A field retains its own prescribed limitations.
- W*t*D/ (Write Display) command of up to 1000 characters.

Syntax: *|Dxx (00 3 xx 3 99)*

Blank

Syntax: `|xx (00 3 xx 3 99 blank spaces)`

Beep

The computer beeps for the length defined by the S*t*D/ command for each occurrence of the |B extension.

Syntax: `|B`

Cursor Position

The cursor (if enabled) appears at this location.

Syntax: `|C`

Pass-Through Hex Stream

This extension enables ASCII representations of hexadecimal values. The "|X|" starts the hex stream. The "|A" ends it.

Syntax: `|X<literal>|A`

Example

"|XDAC4B3|A" translates the literal "DAC4B3" to hex values 0xDA, 0xC4, and 0xB3 and displays the following:

┌─│

Pass-Through Hex Character

This sends hexadecimal characters to the computer. It is a pass-through mode that bypasses the server or gateway's handling of this character.

For example, if you want to send a carriage return to a printer attached to a computer, and if the host program sent the <cr> character, the server or gateway would terminate the command. By sending a ^0D, the host sends these three characters through the server or gateway to the computer. The computer converts the three characters into one byte.

Syntax: `^xx`

Example 1

This command sends the string "Hello World<cr><lf>" out the computer's RS-232 port:

```
W000D/\P\Hello World^0D^0A<cr>
```

Example 2

This command displays "ABCD" on the computer's screen. This command is equivalent to command W000DDBC/ABCD.

```
W000DDBC/^41^42^43^44<cr>
```

WtD// (Extended Write Display)

Use W*t*D// to download more than one data block to a computer and enable computer response mode. W*t*D// concatenates (appends) strings of data. It also permits setting up input parameters for the computer's response. The single input field returns to the host upon data entry.

Command

```
WtD//Fx<cr>
WtD//Mx<cr>
WtDo//Lx<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tKrd<cr>
tSbcm<cr>
tKz<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- **F** First in chain
- **M** Middle of chain
- **L** Last in chain
- ***o*** Options. For more information, see "Write Display Options" on page 259.

  x Data

  |
  (0x7C) Allows all command extensions. For more information, see "Command Extensions" on page 260.

R Response Variables

- ***t*** Computer number *(000-126)*
- **K** Keyboard data
- ***z*** Response code
  0=Good Status
  1=Syntax Error
- **S** Scanned data
- ***b*** Bar code type
- ***c*** Bar code data
- ***m*** MOD 10 check digit *(if enabled)*
- ***d*** Keyboard input data
- ***r*** Optional ASCII display row number

When using the W*t*D// command, designate the first, middle, and last segments of a chain as follows:

- /F=(first in chain) – Computer clears all chained data and replaces it with new data starting with the marked sequence.
- /L=(last in chain) – Computer attaches the current field, and executes the Write Display command using options set for the /L command.
- /M=(middle of chain) – Computer attaches sequence to what is stored.

Example 1

Command: W003D//FFIRST IN CHAIN <cr>
Response: OK<cr>

Example 2

Command: R<cr>
Response: 003K\\0<cr>

Example 3

Command: W003D//MMIDDLE IN CHAIN <cr>
Response: OK<cr>

Example 4

Command: R<cr>
Response: 003K\\0<cr>

Example 5

Command: W003DBS//LLAST IN CHAIN<cr>
Action: Computer 003 sounds its tone and enables its scanner.

```
FIRST IN CHAIN
MIDDLE IN CHAIN
LAST IN CHAIN
```

This illustrates a sample computer display for this command.

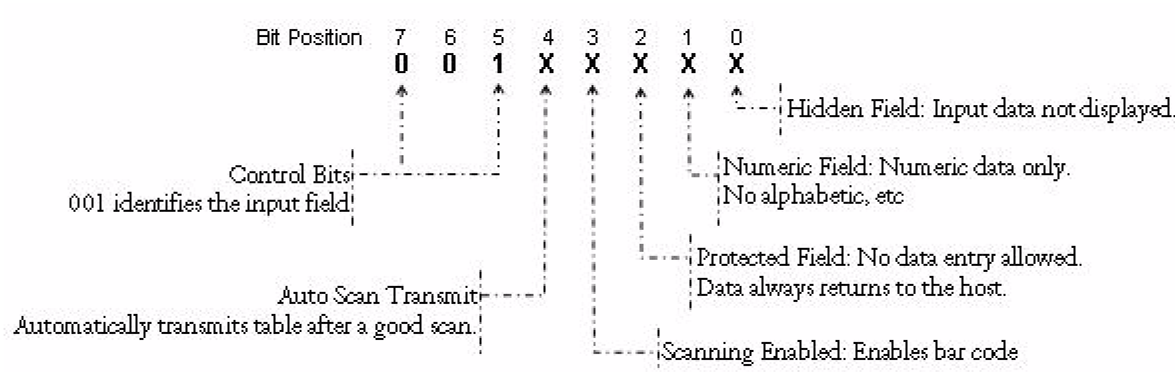
WtD/ \S (Download Table)

A *table* is a block of prompt and input field pairs downloaded to the computer. The computer can store a number of tables at the same time. Host commands activate individual tables.

While a table appears, the computer operator can enter data to any or all its *input* fields before the input data is transmitted to the host.

WtD/\yS defines and sends a table to the computer. The computer stores the table definition until a WtD/\y (Display Table) command is issued.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.



Command

```
WtDo/\ys\p\fx\0\p\fx<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Computer number *(000-126)*
- *o* Erase options
  D=Erase Table
  E=Erase all tables
- *y* Table number *(0-99, Length is 16725 bytes; # pairs is 25)*
- *p* Prompt field *(Maximum field size)*
- *f* Input field control byte (001*XXXXX*)
  For more information, see
  "Input Field Control Byte" on page 265.
- *x* Input field

  |
  (0x7C) Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

R Response Variables

- *t* Computer number *(000-126)*
- *o* Table response codes
  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number
  5=Prompt Field, No Input Field
  6=Field Does Not Exist

7=Missing Prompt and Input Field Pair
8=Field Pair Number Out of Range
9=Format Error or Syntax Error

Input Field Control Byte

The input field control byte sets parameters for the fields. To set field parameters, choose the required options, then convert the control byte to its ASCII character. (Several options combine into 1 byte.) The following image illustrates the definition input field control byte bits.

For example, set the byte for a field that will receive scanned data. The first 3 bits, 001, identify the control byte. The “Scanning Enabled” position (bit 3) is set to 1 while the other four bits remain 0. The complete byte is 00101000 (28 hex). The ASCII character designated by the binary number 00101000 (28 hex) is “(”. Therefore, placing the character “(” into the first position in an input field enables scanning in that field.

Rules for Designing Tables

The following rules apply to tables:

- A table can consist of up to 25 prompt and input field pairs.
- A prompt field must be followed by an input field.
- A prompt, input field pair must be smaller than the computer's screen.
- Table length=(characters)+(field characters)+ 2•(input pairs)

Note: The total number of prompt field characters in tables plus the total number of input field characters in all pairs plus twice the number of pairs must be less than 1600 for tables 0 through 4 and less than 400 for tables 5 through 99.

The maximum length of the command is 1020 bytes. For longer tables, you can use more than one W*t*D/\yS (Download Table) command. The D (erase table) or E (erase all tables) option is required before overwrite.

Example

Host:
W010DD/\0S\1ST BARCODE:|04\(|16\TY?\”|11<cr>
Response: OK<cr>

Table 0 in computer 010 erases and a new set of prompt and input field pairs loads.

The first prompt is “1ST BARCODE:” followed by a blank extension of 4 spaces moving the input field to the next line on a 16-character display.

The first character in the input field is the input field control byte. A “(” (28 hex) enables bar code scanning. A blank extension of 16 characters follows the input field control byte. This defines the length of the input field.

The second prompt is “QTY?”. The control byte is the ASCII character “ (22 hex), which enables numeric keyboard data only. This field contains 11 spaces after the ASCII control byte character, setting the input field length to 11.

```
FIRST BARCODE:
QTY?
```

This is a sample computer display.

WtD/ \y (Display Table)

W*t*D/\y displays table *y* defined by the W*t*D/\yS (Download Table) command, and enables input to the table. Data transmits to the host in response to an R (Read) command.

Command

```
WtDo/\y<cr>
```

Server or Gateway Response

OK<cr>

R Response

```
tk\y\n\gx\0\n\gx<cr>
tk\yM\n\gx\0\n\gx<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** Options *(see the following table)*
- ***y*** Table number *(00-99; Length is 16725 bytes; # pairs is 25)*

WtD/\y (Display Table) Options

Option	Description	Default
B	Sounds tone (S*t*D/ sets the duration).	Tone off
C	Displays computer cursor.	Not displayed
F	Sends all data in a single block, not 112 bytes per R (Read) command.	Maximum of 112 bytes
L*x*	Designates prompt and input field pair to appear at the top of the screen (*x*=prompt and input field pair number).	0
M	Skips protected fields when scrolling through a table.	Not enabled
N	Sets keyboard to SHIFT mode.	Not shifted
O	Resets all modified flags.	Not reset
P*x*	Desired prompt and input field pair starting position in table. *x*=0 up to the number of field pairs in the table minus 1.	0
Q	Enables autoadvance scanning mode. A good scan automatically advances the cursor to the next input field.	Not enabled

WtD/\y (Display Table) Options (continued)

Option	Description	Default
S	When the input field is filled by scanning or keyboard input, the cursor automatically moves to the next input field.	
U	Set all input fields to blank and reset the modified flags (protected fields unaffected).	Not affected
W	Enable display wrap (top-bottom, bottom-top).	Not enabled

R Response Variables

- ***t*** Computer number *(000-126)*
- ***y*** Table number *(0-99; Length is 16725 bytes; # pairs is 25)*
- ***n*** Input field number *(0-99)*
- ***g*** Input data type. If *g*=S and *x*=bar code type
  - *K=Keyboard*
  - *S=Scanner*
- ***x*** Data
- **M** Indicates more data to come. Host continues to send an R (Read) command until it receives a response without the "more data" flag. *(The R response without M means all data was sent.)*

Example

Using the table from the W*t*D/\yS (Download Table) command:

Host command: `W010DQ/\0<cr>`
Response: `OK<cr>`

Computer allows the operator to enter data in table 0 of computer 010.

Scanning autoadvance is enabled. The computer automatically advances to the next input field after getting a valid scan.

After the user presses [Enter], the computer responds to an R (Read) command from the host by uploading the stored input field data.

WtD/ \yT (Download and Display Table)

This combines W*t*D/\yS (Download Table) and W*t*D/\y (Display Table) commands. It downloads prompts and input fields into a buffer memory of the computer for processing, displays the table, and enables data input.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.

Command

```
WtDo/\yT\p\fx\p\fx\p\fx\
0\p\fx<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\y\n\gx\0\n\gx<cr>
tK\yM\n\gx\0\n\gx<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** Erase options
  - *D=Erase Table*
  - *E=Erase All Tables*
- ***y*** Table number *(0-99; Length is 16725 bytes; # pairs is 25)*
- ***p*** Prompt field *(Maximum field size)*
- ***f*** Input field control byte *(001XXXXX)*
  For more information, see
  "Input Field Control Byte" on page 265.
- ***x*** Input field
  |
  (0x7C) Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see
  "Command Extensions" on page 260.

Note: See "Rules for Designing Tables" on page 265 to learn about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- ***t*** Computer number *(000-126)*
- ***y*** Table number *(0-99)*
- ***n*** Input field number *(0-99)*
- ***g*** Input data type. If *g*=S and *x*=bar code type
  - *K=Keyboard*
  - *S=Scanner*
- ***x*** Input field
- **M** Indicates more data to come. Host continues to send an R (Read) command until it receives a response without the "more data" flag. (*The R response without M means all data has been sent.*)

- *z* Response code

  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number
  5=Prompt Field, No Input Field
  6=Field Does Not Exist
  7=Missing Prompt and Input Field Pair
  8=Field Pair Number Out of Range
  9=Format Error or Syntax Error

Example

Command:

```
W000DBCUWQS/\0T\DATE:|03\$02–01–00\TIME:|03\$13:48.00\
PROMPT ONE|05\8|16\ PROMPT TWO|05\8|16<cr>
```

R Response: OK<cr>:

```
DATE:   02-01-00
TIME:   13:48.00
 PROMPT ONE
 PROMPT TWO
```

The illustration is a sample computer display.

WtD/ \yU (Update Table)

W*t*D/\yU allows the operator to change a portion of a table without downloading an entire table. You can accomplish this by one of these methods:

- Replace the prompt and input fields pair together.
- Replace only the prompt field.
- Replace only the input portion of the prompt and input field pair.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.

Command

```
WtDo/\yU\i\{Fp\fx}
```
or
```
{Pp}
```
or
```
{Ifx}0<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Note: Alternative replacements are in braces { }. The "..." indicates that you may use this command multiple times.

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** Erase options
  D=Erase Table
  E=Erase All Tables
- ***y*** Table number *(0-99; Length is 16725 bytes; # pairs is 25)*
- ***i*** Input field number *(0-99)*
- **F** Update the prompt and input fields together
- ***p*** Prompt field *(Maximum field size)*
- ***f*** Input field control byte *(001XXXXX)*
  For more information, see "Input Field Control Byte" on page 265.
- ***x*** Input field
- **P** Update prompt field only
- **I** Update input field only

| *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

Note: See "Rules for Designing Tables" on page 265 to learn about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- ***z*** Response code
  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number
  5=Prompt Field, No Input Field
  6=Field Does Not Exist
  7=Missing Prompt and Input Field Pair
  8=Field Pair Number Out of Range
  9=Format Error or Syntax Error

Example 1

Host command:
W000DD/\0S\DATE:|03\$02–01–00\TIME:|03\$13:48.00\
PROMPT ONE|05\(|16\ PROMPT TWO|05\8|16<cr>
Response: OK<cr>
R response: 000K\\0<cr>

```
DOWNLOADING
TABLE
PLEASE WAIT
```

Example 2

Host command: W000DBCUWQS/\0<cr>
Response: OK<cr>
R response: 000K0\\0<cr>

```
DATE:  02-01-00
TIME:  13:48.00
 PROMPT ONE
 PROMPT TWO
```

Example 3

Host command: W000D/\0U\|\I14:01.32<cr>
Response: OK<cr>
R response: 000K\\0<cr>

```
DATE:  02-01-00
TIME:  14:01.00
 PROMPT ONE
 PROMPT TWO
```

Replacing Pairs Together

The syntax “Fp\fx” updates field “i” with a new prompt and input field pair, “p\fx”, where:

- p=prompt field
- f=input field byte
- x=input field spaces or data

Restrictions

- If the field is already defined but is not the last field entered, the length of the new prompt and input field must be shorter than the original prompt and input field.
- The new pair uses the same screen buffer space as the original pair.
- If the new field is last, and the original field is defined, the new field can be as large as screen size within the 1600 total maximum.
- If the new field is last, and the original field is not defined, the new field must be one or more greater than the number previously assigned but less than 25.
- The table must not exceed the 1600-character limit.

Replacing the Prompt Field

The syntax “Pp” updates prompt field “i” with a new prompt “p”, where:

p=prompt field

Note that the field must be defined and the new prompt field must be less than or equal to the length of the original prompt field.

Replacing the Field Only

The syntax “Ifx” updates input field “i” with a new field, “fx,” where:

- f=field input byte
- x=input field spaces or data

Note that the field must be defined and the new input must be less than or equal to the length of the original input field.

WtD/ \yV (Update and Display Table)

W*t*D/\yV combines the W*t*D/\yU (Update Table) command, which allows the operator to change a portion of a table without downloading an entire table, and the W*t*D/\y (Display Table) command. You can accomplish this by one the following methods:

- Replace the prompt and input field pair together.
- Replace only the prompt field.
- Replace only the input portion of the prompt and input field pair.

Individual hexadecimal characters can be escaped with the “^*xx*” extension. For more information, see “Pass-Through Hex Character” on page 261.

Command

```
WtDo/\yV\i\{Fp\fx}
```
or
```
{Pp}
```
or
```
{Ifx}0<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\y\n\gx\0\n\gx<cr>
tK\yM\n\gx\0\n\gx<cr>
tK\\z<cr>
```

Note: Alternative replacements are in braces { }. The "..." indicates that you may use this command multiple times.

Command Variables

- *t* Computer number *(000-126)*
- *o* Options. For more information, see "WtD/\yR (Update and Print Table) Options" on page 290.
- *y* Table number *(0-99; Length is 16725 bytes; # pairs is 25)*
- *i* Input field number *(0-99)*
- **F** Update the prompt and input fields together
- *p* Prompt field *(Maximum field size)*
- *f* Input field control byte *(001XXXXX)* For more information, see "Input Field Control Byte" on page 265.
- *x* Input field
- **P** Update prompt field only
- **I** Update input field only

  | *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

Note: See "Rules for Designing Tables" on page 265 to learn about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- *t* Computer number *(000-126)*
- *y* Table number *(0-99; Length is 16725 bytes; # pairs is 25)*
- *n* Input field number *(0-99)*
- *g* Input data type *(K=Keyboard; S=Scanner)*
- *x* Input field

- **M** Indicates more data to come. Host continues to send an R (Read) command until it receives a response without the "more data" flag. (*The R response without M means all data has been sent.*)
- *z* Response code
  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number
  5=Prompt Field, No Input Field
  6=Field Does Not Exist
  7=Missing Prompt and Input Field Pair
  8=Field Pair Number Out of Range
  9=Format Error or Syntax Error

Example

Host:
`W000DD/\2S\User Name: |06|3\Department: |05\|16\Job Title: |06\|16 <cr>`

R response: `003K\\0<cr>`
Host: `W000DBCUQ/\2<cr>`
Response: `OK<cr>`

```
User Name:
Department:
Job Title:
```

Suppose the user enters the following data:

```
User Name:
John Doe
Department:
Applications
Job Title:
Programmer II
```

When the user presses [ENTER], the R response is:

```
003K\2\0\KJohn Doe\1\
KApplications\2\KProgrammer II<cr>
```

If you want to change the first prompt and input field pair to receive the user ID number without changing the information displayed, the W*t*D/\yVF (Update and Display Table) command allows the host to transmit to the computer only the necessary changes. A command may be:

Host: `W003DBCQ/\2V\0\FUser Number: |04\|16<cr>`
Response: `OK<cr>`

Note that the computer does not return a status on the display commands until it transmits the data back to the host. The length of the prompt and input fields stayed the same, and the input field control byte changed to accept numeric data only. The computer displays:

```
User Number:
Department:
Applications
Job Title:
Programmer II
```

The user might then input:

```
User Number:
111223333
Department:
Applications
Job Title:
Programmer II
```

When the user presses [Enter], the R response is:

```
003K\2\0\K111223333\1\KApplications\2\KProgrammerII
<cr>
```

WtD/\F (Set Computer Function Keys)

W*t*D/\F defines the computer's 25 function keys. Buffers can contain a maximum of 16 characters.

Command

```
WtD/\F\\f1\f2\0\f25<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Computer number *(000-126)*
- *f1...f25* Operation of function keys 1-25R Response Variables
- *t* Computer number *(000-126)*
- *z* Response code
  0=Message Received
  3=Data Overflow
  4=Bad Function Key (outside of range)
  9=Format or Syntax Error

Example

Host: `W000D/\F\\Vendor #1\Vendor #2\Storage #4\Storage #5<cr>`
`W000DDBCH20/Press the F1 Key`
R Response: `000KVendor #1`

WtD/\B (Set Bank)

W*t*D/\B predefines data strings to be used as macros for the Write Display Table and Print commands, to reduce the number of characters the host transmits to the computer.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.

Command

```
WtD/\B\x\x\0x<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***x*** Blank data (macro)

  | *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

R Response Variables

- ***t*** Computer number *(000-126)*
- ***z*** Response code
  0=Command Executed
  2=Bank Length is 0
  3=Bank Area Filled
  4=Bad Bank Number
  9=Format Error

The following information applies to banks:

- Banks store 100 strings (total of 4000 bytes).
- Bank commands are appended to previous bank commands.
- A command begins a new bank; the delimiter concludes it.
- A bank cannot be nested or recursively call itself.

Example

Host command: `W000D/\Hello \World<cr>`
R response: `000K0<cr>`

Host command: `W000DP16H10C/|D00|D01<cr>`
R response: `OK<cr>`

```
Hello World
```

WtD/\BU (Update Bank)

W*t*D/\BU updates existing macros for use by the Write, Display, Table, and Print commands. The Update Bank data field must be less than or equal to the original bank length.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.

Command

```
WtD/\BU\nn\x<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Computer number *(000-126)*
- *nn* Bank number *(00-99)*
- *x* Bank data (or macro)
  |
  (0x7C) Allows the Blank and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

R Response Variables

- *t* Computer number *(000-126)*
- *z* Response code
  0=Command Executed
  2=Bank Length is 0
  4=Bad Bank Number
  9=Format Error

Example 1

Host: `W000D/\Hello \World\Data String\13 Characters<cr>`
R Response: `000K\\0<cr>`

Example 2

Host: `W000D/\BU\01\Longer String<cr>`
R Response: `000K\\2<cr> (Longer String longer than Data String)`

Example 3

- Host: `W000D/\BU\00\Short<cr>`
  R Response: `000K\\0<cr>`

Example 4

Display command: `W000D/|D00<cr>`

WtD/ \BE (Erase Bank)

This removes all banks, or bank numbers *nn* to the last bank stored. Banks from 00 to *nn*-1 remain in bank storage after the command executes.

Command

`WtD/\BEnn<cr>`

Server or Gateway Response

`OK<cr>`

R Response

`tK\\z<cr>`

Command Variables

- ***t*** Computer number *(000-126)*
- ***nn*** Bank number *(00-99)*

R Response Variables

- ***t*** Computer number *(000-126)*
- ***z*** Response code
  - *0=Command Executed*
  - *4=Bad Bank Number*
  - *9=Format Error*

Example 1

`W000D/\B\Hello World\Data String\13 Characters\Bank Field #3<cr>`
R Response: `000K\\0<cr>`

Example 2

```
W000D/\BE01<cr>
```

R Response: 000K0<cr>
Bank data including and following “Data String” is erased.

WtD/\C (Set Screen Mode)

W*t*D/\C switches the screen mode between the 16-character by 9-line and the 21-character by 15-line displays. The power-up screen appears after this command executes.

Command

```
WtD/\Cx<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Computer number *(000-126)*
- *z* Screen mode
  0=16x9
  1=21x15

The screen mode is set with a 6- or 12-byte string. For the 6-byte string, the first three bytes set the screen Y parameters and the last three are screen X parameters. Using the 12-byte string is the same, except the last six set alternate Y and alternate X screen parameters.

R Response Variables

- *t* Computer number *(000-126)*
- *z* Response code
  0=Command Executed
  1=Bad Command
  9=Format Error

WtD/\P (Print)

W*t*D/\P transmits data via the computer's RS-232 serial port to an external device (for example, a printer). This command can control the port setup options *o*, consisting of P, L, H, W, D, and C variables. The W*t*D/S (Set RS-232 Port Options) command can also preset parameters.

Individual hexadecimal characters can be escaped with the “^*xx*” extension.For more information, see “Pass-Through Hex Character” on page 261.

Command

```
WtDo/\P\x<cr>
```

Server or Gateway Response

`OK<cr>`

R Response

`tK\\z<cr>`

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** RS-232 port options *(see the table on the next page)*
- ***x*** Data

  | *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see “Command Extensions” on page 260.

RS-232 Port Options

Option	Alternative	Default
C	XON/XOFF control	No flow control
D	DTR-DSR	No flow control
H*x*	Time-out; RS-232 port ready in *x* sec	3 sec
L*x*	*x*=speed: • 0=9600 bps • 1=4800 bps • 2=2400 bps • 3=1200 bps • 4=300 bps • 5=19200 bps • 6=38400 bps	0 (9600 bps)
P	Sum of “P value” choices. See the table of “P Values” on page 281.	0 (1 stop bit, 8 data bits, even parity, parity disabled)
W	CTS busy	No wait for CTS

R Response Variables

- ***t*** Computer number *(000-126)*
- ***z*** Response code
  0=Data Printed Correctly
  9=Format Error
  A=CTS time-out
  B=DSR time-out
  C=XON/XOFF time-out

Use the following table to find a “P value” for data bits, stop bits, parity, and baud rate. Add the assigned values to produce a unique sum (see example below). No value combination produces the same sum as any other.

P Values

P Value	Alternatives
1	1=2 stop bits; 0=1 stop bit *(default)*
2	2=7 data bits; 0=8 data bits *(default)*
4	4=odd parity; 0=even parity *(default)*
8	8=parity enabled; 0=parity disabled *(default)*

Example

Initialize the RS-232 port of computer 004, and transmit the message “DATA.” Set RS-232 controls to:

- 7 data bits
- Even parity
- Parity enabled
- 1 stop bit
- 4800 bps
- XON flow control enabled
- P value:
  7 data bits= 2
  Even parity= 0
  Parity enabled= 8
  1 stop bit = 0
  P value= 10 (P10)

L value: 4800 bps=1 (L1)
C value: XON Flow Control enabled=C (C)
The command is: `W004DP10L1C/\P\DATA<cr>`

WtD/\PM (Port in Use Message)

W*t*D/\PM defines a message the computer displays when the RS-232 port is in use. If no message is defined, default message “RS-232 PORT IN USE PLEASE WAIT!” appears.

Individual hexadecimal characters can be escaped with the “^*xx*” extension. For more information, see “Pass-Through Hex Character” on page 261.

Command

`WtD/\PMx<cr>`

Server or Gateway Response

`OK<cr>`

R Response

`tK\\z<cr>`

Command Variables

- *t* Computer number *(000-126)*
- *x* Host-defined message to appear during RS-232 port operation

| *(0x7C)* Allows the Bank and Blank command extensions.
For more information, see “Command Extensions” on page 260.

R Response Variables

- *t* Computer number *(000-126)*
- *z* Response code
  0=Good Status
  2=Port In Use (message is larger than screen size)
  3=Data Error or Buffer Overflow

If using a user-defined message, issue W*t*D/M with a message length of 0 to reset it to the default. The minimum defined character string is one character, with the maximum dependant on screen and font sizes.

Example 1

Host: `W000D/\PM Here is the message to appear during RS-232 port commands.<cr>`
Computer response: `OK<cr>`

Host command: `R<cr>`
Computer response: `000K0<cr>`

Status code “0” indicates “good status” for the command. The message, “Here is the message to appear during RS-232 port commands” appears while the RS-232 port is operating.

Example 2 (For use with 144-character screen)

Host: `W000D/\PMWaiting for Data|16|16|16|16|16|16|16Please be patient<cr>`
Computer response: `OK<cr>`

Host command: `R<cr>`
Computer response: `000K\\2<cr>`

Status code “2” is returned because the message string is one character longer than the large character screen can handle. Switch to the 315-character screen and send the same command:

Host: `W000D/\PMWaiting for Data|16|16|16|16|16|16|16Please be patient<cr>`
Computer response: `OK<cr>`

Host command: `R<cr>`
Computer response: `000K\\0<cr>`

Status code “0” indicates good status. The message “Waiting for Data” appears when the RS-232 port is operating.

Example 3

Host: W000D/\PM|991|992|993|16<cr>
Computer response: OK<cr>

Host command: R<cr>
Computer response: 000K\\3<cr>

Status code “3” is returned because the blank extension caused the message to exceed the screen length. Although the host sent only 15 characters to the computer, the total expanded character count is 316.

WtD/\PS (RS-232 Port Options)

W*t*D/\PS establishes the baud rate, data bits, parity, and flow control at the RS-232 port. This command is sent before a W*t*D/\yP (Print Table) command to initialize the port. For more information, see the “RS-232 Port Options” on page 280.

Command

```
WtDo/\PS<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** RS-232 port options. For more information, see “RS-232 Port Options” on page 280.

R Response Variables

- ***t*** Computer number *(000-126)*
- ***z*** Response code
  0=Parameters Set Correctly

Example

Initialize the RS-232 port with the following parameters:

- 7 data bits
- Even parity
- Parity enabled
- 1 stop bit
- 4800 bps
- XON flow control enabled

P value:

7 data bits	= 2
Even parity	= 0
Parity enabled	= 8
1 stop bit	= 0
P value	= 10 (P10)

L value: 4800 bps=1 (L1)
C value: XON Flow Control enabled=C (C)
The RS-232 Port Control syntax is: `WtDP10L1C/\PS<cr>`

WtD/\PG (Get Data from an External Device)

W*t*D/\PG gets continuous data from an RS-232 device, such as an intelligent printer, that sends a continuous data stream to the computer.

Command

```
WtDo/\PGi<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\x<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** RS-232 port options. For more information, see "RS-232 Port Options" on page 280.
- ***i*** RS-232 input options

RS-232 Input Options

i	Function	Range	Default
E*aa*	Ending (delimiter) character ASCII hex code	*aa*=ASCII code from 00-FF hex	
F	Return the Start Character	No parameters	Not returned
M*xx*	Maximum characters to receive	*xx*=decimal number from 00-99	99
N*xx*	Number of delimiter occurrences	*xx*=decimal number from 00-99	00
P	Return error code on any parity error	No parameters	Ignore parity errors before Start Character
S*aa*	Starting character ASCII hex code	*aa*=ASCII code from 00-FF hex	
T*xx*	Time-out (computer waits *xx* sec for data)	*xx*=decimal number from 00-99. If 00, computer waits indefinitely for RS-232 input or key press to terminate.	05 sec

R Response Variables

- *t* Computer number *(000-126)*
- *x* Data
- *z* External Device Response Codes

*t*K\\9	Syntax error
*t*K\\A	Printer time-out on CTS
*t*K\\B	Printer time-out on DSR
*t*K\\C	Printer time-out XON/XOFF
*t*K\\E	Error with E option (00 < *xx* ≤ FF)
*t*K\\K	If T*xx*=00 and a key is pressed, no input data is returned
*t*K\\M	Error with M option (00 < *xx* ≤ 99)
*t*K\\N	Error with N option (00 < *xx* ≤ 99)
*t*K\\P	Data parity error on RS-232 input
*t*K\\S	Error with S option (00 < *xx* ≤ FF)
*t*K\\T	Error with T option (00 < *xx* ≤ 99)
*t*K\\X	Time-out error waiting for input
*t*K\z*x*	=RS-232 input data

Example

The computer connects to an external scale. The scale continuously sends weight readings to the computer in the following format:
00.00 lbs,08.93 lbs,12.68 lbs,12.43 lbs,12.43 lbs,

The comma is a delimiter or field separator character. Four devices available to manage incoming data are:

- Maximum number of characters
- Delimiter character
- Start and delimiter character
- Start and maximize characters

The choice depends on the form taken by the data and its intended use. Following are examples of each of these devices and a brief description of how they may work with the example.

- **Maximum number of characters.** The data in this example appears in 10 character fields.
  Command: W000D/\PGM10<cr>

Computer stops receiving data after 10 characters. The problem is the computer can start receiving data at any point in the 10-character string.
Data: bs,12.43 l
Here the incorrect starting point makes the data meaningless.

Delimiter characters. Using delimiter characters, computer begins receiving data immediately and stops when a delimiter character is received.

Command: W000D/\FGE2CN01<cr>
Data: .43 lbs

- **Start and Delimiter characters.** Using Start and Delimiter characters, the computer starts saving data once the start character is encountered and stops when the delimiter is read.

  Use the F option if the start character should be part of the response. In the example, the first character in the record is unknown because it depends on the weight reading. However, the previous character (the comma) *is* known and could be a start character.

  Command: `W000D/\PGS2CE2CN01<cr>`
  Data: `12.43 lbs,`

- **Using the F option**

  Command: `W000D/\PGS2CE2CFN01<cr>`
  Data: `,12.43 lbs,`

  The start character in this example is not significant and need not be returned.

- **Start and Max Characters.** Using Start and Max Characters, the computer starts saving data once the start character is encountered, and stops after the maximum number of characters is reached.

  Command: `W000D/\PGS2CM10<cr>`
  Data: `12.43 lbs,`

WtD/\PF (Send and Get Data from an External Device)

W*t*D/\PF transmits data to, and receives data from, an external device. It sends a request and receives its response via the computer's RS-232 port. Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Pass-Through Hex Character" on page 261.

Command

```
WtDo/\PFi\y<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\x<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** RS-232 port options. For more information, see "RS-232 Port Options" on page 280.
- ***i*** RS-232 input options. For more information, see "RS-232 Input Options" on page 284.

- *y* Data transmitted to external device
  |
  (0x7C) Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see **"Command Extensions" on page 260**.

R Response Variables

- *t* Computer number *(000-126)*
- *x* Data from external device
- *z* External Device Response Codes

*t*K\\9	Syntax error
*t*K\\A	Printer time-out on CTS
*t*K\\B	Printer time-out on DSR
*t*K\\C	Printer time-out XON/XOFF
*t*K\\E	Error with E option (00 < *xx* ≤ FF
*t*K\\K	If T*xx*=00 and a key is pressed, no input data is returned
*t*K\\M	Error with M option (00 < *xx* ≤ 99)
*t*K\\N	Error with N option (00 < *xx* ≤ 99)
*t*K\\P	Data parity error on RS-232 input
*t*K\\S	Error with S option (00 < *xx* ≤ FF)
*t*K\\T	Error with T option (00 < *xx* ≤ 99)
*t*K\\X	Time-out error waiting for input
*t*K\\z*x*	=RS-232 input data

Example

```
W000P10L1/\PFS0AE0AFN01M50T20\
DATE: 02-23-00^0D^0A
```

This example sets the following response options:

- P10 7 data bits, 1 stop bit, even parity
- L1 4800 bps
- PF Write "DATE: 02-23-00<cr><lf>" before receiving data input
- S0A Start character is <If>
- E0A End character is <If>
- F Return the start character
- N01 End data input
- M50 Get 50 characters maximum
- T20 Wait for 20 seconds maximum for data input before time-out

WtD/\y P (Print Table)

W*t*D/\yP prints data in a table to the RS-232 port. The RS-232 port setup parameters are set by the W*t*D/\PS (Set RS-232 Port Options) command.

Command

```
WtD/\yP<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Computer number *(000-126)*
- *y* Table and data to print

R Response Variables

- *t* Computer number *(000-126)*
- *z* Response Code
  0=Data Printed Correctly
  9=Syntax Error
  A=CTS Time-out
  B=DSR Time-out
  C=XON/XOFF Time-out

Data in a table can encode characters with hexadecimal values by using ^xx, where xx is the two-digit hexadecimal number for the data. A <cr> must be sent this way. (Carriage return “<cr>” character is sent as ^0D.)

WtD/ \yR (Update and Print Table)

W*t*D/\yR changes a portion of a table and prints it without downloading an entire table. Accomplish this by replacing one of the following: the prompt and input field pair together, the prompt field only, or the input field of prompt and input field pair only.

Individual hexadecimal characters can be escaped with the “^*xx*” extension. For more information, see “Pass-Through Hex Character” on page 261.

Command

```
WtDo/\yR\i\{Fp\fx}
```
or
```
{Pp}
```
or
```
{Ifx}0<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\y\n\gx\0\n\gx<cr>
tK\yM\n\gx\0\n\gx<cr>
tK\\z<cr>
```

Note: Alternative replacements are in braces { }. The “...” indicates that you may use this command multiple times.

Command Variables

- *t* Computer number *(000-126)*
- *y* Table number *(0-99, Length is 16725 bytes; # pairs is 25)*
- *i* Input field number *(0-99)*
- **F** Update prompt and input fields together
- *p* Prompt field *(Maximum field size)*
- *f* Input field control byte (001*XXXXX*)
  For more information, see "Input Field Control Byte" on page 265.
- *x* Input field data
- **P** Update prompt field only
- **I** Update input field only
  | *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see "Command Extensions" on page 260.

Note: See "Rules for Designing Tables" on page 265 to learn about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- *t* Computer number *(000-126)*
- *y* Table number *(0-99)*
- *n* Input field number *(0-99)*
- *g* Input data type
  K=Keyboard
  S=Scanner
- *x* Input field data
- **M** Indicates more data to come. Host continues to send an R (Read) command until it receives a response without the "more data" flag. *(The R response without M means all data has been sent.)*
- *o* Print table controls *(see the table on the next page)*
- *z* Response code
  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number

5=Prompt Field, No Input Field
6=Field Does Not Exist
7=Missing Prompt and Input Field Pair
8=Field Pair Number Out of Range
9=Format Error or Syntax Error
A=CTS Time-out
B=DSR Time-out
C=XON/XOFF Time-out

WtD/\yR** (Update and Print Table) **Options

o	Description	Option
C	XON/XOFF control	
D	DSR XON/XOFF	
H*x*	Time-out for RS-232 port ready in *x* sec	
L*x*	*x*=speed	0=9600 bps *(default)* 1=4800 bps 2=2400 bps 3=1200 bps 4=300 bps 5=19200 bps 6=38400 bps
O	Print prompt fields only	Print all fields
P	Sum of "P value" choices. See the table of "P Values" on page 281.	
Q	Print CR/LF after prompt field	No CR/LF
S	Print CR/LF after input field	No CR/LF
U	Print input fields only	Print all fields
W	CTS busy	

WtD/ \yQ (Update, Display, and Print Table)

W*t*D/\yQ allows change to a portion of a table without downloading an entire table. It displays the table to the computer screen and sends it to the RS-232 port. The update process can replace the prompt field or the input field, or both.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "Command Extensions" on page 260.

Command

```
WtDo/\yQ\i\{Fp\fx}
```
or
```
{Pp}
```
or
```
{Ifx}0<cr>
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\y\n\gx\0\n\gx<cr>
tK\yM\n\gx\0\n\gx<cr>
tK\\z<cr>
```

Note: Alternative replacements are in braces { }. The "..." indicates that you may use this command multiple times.

Command Variables

- ***t*** Computer number *(000-126)*
- ***o*** Options. For more information, see **"Write Display Options" on page 259**.
- ***y*** Table number *(0-99, Length is 16725 bytes; # pairs is 25)*
- ***i*** Input field number *(0-99)*
- **F** Update prompt and input fields together
- ***p*** Prompt field *(Maximum field size)*
- ***f*** Input field control byte (001*XXXXX*)
  For more information, see **"Input Field Control Byte" on page 265**.
- ***x*** Data
- **P** Update prompt field only
- **I** Update input field only
  | *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions. For more information, see **"Command Extensions" on page 260**.

Note: See **"Rules for Designing Tables" on page 265** to learn about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- ***t*** Computer number *(000-126)*
- ***y*** Table number *(0-99)*
- ***n*** Input field number *(0-99)*
- ***g*** Input data type
  K=Keyboard
  S=Scanner
- ***x*** Data

- **M** Indicates more data to come. Host continues to send an R (Read) command until it receives a response without the "more data" flag. (*The R response without M means all data has been sent.*)

- *z* Response code
  0=Message Received
  1=Bad Command
  2=Field Length is 0
  3=Data Overflow
  4=Bad Table Number
  5=Prompt Field, No Input Field
  6=Field Does Not Exist
  7=Missing Prompt and Input Field Pair
  8=Field Pair Number Out of Range
  9=Format Error or Syntax Error
  A=CTS Time-out
  B=DSR Time-out
  C=XON/XOFF Time-out

WtD/\T (Tone)

This sets the volume, frequency (pitch), and length (duration) of the tone.

Command

```
WtD/\Tvvvffflll
```

Server or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z
```

Command Variables

- *t* Computer number *(000-126)*
- *v* Volume *(000-255)*
- *f* Frequency or pitch *(000-030)*
- *l* Length or duration *(001-010)*

R Response Variables

- *t* Computer number *(000-126)*
- *z* Response code
  0=Parameters Set Correctly
  V=Volume Out of Range
  F=Frequency Out of Range
  L=Length Out of Range

Example

This example indicates that the length of "011" is outside the acceptable range of 001-010:

Command: W000D/\T255030011<cr>
R Response: 000K\\L<cr>

DtV (Computer Emulation Version)

Use D*t*V to return the computer emulation version running in the computer. The command variable is *t*, the computer number (000-126). Following are R response variables and characters.

Command

D*t*V<cr>

Server or Gateway Response

OK<cr>

R Response

*t*DFWP<*ab*0H0> V<*v.vv*><cr>

Command and R Response Variables

- ***t*** Computer number *(000-126)*
- **D** Diagnostic command or response
- **FWP** Computer emulation program
- ***ab*** First two numbers in the computer model name
- **H** Represents the combined Native, 3270, 5250, and VT/ANSI computer emulation programs.
- ***v*** Computer emulation program's version number

Example

Command: D003V<cr>
R Response: 003DFWP700H0 V6.17<cr>

Computer 003 is a 700 Color running computer emulation program FWP700H0, version 6.17.

DtE (Echo-Back Diagnostic)

D*t*E tests the communication cycle. The host sends a data string to a computer, which it immediately returns for comparison. The "D" in the R response indicates "diagnostic command or response."

Command

D*t*E/*x*<cr>

Server or Gateway Response

OK<cr>

R Response

`tDx<cr>`

Command and R Response Variables

- *t* Computer number *(000-126)*
- *x* Data *(1-112 characters)*

Example

Command: `D002E/ABCD1234<cr>`
Computer response: `002DABCD1234<cr>`
Computer 002 returns same data responding to next R (Read) command.

VT/ANSI Programming

The following information is for VT/ANSI programming:

- The received codes the computers support. Received codes include C0 and C1 control characters, character sets, and computer modes.
- The transmitted keyboard codes generated by the computers' keys and sent to the host computer.
- The capability of the computers to emulate VT/ANSI computer top-row function, main keypad, editing, and auxiliary keys.
- VT340 applications supported by the computers. Applications include working in local editing and interactive modes, and creating text forms.
- Printing bar code labels and RFID tags using Intermec Printer Language (IPL).

Encoding Characters

The implementation of VT/ANSI computer emulation supports both 7-bit and 8-bit communications environments. The implementation consists of an 8-bit asynchronous character encoding scheme and a 7-bit code extension technique so that the computer is compatible with ANSI and ISO standards. All 8-bit codes can be referenced by using two-byte ESCape sequences in accordance with ANSI standard X3.41 -- 1974.

The document referred to when ANSI is mentioned is ANSI X3.64 -- 1979, *Additional Controls for Use With American National Standard for Information Interchange*. The ANSI X3.64 standard controls character imaging devices such as the VT200 computer. ANSI and ISO determine current standards for character encoding in the communications industry.

Character Sets

The computer processes most characters it receives from the host computer based on characters whose codes and functions are defined and standardized by ANSI. The types are 7-bit and 8-bit character codes.

Multinational Character Set

The DEC multinational character set has C0, GL, C1, and GR codes.

C0 and GL Codes

The left half of the multinational character set consists of control characters (C0 codes) and graphic characters (GL codes).

Control characters (C0 codes) are 7-bit compatible, nondisplayable single-byte characters that perform specific functions during communications and text processing. C0 codes range from 00 to 1F hexadecimal including 7F (DEL) and can be used in a 7- or 8-bit environment.

Graphic characters (GL codes) are 7-bit compatible, displayable characters that represent various alphanumeric characters, punctuation marks, and symbols that appear in the computer's display. GL codes range from 21 to 7E hexadecimal. They can be used in a 7-bit or an 8-bit environment. The space character (20 hexadecimal) may be a graphic or a control character, depending on the context.

The following table lists the C0 and GL codes and their corresponding decimal and hexidecimal values. Shaded cells indicate ignored codes. The table “C0 Control Characters and Computer Action” on page 297 lists the actions taken by the computer when a C0 code is received.

C0 and GL Codes

	column	0		1		2		3		4		5		6		7	
row	B8 B7 B6 B5 B4 B3 B2 B1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 00	DLE	16 10	SP	32 20	0	48 30	@	64 40	P	80 50	‘	96 60	p	112 70
1	0 0 0 1	SOH	1 01	DC1	17 11	!	33 21	1	49 31	A	65 41	Q	81 51	a	97 61	q	113 71
2	0 0 1 0	STX	2 02	DC2	18 12	“	34 22	2	50 32	B	66 42	R	82 52	b	98 62	r	114 72
3	0 0 1 1	ETX	3 03	DC3	19 13	#	35 23	3	51 33	C	67 43	S	83 53	c	99 63	s	115 73
4	0 1 0 0	EOT	4 04	DC4	20 14	$	36 24	4	52 34	D	68 44	T	84 54	d	100 64	t	116 74
5	0 1 0 1	ENQ	5 05	NAK	21 15	%	37 25	5	53 35	E	69 45	U	85 55	e	101 65	u	117 75
6	0 1 1 0	ACK	6 06	SYN	22 16	&	38 26	6	54 36	F	70 46	V	86 56	f	102 66	v	118 76
7	0 1 1 1	BEL	7 07	ETB	23 17	‘	39 27	7	55 37	G	71 47	W	87 57	g	103 67	w	119 77
8	1 0 0 0	BS	8 08	CAN	24 18	(	40 28	8	56 38	H	72 48	X	88 58	h	104 68	x	120 78
9	1 0 0 1	HT	9 09	EM	25 19	)	41 29	9	57 39	I	73 49	Y	89 59	i	105 69	y	121 79
A	1 0 1 0	LF	10 0A	SUB	26 1A	*	42 2A	:	58 3A	J	74 4A	Z	90 5A	j	106 6A	z	122 7A
B	1 0 1 1	VT	11 0B	ESC	27 1B	+	43 2B	;	59 3B	K	75 4B	[	91 5B	k	107 6B	{	123 7B
C	1 1 0 0	FF	12 0C	FS	28 1C	,	44 2C	<	60 3C	L	76 4C	\	92 5C	l	108 6C	|	124 7C
D	1 1 0 1	CR	13 0D	GS	29 1D	-	45 2D	=	61 3D	M	77 4D	]	93 5D	m	109 6D	}	125 7D
E	1 1 1 0	SO	14 0E	RS	30 1E	.	46 2E	>	62 3E	N	78 4E	^	94 5E	n	110 6E	~	126 7E
F	1 1 1 1	SI	15 0F	US	31 1F	/	47 2F	?	63 3F	O	79 4F	_	95 5F	o	111 6F	DEL	127 7F
		<-------- C0 Codes ------->				<------------------------------------- GL Codes --> ASCII Graphics Character Set											

Ignored codes are removed from the data stream with no action taken by the computer.

C0 Control Characters and Computer Action

Mnemonic	Hex	Name	Action
NUL	00	Null	Ignored.
SOH	01	Start of heading	Ignored.
STX	02	Start of text	Ignored.
ETX	03	End of text	Ignored.
EOT	04	End of transmission	Ignored.
ENQ	05	Enquiry	Generates Answerback message. Set number of characters in the message (0-30) through the computer's firmware. If the computer is operating in ANSI mode, it ignores this character.
ACK	06	Acknowledge	Ignored.
BEL	07	Bell	Generates bell tone if bell is enabled.
BS	08	Backspace	Moves cursor one column to the left, unless it is in Column 1.
HT	09	Horizontal tab	Moves cursor to predetermined tab stop or to right margin.
LF	0A	Line feed	Causes LF or new line operation, based on LNM received code.
VT	0B	Vertical tab	Processes as LF (line feed).
FF	0C	Form feed	Processes as LF (line feed).
CR	0D	Carriage return	Moves cursor to Column 1 of the current row.
SO (LS1)	0E	Shift out (Lock shift G1)	Invokes G1 character set into GL.
SI (LS0)	0F	Shift in (Lock shift G0)	Invokes G0 character set into GL.
DLE	10	Data link escape	Ignored.
DC1 (XON)	11	Device control 1	Clears DC3 if XOFF support is enabled, which causes the computer to continue sending characters (keyboard unlocks).
DC2	12	Device control 2	Ignored.
DC3 (XOFF)	13	Device control 3	If XOFF support is enabled, causes the computer to stop sending characters until it receives a DC1 control character.
DC4	14	Device control 4	Ignored.
NAK	15	Negative acknowledgment	Ignored.
SYN	16	Synchronous table	Ignored.
ETB	17	End transmission block	Ignored.
CAN	18	Cancel	Terminates and cancels any sequence in progress.

Ignored codes are removed from the data stream with no action taken by the computer.

C0 Control Characters and Computer Action

Mnemonic	Hex	Name	Action
EM	19	End of medium	Ignored.
SUB	1A	Substitute	Terminates, cancels escape or control sequence, or terminates the device control string and displays reverse question mark.
ESC	1B	Escape	Processes as escape sequence introducer. Terminates any escape, control, or device control sequence in process.
FS	1C	File separator	Ignored.
GS	1D	Group separator	Ignored.
RS	1E	Record separator	If GATM reset, record separator processed as protected field entry.
US	1F	Unit separator	Ignored.
DEL	7F	Delete	Ignored; it cannot be used as a fill character.

C1 and GR Codes

The right half of the DEC multinational character set consists of eight-bit control codes (C1 codes) and supplemental graphic characters (GR codes).

C1 codes are non-display codes that perform additional functions beyond the C0 codes. C1 codes range from 80-9F hexadecimal.

The supplemental graphic characters set (GR codes) range from A0 hexadecimal to FF hexadecimal. The set has alphabetic characters with accents and diacritical marks that appear in the major Western European alphabets, plus other symbols not included in the ASCII graphics set. The table also includes the corresponding decimal and hexidecimal values.

C1 and GR Codes

8		9		10		11		12		13		14		15		column	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		B8 B7 B6 B5 B4 B3 B2 B1	row
	128 80	DCS	144 90	(nbsp)	160 A0	°	176 B0	à	192 C0	Ð	208 D0	à	224 E0	ð	240 F0	0 0 0 0	0
	129 81	PU1	145 91	¡	161 A1	±	177 B1	á	193 C1	Ñ	209 D1	á	225 E1	ñ	241 F1	0 0 0 1	1
	130 82	PU2	146 92	¢	162 A2	²	178 B2	â	194 C2	Ò	210 D2	â	226 E2	ò	242 F2	0 0 1 0	2
	131 83	STS	147 93	£	163 A3	³	179 B3	Ã	195 C3	Ó	211 D3	ã	227 E3	ó	243 F3	0 0 1 1	3
IND	132 84	CCH	148 94	¤	164 A4	´	180 B4	Ä	196 C4	Ô	212 D4	ä	228 E4	ô	244 F4	0 1 0 0	4
NEL	133 85	MW	149 95	¥	165 A5	µ	181 B5	Å	197 C5	Õ	213 D5	å	229 E5	õ	245 F5	0 1 0 1	5
SSA	134 86	SPA	150 96	¦	166 A6	¶	182 B6	Æ	198 C6	Ö	214 D6	æ	230 E6	ö	246 F6	0 1 1 0	6
ESA	135 87	EPA	151 97	§	167 A7	·	183 B7	Ç	199 C7	×	215 D7	ç	231 E7	÷	247 F7	0 1 1 1	7
HTS	136 88		152 98	¨	168 A8	¸	184 B8	è	200 C8	Ø	216 D8	è	232 E8	ø	248 F8	1 0 0 0	8
HTJ	137 89		153 99	©	169 A9	¹	185 B9	É	201 C9	ù	217 D9	é	233 E9	ù	249 F9	1 0 0 1	9
VTS	138 8A		154 9A	ª	170 AA	º	186 BA	ê	202 CA	ú	218 DA	ê	234 EA	ú	250 FA	1 0 1 0	A
PLD	139 8B	CSI	155 9B	«	171 AB	»	187 BB	ë	203 CB	û	219 DB	ë	235 EB	û	251 FB	1 0 1 1	B
PLU	140 8C	ST	156 9C	¬	172 AC	¼	188 BC	ì	204 CC	Ü	220 DC	ì	236 EC	ü	252 FC	1 1 0 0	C
RI	141 8D	OSC	157 9D	-	173 AD	½	189 BD	í	205 CD	Ý	221 DD	í	237 ED	ý	253 FD	1 1 0 1	D
SS2	142 8E	PM	158 9E	®	174 AE	¾	190 BE	î	206 CE	Þ	222 DE	î	238 EE	þ	254 FE	1 1 1 0	E
SS3	143 8F	APC	159 9F	–	175 AF	¿	191 BF	ï	207 CF	ß	223 DF	ï	239 EF	ÿ	255 FF	1 1 1 1	F
<-------- C1 Codes ------->				<-- GR Codes --> ASCII Graphics Character Set													

Ignored and undefined codes are removed from the data stream with no action taken. Note that some letters that are normally uppercase are converted to lowercase. The hexadecimal values for these letters are C0, C1, C2, C8, CA, CB, CC, CD, CE, CF, D9, DA, and DB.

C1 Control Characters and Computer Action

Mnemonic	Hex	Name	Action
	80-83		Ignored.
IND	84	Index	Moves cursor down one line in same column. If cursor is at bottom margin, display performs a scroll up.
NEL	85	Next line	Moves cursor to first position on next line. If cursor is at bottom margin, display performs a scroll up.
SSA	86	Start of selected area	Selects which current page characters the computer can send to the host.
ESA	87	End of selected area	Ends the selected area that is sent to the host when forms are created for VT340 computer emulation.
HTS	88	Horizontal tab set	Sets one horizontal tab stop at column with cursor.
HTJ	89	Horizontal tab w/justify	Ignored.
VTS	8A	Vertical tabulation set	Ignored.
PLD	8B	Partial line down	Ignored.
PLU	8C	Partial line up	Ignored.
RI	8D	Reverse index	Moves cursor up one line in same column. If cursor is at top margin, display performs a scroll down.
SS2	8E	Single shift 2	Temporarily invokes G2 character set into GL for the next graphic character. G2 is designated by a select character set (SCS) sequence.
SS3	8F	Single shift 3	Temporarily invokes G3 character set into GL for the next graphic character. G3 is designated by an SCS sequence.
DCS	90	Device control string	Processes as opening delimiter of a device control string for device control use.
PU1	91	Private use 1	Ignored.
PU2	92	Private use 2	Ignored.
STS	93	Set transmit state	Ignored.
CCH	94	Cancel character	Ignored.
MW	95	Message waiting	Ignored.
SPA	96	Start of protected area	Starts the protected area in page memory that cannot be edited when forms are created from the host.

C1 Control Characters and Computer Action (continued)

Mnemonic	Hex	Name	Action
EPA	97	End of protected area	Ends the protected area in page memory that cannot be edited when forms are created from the host.
	98-9A		Ignored.
CSI	9B	Ctrl sequence introducer	Processes as control sequence introducer.
ST	9C	String terminator	Processes as the string closing delimiter opened by DCS and processes code as the EOL of extended command characters.
OSC	9D	O/S command	Ignored.
PM	9E	Privacy message	Ignored.
APC	9F	App program command	Processes code as the SOL of extended command characters.

Display Controls Mode

The computer does not support the VT/ANSI computer display controls mode, which displays control codes as graphic characters for debugging.

Dynamically Redefinable Character Set

The computer does not support the dynamically redefinable character set (DRCS), which is a 94-character set created on the VT/ANSI computer and down-line loaded into the computer DRCS buffer.

DEC Special Graphics Character Set

The computer supports most of the DEC special graphics character set. The set contains ASCII C0 codes and most of the GL codes. It also has special symbols and short line segments. You would use these characters to create a limited range of drawings while working in text mode.

This table shows special graphics characters recognized by the computer, including the corresponding hexidecimal values. Ignored and undefined codes are removed from the data stream with no action taken.

DEC Special Graphics Character Set

	column	0		1		2		3		4		5		6		7	
	B8	0		0		0		0		0		0		0		0	
	B7	0		0		0		0		1		1		1		1	
	B6	0		0		1		1		0		0		1		1	
	B5	0		1		0		1		0		1		0		1	
row	B4 B3 B2 B1																
0	0 0 0 0	NUl	00	DLE	10	SP	20	0	30	@	40	P	50		60		70
1	0 0 0 1	SOH	01	DC1	11	!	21	1	31	A	41	Q	51		61	SCAN 5	71
2	0 0 1 0	STX	02	DC2	12	“	22	2	32	B	42	R	52		62		72
3	0 0 1 1	ETX	03	DC3	13	#	23	3	33	C	43	S	53		63		73
4	0 1 0 0	EOT	04	DC4	14	$	24	4	34	D	44	T	54		64		74
5	0 1 0 1	ENQ	05	NAK	15	%	25	5	35	E	45	U	55		65		75
6	0 1 1 0	ACK	06	SYN	16	&	26	6	36	F	46	V	56		66		76
7	0 1 1 1	BEL	07	ETB	17	‘	27	7	37	G	47	W	57		67		77
8	1 0 0 0	BS	08	CAN	18	(	28	8	38	H	48	X	58		68		78
9	1 0 0 1	HT	09	EM	19	)	29	9	39	I	49	Y	59		69		79
A	1 0 1 0	LF	0A	SUB	1A	*	2A	:	3A	J	4A	Z	5A	┘	6A		7A
B	1 0 1 1	VT	0B	ESC	1B	+	2B	;	3B	K	4B	[	5B	┐	6B		7B
C	1 1 0 0	FF	0C	FS	1C	,	2C	<	3C	L	4C		5C	┌	6C		7C
D	1 1 0 1	CR	0D	GS	1D	-	2D	=	3D	M	4D	]	5D	└	6D		7D
E	1 1 1 0	SO	0E	RS	1E	.	2E	>	3E	N	4E	^	5E		6E		7E
F	1 1 1 1	SI	0F	US	1F	/	2F	?	3F	O	4F	(blank)	5F		6F		7F
		<-------- C0 Codes -------> ASCII Non-display Control Character set				<-------------------------------------- GL Codes --> ASCII Graphics Character Set											

National Replacement Character Sets

The computer supports the 7-bit national replacement character (NRC) sets for European languages. The NRC sets are similar to the 7-bit ASCII set, except for a few characters. You designate the character sets as hard using escape sequence formats. This table lists characters in each NRC set that are different from the ASCII set. The computer does not support “¾” (40 hexadecimal) in the Dutch NRC set.

<table>
<tr><th></th><th colspan="12">Hexadecimal</th></tr>
<tr><th>Character Set</th><th>23</th><th>40</th><th>5B</th><th>5C</th><th>5D</th><th>5E</th><th>5F</th><th>60</th><th>7B</th><th>7C</th><th>7D</th><th>7E</th></tr>
<tr><td>ASCII</td><td>#</td><td>@</td><td>[</td><td>\</td><td>]</td><td>^</td><td>_</td><td>`</td><td>{</td><td>|</td><td>}</td><td>~</td></tr>
<tr><td>British</td><td>£</td><td>@</td><td>[</td><td>\</td><td>]</td><td>^</td><td>_</td><td>`</td><td>{</td><td>|</td><td>}</td><td>~</td></tr>
<tr><td>Dutch</td><td>£</td><td></td><td>ÿ</td><td>½</td><td>|</td><td>^</td><td>_</td><td>`</td><td>¨</td><td></td><td>¼</td><td>'</td></tr>
<tr><td>Finnish</td><td>#</td><td>@</td><td>Ä</td><td>Ö</td><td>Å</td><td>Ü</td><td>_</td><td>é</td><td>ä</td><td>ü</td><td>å</td><td>ü</td></tr>
<tr><td>French</td><td>£</td><td>à</td><td>º</td><td>ç</td><td>§</td><td>^</td><td>_</td><td>`</td><td>é</td><td>ù</td><td>è</td><td>¨</td></tr>
<tr><td>French Canadian</td><td>#</td><td>à</td><td>â</td><td>ç</td><td>ê</td><td>î</td><td>_</td><td>ô</td><td>é</td><td>ù</td><td>è</td><td>û</td></tr>
<tr><td>German</td><td>#</td><td>§</td><td>Ä</td><td>Ö</td><td>Ü</td><td>^</td><td>_</td><td>`</td><td>ä</td><td>ö</td><td>ü</td><td>ß</td></tr>
<tr><td>Italian</td><td>£</td><td>§</td><td>º</td><td>ç</td><td>é</td><td>^</td><td>_</td><td>ù</td><td>à</td><td>ò</td><td>è</td><td>ì</td></tr>
<tr><td>Norwegian/Danish</td><td>#</td><td>@</td><td>Æ</td><td>Ø</td><td>Å</td><td>^</td><td>_</td><td>`</td><td>æ</td><td>ø</td><td>å</td><td>~</td></tr>
<tr><td>Portuguese</td><td>#</td><td>@</td><td>Ã</td><td>Ç</td><td>Ô</td><td>^</td><td>_</td><td>`</td><td>ã</td><td>ç</td><td>ő</td><td>~</td></tr>
<tr><td>Spanish</td><td>£</td><td>§</td><td>í</td><td>Ń</td><td>¿</td><td>^</td><td>_</td><td>`</td><td>'</td><td>º</td><td>ń</td><td>ç</td></tr>
<tr><td>Swedish</td><td>#</td><td>Ê</td><td>Ä</td><td>Ö</td><td>Å</td><td>Ü</td><td>_</td><td>é</td><td>ä</td><td>ö</td><td>å</td><td>ü</td></tr>
<tr><td>Swiss</td><td>ù</td><td>à</td><td>é</td><td>ç</td><td>ê</td><td>î</td><td>è</td><td>ô</td><td>ä</td><td>ö</td><td>ü</td><td>û</td></tr>
</table>

Character Set Selection

The computer supports hard character sets. It does not support designating soft (down-line loadable) character sets.

Designating Hard Character Sets

Designate hard character sets using the escape sequence format in the following table.

Hard Character Set Escape Sequences

Escape Sequence		Designated As
Supplemental and special graphics sets:		
1B hexadecimal ESC)	28 hexadecimal (Final)	G0
1B hexadecimal ESC)	29 hexadecimal (Final)	G1
1B hexadecimal ESC *	2A hexadecimal (Final)	G2
1B hexadecimal ESC *	2B hexadecimal (Final)	G3
NRC sets:		
1B hexadecimal ESC -	2D hexadecimal (Final)	G1
1B hexadecimal ESC -	2E hexadecimal (Final)	G2
1B hexadecimal ESC *	2F hexadecimal (Final)	G3

The final hard character in the escape sequence represents the character set to designate. For computers, final characters are indicated in this table.

Final Characters

Character Set		Final Character (Hex)
ASCII		B (42)
Special graphics		0 (30)
National replacement	British	A (41)
	Dutch	4 (34)
	Finnish*	5 (35) *or* C (43)
	French	R (52)
	French Canadian*	9 (39) *or* Q (51)
	German	K (4B)
	Italian	Y (59)
	Norwegian/Danish*	` (60) *or* E (45)
	Portuguese	% (25)
	Spanish	Z (5A)
	Swedish*	7 (37) *or* H (48)
	Swiss	= (3D)
* The first code shown is recommended.		

Locking Shifts

The computer applies the locking shift (LS) control functions in this table. *LS1R, LS2, LS2R, LS3, LS3R not available in VT100 or ANSI modes.*

Locking Shift Control Functions

Control Name	Code	Function
LS0 (Locking shift 0)	SI (0F hexadecimal)	Map G0 into GL *(default)*
LS1 (Locking shift 1)	SO (0E hexadecimal)	Map G1 into GL
LS1R* (Locking shift 1, right)	ESC ~ (1B, 7E hexadecimals)	Map G1 into GR
LS2* (Locking shift 2)	ESC n (1B, 6E hexadecimals)	Map G2 into GL
LS2R* (Locking shift 2, right)	ESC } (1B, 7D hexadecimals)	Map G2 into GR
LS3* (Locking shift 3)	ESC o (1B, 6F hexadecimals)	Map G3 into GL
LS3R* (Locking shift 3, right)	ESC | (1B, 7C hexadecimals)	Map G3 into GR
* *Available only in VT320 and VT340 modes.*		

Received Codes

Received codes are codes the computer receives from an application or host computer. These pages describe received codes, and the action taken by the computer when it receives the codes from an application or host.

Select C1 Controls

Select C1 controls (code extension announcers) control the representation of C1 control codes returned to the application. The computer supports select 7-bit and 8-bit C1 control transmission as shown in this table.

Note: The computer does not support DECSCL sequences.

Select C1 Controls

Transmission	Sequence	Action
Select 7-bit C1 Control (S7C1T)	ESC sp F	Converts all C1 codes returned to the application to their equivalent 7-bit code extensions.
Select 8-bit C1 Control (S8C1T) *(Ignored in VT100 and ANSI modes.)*	ESC sp G	Returns C1 codes to the application without converting them to their equivalent 7-bit code extensions.

Computer Modes

A mode is one of several operating states used by the computer. The following table lists selectable computer modes and the action taken by the computer in set mode and reset mode. *Last character of reset mode sequences is lowercase L (6C hexadecimal).*

Selectable Computer Modes

Mode (Mnemonic)	Set Mode Reset Mode	Sequence	Action
Keyboard action (KAM)	Locked	CSI 2 h	Locks the keyboard for all following keystrokes.
	Unlocked	CSI 2 l	Unlocks the keyboard.
Insert/Replace (IRM)	Insert	CSI 4 h	Selects insert mode. New display characters move old display characters to the right. Characters moved past the right margin are lost. Text moved into a protected character field is lost. If erasure mode (ERM) is reset, text moved into a protected character field is lost.
	Replace	CSI 4 l	Selects replace mode. New display characters replace old display characters at cursor position. Old character is erased.
Send/Receive (SRM))	Off	CSI 1 2 h	Disables local echo. When the computer sends characters to the host, the host must echo them back to the display.
	On	CSI 1 2 l	Enables local echo. When the computer sends characters, they are automatically sent to the display.
Line feed/New line (LNM)	New line	CSI 2 0 h	Causes a received LF, FF, or VT code to move the cursor to the first column of the next line. “Return” sends both a CR and an LF code.
	Line feed	CSI 2 0 l	Causes received LF, FF, or VT code to move cursor to the next line in the current column. “Return” sends a CR code only.
Cursor key (DECCKM)	Application	CSI ? 1 h	Causes cursor keys to send application control functions.
	Cursor	CSI ? 1 l	Has cursor keys to send ANSI cursor control sequences.
ANSI/VT52 (DECANM)	(None)	(None)	Not applicable.
	VT52	CSI ? 2 l	Ignored.
Column (DECCOLM) **see Note next page*	132 column	CSI ? 3 h	Selects 132 columns per line, *supported as of TE 2000 version 8.10.*

Selectable Computer Modes (continued)

Mode (Mnemonic)	Set Mode Reset Mode	Sequence	Action
	80 column	CSI ? 3 l	Selects 80 columns per line.
Scrolling (DECSCLM)	Smooth	CSI ? 4 h	Ignored.
	Jump	CSI ? 4 l	Has computer add lines to the display as fast as possible.
Screen (DECSCNM)	Reverse	CSI ? 5 h	Ignored.
	Normal	CSI ? 5 l	Selects normal (light characters on a dark background).
Origin (DECOM)	Origin	CSI ? 6 h	Ignored.
	Absolute	CSI ? 6 l	Ignored.
Auto wrap (DECAWM)	On	CSI ? 7 h	Graphic display characters received when the cursor is at right margin appear on the next line. The display scrolls up if cursor is at end of the scrolling region.
	Off	CSI ? 7 l	Graphic display characters received when cursor is at right margin replace previously displayed characters.
Auto repeat (DECARM)	On	CSI ? 8 h	Ignored.
	Off	CSI ? 8 l	Ignored.
Print form feed (DECPFF)	On	CSI ? 18 h	Selects form feed (FF) as print termination character. The computer sends this character to the printer after each print screen operation.
	Off	CSI ? 18 l	Selects no termination character. Computer does not send an FF to the printer after each print screen operation.
Print extent (DECPEX)	Full screen	CSI ? 19 h	Selects full screen to print during a print screen operation.
	Scrolling region	CSI ? 19 l	Selects scrolling region to print during print screen operation.
Text cursor enable (DECTCEM)	Visible	CSI ? 25 h	Makes the cursor visible.
	Not visible	CSI ? 25 l	Makes the cursor not visible.
Keypad (DECKPAM/ DECPNM)	Application	ESC =	Keypad keys send application control functions.
	Numeric	ESC >	Keypad keys send characters that match the numeric, comma, period, and minus sign keys on main keypad. PF1-PF4 send control functions.

Selectable Computer Modes (continued)

Mode (Mnemonic)	Set Mode Reset Mode	Sequence	Action
Character set (DECNRCM)	National	CSI ? 42 h	Ignored.
	Multinational	CSI ? 42 l	Ignored.
Back arrow key mode (DECBKM)	Backspace	CSI ? 6 7 h	The arrow key moves cursor back one space and deletes the character. It sends a BS character to the host.
	Delete	CSI ? 6 7 l	The arrow key deletes the character at the cursor position. It sends a DEL character to the host.

Note: VT/ANSI DECCOLM is supported as of TE 2000 version 8.10. VT220 132 column support does not require any menu options. The host computer can enable and disable the 132 column mode using the VT Column Mode (DECCOLM) commands of CSI ?3h (132 columns) and CSI ?3l (80 columns).

Cursor Positioning

The cursor indicates the position where the next character appears. The computer supports all cursor positioning sequences as shown in this table.

Note: Pn is a variable, ASCII-coded, numeric parameter.

Cursor Positioning

Name (Mnemonic)	Sequence	Action
Cursor up (CUU)	CSI Pn A	Moves cursor up Pn lines in same column, stops at top.
Cursor down (CUD)	CSI Pn B	Moves cursor down Pn lines in same column, stops at bottom.
Cursor forward (CUF)	CSI Pn C	Moves cursor right Pn columns. Cursor stops at right margin.
Cursor backward (CUB)	CSI Pn D	Moves cursor left Pn columns. Cursor stops at left margin.
Cursor position (CUP)	CSI Pl ; Pc H	Moves cursor to Line P1, Column Pc.
Horizontal/vertical positions (HVP)	CSI Pl ; Pc f	Moves cursor to Line P1, Column Pc.
Index (IND)	ESC D	An 8-bit control character (84 hexadecimal) to be expressed as an escape sequence for a 7-bit environment. IND moves the cursor down one line in the same column. If cursor is at bottom margin, display scrolls up.

Cursor Positioning (continued)

Name (Mnemonic)	Sequence	Action
Reverse index (RI)	ESC M	An 8-bit control character (8D hexadecimal) to be expressed as an escape sequence for a 7-bit environment. RI moves the cursor up one line in the same column. If the cursor is at top margin, the display scrolls down.
Next line (NEL)	ESC E	Is an 8-bit control character (85 hexadecimal) that can be expressed as an escape sequence for a 7-bit environment. NEL moves the cursor to the first position on the next line. If the cursor is at bottom margin, the display scrolls up.
Save cursor (DECSC)	ESC 7	Saves the following into memory: cursor position, graphic rendition, character set shift state, state of wrap flag, state of origin mode, state of selective erase.
Restore cursor (DECRC)	ESC 8	Restores the states described for DECSC. If characteristics were not saved, the cursor moves to home position, origin mode is reset, character attributes are not assigned, default character set mapping is established.

Tab Stops

Tab stops are selected based on the vertical display column. When the computer receives a horizontal tab code (HT, 09 hexadecimal), the cursor tabs to the next stop. If no tab stops exist, the cursor tabs to the right margin.

The computer supports both tab stop sequences . The computer supports “global” line tab stops only. It does not allow tab set and clear on every character on every line as for a VT/ANSI.

Tab Stops

Name (Mnemonic)	Sequence	Action
Horizontal tab set (HTS)	ESC H	Sets a tab stop at the current column. HTS is an 8-bit control character (88 hexadecimal) that can also be expressed as an escape sequence when coding for a 7-bit environment.
Tabulation clear (TBC)	CSI g	Clears a horizontal tab stop at cursor position.
	CSI 0 g	Clears a horizontal tab stop at cursor position.
	CSI 3 g	Clears horizontal tab stops.

Character Rendition and Attributes

Character rendition and attributes affect how a character is displayed, without changing the character. Select graphic rendition sequences change character rendition. The select character attribute sequence designates characters as erasable or non-erasable.

Select Graphic Rendition (SGR)

This lists select graphic rendition sequences and computer action taken.

Select Graphic Rendition Sequences

Graphic Rendition	Sequence and Ps (Single Parameter)	Action
All attributes off	CSI 0 m	Turns all attributes off.
Display bold	CSI 1 m	Translates bold to reverse video.
Display underscored	CSI 4 m	Displays underscored.
Display blinking	CSI 5 m	Ignored.
Display negative (reverse) image	CSI 7 m	Displays negative (reverse) image.
Display normal intensity	CSI 2 2 m	Displays normal intensity.
Display not underlined	CSI 2 4 m	Displays not underlined.
Display not blinking	CSI 2 5 m	Ignored.
Display positive image	CSI 2 7 m	Displays positive image.
Display black foreground*	CSI 30 m	Display black foreground
Display red foreground*	CSI 31 m	Display red foreground
Display green foreground*	CSI 32 m	Display green foreground
Display yellow foreground*	CSI 33 m	Display yellow foreground
Display blue foreground*	CSI 34 m	Display blue foreground
Display magenta foreground*	CSI 35 m	Display magenta foreground
Display cyan foreground*	CSI 36 m	Display cyan foreground
Display white foreground *(default)**	CSI 37 m	Display white foreground
Display black background *(default)**	CSI 40 m	Display black background
Display red background*	CSI 41 m	Display red background
Display green background*	CSI 42 m	Display green background
Display yellow background*	CSI 43 m	Display yellow background
Display blue background*	CSI 44 m	Display blue background
Display magenta background*	CSI 45 m	Display magenta background

Select Graphic Rendition Sequences (continued)

Graphic Rendition	Sequence and Ps (Single Parameter)	Action
Display cyan background*	CSI 46 m	Display cyan background
Display white background*	CSI 47 m	Display white background
* Supported on 700 Colors, CK30 Colors, CK31, CK32, CN3, CV30, and CV60.		

Select Character Attributes (DECSCA)

The computer ignores select character attributes.

Select Character Attributes

Character Attribute	Sequence and Ps	Action
All attributes off	CSI 0 " q	Ignored
Designate characters as non-erasable by DECSEL/ DECSED	CSI 1 " q	Ignored
Designate character as erasable by DECSEL/ DECSED	CSI 2 " q	Ignored

Line Attributes

Line attributes affect an entire display line. The cursor selects the line affected by the attribute. It stays in the same character position when the attribute changes, unless the attribute would move the cursor past the right margin. In this case, the cursor stops at the right margin. When scrolling, the attribute moves with the line.

Double-height line sequences make the cursor line the top or bottom half of a double-height, double-width line. The computer ignores these sequences. Double-width line makes the line with the cursor double-width, single-height. The computer ignores this sequence as well. The computer does support the single-width line.

Select Line Attributes

Line Attribute	Half	Sequence	Action
Double-height, double-width	Top	ESC # 3	Ignored
	Bottom	ESC # 4	Ignored
Double-width, double-height		ESC # 6	Ignored
Single-width, single-height		ESC # 5	Makes the line with the cursor single-width, single-height. This is the line attribute for all new lines on the display.

Erasure Mode

Erasure mode (ERM) determines whether erasing functions (ECH, erase character; EL, erase in line; and ED, erase in display) can edit protected characters. The following table lists erasure mode sequences.

Note: In edit mode (DECEDM), ERM affects editing functions IL, DL, ICH, and DCH.

Erasure Mode

Mode	Sequence	Comments
Set	CSI 6 h	Erasing functions ECH, EL, and ED can affect all protected and unprotected characters.
Reset	CSI 6 l	Erasing functions ECH, EL, and ED can affect only unprotected characters.

The following table lists the erasing and editing functions that ERM affects. ERM also affects the independent style of character protection.

Control Functions Affected by Character Protection

Always Affected	Affected if Edit Mode (DECEDM) is Set
Erase character (ECH)	Insert character (ICH)
Erase in line (EL)	Insert line (IL)
Erase in display (ED)	Delete character (DCH) Delete line (DL)

Editing

Editing sequences insert and delete characters and lines of characters at the cursor position. The position does not change when lines are inserted or deleted. The computer supports all editing sequences.

Note: Pn is a variable, ASCII-coded, numeric parameter.

Editing Sequences

Name (Mnemonic)	Sequence	Action
Insert line (IL)	CSI Pn L	Inserts Pn lines at the cursor. If fewer than Pn lines remain from the current line to the end of the scrolling region, the number of lines inserted is the lesser number. Lines within the scrolling region at and below the cursor move down. Lines moved past bottom margin are lost. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region. In edit mode (DECEDM), if erasure mode (ERM) is reset, lines that move down into a line with a protected character field are lost.
Delete line (DL)	CSI Pn M	Deletes Pn lines, starting at the line with the cursor. If fewer than Pn lines remain from the current line to the end of the scrolling region, the number of lines deleted is the lesser number. As lines are deleted, the lines within the scrolling region and below the cursor move up, and blank lines are added at the bottom of the scrolling region. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region. In edit mode (DECEDM), if erasure mode (ERM) is reset, DL cannot delete lines that have protected characters. As lines are deleted, the area moved up is bounded by the bottom of the scrolling region, or by the next line with a protected character field.
Insert characters (ICH) (*Applies only to ANSI, VT220, VT320, and VT340 modes. Is ignored in VT100 mode.*)	CSI Pn @	Inserts Pn blank characters at the cursor position, with the character attributes set to normal. The cursor does not move and remains at the beginning of the inserted blank characters. A parameter of 0 or 1 inserts one blank character. Data on the line is shifted forward as in character insertion. In edit mode (DECEDM), if erasure mode (ERM) is reset, text moved into a protected character field is lost.
Delete character (DCH)	CSI Pn P	Deletes Pn characters, starting with the character at the cursor position. When a character is deleted, all characters to the right of the cursor move to the left. This creates a space character at the right margin for each character deleted. Character attributes move with the characters. Spaces created at the end of the line have all of their character attributes off. In edit mode (DECEDM), if erasure mode (ERM) is reset, DCH cannot delete protected characters.

Erasing

Erasing deletes characters in the computer's display without affecting other characters. Erased characters are lost. Cursor positioning does not change when characters or lines are erased. The following table lists erasing sequences and the action taken by the computer.

Erasing Sequences

Name (Mnemonic)	Sequence	Action
Erase character (ECH) *(Applies to ANSI, VT220, VT320, VT340 modes. Ignored in VT100 mode.)*	CSI Pn X	Erases characters at the cursor position and the next Pn-1 character. A parameter of 0 or 1 erases a single character. Character attributes are set to normal. No reformatting of data on the line occurs. The cursor remains in the same position. If erasure mode (ERM) is reset, ECH cannot erase protected characters.
Erase in line (EL)	CSI K	Erases from the cursor to the end of the line, including the cursor position. Line attribute is not affected. If erasure mode (ERM) is reset, EL cannot erase protected characters.
	CSI 0 K	Same as CSI K.
	CSI l K	Erases from the beginning of the line to the cursor, including the cursor position. Line attribute is not affected.
	CSI 2 K	Erases the complete line.
Erase in display (ED)	CSI J	Erases from cursor to the end of display, including the cursor position. Line attribute is single-height, single-width for all completely erased lines. If erasure mode (ERM) is set, ED cannot erase protected character positions.
Erase in display (ED)	CSI 0 J	Same as CSI J.
	CSI 1 J	Erases from the beginning of the display to the cursor, including the cursor position. Line attribute becomes single-height, single-width for all completely erased lines.
	CSI 2 J	Erases complete display. All lines are erased and changed to single-width. The cursor does not move.
Selective erase in line (DECSEL)	CSI ? K	Ignored by the computer.
	CSI ? 0 K	
	CSI ? 1 K	
	CSI ? 2 K	
Selective erase in display (DECSED)	CSI ? J	Ignored by the computer.
	CSI ? 0 J	
	CSI ? 1 J	
	CSI ? 2 J	

Scrolling Margins (Top and Bottom)

The scrolling region is the area of the computer's display that can receive new characters by scrolling old characters off the display. The area is defined by the top and bottom display margins. The smallest scrolling region allowed is two lines. The number of the top margin must be at least one less than the number of the bottom margin.

Scrolling Margins

Name (Mnemonic)	Sequence	Action
Set top and bottom margins (DECSTBM)	CSI Pt ; Pb r	Sets top and bottom scrolling margins. The value of Pt must be less than Pb. Moves the cursor to Column 1, Line 1 of the page.

Printing

Select print operations with control sequences. When characters are printed on the screen, printer tab stops are ignored. Print characters are spaced with the SP character. The computer sends a carriage return and line feed, vertical tab, or form feed after the last printable character on a line (not a space character).

Note: The computer ignores printer port DSR exchanges.

Printing Sequences

Name (Mnemonic)	Sequence	Action
Auto print mode	CSI ? 5 i	Turns on auto print mode. Display lines print when you move the cursor off the line with a line feed, form feed, vertical tab, or auto wrap. The printed line ends with a carriage return and the character that moved the cursor off the previous line (LF, FF, or VT). Auto wrap lines end with a line feed.
	CSI ? 4 i	Turns off auto print mode.
Printer controller	CSI 5 i	Turns on printer controller mode. The computer sends received characters to the printer without displaying them on the screen. All characters and character sequences except NUL, XON, XOFF, CSI 5 i, and CSI 4 i are sent to the printer. The computer does not insert or delete spaces, provide line delimiters, or select the correct printer character set. Keyboard activity is still directed to the host.
	CSI 4 i	Turns off printer controller mode.
Print cursor line	CSI ? l i	Prints the data on the display line with the cursor. Cursor position does not change. Print-cursor-line sequence is complete when the line prints. No CR/LF is attached to the line.
Print screen	CSI i	Prints the screen display (full screen or scrolling region, depending on the print extent DECPEX selection). Printer form feed mode (DECPFF) selects either a form feed (FF) or nothing as the print terminator. Print screen sequence is complete when the screen prints.
	CSI 0 i	Same as CSI i.

User-Defined Keys (DECUDK)

Note: The DECUDK command is ignored in VT100 mode.

The computer has 20 user-defined keys (UDKs). You can define the codes of 15 of these keys. The keys are F6 through F14, F15 (Help), F16 (Do), and F17 through F20.

Use UDKs to store and recall text and commands that you often use with applications. You should refer to your application's software manual for the commands you can store in user-defined keys.

If you use dual sessions, you can define UDKs for each session. However, you can only save one set of UDK definitions. For each session you can use a total of 256 characters to define UDKs.

Using UDKs

To enter definitions for the 15 UDKs, you program the definitions with DECUDK device control strings. To use a UDK after you have defined it, the computer must be in caps lock mode. For example, if you defined the [F6] key, you can use it by pressing the caps lock key for the computer and then the key sequence for the [F6] key. The following chart shows key combinations for CAPS lock.

Note: The host system can also define function keys.

CAPS Lock Combinations

Computer	Key Sequence
700 Color	[Alpha] [1] *(numeric keypad)* [Alpha] [CapLock] *(alphanumeric keypad)*
CK30	■□ [Shift] *(52-key keypad)* [Func] [Shift] *(50-key keypad)* ■□ [A] *(42-key keypad)*
CK31	■□ [Shift] *(52-key keypad)*
CK32	■□ [C]
CN3	■□ [Shift] *(QWERTY keypad)* [Green] [1] *(numeric keypad)*
CV30	[CAPS]
CV60	[Caps Lock]

UDK Memory Space

There are 256 bytes of memory space available for the 15 UDKs. Space is supplied on a first-come, first-serve basis. When the 256 bytes are full you cannot define any more keys until you clear some of the memory space. Three ways that you can clear space are:

- Redefine one or more UDKs by using a DECUDK control string.
- Clear one or more UDKs by using a DECUDK control string.
- Clear all UDKs with a computer power-up or hard computer reset (RIS) operation.

Note: All UDK key definitions are lost when power is lost.

Programming UDKs

Use the following Device Control String (DCS) format to load UDK definitions from the host.

DCS Format

DCS	Pc ; Pl	|	Ky1/St1;0Kyn/Stn	ST
Device control string introducer	Clear and lock parameters	Final character	Key definition string	String terminator

- **DCS** indicates the beginning of a device control string. DCS is an 8-bit C1 character (90 hexadecimal). You can use ESC P (1B and 50 hexadecimals) for a 7-bit environment.
- **Pc** is the clear parameter that selects how to clear key definitions.

Parameters

Pc	Meaning
0 *(default)* or none	Clear all keys before loading new values.
1	Clear one key at a time, before loading a new value.

When Pc is 1, the computer only clears the keys you are loading. By using a Pc value of 1, you can redefine some keys without redefining them all.

Note: There are 256 bytes of memory for all UDKs. A key definition can only use the number of bytes available when that key is loaded.

If Pc is 1, a key load may fail because no memory space is available. The reason for this is, with Pc set to 1, keys are cleared and loaded sequentially. If the new definition for a key is larger than the old one you may exceed the 256-byte limit.

Example

Suppose F6 contains 120 bytes, F7 contains 110 bytes, and F8 contains 20 bytes. You try to load F8 with 40 bytes, F6 with 1 byte, and F7 with 1 byte, in that order. This works if all keys are cleared first (Pc is 0), but not if keys are cleared one at a time (Pc is 1). When you try to load F8 with 40 bytes, the load fails because only 26 bytes are free at that time. (256 maximum – 120 in F6 – 110 in F7 = 26.)

Pl is the lock parameter. Pl determines whether the key definitions are locked or unlocked after you load them.

Lock Parameter

Pl	Meaning
0	Lock the keys. If you want to load new values into the keys, you must unlock the keys through the computer's firmware menus.
1 *(default)* or none	Do not lock the keys. Keys are unlocked and can be redefined with another DECUDK string.

Note: If Pl is 1 and the keys are already locked, nothing happens. The computer ignores UDK DSR exchanges.

The computer uses a special lock to allow or prevent the programming of UDKs. You can turn on this lock through the computer's firmware menus or from the host (with a DECUDK device control string). The lock affects all programmable keys. When using the lock follow these guidelines:

- Unlock the keys to define them. The keys must be unlocked before you can define them. You can only unlock the keys through the computer's firmware menus. If a key is locked and an application tries to redefine the key with a DECUDK sequence, the computer ignores the sequence.
- Lock the keys to prevent redefinition. You can lock the keys through the computer's firmware menus or from the host (with a DECUDK sequence). New key definitions are unlocked by default.

The **vertical bar** (|, 7C hexadecimal) is the final character. It identifies this control string as a DECUDK.

Ky1/St1;0Kyn/Stn are the key definition strings. You include these strings between the final character (|) and the string terminator (ST). Each string consists of a key selector number (Kyn) and a string parameter (Stn), separated by a slash (/, 2F hexadecimal). A semicolon (3B hexadecimal) separates different strings. The key selector number (Kyn) indicates which key you are defining.

Below are the definable keys and identifying values.

Definable Keys

Key	Value	Key	Value
F6	17	F13	25
F7	18	F14	26
F8	19	Help	28
F9	20	Do	29
F10	21	F17	31
F11	23	F18	32
F12	24	F19	33
F13	25	F20	34
F14	26		

String parameters (Stn) are the encoded definition of the keys. String parameters consist of hex pairs in the following ranges:

- 30 through 39 hexadecimal (0 through 9)
- 41 through 46 hexadecimal (A through F)
- 61 through 66 hexadecimal (a through f)

When you combine these hexadecimal values they represent an 8-bit quantity. This method lets you use any of the 256 character codes in the key string. You can enter key definition strings in any order. The default for Stn is “empty.” The key is undefined.

ST is the string terminator. ST (9C hexadecimal) is a C1 8-bit character. You can use ESC (1B and 5C hexadecimals) for a 7-bit environment.

Loading UDKs

Following are some guidelines for loading UDKs.

- Clear UDK memory space before loading new definitions. Use a DECUDK string to clear keys without locking them. Then you can use another DECUDK string to redefine the keys and lock them.
- If you redefine a key, the old definition is lost. This may clear some space if the new definition uses fewer bytes than the old one.
- There are two ways to lock UDKs, but only one way to unlock them. To lock UDKs, you can use either the computer's firmware menus or a DECUDK control string. To unlock UDKs, you must use the computer's firmware menus.
- The default value for each key definition is empty. When you clear UDKs, they are empty.
- An invalid hex pair in a DECUDK string is ignored.

Examples of Device Control Strings

This sequence clears UDKs: `DCS 0 ; 1 | ST`

This sequence locks UDKs: `DCS 1 ; 0 | ST`

Suppose you want to define the F20 key to be PRINT without clearing or locking any other keys and you are using 8-bit mode. The first part of the sequence would look like this: `É1;1|34/` where 34 is the code for the F20 key and “É” is the ASCII character for 90 hexadecimal.

After the slash character “/” (2F hexadecimal) you would include the definition. The rest of the sequence after the slash character would be, `5052494E54£` where the hex encoding for PRINT is:

50 = P
52 = R
49 = I
4E = N
54 = T

The ST ASCII character “£” (9C hexadecimal) marks the end of the control string. The complete string is: `É1;1|34/5052494E54£`

Down-Line Loadable Character Set

The computer does not support the ability to create and down-line load a character set of up to 94 characters. This character set is also called a dynamically redefinable character set (DRCS).

DRCS

Function	Action
Down-line load DRCS characters	Ignored
Clear a down-line loaded character set	Ignored

Reports

The computer sends reports in response to host computer requests. Reports provide identification (type of computer). Two categories of reports are available: primary/secondary device attributes, and device status reports.

Device Attributes (DA)

The following table shows the action taken by the computer when it receives primary and secondary device attribute exchanges.

During a primary exchange, the host asks for the computer's service class code. The computer responds with "I am a service class 2 computer." During a secondary exchange the computer responds with "I am a VT (identification code of 1) family computer."

DA Actions

Communication	DA Exchange	Sequence	Meaning
Host to VT220 (request)	Primary	CSI > c *or* CSI > 0 c	What type of computer are you?
	Secondary	CSI c *or* CSI 0 c	What type of computer are you?
VT220 to host (response)	Primary	CSI > c	I am a VT100 Computer
		CSI > 1 c	I am a VT220 Computer.
		CSI > 24 c	I am a VT320 Computer.
		CSI > 19 c	I am a VT100 Computer.
	Secondary	CSI ? 61 c	I am a VT100 Computer.
		CSI ? 62 ; 8 ; 9 c	I am a VT220 Computer supporting user-defined keys and national replacement character sets.
		CSI ? 63 ; 8 ; 9 c	I am a VT320 Computer supporting user-defined keys and national replacement character sets.
		CSI ? 63 ; 8 ; 9 ; 13 c	VT340 Computer supporting user-defined keys, national replacement character sets, local editing.

Device Status Reports (DSR)

This table shows the action taken by the computer when it receives device status report exchanges, when the host computer asks for the general operating status of the computer or printer, or both. If the computer is in printer controller mode, the printer receives the DSR request but cannot answer.

DSR Actions

DSR Exchange	Communication	Sequence	Action
VT220	Host to VT220 (Request for status)	CSI 5 n	Ignores this code.
	VT220 to host (DA response)	CSI 0 n *or* CSI 3 n	Ignores these codes.
	Host to VT220 (Request for cursor position)	CSI 6 n	"Please report your cursor position using a CPR (not DSR) control sequence."
	VT220 to host (CPR response)	CSI Pv; Ph R	"My cursor is positioned at ______ (Pv); ______ (Ph)." (Pv=row, Ph=column)
Printer port	Request for printer status	CSI ? 15 n	What is the printer status?
	Response	CSI ? 13 n	No printer.
User-defined keys	Request for UDK status	CSI ? 25 n	Ignored in VT100 mode.
	Response	CSI ? 20 n	UDKs are unlocked.
		CSI ? 21 n	UDKs are locked.
Keyboard language	Request for keyboard language	CSI ? 26 n	What is the keyboard language?
	Response	CSI ? 27; Pn n	North American keyboard dialect.

Identification

The computer supports the identification sequence.

Identification Sequence

Name (Mnemonic)	Sequence	Action
Identification (DECID)	ESC Z	Ignored

Computer Reset

Computer reset escape sequences cause either a soft or hard computer reset.

Soft Computer Reset (DECSTR)

Note: The DECSTR control function is available for all VT/ANSI modes.

DECSTR changes most of the computer's current settings to the power-up default settings listed in the following table. The escape sequence is:

```
CSI ! p
```

DECSTR affects only those functions listed in the following table.

Soft Computer Reset (DECSTR) States

Mode	Mnemonic	State After DECSTR
Text cursor enable	DECTCEM	Cursor enabled.
Insert/replace	IRM	Replace. ("Insert" if local editing mode.)
Origin	DECOM	Absolute (cursor origin at upper-left of screen).
Auto wrap	DECAWM	No auto wrap.
Keyboard action	KAM	Unlocked.
Numeric Keypad	DECNKM	Numeric characters.
Cursor keys	DECCKM	Normal (arrow keys).
Edit	DECEDM	Interactive.
Transmit execution	DECTEM	Immediate.
Erasure	ERM	All characters.
Guarded area transfer	GATM	All characters.
Multiple area transfer	MATM	All selected areas.
Selected area transfer	SATM	All areas.
Scanner lock		Reset.

The following table lists other control functions. All tab stops are reset. User-defined keys are cleared.

Soft Computer Reset (DECSTR) States

Mode	Mnemonic	State After DECSTR
Set top and bottom margins	DECSTBM	Top margin = 1. Bottom margin = page length.
All character sets	G0, G1, G2, G3, GL, GR	VT/ANSI default settings.
Select graphic rendition	SGR	Normal rendition.
Start of selected area	SSA	Cleared.
End of selected area	ESA	Cleared.
Start of protected area	SPA	Cleared.
End of protected area	EPA	Cleared.
Save cursor state	DECSC	Home position with VT/ANSI defaults.

Hard Computer Reset (RIS)

When the RIS is complete, the computer sends XON to resume communication. The RIS sequence is: `ESC c`

RIS resets values to factory default settings. It is the same as DECSTR, but also clears the screen, returns the cursor to the upper-left corner of the screen, and sets the SGR function to normal rendition.

Tests and Adjustments

The computer has adjustment patterns you can invoke from the host computer with escape sequences. Adjustment sequences send uppercase Es to the computer's display. Only qualified technicians perform adjustment procedures.

This table shows the action taken by the computer when it receives test and adjustment sequences.

Test and Adjustment Sequence Actions

Name (Mnemonic)	Sequence	Action
Tests (DECTST)	CSI 4 ; Ps ; ..... ; Ps y	Ignored
Adjustments (DECALN)	ESC # 8	Fills display with uppercase Es

VT52 Mode Escape Sequence

The computer does not support VT52 mode, which allows the VT/ANSI Computer to operate with DEC software written for VT52 Computers.

Define Area Qualification

The define area qualification (DAQ) command indicates that the active position is the first character position of a qualified area. The end of a qualified area is indicated by the beginning of the following qualified area. The qualifications of the area are specified according to the parameter(s). The following table lists qualification sequences.

Note: DAQ is ignored unless the computer is in VT340 mode.

Define Area Qualification

Qualification	**Sequence and Ps** (Single Parameter)	Action
Accept all input	CSI 0 o	Accepts all input
Accepts no input (protected) and does not transmit (unguarded)	CSI 1 o	Accepts no input
Accept graphics	CSI 2 o	Accepts graphics
Accept numerics	CSI 3 o	Accepts numerics
Accept alphabetics	CSI 4 o	Accepts alphabetics
Right justify in area	CSI 5 o	Right justifies area
Zero fill in area	CSI 6 o	Fills area with zeros.
Horizontal tab stop at start of area	CSI 7 o	Puts horizontal tab stop at start of area
Accepts no input (protected), but selects for transmission (unguarded)	CSI 8 o	Accepts no input
Space fill in area	CSI 9 o	Fills area with spaces
Invisible	CSI 1 0 o	Makes passwords invisible

Private Sequences

Private sequences enable or disable autosend.

Private Sequences

Mode	Sequence*	Action
Set	ESC [= 3 l	Enables autosend. Transmits the screen when the last field is filled. If you are on the last defined field and get data from the bar code reader or RS-232 port, the display will also be automatically sent even if the field not filled.
Reset	ESC [= 3 h	Disables autosend. Only transmits when a terminating key is pressed.

** The last character of the set mode sequence is lowercase L (6C hexadecimal).*

Proprietary Sequences

Following are INTERMEC® extensions to the data stream.

Line Edit and Character Modes

Two proprietary escape sequences enable the host to put the computer into Line Edit (block) mode or Character mode. Note that the computer enters Character mode when it is cold-started.

Line Edit and Character Modes

Mode	Sequence*
Line Edit	CSI = 1 l
Character	CSI = 1 h
** The last character of the Line Edit sequence is lowercase L (6C hexadecimal).*	

Norcompress

The Norcompress function handles the norrc command, where “a” is the number of iterations that “b” occurs in the data stream. The sequence for Norcompress is **CSI a;bz**.

Scanner Lock Mode

Scanner lock mode is similar to the KAM function; however, you do not need to repeatedly send the Locked command (this command is assumed after each scan). The scanner lock state can also be reset by the RIS command (ESC c) or the DECSTR command (CSI ! p).

Scanner Lock Mode

Mode	Sequence*	Action
Set	CSI = 4 h	The host sends this sequence, which puts the computer into Scanner Lock mode. When the computer is in this mode, only one scan is allowed; the scanner is then inhibited. After this sequence is sent, the user can scan only once (no additional scans are allowed). To re-enable the scanner, the host must send CSI 2 l to unlock the scanner.
Reset	CSI = 4 l	Resets Scanner Lock mode (default).
** The last character of the reset sequence is lowercase L (6C hexadecimal).*		

VT340 Applications

Supported VT340 computer applications include setting the right margin, using control functions, and creating text forms.

Right Margin

The computer's display contains 2 to 80 columns. To ensure that the cursor will not go beyond Column Pn, use the following sequence to set the number of columns (Pn is a number from 2 to 80):

```
CSI Pn $ |
```

Control Functions

In editing mode you can use the erasure, deletion, and insertion control functions. The character protection field lets you design text forms that cannot be changed or overwritten. In edit mode, character protection also depends on the setting of erasure mode (ERM):

- If ERM is set all characters can be changed, erased, and moved, regardless of protection.
- If ERM is reset only unprotected characters can be changed, erased, and moved. These control functions do not affect protected areas in page memory:
  - Insert line (IL)
  - Delete line (DL)
  - Insert character (ICH)
  - Delete character (DCH)
  - Erase in display (ED)
  - Erase in line (EL)
  - Erase character (ECH)

Text Forms

When using C1 control characters SPA, EPA, SSA, and ESA to create forms on the host, the rows and columns to which the cursor moves must be in sequential order. The following shows some sequences that create a text form, start with the left column.

Control Characters for Auto-Login Script File

Sequence	Definition	Sequence	Definition
ESC [1; 1 H	Home cursor	ESC V	Start protected area
ESC [2 J	Clear screen	ESC [4; 1 H	Move cursor to Row 4, Col 1
ESC [? 10 h	Start local editing mode	“Lot:”	Display literal
ESC V	Start protected area	ESC [1 D	Move cursor back 1 position
ESC [1; 2 H	Move cursor to Row 1, Col 2	ESC W	End protected area
“Manufacturing”	Display literal	ESC [1 C	Move cursor forward 1 position
ESC [1 ; 17 H	Move cursor to Row 1, Col 17	ESC F	Start selected area
“Receiving”	Display literal	ESC [4; 9 H	Move cursor to Row 4, Col 9
ESC [2; 1 H	Move cursor to Row 2, Col 1	ESC G	End selected area
“LIP:”	Display literal	ESC V	Negative Acknowledge
ESC W	End protected area	ESC [5; 1 H	Move cursor to Row 5, Col 1
ESC [1 D	Move cursor back 1 position	“Qty:”	Display literal
ESC W	End protected area	ESC [1 D	Move cursor back 1 position
ESC [l C	Move cursor forward 1 position	ESC W	End protected area
ESC F	Start selected area	ESC [1 C	Move cursor forward 1 position

Control Characters for Auto-Login Script File (continued)

Sequence	Definition	Sequence	Definition
ESC [2; 15 H	Move cursor to Row 2, Col 15	ESC F	Start selected area
ESC G	End selected area	ESC [5; 13 H	Move cursor to Row 5, Col 13
ESC V	Start protected area	ESC G	End selected area
ESC [3; 1 H	Move cursor to Row 3, Col 1	ESC V	Start protected area
"Item:"	Display literal	ESC [24; 80 H	Move cursor to Row 24, Col 80
ESC [1 D	Move cursor back 1 position	ESC W	End protected area
ESC W	End protected area	ESC [26 $ |	Set the number of columns to 26
ESC [1 C	Move cursor forward 1 position	ESC [1; 8 r	Set top at 1, bottom at 8 Rows
ESC F	Start selected area	ESC [3 g	Clear tab stops
ESC [3; 16 H	Move cursor to Row 3, Col 16	ESC [2; 5 H	Move cursor to Row 2, Col 5
ESC G	End selected area		

If this information is entered on the form:

`111111111` for "LIP:"
`2222222222` for "Item:"
`3333` for "Lot:"
`44444444` for "Qty:"

The data stream will be sent to the host as indicated in the following chart. (If guarded area transfer mode (GATM) is reset, then the record separator is processed as a protected field entry.)

GATM Data Stream

Sequence	Definition	Sequence	Definition
RS	Protected field Line 1	3333	
RS	First protected field Line 2	RS	
111111111	Selected field Line 2	RS	
RS	First protected field Line 3	44444444	
RS	First protected field Line 3	RS	
2222222222	Selected field Line 3	RS	
RS	Second protected field Line 3	RS	
RS	First protected field Line 4	RS	

IBM 330X Sequences

The following escape sequences are supported for IBM 330X computers:

IBM 330X Escape Sequences

Escape Sequence	Action
ESC n A	Moves cursor up “n” lines. Cursor stops at top margin.
ESC n B	Moves cursor down “n” lines. Cursor stops at bottom margin.
ESC n C	Moves cursor right “n” columns. Cursor stops at right margin.
ESC n D	Moves cursor left “n” columns. Cursor stops at left margin.
ESC I	Erase from the cursor to the end of the line, including the cursor position. Line attribute is not affected. If erasure mode (ERM) is reset, this function cannot erase protected characters.
ESC n O	Deletes “n” lines, starting at the line with the cursor. If fewer than “n” lines remain from the current line to the end of the scrolling region, the number of lines deleted is the lesser number. As lines are deleted, the lines within the scrolling region and below the cursor move up, and blank lines are added at the bottom of the scrolling region. The cursor is reset to the first column.
ESC L	Erases the entire display. All lines are erased and changed to single width. The cursor does not move.
ESC n P	Inserts “n” characters starting with the character at the cursor location, with the character attribute set to normal. The cursor does not move and remains at the beginning of the inserted characters. A parameter of “0” or “1” inserts one blank character. Data on the line is shifted forward as in character insertion. In edit mode (DECEDM), if erasure mode (ERM) is reset, text moved into a protected character field is lost.
ESC n Q	Deletes “n” characters, starting with the character at the cursor position. When a character is deleted, all characters to the right of the cursor move to the left. This creates a space character at the right margin for each character deleted. Character attributes move with the characters. Spaces created at the end of the line have all of the character attributes off. In edit mode (DECEDM), if erase mode (ERM) is reset, then this command cannot delete protected characters.
ESC n N	Inserts “n” lines at the cursor. If fewer than “n” lines remain, from the current line to the end of the scrolling region, the number of lines inserted is the lesser number. Lines within the scrolling region at and below the cursor move down. Lines past the bottom margin are lost. The cursor is reset to the first column. This sequence is ignored if the cursor is outside of the scrolling region. In edit mode (DECEDM), if erasure mode (ERM) is reset, lines that move down into a line with a protected character field are lost.
ESC n X	Moves the cursor to the “nth” column.
ESC n Y	Moves the cursor to the “nth” row.
ESC H	Moves the cursor to the top left of the memory screen.
ESC J	Erases from the cursor to the end of the display, including the cursor position. Line attributes become single height/single width for all completely erased lines.

ANSI Mode Sequences

Cursor Positioning

The computer supports cursor positioning sequences in this table.

Cursor Positioning

Name (Mnemonics)	Sequence	Action
Cursor backward tab (CVT)	CSI Ps Z	Moves the cursor to previous Ps tab stop(s). If there is no previous tab stop, then the cursor moves to the left margin.
Cursor horizontal absolute (CHA)	CSI Ps G	Moves the cursor to the Ps column on the current line.
Cursor horizontal tab (CHT)	CSI Ps I	Moves the cursor to the next Ps tab stop(s). If there is not next tab stop, then the cursor moves to the right margin.
Cursor next line (CNL)	CSI Ps E	Moves the cursor down Ps lines. If at the bottom margin of the screen, then the cursor is not moved.
Cursor previous line (CPL)	CSI Ps F	Moves the cursor up Ps lines. If at the top margin of the screen, then the cursor is not moved.
Horizontal position absolute (HPA)	CSI Ps	Moves the cursor to Ps column on the current line.
Horizontal position relative (HPR)	CSI Ps a	Moves the cursor Ps columns from the current location. If past the right margin, then the cursor is stopped at the right margin.
Vertical position absolute (VPA)	CSI Ps d	Moves the cursor to Ps row using the current column.
Vertical position relative (VPR)	CSI Ps e	Moves the cursor Ps rows from the current location. If at the bottom margin, the cursor is stopped at the bottom margin.
Cursor vertical tab (CVT)	CSI Ps Y	Moves the cursor down Ps vertical tab stops. If at the bottom margin, then the cursor is stopped.

Cursor Tabulation Control

This table lists cursor tabulation control (CBT) sequences in ANSI mode.

Cursor Tabulation Control

Sequence	Action
CSI 0 W	Sets horizontal tab stop at current location.
CSI 1 W	Sets vertical tab stop at current location.
CSI 2 W	Clears horizontal tab stop.
CSI 3 W	Clears vertical tab stop.
CSI 4 W	Clears all horizontal tab stops this line.
CSI 5 W	Clears all horizontal tabs stops in the machine.
CSI 6 W	Clears all vertical tab stops

Scrolling

This table lists scrolling sequences in ANSI mode.

Scrolling Sequences

Name (Mnemonics)	Sequence	Action
Scroll up (SU)	CSI Ps S	Scrolls display up Ps lines.
Scroll down (SD)	CSI Ps T	Scrolls display down Ps lines.

Transmitted Keyboard Codes

Transmitted codes are generated by computer keys and sent to the used host computer or software application. The next few pages describe computer capability to emulate codes generated by the VT/ANSI computer keyboard main keypad, editing keypad, auxiliary keypad, and top-row function keys. Codes are sent via Line Edit (block) mode or optimized Character mode selected through the keyboard function labeled MODE on the overlay.

Line Edit (Block) Mode

Line Edit (block) mode is a synchronous condition where the computer temporarily stores or "buffers" keys you press. It sends the cumulative data to the host computer when you press one of these terminating keys:

- [Enter]
- [F6] through [F20], [PF1] through [PF4]
- [CTRL]+[A] through [CTRL]+[Z]
- Forward Tab
- DEL (7F hexadecimal)
- Backspace
- The following also apply to Line Edit (block) mode. For codes generated by the cursor control key, see "Cursor Keys" on page 333. For codes generated by the auxiliary keypad keys, see "Auxiliary Keypad" on page 334.
- When you press [PF1], the following key will also be transmitted directly to the host computer.
- When scanning, the computer automatically enters Line Edit (block) mode for each individual scan. When it completes scanning, it returns to the mode it was set at before it started scanning.

Character Mode

Character mode (the default) is a condition where the computer sends, to the host computer, each key as you press it. You should place the computer into Character mode during these situations:

- When the computer should immediately send information to the host computer. This operation is called “type-ahead.”
- When the computer's keyboard sends a character to the display as well as to the host. This operation is called “local echo.” If local echo is disabled, the host receives the character from the computer and then sends the return response.

To alternate between Line Edit (block) mode and Character mode, press the mode key as defined in Chapter 3, “Using the Computer Keypad.”

Transmitted Keypad Keys

The following pages describe codes generated by main keypad keys, editing keys, auxiliary keys, and top-row function keys.

Main Keypad

The computer emulates all of the standard and most of the special function keys on the VT/ANSI main keypad. The standard keys generate letters, numbers, and symbols, either alone or in combination with other keys. The function keys generate special function codes listed in this table.

Main Keypad Function Keys

VT/ANSI Key	Code
Ctrl	Does not send a code when used alone. Is always used in combination with another key to send a control code.
Delete	DEL (7F hexadecimal).
Lock	None.
Return	CR (0D hexadecimal) or CR (0D hexadecimal) LF (0A hexadecimal).
Shift	Does not send a code when used alone; sends uppercase characters when used with other standard keys.
Space bar	SP (20 hexadecimal).
Tab	HT (09 hexadecimal).

Editing Keypad

The computer has editing and cursor control keys. Editing keys have functions assigned to them by the application software in use. Refer to your VT/ANSI application software manual for the uses of the editing keys.

Editing Keys

The following table lists the codes generated by the editing keys.

Note: The editing keys do not send codes in VT100 mode.

Editing Keys

VT/ANSI Key	Code	VT/ANSI Key	Code
Find	CSI 1 ~	Prev screen	CSI 5~
Insert here	CSI 2 ~	Remove	CSI 3~
Next screen	CSI 6 ~	Select	CSI 4~

Cursor Keys

The following table lists codes generated by the computers' cursor keys.

Cursor Keys

Key	Cursor Key Mode Reset (Normal)	Cursor Key Mode Set (Application)
Down arrow	CSI B	SS3 B
Left arrow	CSI D	SS3 D
Right arrow	CSI C	SS3 C
Up arrow	CSI A	SS3 A

Sending Host Cursor Keys

In keyboard unshifted and shifted modes, the cursor control key sends a host cursor key to the host computer. The cursor control key emulates the arrow keys of a VT/ANSI Computer in either their normal arrow key mode or Application mode, depending on the state set by the programmer.

Sending Host Cursor Keys

To Send	Press Computer Keys	To Send	Press Computer Keys
Host cursor down key	[▼]	Host cursor down key	[SFT]+[▼]
Host cursor left key	[**L**]	Host cursor left key	[SFT]+[**L**]
Host cursor right key	[**R**]	Host cursor right key	[SFT]+[**R**]
Host cursor up key	[▲]	Host cursor up key	[SFT]+[▲]

Auxiliary Keypad

Auxiliary keypad keys enter numeric data. The application software in use can also assign functions to these keys; refer to your VT/ANSI application software manual for their uses. The following table shows codes generated by the auxiliary keys in keypad Application mode.

Auxiliary Keypad

VT/ANSI Key	Keypad Mode	VT/ANSI Key	Keypad Mode
0	SS3 p	9	SS3 y
1	SS3 q	- (hyphen)	SS3 m
2	SS3 r	, (comma)	SS3 l
3	SS3 s	. (period)	SS3 n
4	SS3 t	Enter	SS3 M
5	SS3 u	PF1	SS3 P
6	SS3 v	PF2	SS3 Q
7	SS3 w	PF3	SS3 R
8	SS3 x	PF4	SS3 S

Top-Row Function Keys

The following table lists function keys and the codes generated.

Top-Row Function Keys

Key	VT220/320/340 Mode	VT100 Mode
F5 (Break)	No code	No code
F6	CSI 1 7 ~	No code
F7	CSI 1 8 ~	No code
F8	CSI 1 9 ~	No code
F9	CSI 2 0 ~	No code
F10	CSI 2 1 ~	No code
F11	CSI 2 3 ~	Esc
F12	CSI 2 4 ~	BS
F13	CSI 2 5 ~	LF
F14	CSI 2 6 ~	No code
F15	CSI 2 8 ~	No code
F16	CSI 2 9 ~	No code
F17	CSI 3 1 ~	No code
F18	CSI 3 2 ~	No code
F19	CSI 3 3 ~	No code
F20	CSI 3 4 ~	No code

Local Edit Mode

If your application software program supports local editing, the computer can be programmed to operate in Local Edit mode. Local Edit mode is a feature of the VT340 Computer. Use the mode to send characters to page memory instead of to the host. The computer sends a block of edited text to the host after you press a terminating key. Terminating keys in Local Edit mode are [Enter] and [F1] through [F20].

Edit Mode and Interactive Mode

The edit mode control function (DECEDM) selects edit mode or interactive mode. The function determines when the computer sends data to the host. The following Local Edit Modes table describes the modes.

Edit and Interactive Modes

Mode	Sequence*	Action
Set	CSI ? 1 0 h	Selects edit mode to turn on the display annunciator. Computer stores all typed characters in page memory for local editing. After data is edited, it can be sent in a block to the host.
Reset	CSI ? 1 0 l	Selects interactive mode. (Turns off the annunciator in the display.) The computer immediately sends typed characters to the host.
* *The last character in the reset sequence is lowercase L (6C hexadecimal).*		

Local Editing Setup

To use the computer in Local Edit mode, the local editing features must be set up for the application software as described in the following table.

Local Editing Setup

Feature	Setting	Description
Edit mode	Edit	Edited text is stored in memory until terminating key is pressed, which sends text to the host.
Edit key execution mode		Not supported.
Transmit execution mode	Immediate	Computer sends data to host right after a terminating key is pressed.
Local editing application keys	Suffix transmit	When [F1] through [F20] is pressed, computer sends that function to host after sending a block of data.
Line transmit mode	Disabled	Computer sends full page of data to host after terminating key is pressed.
Transfer termination mode	Enabled	After terminating key is pressed, computer sends scrolling region (area inside scrolling margins).
VT131 transfer mode		Not supported.
Space compression	Disabled	Computer sends a space character for each unused character position
End of line characters		On the computer, no characters indicate the EOL in a data block.
End of block characters		On the computer, no characters indicate the end of a data block.

Selecting Characters to Send

Three control functions allow you to define which characters the computer can send to the host. The following table shows how the control functions select which characters the computer sends.

Selecting Character Fields for Transmission

Fields Selected	GATM	SATM	MATM
All fields	Set	Set	Unavailable
Unprotected fields only	Reset	Set	Unavailable
Selected fields only	Set	Reset	Set
Selected field with cursor only	Set	Reset	Reset
Unprotected and selected fields	Reset	Reset	Set
Selected field with cursor only	Reset	Reset	Reset

Guarded Area Transfer Mode (GATM)

The GATM control function selects whether the computer sends all characters or only unprotected characters to the host. When GATM is unprotected (reset), the computer sends a record separator (RS, 1E hexadecimal) to the host in place of a protected field. This lists GATMs.

Guarded Area Transfer Mode

Mode	Sequence*	Action
Set (All)	CSI 1 h	Selects all characters. During block transmission, the computer can send all protected and unprotected characters to the host.
Reset (Unprotected)	CSI 1 l	Selects unprotected characters. During a block transmission, the computer can send only unprotected characters to the host.
** The last character of the reset sequence is lowercase L (6C hexadecimal).*		

Selected Area Transfer Mode (SATM)

The SATM control function determines whether the computer can send all characters or only selected characters to the host. Selected characters are characters defined as eligible to send to the host. This table lists SATMs.

Selected Area Transfer Mode

Mode	Sequence*	Action
Set (All)	CSI 1 7 h	Selects all characters. The computer can send selected and unselected characters on the current page to the host.
Reset (Unprotected)	CSI 1 7 l	Selects only selected characters. The computer can only send selected characters on the current page to the host.
** The last character of the reset sequence is lowercase L (6C hexadecimal).*		

Multiple Area Transfer Mode (MATM)

The MATM control function determines what selected character areas the computer can send to the host. MATM work only when SATM is reset.

Multiple Area Transfer Mode

Mode	Sequence*	Action
Set (All)	CSI 1 5 h	The computer can send all selected areas on the page to the host.
Reset (Unprotected)	CSI 1 5 l	Selects one area. The computer can send only the selected area with the cursor. If the cursor is not in a selected field, cursor moves to the next selected field.
** The last character of the reset sequence is lowercase L (6C hexadecimal).*		

Defining Selected Areas

Start selected area (SSA) and end selected area (ESA) control functions select which characters on the current page the computer can send to the host. SSA and ESA are 8-bit C-1 control characters that can also be coded as 7-bit escape sequences. The following conditions apply to SSA and ESA:

- They work only when SATM is reset.
- If the computer receives ESA before SSA, it ignores ESA.
- If SSA is not followed by ESA on the same page, the SSA has no effect on that page. Selected areas must always end with ESA.
- ED, EL, or ECH control functions cannot change selected areas.

SSA and ESA

Name (Mnemonic)	8-Bit	7-Bit Equivalent	Action
SSA	SSA (86 hex)	ESC F (1B, 46 hex)	Marks the cursor position as the first of a string of character positions the computer can send to the host.
ESA	ESA (87 hex)	ESC G (1B, 47 hex)	Defines the cursor position as the last of a string of character fields the computer can send to the host.

Local Edit Mode Keys

Keys with special functions in Local Edit mode are described in this table.

Local Edit Mode Keys

Keys	Function
Find	Advances cursor to the top margin, Column 1.
Tab	Advances cursor to the next tab stop. If in protected area, it advances to the next selectable field. If tab stop is selected, it advances to the next selectable field.
Insert here	Default mode is "insert." Key toggles between insert and overstrike. If insert mode, all following characters in the field are shifted right one position. If the field is full the last character is deleted.
Remove	If in protected area, computer emits beep and cursor advances to first position of next selectable field. If not found, it advances to bottom margin (default is Row 24, Column 80). If in unprotected field, it is cleared and the cursor is positioned to the beginning of the selectable field.

Local Edit Mode Keys (continued)

Keys	Function
Prev screen Next screen Select	Ignores these keys.
Backspace	Moves cursor one position to the left until it reaches the left margin.
Delete	Deletes the previous selectable character.
A-Z, 0-9, Auxiliary keypad	If a key is pressed while in a protected area, computer emits beep, advances cursor to next selectable field, enters characters. If field not found, cursor advances to bottom margin.
Back Tab	Cursor moves back to the first occurrence of the following: • Previous tab stop. • Beginning of the current unprotected field. • Beginning of the previous unprotected field. • Beginning of scrolling region. (Also called top margin.)
Cursor [▲]	Cursor moves up one line until it reaches the top margin. The computer beeps when the cursor reaches the top margin.
[▼]	Cursor moves down one line until it reaches the bottom margin. The computer beeps when the cursor reaches the bottom margin.
[▶]	Cursor moves right one character position until it reaches the right margin. The computer beeps when the cursor reaches the right margin.
[◀]	Cursor moves left one character position until it reaches the left margin. The computer beeps when the cursor reaches the left margin.

Scanning in Local Edit Mode

When scanning is enabled, data from the scanner is placed into the field where the cursor is located. If the cursor is not in an input field, the data from the scanner is placed into the next input field. If scanner data fills the first input field, remaining characters are placed into the next input field.

When the bar code does not fill the input field, the cursor stays where it ended. If autotab scanning is enabled, the cursor is placed at the beginning of the next input field. If that field is not found, extra data is lost.

Set Transmit Termination Character (DECTTC)

The computer supports the set transmit termination character (DECTTC) control function. Use the control function to select a character to indicate the end of a block transmission. You do not need to use an end-of-block character. The computer sends the end-of-block character to the host at the end of each block transmission. The control sequence is: `CSI Ps |`

Set Transmit Termination Characters

Ps	End-of-Block Character	Action
0 (default)	No character (DECTTC disabled)	
1	FF (0C hexadecimal)	Form feed
2	ETX (03 hexadecimal)	End of text
3	EOT (04 hexadecimal)	End of transmission
4	CR (0D hexadecimal)	Carriage return
5	DC3 (13 hexadecimal)	XOFF

Use an extended form of DECTTC to select a string of characters to indicate the end of a block. The extended form uses decimal codes to represent characters. You can use the extended form to send a control function at the end of a block transmission, instead of a single character. You can send a control sequence of up to six characters (Pn1 through Pn6) at the end of a block. The control sequence is: CSI ? Pn1 ; 0 Pn6

Pn1 through Pn6 are decimal codes for characters you can define as end-of-block. For example, the decimal code for ESC is 27 (1B hexadecimal). A code outside the range of 0 to 254 is ignored. The "GL Codes" table on page and "C1 and GR Codes" table on page contain decimal codes for characters.

For example, to send the default code of the [PF1] key on the computer's numeric keypad at the end of a block transmission, use this procedure.

To send the default code

1. The [PF1] key sends this default code: ESC O P
2. Translate each character in the sequence to decimal code.

   ASCII characters: `ESC O P`

   Decimal codes: `27 79 80`
3. Insert the decimal codes into the extended DECTTC sequence:

```
CSI ? 27 ; 79 ; 80 |
```

Telxon Private Use

Telxon has defined its set of private use 2 escape sequences. For more information about these sequences, see the *AIRVU ANSI – AE User's Guide* (Telxon Part Number: 24627-000) from Telxon Corporation. Each escape sequence is introduced by either PU2 in 8-bit mode (0x 92) or ESC,R in 7-bit mode.

The two bytes that follow the introducer are interpreted as an ASCII decimal value that describes the specific type of escape operation. Intermec supports the following escape sequence for Telxon private use only:

```
(PU2 0x92 is ESC 7 (1B 37 in 7 bit mode), ST  0x9c is ESC (1B and 5C) in
7 bit mode))
```

Telxon Private Use

PU2 0 0 0 0 ST	Sets screen size to 26x16 instead of 21x16 (Intermec's closest match)
PU2 0 0 0 1 ST	Sets screen size to 16x16
PU2 0 0 0 2 ST	Sets screen size to 26x8 instead of 21x8 (Intermec's closest match)
PU2 0 0 0 3 ST	Sets screen size to 16x8
PU2 0 0 0 4 ST	(16x4 screen size) Ignored, not supported by Intermec
PU2 0 0 0 5 ST	(21x4 screen size) Ignored, not supported by Intermec
PU2 0 5 0 0 ST	Disables scanner
PU2 0 5 0 2 ST	Enables scanner without Auto Enter
PU2 0 5 0 3 ST	Enables scanner with Auto Enter *
ESC R 1 3 ST	Data entry allowed from scanner only
ESC R 1 4 ST	Data entry allowed from keyboard and scanner only*
** Default mode. PU2 0 0 0 0 ST thru PU2 0 0 0 5 ST are for the 700 Colors, CK30, CK31, CK32, and CN3.*	

Using Intermec Printer Language

This example shows how to create a bar code, the human readable form of that bar code, and its RFID tag-write using the Intermec Printer Language (IPL). For a further definition of IPL, see the *Intermec Printer Language (IPL) Programmer's Reference Manual* (P/N 934-013-xxx).

```
/* Commands to setup PM4i Printer to write a barcode label, human readable info
and an RFID tag. */

/* Turn on RFID mode, with a tag adjust of 0 dots, retry 2 times */
APC "#P4N8100000=02=0fJ1,0,2=03#" ST

/* Place the printer in advanced mode */
APC "#P4N8100000=02=1bC0=03#" ST

/* Place the printer in program mode */
APC "#P4N8100000=02=1bP;=03#" ST

/* Erase format field 4 and then create format field 4.  Make a workspace area
for the printer to use to store the information we are going to print. */
APC "#P4N8100000=02=45=34;F4;=03#" ST

/* Make a human readable entry at o100,200 orgin 100,200. f0 means field
direction is horizontal, h1,w1 is use height *1 and width *1.  C21 means use 12
point monspace. The r0 says character rotation is horiontal.  b0 is saying no
border. Finally the d3 is the text you want to display. */
APC "#P4N8100000=02H0;o100,200;f0;h1,w1;c21;r0;b0;d3,TEST LABEL;=03#" ST

/* Edit and create line field 1.  o11,250 Orgin of 11,250. f0 field direction
is horizontal.  1500;w4 is length of 1500 dots and width is field is 4 dots.
APC "#P4N8100000=02L1;o11,250;f0;1500;w4;=03#" ST

/* B2 Defines field 2 as Barcode data. o20,300 with an origin of 20,300. f0
means field direction is horizontal. Height is 102 dots and width is 2 dots.
The c0,0 says to print the barcode as code 39 with no check digit. i1 (one)
determines that an interpretation of the barocde prints with a start and stop
character included. r1 means rotation of the barcode is vertical. d3,ABCD says
to print a barcode containing the data ABCD. */
APC "#P4N8100000=02=42=32;o20,300;f0;h102;w2;c0,0;i1;r1;d3,ABCD;=03#" ST

/* I2 Edit interpretive field for barcode field 2. 020,402 with an orgin of
20,402. f0 means field direction is horizontal. h1,w1 height*1, width*1. c20
means use 8 point monospace font.  r0 - character rotation is horizontal. */
APC "#P4N8100000=02I2;o20,402;f0;h1,w1;c20;r0;b0=03#" ST

/* Q3 Edit and create RFID field 3. a2 tells us to write ASCII data. 2,5,4
tells us to write to segment 2 the data area of the chip starting at address 5
of this segment for four bytes. d3,ABCD says to write ABCD to the tag.*/
APC "#P4N8100000=02Q3;a2,2,5,4;d3,ABCD;=03#" ST

/* Actually print the data we created above. */
APC "#P4N8100000=02R=03#" ST

/* Upload data from format table 4 created above and write data to the tag. */
APC "#P4N8100000=02=1b=45=34=18=03#" ST

/* Actually write the data to the tag. */
APC "#P4N8100000=02=17=03#" ST
```

7 Extended Commands

Extended commands govern abilities unique to computers. You can use extended commands to transmit or receive data over the computer's RS-232 port, send information to an RS-232 device (such as a printer or bar code printer), or collect data. The following chart lists the commands.

Extended Commands

Name	Character
Transmit and Receive On RS-232 Port	#F
Transmit Only On RS-232 Port	#P
Receive Only On RS-232 Port	#G
Return Version	#V
Scan	#S
Tone *(CK30, CK31)*	#T
Speech XML Parsing *(700 Color)*	#X
Magnetic Card Reader *(700 Color)*	#M
RFID – Set Attributes *(700 Color with PPC 2003, CV30, CV60)*	#A
RFID – Define Format *(700 Color with PPC 2003, CV30, CV60)*	#D
RFID – Write Tags *(700 Color with PPC 2003, CV30, CV60)*	#W
RFID – Set Filter *(700 Color with PPC 2003, CV30, CV60)*	#J

Note: For software version 8.25 or greater, the #A, #D, #W, and #J commands apply to all computers and are allowed when connecting to an RFID reader.

To use extended commands, you must enable the extended command option through the TE 2000 configuration menus. For information about enabling the command on a computer, see "Using theTerminal Emulation Menus" on page 97.

Transmit and Receive On RS-232 Port (#F)

Use the #F extended command to transmit and receive data on the computer's RS-232 port.

When constructing a Transmit and Receive command, the data to be sent should be placed at Line 2, Column 1. The data must be less than 1840 bytes in length (after "=yy" compression, where "yy" is the hexadecimal representation of the output byte). The last character must be a "#" (pound sign). The computer sends the data to the RS-232 port until it detects a "#" character.

3270 and 5250 Data Streams

Line 2, Column 1, begins the start of transmit data. The host application must set up the screen with a transmit field to locate the RS-232 port I/O data. The field can be anywhere on the screen, beginning at position 80. The host application must supply the receive field (for response from the RS-232 port to the host). The receive field can be anywhere on the screen except for the position allocated for the return status field.

The output stream appears in ASCII character format. For bytes that are not displayable ASCII characters, you may insert the "=yy" characters, where "yy" is the hexadecimal representation of the output byte. For example, insert "=0D" if your printer requires a carriage return, or insert "=0A" for a line feed. The table titled "Computer Font Set" on page 408 in Appendix B contains other hexadecimal values.

The following table describes the line and columns where characters must appear, and their meanings. Note that "ƀ" indicates a 1-byte space.

3270/5250 Data Streams

Column	Description	Character
2	Begin extended command	#
3	Transmit and Receive on RS-232 Port command	F
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ƀ=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8	CTS flow	0=Disable; 1=Enable
9	DSR flow	0=Disable; 1=Enable

3270/5250 Data Streams (continued)

Column	Description	Character
10	XON/XOFF flow control	0=Disable; 1=Enable
11-12	Flow control timeout value	XX=Number of seconds
13	Return AID key	~~b~~=AID key is Enter (default). X=AID key character. For return AID key characters, see the next section, “Return AID Key Characters”.
14-15	Maximum characters to receive	dd *or* Xd...dX, where: dd=00-99. *Default: 99.* d...d=Any number of decimal digits from 0-2000, inclusive. *Default: 99.*
16-17	Delimiter character	AA=Hexadecimal ASCII code that marks the end of data to be received. Range: 00-7E. *Default of 00 or ~~bb~~ implies no start character.*
18-19	Number of delimiter characters	XX=Number of characters accepted before sending return code to host. Range: 00-99. *Default of 00 or ~~bb~~ implies no start character.*
20-21	Start character	AA=Hexadecimal ASCII code. Range: 00-99. *Default of 00 or ~~bb~~ implies no start character.*
22	Return start character to host	F=Return character; ~~b~~ = Do not return character
23	Flag parity errors	P=Flag; ~~b~~ = Do not flag
24-25	Receive timeout length	XX=Number of seconds the computer waits for input from the RS-232 port before it sends a timeout error. *Uses a 5-second default when field contains spaces.*

Return AID Key Characters

The following table lists the extended command AID keys and characters returned to the host. For example, if you want an F1 AID key, use “A” as the character. Note that “~~b~~” indicates a 1-byte space.

Return AID Key Characters

AID Key	Character	AID Key	Character	AID Key	Character
Enter	b	F6	F	F18	R
Help *(5250)*	)	F7	G	F19	S
Roll down/Page up *(5250)*	*	F8	H	F20	T
Roll up/Page down *(5250)*	+	F9	I	F21	U
Print *(5250)*	'	F10	J	F22	V
Home *(5250)*	.	F11	K	F23 *(5250)*	W
Clear	7	F12	L	F24 *(5250)*	X
F1	A	F13	M	PA1 *(3270)*	Y
F2	B	F14	N	PA2 *(3270)*	Z
F3	C	F15	O	PA3 *(3270)*	,
F4	D	F16	P		
F5	E	F17	Q		

Return Codes for Transmit and Receive

Line 24, Column 1, has the return status field. The code returned in this position tells the host the extended command status.

Return Codes for Transmit and Receive

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed, number of data bits or stop bits, or flow control timeout.	4, 6, 7, 11-12
2	No delimiter (#) on data stream.	(None)
3-4	Reserved	(None)
5	Timeout using flow control.	(None)
6	Improper return field.	(None)
7	Hexadecimal value is outside the range of 00-0F.	15-16, 19-20
8	Error in AID code.	(None)
9	Incorrect setting for maximum characters to receive.	13-14
A	Incorrect setting for delimiter character.	15-16
B	Incorrect setting for number of delimiter characters.	17-18
C	Incorrect setting for flow control.	8, 9, 10
D	Incorrect setting for start character.	19-20
E	Incorrect setting for data parity.	5, 22
F	Timeout.	(None)
K	Returned if the RS-232 receive is aborted after a key is pressed.	(None)
O	Overrun of UART receive register; an error from the RS-232 device.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Example of Transmit and Receive

The following example and chart show what would be sent to a computer to transmit and receive data. *Note that "~~b~~" indicates a 1-byte space.*

Column 1 Column 11 Column 23

~~b~~#F4N8100000~~b~~500A0102~~bb~~10

This is data to send. #

Line 1

Line 2

Example of Transmit and Receive

Column	Character	Description
1	~~b~~	This column typically contains a space.
2	#	Begin extended command.
3	F	Transmit and Receive command.
4	4	9600 baud.

Example of Transmit and Receive

Column	Character	Description
5	N	No parity.
6	8	Eight data bits.
7	1	One stop bit.
8	0	CTS flow disabled
9	0	DSR flow disabled
10	0	XON/OFF flow control disabled.
11-12	00	No flow control timeout value.
13	ƀ	AID key is Enter.
14-15	50	Receive a maximum of 50 characters.
16-17	0A	Delimiter character is 0a hexadecimal (line feed).
18-19	01	Data is sent to host after 1 delimiter character is received.
20-21	02	Start character is 02 hexadecimal (STX).
22	ƀ	Start character is not returned.
23	ƀ	Do not flag parity errors.
24-25	10	Receive timeout length is 10 seconds.

VT/ANSI Data Stream

Due to possible system fragmentation of a data stream, the "#" in Line 1, Column 2, should be the last character placed on the screen. This assures that all data is present before the extended command is parsed (removed).

There are two ways to send the #F extended command. The conventional method involves writing the command to the screen on line one, writing the data to be printed on the second line, then writing a "#" character on the first line to activate the print. The other method is to use the control characters APC (Application Program Command) (0x9F) and ST (String Terminator) (0x9C) to encapsulate the command. Using the control characters method is faster since they are not written to the display.

Conventional Method

This is an example of Transmit and Receive using the conventional method.

Note: The descriptions for the "This is the Data to be sent" command are listed under the control character sequence method.

Conventional Method

Command	Action
`CSI 2J`	Clears the screen.
`CSI 1;3H`	Move the cursor to top row, column 3.
`F4N8100005500A0102ƀƀ00`	Start the print command.
`CSI 2;1HThis is the Data to be sent#`	Moves cursor to row 2, column 1 - sends the data to print
`CSI 1;2H#`	Move the cursor to top row, column 2 - start the print job.

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background.

Control character sequences for mnemonics APC and ST can also start and end data. The last data character must be a "#" (pound sign). The computer sends the data to the RS-232 port until it detects a "#" character and the string terminator sequence.

Below are the descriptions of each group of characters. Note that "ƀ" indicates a 1-byte space.

^9F#F4N8100005500A0102ƀƀ00This is the Data to be sent#^9C

Data to send

- ^9F APC character 0x9F
- #F Begin extended command for transmit and receive
- 4N81 9600 baud, no parity, 8 data bits, 1 stop bit
- 000 Flow control options *(disabled)*
- 05 Flow timeout *(5 seconds)*
- 50 Maximum receive characters
- 0A Delimiting character
- 01 Number of delimiting characters to wait for
- 02 Start character *(STX)*
- ƀ tart character return *(do not return start character)*
- ƀ Parity error flags *(do not flag parity errors)*
- 00 Receive timeout *(use 5-second default)*
- Data to send
- # Data termination character
- ^9C ST character 0x9C

The output stream appears in ASCII character format. For bytes that are not displayable ASCII characters, you may insert the “=yy” characters, where “yy” is the hexadecimal representation of the output byte.

The following table describes the line and columns where characters must appear, and their meanings. *Note that “ƀ” indicates a 1-byte space.*

Control Character Sequence Method

Column	Description	Character
2	Begin extended command	#
3	Transmit and Receive on RS-232 Port command	F
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ƀ=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8	CTS flow	0=Disable; 1=Enable
9	DSR flow	0=Disable; 1=Enable
10	XON/XOFF flow control	0=Disable; 1=Enable
11-12	Flow control timeout value	XX=Number of seconds
13-14	Maximum characters to receive	dd *or* Xd...dX, where: dd=00-99. *Default: 99.* d...d=Any number of decimal digits from 0-2000, inclusive. *Default: 99.*
15-16	Delimiter character	AA=Hexadecimal ASCII code that marks the end of data to receive. Range: 00-7E. *Default of 00 or ƀƀ implies no start character.*
17-18	Number of delimiter characters	XX=Number of characters accepted before sending return code to host. Range: 00-99. *Default of 00 or ƀƀ implies no start character.*
19-20	Start character	AA=Hexadecimal ASCII code. Range: 00-99. *Default of 00 or ƀƀ implies no start character.*
21	Return start character to host	F=Return character; ƀ = Do not return character
22	Flag parity errors	P=Flag; ƀ = Do not flag
23-24	Receive timeout length	XX=Number of seconds the computer waits for input from the RS-232 port before it sends a timeout error. *Uses a 5-second default when field contains spaces.*

Flow Control

The Transmit command supports the XON/XOFF flow control. This is the same XON/XOFF flow control most devices support. The timeout value tells the computer how long to wait for the flow control handshake before returning a one-byte error value.

Return Codes for Transmit and Receive

The return code is the status sent to the host computer. The computer returns data and the extended command's status to the host computer in the XCATA<CR> or XCCCATA<CR> format.

- X
  The return code listed in the following chart.
- CC or CCCC
  The character count of the data returned. CC is 00-99.
  CCCC is 100-2000.
- DATA
  The RS-232 data received from the RS-232 device attached to the computer (if any data was received). The computer simulates the [Enter] key to return the code to the host.

Return Codes for Transmit and Receive

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed, number of data bits or stop bits, or flow control timeout.	4, 6, 7, 11-12
2	No delimiter (#) on data stream.	(None)
3-4	Reserved	(None)
5	Timeout using flow control.	(None)
6	Improper return field.	(None)
7	Hexadecimal value is outside the range of 00-0F.	15-16, 19-20
8	Reserved	(None)
9	Incorrect setting for maximum characters to receive.	13-14
A	Incorrect setting for delimiter character.	15-16
B	Incorrect setting for number of delimiter characters.	17-18
C	Incorrect setting for flow control.	8, 9, 10
D	Incorrect setting for start character.	19-20
E	Incorrect setting for data parity.	5, 22
F	Timeout.	(None)
K	Returned if the RS-232 receive is aborted after a key is pressed.	(None)
O	Overrun of UART receive register; an error from the RS-232 device.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Transmit Only On RS-232 Port (#P)

The Transmit Only On RS-232 Port extended command lets the computer send information to a slaved RS-232 device, such as a receipt printer or bar code printer. The command uses the RS-232 communications port to send data to the device. The computer checks data from the host computer for a transmit sequence, then sends the requested data. The host computer signals the computer for a transmit command by inserting the characters "#P" into the display at Line 1, Columns 2 and 3. Characteristics of the transmission immediately follow the #P command.

When constructing a Transmit Only command, the data to be sent should be placed at Line 2, Column 1. The data must be less than 1840 bytes in length (after "=yy" compression). The last character must be a "#" (pound sign). The computer sends the data to the RS-232 port until it detects a "#" character.

3270 and 5250 Data Streams

Line 2, Column 1, begins the start of transmit data. The host application must set up the screen with a transmit field to locate the RS-232 port I/O data. The field can be anywhere on the screen, beginning at position 80.

The output stream takes the form of ASCII characters. For nondisplayable ASCII characters, you may insert the "=yy" characters, where "yy" is the hexadecimal representation of the output byte. For example, insert "=0D" if your printer requires a carriage return, or insert "=0A" for a line feed. The table titled "Computer Font Set" on page 408 in Appendix B contains other hexadecimal values.

The following table describes the line and columns where characters must appear, and their meanings. Note that the line and column information refers to a 3278 unit with a 24-line by 80-column display and that "ƀ" indicates a 1-byte space. The display buffer position refers to the same buffer but with a linear array ranging from 0-1919.

3270 and 5250 Data Streams

Column	Description	Character
2	Begin extended command	#
3	Transmit Only on RS-232 Port command	P
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ƀ=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8-9	Reserved	ƀƀ
10	XON/XOFF flow control	0=Disable; 1=Enable

3270 and 5250 Data Streams (continued)

Column	Description	Character
11-12	Flow control timeout value	XX=Number of seconds
13	Return AID key	~~b~~=AID key is Enter (default). X=AID key character. For return AID key characters, see “Return AID Key Characters” on page 345.

Return Codes for Transmit Only

The computer puts the return code response in an input field in the display buffer at Line 24, Column 1. It simulates the [Enter] key to return the value to the host. Use the Start Field order to return a code for this operation. The following chart lists return codes.

Return Codes for Transmit Only

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed, number of data bits or stop bits, or flow control timeout.	4, 6, 7, 11-12
2	No delimiter (#) on data stream.	(None)
3-4	Reserved	(None)
5	Timeout using flow control.	(None)
6	Improper return field.	(None)
8	Error in AID code.	(None)
C	Incorrect setting for flow control.	8, 9, 10
E	Incorrect setting for data parity.	5
O	Overrun of UART receive register.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Example of Transmit Only

The following example and chart show what would be sent to a computer to cause data to transmit. *Note that “~~b~~” indicates a 1-byte space.* Following the example is an explanation of Line 1.

Column 1 Column 11

Line 2

~~b~~#P3N81~~bb~~005A

This is data to send. #

Line 1

Example of Transmit Only

Column	Character	Description
1	~~b~~	This column typically contains a space.
2	#	Begin extended command.
3	P	Transmit Only command.
4	3	4800 baud.
5	N	No parity.
6	8	Eight data bits.
7	1	One stop bit.
8	0	(None)
9	0	(None)
10	0	XON/OFF flow control disabled.
11-12	05	Flow control timeout is 5 seconds.
13	A	AID key is [F1].

VT/ANSI Data Stream

Conventional Method

This is an example of Transmit Only using the conventional method.

Conventional Method

Command	Action
`CSI 2J`	Clears the screen.
`CSI 1;3HP5E72010008Data to send=0A=0D#`	Moves the cursor to top row, column 3
`CSI 1;2H #`	Moves the cursor to the beginning.

Due to possible system fragmentation of a data stream, the "#" in Line 1, Column 2, should be the last character placed on the screen. This assures that all data is present before the extended command is parsed (removed).

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background.

Control character sequences for mnemonics APC and ST can also start and end data. The last data character must be a "#" (pound sign). The computer sends the data to the RS-232 port until it detects a "#" character and the string terminator sequence.

Below are the descriptions for each group of characters.

^9F#P5E7201008Data to send=0A=0D#^9C

- ^9F PC character 0x9F

- #P Begin extended command for transmit only
- 5E72 19200 baud, even parity, 7 data bits, 2 stop bits
- 010 Enable DTR Flow control
- 08 Wait up to 8 seconds for return of DSR
- Data to send Sending string with CR, LF
- # Begin extended command terminating character
- ^9C ST character 0x9C

The output stream takes the form of ASCII characters. For nondisplayable ASCII characters, you may insert the “=yy” characters, where “yy” is the hexadecimal representation of the output byte.

The following table describes the line and columns where characters must appear, and their meanings. *Note that “~~b~~” indicates a 1-byte space.*

Control Character Sequence Method

Column	Description	Character
2	Begin extended command	#
3	Transmit Only On RS-232 Port command	P
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ~~b~~=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8-9	Reserved	~~bb~~
10	XON/XOFF flow control	0=Disable; 1=Enable
11-12	Flow control timeout value	XX=Number of seconds

Flow Control

Transmit Only supports XON/XOFF.

Return Codes for Transmit Only

The Transmit Only return code is the status sent to the host computer. The computer returns the extended command's status to the host computer in the \\X<CR> format. X is the return code listed in the following chart. The computer simulates the [Enter] key to return the code to the host.

Return Codes for Transmit Only

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed, number of data bits, number of stop bits, or flow control timeout.	4, 6, 7, 11-12
2	No delimiter (#) on data stream.	(None)
3-4	Reserved	(None)
5	Timeout using flow control.	(None)
6	Improper return field.	(None)
C	Incorrect setting for flow control.	8, 9, 10
E	Incorrect setting for data parity.	5
O	Overrun of UART receive register.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Receive Only On RS-232 Port (#G)

The Receive Only On RS-232 Port extended command provides a way to use the RS-232 port on the computer to collect data. A scale is one example of a use for this command. The host computer sends “#G” characters to alert the computer for activity on the port.

3270 and 5250 Data Streams

Line 2, Column 1, is the start of the area on the screen where an input field may be placed for receive data. The host application must supply the input field for the data received on the RS-232 port. The host can place the input field anywhere on the screen except the position for the return status field.

The following table describes the line and columns where characters must appear, and their meanings. *Note that “ƀ” indicates a 1-byte space.*

3270 and 5250 Data Streams

Column	Description	Character
2	Begin extended command	#
3	Receive Only on RS-232 Port command	G
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ƀ=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8	CTS flow	0=Disable; 1=Enable
9	DSR flow	0=Disable; 1=Enable

3270 and 5250 Data Streams (continued)

Column	Description	Character
10	XON/XOFF flow control	0=Disable; 1=Enable
11-12	Flow control timeout value	XX=Number of seconds
13	Return AID key	~~b~~=AID key is Enter (default). X=AID key character. For return AID key characters, see "Return AID Key Characters" on page 345.
14-15	Maximum characters to receive	~~bb~~=Default dd *or* Xd...dX, where: dd=00-99. *Default: 99.* d...d=Any number of decimal digits from 0-2000, includsive. *Default: 99.*
16-17	Delimiter character	AA=Hexadecimal ASCII code that marks the end of data to receive. Range: 00-7E. *Default of 00 or ~~bb~~ implies no start character.*
18-19	Number of delimiter characters	XX=Number of characters accepted before sending return code to host. Range: 00-99. *Default of 00 or ~~bb~~ implies no start character.*
20-21	Start character	AA=Hexadecimal ASCII code. Range: 00-99. *Default of 00 or ~~bb~~ implies no start character.*
22	Return start character to host	F=Return character; ~~b~~=Do not return character
23	Flag parity errors	P=Flag; ~~b~~=Do not flag
24-25	Receive timeout length	XX=Number of seconds the computer waits for input from the RS-232 port before it sends a timeout error. *Uses 5-second default when this field contains spaces.*

Return Codes for Receive Only

Line 24, Column 1, is the return status byte. The host application defines the 1-byte return status field location. This chart lists return status codes.

Return Codes for Receive Only

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed, number of data bits or stop bits, or flow control timeout.	4, 6, 7
2	No delimiter (#) on data stream.	(None)
3-5	Reserved	(None)
6	Improper return field.	(None)
7	Hexadecimal value is outside the range of 00-0F.	15-16, 19-20
8	Error in AID code.	(None)
9	Incorrect setting for maximum characters to receive.	13-14
A	Incorrect setting for delimiter character.	15-16
B	Incorrect setting for number of delimiter characters.	17-18
C	Incorrect setting for flow control.	8, 9, 10
D	Incorrect setting for start character.	19-20
E	Incorrect setting for data parity.	5, 22
F	Timeout.	(None)

Return Codes for Receive Only (continued)

Code	Description	Column
K	Returned if the RS-232 receive is aborted after a key is pressed.	(None)
O	Overrun of UART receive register.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Example of Receive Only

The following example and chart show what would be sent to the computer to receive data. *Note that "ƀ" indicates a 1-byte space.*

Column 1 Column 11 Column 23

```
ƀ#G3N8100000ƀ500A0102FP10
```

Example of Receive Only

Column	Character	Description
1	ƀ	This column typically contains a space.
2	#	Begin extended command.
3	G	Receive Only command.
4	3	4800 baud.
5	N	No parity.
6	8	Eight data bits.
7	1	One stop bit.
8	0	CTS flow disabled.
9	0	DSR flow disabled.
10	0	XON/OFF flow control disabled.
11-12	00	No flow control timeout value.
13	ƀ	AID key is Enter.
14-15	50	Receive a maximum of 50 characters.
16-17	0A	Delimiter character is 0A hexadecimal (line feed).
18-19	01	Data sent to host after one delimiter character is received.
20-21	02	Start character is 02 hexadecimal (STX).
22	F	Start character is returned to host.
23	P	Parity errors are flagged.
24-25	10	Receive timeout length is 10 seconds.

VT/ANSI Data Stream

Conventional Method

This is an example of Receive Only using the conventional method. *Note that the descriptions for the middle command are listed under the control character sequence method and that “b̸” indicates a 1-byte space.*

Conventional Method

Command	Action
CSI 2J	Clears the screen.
CSI 1;3HG3E81~~bbbbb~~X1000X03010202FP06	Moves the cursor to the top row, column 3
CSI 1,2H #	Moves the cursor to the beginning.

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background. *Note that “b̸” indicates a 1-byte space.*

^9F#G3E8100005X1000X03010202FP06^9C

- ^9F APC character 0x9F
- #G Begin extended command for receive only
- 3E81 9600 baud, even parity, 8 data bits, 1 stop bit
- 000 Flow control options *(disabled)*
- 05 Flow timeout *(5 seconds)*
- X1000X Receive 1000 bytes
- 03 Delimiting character *(ETX)*
- 01 One delimiting character
- 02 Start character *(STX)*
- 02 Number of start characters *(2)*
- F Return start characters
- P Flag parity errors
- 06 Receive all data within 6 seconds
- ^9C ST character 0x9C

The following table describes the line and columns where characters must appear, and their meanings. *Note that “b̸” indicates a 1-byte space.*

Control Character Sequence Method

Column	Description	Character
2	Begin extended command	#
3	Receive Only On RS-232 Port command	G
4	Speed (bits per second)	1=1200; 2=2400; 3=4800; 4=9600; 5=19200; 6=38400
5	Data parity	N=None; 0 (zero)=None; ~~b~~=None; E=Even; O=Odd
6	Data bits	7=Seven; 8=Eight
7	Stop bits	1=One; 2=Two
8	CTS flow	0=Disable; 1=Enable
9	DSR flow	0=Disable; 1=Enable
10	XON/XOFF flow control	0=Disable; 1=Enable
11-12	Flow control timeout value	XX=Number of seconds
13-14	Maximum characters to receive	dd *or* Xd...dX, where: dd=00-99. *Default: 99.* d...d=Any number of decimal digits from 0-2000, inclusive. *Default: 99.*
15-16	Delimiter character	AA=Hexadecimal ASCII code that marks end of valid data to be received. Range: 00-7E. *Default: 00.*
17-18	Number of delimiter characters	XX=Delimiter characters received before transmitting return code to host. Range: 00-99. *Default: 00.*
19-20	Start character	AA=Hexadecimal ASCII code for start character. Range: 00-99. *Default of 00 or ~~bb~~ implies no start character.*
21	Return start character to host	F=Return character; ~~b~~=Do not return character
22	Flag parity errors	P=Flag; ~~b~~=Do not flag
23-24	Receive timeout length	XX=Number of seconds the computer waits for input from the RS-232 port before it sends a timeout error. *Uses 5-second default when this field contains spaces.*

Return Codes for Receive Only

The Receive Only return code is the status sent to the host computer. The computer returns data and the extended command's status to the host computer in the XCATA<CR> or XCCCATA<CR> format.

- X
  The return code listed in the following chart.
- CC or CCCC
  The character count of the received data. CC is 00-99.
  CCCC is 100-2000.
- DATA
  The RS-232 data received from the RS-232 device attached to the computer (if any data was received). The computer simulates the [Enter] key to return the code to the host.

Return Codes for Receive Only

Code	Description	Column
0	Good status, transaction complete.	(None)
1	Not enough memory. Or, incorrect setting for speed or number of data bits or of stop bits.	4, 6, 7
2	No delimiter (#) on data stream.	(None)
6	Improper return field.	(None)
7	Hexadecimal value is outside the range of 00-0F.	15-16, 19-20
8	Reserved.	(None)
9	Incorrect setting for maximum characters to receive.	13-14
A	Incorrect setting for delimiter character.	15-16
B	Incorrect setting for number of delimiter characters.	17-18
C	Incorrect setting for flow control.	8, 9, 10
D	Incorrect setting for start character.	19-20
E	Incorrect setting for data parity.	5, 22
F	Timeout.	(None)
K	Returned if the RS-232 receive is aborted after a key is pressed.	(None)
O	Overrun of UART receive register.	(None)
P	Data parity or framing error.	(None)
Z	Command not supported for this device	(None)

Return Version (#V)

The Return Version extended command returns the current terminal emulation name and version to the host computer. The following table lists columns in which characters must appear.

Return Version Characters

Column	Description	Character
2	Begin extended command	#
3	Return Version command	V

3270 Data Stream

Line 2, Column 1, is the input field large enough to hold the program name and version number followed by the # sign, as shown below, where “s” is the status.

‘0000/A0080 “<program name><version>#”/A1841s

Line 24, Column 1, has the return status field. The code returned in this position tells the host the status of the extended command. The code is 0, which means good status or transaction complete.

5250 Data Stream

The computer returns data and the extended command's status to the host computer in the following format:

//rrccfl <program name><version>#s

- **rr** is the row.
- **cc** is the column.
- **fl** is the AID key return value.
- **s** is the status.

VT/ANSI Data Stream

The return code is the status sent to the host computer and indicates if the extended command was successful. The computer returns data and the extended command's status to the host computer in the following format. For example: `\\0\FWP240H0 V6.52 <CR>`

\\0\<Program name> <Version> <CR>

- **0**
  Return code, which indicates "good status, transaction complete."
- **<Program name>**
  TE program name.
- **<Version>**
  Program version.

Conventional Method

Note: The descriptions for the second command are listed under the control character sequence method.

Conventional Method

Command	Action
`CSI 2J`	Clears the screen.
`CSI 1;2H #V#`	Moves the cursor to the top row, column 2

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background.

^9F#V#^9C

- ^9F — APC character 0x9F
- #V# — Begin extended command to get program name/version
- ^9C — ST character 0x9C

Scan (#S)

The Scan extended command allows host systems to have the same capabilities as computers using the Native data stream to set bar code parameters and scan.

By inserting characters "#S" in the display buffer at Line 1, Columns 2 and 3, the host computer tells the computer to expect a Bar Code Scanner extended command. Appropriate descriptive characteristics should follow the "#S" characters. The computer returns a code to indicate if the command was successful. Then it simulates the [Enter] key to return a value to the host computer. *Note that "~~b~~" indicates a 1-byte space.*

Conventional Method

Note that the descriptions for the middle command are listed under the control character sequence method.

Conventional Method

Command	**Action**
`CSI 2J`	Clears the screen.
`CSI 1;3H S07F3201221131001208010806`~~bbbbb~~`#`	Moves the cursor to the top line, Column 3
`CSI 1;2H #`	Moves the cursor to the beginning.

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background.

^9F#S08F32012211310012080108060~~bbbb~~#^9C

- ^9F — APC character 0x9F
- #S — Begin extended command for setting scan parameters
- 0 — Laser with no redundancy
- 7 — Auto enter scan
- F — Enable stream scanning, return bar code type and concatenate bar code
- 32 — Set maximum length for all bar codes to 32 characters
- 01 — Set minimum length for all bar codes to 1 character
- 2 — Decode UPC-E system 0 and UPC-A bar codes
- 2 — Enable EAN bar code symbology
- 1 — Enable Encoded 39 bar code symbology
- 1 — Enable Plessey bar code symbology

- 3 Keep MOD 10 first check digit for Plessey symbology
- 1 Enable Codabar bar code symbology
- 0 Disable Code 11 bar code symbology
- 0 Disable Code 93 bar code symbology
- 1 Enable Code 128 bar code symbology
- 2 Enable Straight 2 of 5 bar code symbology
- 08 Set maximum length to 8 characters
- 01 Set minimum length to 1 character
- 08 Set first fixed length to 8 characters
- 06 Set second fixed length to 6 characters
- 0 Disable Interleave 2 of 5 bar code symbology
- ~~bbbb~~# Keep current value
- ^9C ST character 0x9C

Due to possible system fragmentation of a data stream, the "#" in Line 1, Column 2, should be the last character placed on the screen. This will ensure that all data is present before the extended command is parsed (removed). The following table lists the command. Other tables list options.

Control Character Sequence Method

Column	Description	Character
2	Begin extended command	#
3	Scan command	S

CC Byte 1

Control Character Byte 1 (CC Byte 1) implementation differs among the computers. The tables below and on the next page show the CC Byte 1 options for scanning.

Options Supported by Characters – Line 1, Column 4

CC Byte 1	Laser	Disable Scanner	No Change From Current Setting
0	X		
1	X		
2			
3	X		

Options Supported by Characters – Line 1, Column 4 (continued)

CC Byte 1	Laser	Disable Scanner	No Change From Current Setting
4			
5	X		
6			
7	X		
8			
9	X		
A			
B	X		
C			
D	X		
E			
F		X	
ƀ *1-byte space*			X

CC Byte 2

CC Byte 2 implementation differs among the types of computers. The following tables show CC Byte 2 options.

Options Supported by Characters

CC Byte 2	Scan Termination Character
0-3	None
4-7	Auto Enter Scan *(not for 5250)*
8-F	Auto Tab Scan *(not for 5250)*
ƀ	*1-byte space.* No change from current setting

Options Supported by Characters – Line 1, Column 5

CC Byte 2	Scan Termination Character	Modulo 10 Check Digit
0	None	
1		X
2		
3		X
4	[Enter] key	
5		X
6		
7		X
8	[Tab] key	
9		X

Options Supported by Characters – Line 1, Column 5 (continued)

CC Byte 2	Scan Termination Character	Modulo 10 Check Digit
A		
B		X
C		
D		X
E		
F		X

CC Byte 3

Options Supported by Characters – Line 1, Column 6

CC Byte 3	Enable Stream Scanning (not for 5250)	Reserved	Return Bar Code Type	Bar Code Concatenated
0				
1				X
2			X	
3			X	X
4		X		
5		X		X
6		X	X	
7		X	X	X
8	X			
9	X			X
A	X		X	
B	X		X	X
C	X	X		
D	X	X		X
E	X	X	X	
F	X	X	X	X
ƀ	*1-byte space* No change from current setting.			

Bar Code Length

Bar Code Length sets the minimum and maximum character lengths for all types of bar codes scanned. Setting the minimum and maximum values to their optimum can increase scanning performance. If the computer scans bar codes that are outside the minimum and maximum value, the computer ignores the bar code.

Bar Code Length

Column	Character and Description
7-8	XX=Bar code length. Maximum length: 99. ~~bb~~=*1-byte spaces* No change from current setting
9-10	XX=Bar code length. Minimum length: 00. ~~bb~~=*1-byte spaces* No change from current setting

UPC

Use the UPC command to select combinations of characters in this table.

UPC Command Characters

Character	UPC-E # System 1	Expand UPC-E to UPC-A	UPC-E # System 0	Add-ons	UPC-A
0	Disables all.				
1				X	X
2			X		X
3			X	X	X
4		X			X
5		X		X	X
6		X	X		X
7		X	X	X	X
8	X				X
9	X			X	X
10	X		X		X
11	X		X	X	X
12	X	X			
13	X	X		X	X
14	X	X	X		X
15	X	X	X	X	X
~~b~~	*1-byte space* No change from current setting.				

EAN Algorithms

Use EAN Algorithms to select combinations of EAN options.

EAN Algorithms – Line 1, Column 12

Character	Description
0	EAN disabled
1	EAN with Add-ons enabled
2	EAN enabled
3	EAN and EAN with Add-ons enabled
~~b~~	*1-byte space* No change from current setting

Code 39

Code 39 sets the scanner to read simple Code 39 bar codes that do not include extended or encoded sequences.

Extended Code 39 is a superset of Code 39 and scans all regular Code 39 bar codes. You cannot select both Code 39 and Extended Code 39. Encoded Code 39 combines key presses with normal bar code data. Appendix A, “Bar Code Scanning,” contains Encoded Code 39 sequences.

Code 39 Algorithms Characters – Line 1, Column 13

Character	Description
0	Code 39 disabled
1	Encoded Code 39 enabled
2	Extended Code 39 enabled
3	Code 39 enabled
~~b~~	*1-byte space* No change from current setting

Plessey

If the Plessey bar code scanning algorithm is enabled, set its check digits according to your requirements. Refer to the manufacturer's bar code specifications for more information on check digits.

Plessey Characters – Line 1, Column 14

Character	Description
0	Plessey disabled
1	Plessey enabled
~~b~~	*1-byte space* No change from current setting

The Xs in this table indicate the Plessey check digits characters support.

Plessey Check Digit Characters– Line 1, Column 15

Character	Do Not Validate 1st Check Digit	Keep 1st Check Digit	Mod 10 1st Check Digit
1			X
2		X	
3		X	X
4	X		
5	X		X
6	X	X	
7	X	X	X
8		X	
9		X	X

Plessey Check Digit Characters– Line 1, Column 15

Character	Do Not Validate 1st Check Digit	Keep 1st Check Digit	Mod 10 1st Check Digit
A		X	
B		X	X
C	X	X	
D	X	X	X
E	X	X	
F	X	X	X
b	*1-byte space* No change from current setting.		

Codabar

The Codabar options (Codabar and ABC Codabar) are mutually exclusive coding algorithms and cannot be selected at the same time.

Codabar Characters – Line 1, Column 16

Character	Description
0	Codabar disabled
1	Codabar enabled
ƀ	*1-byte space* No change from current setting

Code 11

Code 11 Characters – Line 1, Column 17

Character	Description
0	Code 11 disabled
1	Code 11 enabled
ƀ	*1-byte space* No change from current setting

Code 93

Code 93 options can be enabled.

Code 93 Characters – Line 1, Column 18

Character	Description
0	Code 93 disabled
1	Code 93 enabled
ƀ	*1-byte space* No change from current setting

Code 128

Code 128 options can be enabled.

Code 128 Characters – Line 1, Column 19

Character	Description
0	Code 128 disabled
1	Code 128 enabled
~~b~~	*1-byte space* No change from current setting

Straight or Computer Identics 2 of 5

If the Straight or Computer Identics bar code is enabled, select the maximum and minimum lengths and the first and second fixed bar code lengths according to your requirements. *Note that a "~~b~~" indicates a 1-byte space. Refer to the manufacturer's bar code specifications for more information.*

Straight or Computer Identics 2 of 5

Character	Description
20	0=Straight and Computer Identics 2 of 5 disabled; 1=Computer Identics 2 of 5 enabled; 2=Straight 2 of 5 enabled; ~~b~~=No change from current setting
21-22	XX=Maximum length Straight or Computer Identics 2 of 5; ~~bb~~=No change from current setting
23-24	XX=Minimum length Straight or Computer Identics 2 of 5; ~~bb~~=No change from current setting
25-26	XX=1st fixed bar code length for Straight or Computer Identics 2 of 5; ~~bb~~=No change from current setting
27-28	XX=2nd fixed bar code length for Straight or Computer Identics 2 of 5; ~~bb~~=No change from current setting

Interleaved 2 of 5

If Interleaved 2 of 5 is enabled, select the maximum and minimum lengths and the first and second fixed bar code lengths according to requirements. *Note that a "~~b~~" indicates a 1-byte space. Refer to the manufacturer's bar code specifications for more information.*

Interleaved 2 of 5

Character	Description
29	0=Interleaved 2 of 5 disabled; 1=Interleaved 2 of 5 enabled; ~~b~~=No change from current setting
30-31	XX=maximum length Interleaved 2 of 5; ~~bb~~=No change from current setting
32-33	XX=minimum length Interleaved 2 of 5; ~~bb~~=No change from current setting
34-35	XX=1st fixed length for Interleaved 2 of 5 Bar Code; ~~bb~~=No change from current setting
36-37	XX=2nd fixed length for Interleaved 2 of 5 Bar Code; ~~bb~~=No change from current setting

Return Codes for Scan Bar Code

The Scan Bar Code Parameters return code is the status sent to the host computer; the code indicates if the extended command was successful. The computer returns data and the extended command's status to the host computer in the **X<CR>** format where **X** is the return code listed in the following chart. The computer simulates the [Enter] key to return a value to the host computer. If a hexadecimal number is entered wrong, the return code defaults to “0.”

Return Codes for Scan Bar Code

Code	Description
0	Good status, transaction complete
1	Bad status, transaction incomplete
Z	Command not supported for this device

Tone (#T)

Note: The #T extended command is only supported on CK30s and CK31s. All other computers ignore this command and subsequently play sounds from their default.wav files.

The Tone extended command causes the computer to make a tone of a specified volume, frequency, and length ().

3270 and 5250 Data Streams

The following table describes the line and columns where characters must appear, and their meanings. *Note that a “~~b~~” indicates a 1-byte space.*

3270 and 5250 Data Streams

Column	Description	Character
2	Begin extended command	#
3	Tone command	T
4-6	Volume	~~bbb~~=No change from current setting; 000-255=Range from quiet to loud.
7-9	Frequency	~~bbb~~=No change from current setting; 000-030=Range from low to high.
10-12	Length	~~bbb~~=No change from current setting; 001-010=Duration in 500 ms from short to long *(keyboard response time)*.

VT/ANSI Data Stream

Conventional Method

Note that the descriptions for the middle command are listed under the control character sequence method.

Conventional Method

Command	Action
CSI 2J	Clears the screen.
CSI 1;3HT25015008	Moves the cursor to the top line, column 3
CSI 1,2H #	Moves the cursor to the beginning.

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the print command executes in the background.

^9F#T125015008^9C

- ^9F APC character 0x9F
- #H Begin extended command for the tone command
- 125 Sets beeper volume to 125
- 015 Sets beeper frequency to 15
- 008 Set beeper length to 8 seconds
- ^9C ST character 0x9C

The following table describes the line and columns where characters must appear, and their meanings. *Note that a “~~b~~” indicates a 1-byte space.*

Control Character Sequence Method

Column	Description	Character
2	Begin extended command	#
3	Tone command	T
4-6	Volume	~~bbb~~=No change from current setting; 000-255=Range from quiet to loud
7-9	Frequency	~~bbb~~=No change from current setting; 000-030=Range from low to high
10-12	Length	~~bbb~~=No change from current setting; 001-010=Duration in 500 ms from short to long *(keypad response time)*

Return Codes for Tone

The return code is the status sent to the host computer; the code indicates if the extended command was successful. The computer returns data and the extended command's status to the host computer in the **\\X<CR>** format. X is the return code listed in the following chart.

Return Codes for Tone

Code	Description	Column
0	Good status, transaction complete.	(None)
8	Incorrect setting for volume.	4-6
9	Incorrect setting for frequency.	7-9
A	Incorrect setting for length.	10-12
Z	Command not supported for this device	(None)

Speech XML Parsing (#X) *(700 Color)*

TE 2000 with Speech integrates the SyVox Speech client with TE 2000 for text to speech (TTS) and automatic speech recognition (ASR) functionality. The SyVox Speech client uses speaker-independent technology which means that there is no need to train the speech engine to recognize your voice before using it. There is also no need for an intermediate server to manage voice templates or control speech logic.

You can add speech commands to your existing business logic by using the new speech extended commands. These commands indicate prompts that should be spoken to the user by way of text to speech. The commands also determine the set of recognized words that the user may speak for an acceptable response, such as through automatic speech recognition.

Speech XML Parsing Characters

Line	Column	Description	Character
1	2	Begin extended command	#
1	3	Speech command	X
2	1	Prompt tags	<prompt>Text to Say</prompt>
following prompt tags		Grammar tags	<grammar>Standard Grammar to Listen for</grammar> For more information, see “Grammars” on page 375.
following grammar tags		End extended command	#

Note: Meta tags for prompt, /prompt, grammar, /grammar, and the value for the grammar are case-sensitive and must be entered exactly.

Return Codes for Speech

The return code is the status sent to the host computer; the code indicates if the extended command was successful. The computer returns data and the extended command's status to the host computer in the \\X<CR> format. X is the return code listed in the following chart.

Return Codes for Speech

Code	Description
0	Good status, transaction complete.
2	No delimiter # on data stream.
3	No prompt or ending prompt.
4	No grammar or ending grammar.
5	No license
6	The text-to-speech generator has failed.
Z	Command not supported for this device.

Example for 3270 and 5250 Data Streams

The following table describes the line and columns where characters must appear, and their meanings. *Note that you must define the <prompt> data on line 2 or greater, then define the grammar to follow the <prompt>.*

Line 2, column 1 begins the XML commands to select the prompt to say and grammar to listen for from the speech engine. You must terminate the XML commands with a # character.

Example for 3270 and 5250 Data Streams

Line	Column	Description	Character
1	2	Begin extended command	#
	3	Speech command	X
2	1-55	<prompt>Go to location C 1 2 and say check digit</prompt>	
3	1-32	<grammar>Utility2Digit</grammar>	
	33	Terminator for speech extended command	#

Examples for VT/ANSI Data Streams

Conventional Method

Command	Action
CSI 2J	Clears the screen.
CSI 1;3H	Moves the cursor to row 1, column 3
X	Speech extended command
CSI 2;1H	Move cursor to row 2, column 1
<prompt>Go to location C 1 2 and say check digit</prompt>	Data for the speech engine to say

Conventional Method (continued)

Command	Action
CSI 3;1H	Move cursor to row 3, column 1
<grammar>Utility2Digit</grammar>	Speech engine to wait for two numbers
CSI 4;1H	Move cursor to row 4, column 1
#	Ending terminator for command
CSI 1,2H	Move cursor to row 1, column 2
#	Activate speech engine command parsing

Control Character Sequence Method

This is the same example using the control character sequence method. You do not need to clear the screen or move the cursor in the example because the speech command executes in the background.

^9F#X
<prompt>Go to location c 1 2 and say check digit</prompt>
<grammar>Utility2Digit</grammar>#^9C

- ^9F APC character 0x9F
- #X Begin extended command for the speech command
- <prompt>Go to location c 1 2 and say check digit</prompt>
  Speech engine says "Go to location C 1 2 and say check digit"
- <grammar>Utility2Digit</grammar>
  Speech engine "listens" or waits for two numbers
- # End extended command for the speech command
- ^9C ST character 0x9C

Phonetic Alphabet Keywords

The following alpha characters must be spoken using the corresponding phonetic word. Note that the SyVox Client returns only the single alpha character to the host.

Phonetic Alphabet Keywords

Say This	To Get This	Say This	To Get This	Say This	To Get This
Alpha	a	Juliet	j	Sierra	s
Bravo	b	Kilo	k	Tango	t
Charlie	c	Lima	l	Uniform	u
Delta	d	Mike	m	Victor	v
Echo	e	November	n	Whiskey	w
Foxtrot	f	Oscar	o	X-Ray	x
Golf	g	Papa	p	Yankee	y
Hotel	h	Quebec	q	Zed	z
India	i	Romeo	r		

Grammars

Grammar files contain a word or words which the speech engine is configured to recognize. Grammars may be activated at each prompt as needed. The highest recognition rates are achieved when the set of possible responses is limited. Although a large number of grammar files are available, the TE 2000 application is not designed to use the full vocabulary, such as all grammar lists, at an individual prompt.

The following grammars are included. Custom grammars are available. Contact your Intermec representative to learn how you can purchase the custom grammars.

Global Grammars

Global grammars are available at all prompts inside the speech extended command. There is no need to specify a global grammar.

- Change volume louder or softer
- Please repeat
- Change speed faster or slower
- Give me the time
- Give me the battery charge

Standard Grammars

Standard grammars indicate the range of acceptable verbal responses to a prompt. Function keys are commonly used to navigate within TE 2000 applications. The function and field grammars allow you to use speech commands in place of pressing function keys. This provides hands-free navigation of your application using familiar user commands.

Standard Grammars

Grammar	Values to Say
Function2Digit	Function 1 through Function 24 *see note below
Field	Field Plus; Field Minus; Field Exit
Utility1Digit	1 numeric phonetic character
Utility2Digit	2 numeric phonetic characters
Utility3Digit	3 numeric phonetic characters
Utility4Digit	4 numeric phonetic characters
Utility5Digit	5 numeric phonetic characters
Utility6Digit	6 numeric phonetic characters
UtilityYesNo	yes, no
UtilityAlphaNum1	1 alpha and 1 numeric phonetic character
UtilityAlphaNum2	1 alpha and 2 numeric phonetic characters
UtilityAlphaNum3	1 alpha and 3 numeric phonetic characters
Utility1AlphaNum	1 alpha or 1 numeric phonetic character

Standard Grammars (continued)

Grammar	Values to Say
Utility2AlphaNum	mix of 2 alpha or numeric phonetic characters
Utility3AlphaNum	mix of 3 alpha or numeric phonetic characters
Utility4AlphaNum	mix of 4 alpha or numeric phonetic characters
Utility5AlphaNum	mix of 5 alpha or numeric phonetic characters
Utility6AlphaNum	mix of 6 alpha or numeric phonetic characters

Note: When entering a Function2Digit grammar, you need to specify two digits to get the desired function key value, rather than its value. For example, for Function 1, you say "Function zero one," through to Function 24, where you say "Function two four." Contact your Intermec representative for more information.

Magnetic Card Reader (#M) *(700 Color)*

On the following Intermec printers, you can access the Magnetic Card Reader from the host by sending an extended #M command to the TE 2000 application.

- 782T CR
- PW40 with CR
- 6808 CR

The Magnetic Card Reader extended command is supported for 3270, 5250, and VT/ANSI terminal emulations but not the Native terminal emulation. 3270 and 5250 terminal emulations require an input field to hold the card data. Data beyond the size of the field is discarded.

When the 700 Color with the TE 2000 application receives the Magnetic Card Reader (#M) command, the display shows:

```
Please SWIPE Card
      NOW!
 (Press any key)
   to cancel.)
```

The operator should then swipe the card through the reader to cancel the process. The card reader data buffer is cleared just after displaying the waiting message, thus do not swipe the card until prompted by the display.

Pressing a key instead of swiping a card cancels the process *(see return code "K" on the next page).*

Extended Command Syntax

The Magnetic Card Reader extended command syntax is in the **#Mttddd** format.

Extended Command Syntax

Column	Description	Character
2	Begin extended command	#
3	Magnetic Card Reader command	M
4-5	Timeout	01-99=2-digit data receive timeout in seconds. *Default is 30 seconds when "tt" has two blank characters.*
6-8	Maximum characters to accept	001-250=Limit of maximum characters to accept. *Default is 250 when "ddd" has three blank characters.* Data characters received past the maximum are discarded. For 3270 and 5250 emulation, data characters past the input field size are discarded.

Return Codes for Magnetic Card Reader

The return code is the status sent to the host computer; the code indicates if the extended command was successful. The computer returns data and the extended command's status to the host computer in the **`\\X<CR>`** format. **X** is the return code listed in the following chart.

Return Codes for Magnetic Card Reader

Code	Description
0	Good status, transaction complete.
1	Not enough memory for card data buffer.
6	No 3270 or 5250 input field available to hold the magnetic card data.
9	Incorrect setting for maximum characters to receive.
C	Incorrect setting for timeout value.
F	Timeout error, no data.
K	Key pressed to abort card read.
P	Communication with the Magnetic Card Reader failed, check cabling, reader power, no data.
T	OK, the data is returned, but is truncated by the lesser of the input field size (for 3270 and 5250) or "ddd."
Z	Command not supported for this device

#M returns the data in the following format:

```
"{T1:" <Track 1> "|T2:" <Track 2> "|T3:" <Track 3> "}"
```

where <*Track n*> is:

- 0 to 81 alphanumeric data characters for Track 1.
- 0 to 42 numeric data digits for Track 2.

- 0 to 109 alphanumeric or numeric data characters *(depending on ISO or AAMVA conformance)* for Track 3.

Note: If the track data is "B," then the track was not read. An "N" means the track is not on the magnetic card.

The maximum possible characters is 245.

Magnetic Card Reader CAB File

For the Magnetic Card Reader to work, the MAGCARD.CAB file must be installed (which installs and registers the MAGCARD.DLL file). Contact your Intermec Representative to get this file, then install this file in your 700 Color. The TE 2000 application will only look for the MAGCARD.DLL file when it gets a Magnetic Card Reader (#M) extended command from the host.

RFID (#A) Set Attributes

This section applies to the 700 Color series with PPC 2003 (CV30, CV60).

Note: In software version 8.25 or greater, this command applies to all computers.

The #A extended command modifies the interface between the computer and the RFID reader. The reader does not need to be attached to the computer to complete these changes. The data collection engine caches the changes and applies the changes when a new connection to the RFID reader is opened.

On the 700 Color

A TE 2000 "Reader State" button is available to add to the TE 2000 toolbar which shows the current setting of the connected RFID reader. This button displays the following:

- "NO" when communication to the reader is not available. Use the button in this state to establish a connection to the reader.
- "RF" when the computer has an open connection to the reader. Pull the trigger or push the scan buttons to activate the reader. Use the button in this state to toggle from reader mode to scanner mode.
- "SC" when the computer is in scanner mode. Pull the trigger or push the scan buttons to activate the scanner. Use the button in this state to toggle from scanner mode to reader mode. If the reader is available, the button changes to "RF." If not, the button changes to "NO."

On the CV30

A TE 2000 “Reader State” button is available to add to the TE 2000 toolbar which shows the current setting of the connected RFID reader. This button displays the following:

- “NO” when communication to the reader is not available. Use the button in this state to establish a connection to the reader.
- “RF” when the computer has an open connection to the reader. Use the button in this state to activate the reader.

On the CV60

A TE 2000 “Reader State” button is available to add to the TE 2000 toolbar which shows the current setting of the connected RFID reader. This button displays the following:

- “ERROR” when communication to the reader is not available. Use the button in this state to establish a connection to the reader.
- “READ” when the computer has an open connection to the reader. Use the button in this state to activate the reader.
- “WRITE” when the #W write enabled command is processed. Use the button in this state to write the pending data to tags. A beep is emitted for each successfully written tag.

3270, 5250, VT/ANSI Data Streams Conventional Method

This writes to the display to execute the extended command. *Note that “~~b~~” indicates a 1-byte space.* Use spaces to retain the current parameter value.

3270, 5250, VT/ANSI Data Streams Conventional Method

Column	Description	Characters	Recommended Setting
2	Begin extended command	#	
3	RFID Set Attributes command	A	
4	Hand reader trigger state	1 = SCAN; 2 = RFID	
5-7	Delimiter character between tags	0-255 or ~~b~~	009
8-10	Data terminating character	0-255 or ~~b~~	013
11-12	Read tries per antenna	0-99 or ~~b~~	03
13-14	Write tries per antenna	0-99 or ~~b~~	03
15-16	Identify tries per antenna	0-99 or ~~b~~	03
17-18	Antenna tries - multiply read, write, and identify tries per antenna	0-99 or ~~b~~	03
19-20	Auto_id_timeout - milliseconds for identifies in autonomous mode	0-99 or ~~b~~	50
21-22	Initialization tries	0-99 or ~~b~~	01
23-25	Reserved		
26-29	Antennas to use and order to use them	1~~bbb~~-4444 or ~~bbbb~~	1~~bbb~~

3270, 5250, VT/ANSI Data Streams Conventional Method (continued)

Column	Description	Characters	Recommended Setting
30	Tag type. *Note that you can combine these hex values to "ALL" to read or write multiple tag types.*	1 = EPC Class 1 Gen 2 2 = ISO6B/G1 4 = ISO6B/G2 8 = Phillips V1.19	1 = EPC Class 1 Gen 2
31-40	Reserved		
41	End extended command delimiter	#	

Observe the following when using the #A command:

- The command terminating # character may be inserted after the final parameter to change without specifying the remaining parameters.
- The Antennas parameter (position 26-29) uses the numbers 1 through 4 that corresponds to the antennas of the reader. The default setting of 1 followed by three blanks tells the reader to use antenna 1 only where a setting of 213 with a blank would instruct the reader to read antenna 2, then 1, then 3.
  If all four bytes are blank, then the antenna is not modified.
  If any of the settings are not blank, then the antenna setting is modified to only enable those antennas in the order specified in this command.
- If you know what tags you plan to read or write frequently, enable just that tag type to improve performance, otherwise you will slow down the performance of your reader.
- Changing the initialization tries from the default to any value greater than the default allows for multiple identifies to occur for each tag in the antenna field. Changing the value decreases performance.
  Setting the initialization tries to 0 places the reader in autonomous mode, in which the reader continually attempts to read tags without requiring a button or trigger event. Only set this to 0 when tags are moving in and out of the antenna field.

Example #A Commands - *Note that "ƀ" indicates a 1-byte space*

Example 1

Set the data delimiter character to tab and the terminating character to a carriage return:

`ƀ#A2009013#`

Example 2

Set the terminating character to a carriage return, reduce read tries to 2, and reduce write tries to 4:

`^9F#A2ƀƀƀ0130204#^9C`

VT/ANSI Data Stream Control Character Method

This method uses the control characters APC (0x9f) and ST(0x9c) to encapsulate the extended command. Using the control characters method is faster since it is not written to the display. You do not need to clear the screen or move the cursor because the encapsulated command executes in the background. The format is exactly the same as the conventional method within the APC and ST control characters.

Note: For VT/ANSI data stream, send the # character to row 1, column 2 *last* to ensure all command data is received before processing the extended command.

Example #A Command - *Note that "~~b~~" indicates a 1-byte space*

This takes Example 2 and uses the control character method:

^9F#A2~~bbb~~0130204#^9C

Return Codes for the RFID (#A) Set Attributes Command

The return code is a status sent to the host computer in the **\\x<cr>** format, which indicates whether the extended command was successful. The return codes are listed in the following chart.

Return Codes for RFID (#A) Set Attributes Command

Code	Description
0	Good status, transaction complete
1	Incorrect trigger type setting. Must be "1" or "2"
2	Terminating '#' character missing
4	Incorrect value for read tries (must be 0-99 or spaces)
5	Incorrect value for write tries (must be 0-99 or spaces)
6	Incorrect value for identify tries (must be 0-99 or spaces)
7	Incorrect value for antenna tries (must be 0-99 or spaces)
8	Incorrect value for auto_id_timeout tries (must be 0-99 or spaces)
9	Incorrect value for initialization tries (must be 0-99 or spaces)
B	Incorrect setting for antennas (must be 1~~bbb~~-4444 or spaces)
C	Incorrect setting for TagType (must be 0-F) or space)
F	Communication error with reader
P	Socket interface generated error. Settings are ignored.
Z	Command not supported for this device.

RFID (#D) Define Format

Note: In software version 8.25 or greater, this command applies to all computers. Prior to version 8.25, this command applies to 700 Color series computers running Windows Mobile 2003 and the CV30 and CV60 regardless of operating system.

The #D extended command defines an SWTT mask to say what tags to identify, then defines formats up to eight fields to read from the tag. The command remains in effect until a new write, read, or filter command is sent to the computer.

3270, 5250, VT/ANSI Data Streams Conventional Method

This method writes to the display to execute the extended command.

3270, 5250, VT/ANSI Data Streams Conventional Method

Column	Description	Character
2	Begin extended command	#
3	RFID Define Format command	D
4-15	SWTT mask	12 Hex characters (0-F), X = wildcard
16	Field type	S = String; I = Integer (max 4 bytes)
17	Bank	Character “0” - “3”
18-22	Starting position of field to read	0-65535 (check tag data size for physical limit)
23-27	Length of field to read	0-65535 (32 maximum supported size for ISO tags - 12 for EPC Class 1 Gen 2 tags)
28	Field delimiter or end extended command	, or # Use # to end the command or comma to define up to 7 additional fields.

Note: If no fields are defined (# in column 16), then the TagID is read for ISO tags. If you are reading Electronic Product Code (EPC) Class 1 Gen 2 tags, then the EPCID is returned.

Note: If the command extends past the screen width (80 or 132), then the remaining data must go on the next line.

Note: Since the TagID or EPCID is a setting of hex pairs, you need to ensure that you send an even number of characters. If not, then communications between the computer and the RFID reader generates an error and the reader state on the TE 2000 toolbar is modified to show the new error state.

Note: If the TagType is set to multiple types, then setting a filter on TagID or EPCID generates error #9.

Note: The SWTT mask does not apply to Phillips U-Code v1.19 tags.

Example #D Commands - *Note that "ƀ" indicates a 1-byte space*

Example 1

Set the reader to return only the tagID from ISO tags or the EPC ID from EPC Class 1 Gen 2 tags:

To support variable length EPC IDs, the SWTT can start with a "V" character and terminate with a "V" character. The EPC Class 1 Gen 2 tags do not support a wild card in the EPCID filter. If a wild card is part of the EPCID, then error #9 is returned from the #D command.

```
ƀ#DXXXXXXXXXXXX#
```

Example 2

Set the reader to return an 8-byte string and a 2-byte integer from any tag: *Use the SWTT mask once during the entire command.*

```
ƀ#DXXXXXXXXXXXXS20001800008,I20002600002#
```

Example 3

Set the reader to read EPC Class 1 Gen 2 tags which have an EPCID starting with 0001:

```
ƀ#DV0001V#
```

VT/ANSI Data Stream Control Character Method

This method uses the control characters APC (0x9f) and ST(0x9c) to encapsulate the extended command. Using the control characters method is faster since it is not written to the display. You do not need to clear the screen or move the cursor because the encapsulated command executes in the background. The format is exactly the same as the conventional method within the APC and ST control characters.

Note: For VT/ANSI data stream, send the # character to row 1, column 2 *last* to ensure all command data is received before processing the extended command.

Example #D Command - *Note that "ƀ" indicates a 1-byte space*

This takes Example 2 from the top of this page and uses the control character method:

```
^9F#ƀ#DXXXXXXXXXXXXS20001800008,I20002600002#^9C
```

Return Codes for the RFID (#D) Define Format Command

The return code is a status sent to the host computer in the **\\x<cr>** format that indicates whether the extended command was successful. The return codes are listed in the following chart.

Return Codes for RFID (D#) Define Format Command

Code	Description
0	Good status, transaction complete.
1	Incorrect setting for SWTT not '0' - '9' or 'A' - 'F' or 'X' or “V”
2	Terminating '#' character missing or ',' character missing from separating field entries.
4	More then eight field definitions
6	No input field to hold RFID data for 3270 or 5250
7	Field type not 'I' or 'S'
8	Error in AID code (3270 or 5250 only)
9	Attempt to set a filter on TagID or EPCID when multiple tag types are selected or attempt to place a wild card in the EPCID filter.
A	Read Bank, not “0” - “3”
B	Field Start Position not in valid range 0-65535
C	Field Length is not in valid range (0-4 for Integer, 0-65535 for string)
Z	Command not supported for this device.

RFID (#W) Write Tags *(700 Color with PPC 2003, CV30, CV60)*

Note: In software version 8.25 or greater, this command applies to all computers.

The #W extended command initiates a write sequence to program an RFID tag. A “RFID Write Please pull trigger” message is shown on the screen until the RFID reader is triggered to complete the write. When the reader is activated, the computer emits a beep for each successfully written tag. If any tags were not written to correctly, an error beep is emitted. When the write command is complete, a “0” (GOOD) status and the number of successfully written tags are returned to the host application. If no tags were successfully written, then a “3” (Communication Error) value is returned to the host application.

3270, 5250, VT/ANSI Data Streams Conventional Method

This method writes to the display to execute the extended command.

Data Streams Conventional Method

Column	Description	Character
2	Begin extended command	#
3	RFID Write command	W
4-15	SWTT mask	12 Hex characters (0-F), X = wildcard
16	Field type	S = String; I = Integer (max 4 bytes)
17	Bank	Character “0” - “3”
18-22	Starting position of field to read	0-65535 (check tag data size for physical limit)
23-27	Length of field to read	0-65535 (32 maximum supported size for ISO tags, 4 if integer type - 12 for EPC Class 1 Gen 2 tags)
28-X	Data to write	String or integer type
X + 1	Field delimiter or end extended command	, or # Use # to end the command or comma to define up to 7 additional data fields with data

Note: If the command extends past the screen width (80 or 132), then the remaining data must go on the next line.

Example #W Command - *Note that “ƀ” indicates a 1-byte space*

Example 1

Write 1 11-byte string and 1 integer to any tag: *Use the SWTT mask once during the entire command.*

```
ƀ#WXXXXXXXXXXXXS20001800011HELLO WORLD,I200029000041234#
```

Example 2

To support variable length EPCIDs, the SWTT can start with a “V” character and terminate with a “V” character. EPC Class 1 Gen 2 tags do not support a wild card in the EPCID filter. If a wild card is part of the EPCID, then error #9 is returned from the #W command

Set the reader to write EPC Class 1 Gen 2 tags which have an EPCID starting with 0001. Only to these tags write the string DATA to bank 3 (user data):

```
ƀ#WV0001VS30000000004DATA#
```

Note: Since the TagID or EPCID is a setting of hex pairs, you need to ensure that you send an even number of characters. If not, then communications between the computer and the RFID reader generates an error and the reader state on the TE 2000 toolbar is modified to show the new error state.

Note: If the TagType is set to multiple types, then setting a filter on TagID or EPCID generates error #9.

Note: If you write data to EPC Class 1 Gen 2 tags, then all data written to the tags must be in words. If you attempt to write data that is not in a word format, then a communication error occurs between the reader and the computer. The reader state on the TE 2000 toolbar is modified to show the new error state.

Note: The SWTT mask does not apply to Phillips U-Code v1.19 tags.

VT/ANSI Data Stream Control Character Method

This method uses the control characters APC (0x9f) and ST(0x9c) to encapsulate the extended command. Using the control characters method is faster since it is not written to the display. You do not need to clear the screen or move the cursor because the encapsulated command executes in the background. The format is exactly the same as the conventional method within the APC and ST control characters.

Example #W Command - *Note that "b̶" indicates a 1-byte space*

This takes the previous example and uses the control character method. It writes the "Hello World" string to all tags starting at offset 18. At the 29th location in the tag, the "4660" data is written because the integer data is sent as hex data and hex "1234" is a decimal "4660."

```
^9F#WXXXXXXXXXXXXS20001800011HELLO WORLD,I200029000041234#^9C
```

Return Codes for the RFID (#W) Write Command

The return code is a status sent to the host computer in the **`\\x<cr>`** format indicating whether the extended command was successful. The return codes are listed in the chart on the next page.

Return Codes for RFID (#W) Write Command

Code	Description
0	Good status, transaction complete
1	Incorrect setting for SWTT not '0' - '9' or 'A' - 'F' or 'X' or "V"
2	Terminating '#' character missing or ',' character missing from separating field entries
3	Write operation timed out, a write error returned from RFID reader, or no tags found in the location being written.
4	More then eight field definitions
6	No input field to hold RFID data for 3270 or 5250
7	Field type not 'I' or 'S'
8	Error in AID code (3270 or 5250 only)
9	Attempt to set a filter on TagID or EPCID when multiple tag types are selected or attempt to place a wild card in the EPCID filter.
A	Bank not "0" - "3"

Return Codes for RFID (#W) Write Command (continued)

Code	Description
B	Field Start Position not in valid range 0-65535
C	Field Length is not in valid range (0-4 for Integer, 0-65535 for string)
D	Integer value to write to the tag is not a valid hex character
Z	Command not supported for this device.

When the return code is good, “0” the format of the return value is changed to `\\0\04\9999<cr>` where `\\0` is the good return status code, `\04` identifies the field size of the successful write count (min 4), and `\9999` is the count of the number of successfully written tags.

RFID (#J) Set Filter *(700 Color with PPC 2003, CV30, CV60)*

Note: In software version 8.25 or greater, this command applies to all computers.

The #J extended command removes or sets a filter the tags that can be read or written to. Tags that do not match the filter are ignored. A field length of zero removes the filter. If the bank is 0, then the tag filter is assumed to be in hex pairs.

3270, 5250, and VT/ANSI Data Streams Conventional Method

Data Streams Conventional Method

Column	Description	Character
2	Begin extended command	#
3	RFID Set Filter command	J
4	Bank	0-3
5-9	Starting position	
10-14	Length of filter	0-255
15-X	Filter comparison value	
X + 1	End extended command	#

Example #J Commands - *Note that “ƀ” indicates a 1-byte space*

Example 1

Set a writing filter to match the first two bytes of tags with their tag ID starting with 0211:

ƀ#J000000000040211#

Example 2

Set a reading filter to match ISO/G1 tags that have DATA in position 18:

b#J2000180004DATA#

Example 3

Set a reading filter to match EPC Class 1 Gen 2 tags that have DATA at location 0 of the user data bank (3) of the tag:

b#J30000000004DATA#

VT/ANSI Data Stream Control Character Method

This method uses the control characters APC (0x9f) and ST(0x9c) to encapsulate the extended command. Using the control characters method is faster since they are not written to the display. You do not need to clear the screen or move the cursor because the encapsulated command executes in the background. The format is exactly the same as the conventional method within the APC and ST control characters.

Note: For VT/ANSI data stream, send the # character to row 1, column 2 *last* to ensure all command data is received before processing the extended command.

Example #J Command - *Note that “b̸” indicates a 1-byte space*

This takes Example 2 on the previous page and uses the control character method:

^9F#J0001800004DATA#^9C

Return Codes for the RFID (#J) Set Filter Command

The return code is a status sent to the host computer in the **\\x<cr>** format indicating whether the extended command was successful. The return codes are listed in the following chart.

Return Codes for RFID (#J) Set Filter Command

Code	Description
0	Good status, transaction complete
1	Invalid starting position, must be 0-65535
2	Terminating '#' character missing
3	Error in AID code (3270 or 5250 only)
4	Invalid filter length, must be 0-255
A	Bank not “0” - “3”
Z	Command not supported for this device.

About RFID *(700 Color with PPC 2003, CV30, CV60)*

Note: In software version 8.25 or greater, this applies to all computers.

By default, the reader reads EPC Class 1 Gen 2 tags. The EPCID from all tags will be the data that is read. EPC Class 1 Gen 2 Tags are mapped as follows:

EPC Class 1 Gen 2 Tags

Memory Bank Name	Value
Reserved (passwords)	0
EPC	1 *(default is x)*
TagID	2
User memory	3

The EPC bank is set up as Bytes 0-1 to correspond to the CRC-16. The tag recalculates the CRC-16 value each time the tag is powered on. Bytes 2-3 correspond to the protocol control (PC) word which includes the data length field, the EPC/ISO bit, and the header bits. Users must be careful to encode the PC correctly. The remainder of this bank, up to 66 bytes, contains the EPC. The tag manufacturer may supply less then 31 EPC data words.

The address parameter can range from 0 to the maximum address for the defined field. When the address is larger then what is available on the tag, the address wraps to the beginning of the tags memory for ISO 18000-6B tags, but an error is returned for out of range addresses on EPC Class 1 Gen 2 tags.

Not all EPC Class 1 Gen 2 tags have user a data bank.

When writing to EPC Class 1 Gen 2 tags, there is a limitation on the addresses and lengths that can be written. EPC Class 1 Gen 2 tags only support writing to words or 16 bit values. Thus, it is only possible to write to even byte addresses that have even length values.

The filters on the EPC Class 1 Gen 2 tags do not support wild cards. This must match represent a 2-character hex value. Also, if multiple TagTypes are enabled, then filters on the TagID or EPCID are not allowed and return an error result.

ISO tags are designated as ISO18000-6b. There are two sizes of ISO tags available:

- Phillips chipset tags are in flexible media (printer stock) and are 223 bytes in size. With these tags, the TagID is not protected and can be overwritten. These are referred to as Phillips V1.19 tags.

- Fairchild chipset tags are molded in solid media (bin labels/credit card style) and are 128 bytes in size. Reading from and writing to tag locations that exceed the maximum tag size do not cause an error, it does wrap the command to the offset location.

The ISO tag memory is mapped out like this:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18-12 7	128-222

Bytes 0 - 7 are the tag ID (16 hex characters) read only
Bytes 8 - 11 are reserved and not writable.
Bytes 12 - 17 make up the SWTT (12 hex characters) rewritable
Bytes 18 - 127 are for general use
Bytes 128 - 222 are for general use (Phillips chipset only)

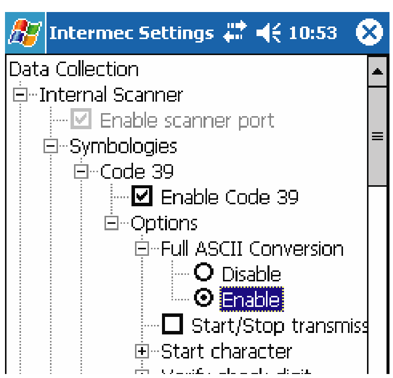
A Bar Code Scanning

This appendix lists bar code labels for TE 2000 commands and functions. It also contains Encoded Code 39 key press sequences. Code 39 Full ASCII must be enabled in the firmware (default is disabled).

Note: For 5250 emulation, when Encoded Code 39 is enabled, a Field Exit key is generated automatically when the value of the last scanned data character is between 0-255.

To scan the bar code labels, configure the computer to use Code 39 in Full ASCII mode using one of two methods:

Via the Intermec Settings applet, click (+) to expand **Data Collection > Internal Scanner > Symbologies > Code 39 > Full ASCII Conversion**, then check **Enable**. *This sample is from the 700 Color.*



- Within the TE 2000 menus, you can select Full ASCII under Code 39. However, you also have to enable **ParmsBySession**. For more information, see "4) ParmsBySession" on page 109.

Cursor Keys

Window/Viewport up *(up one line)*

%UP

Window/Viewport down *(down one line)*

%DN

Window/Viewport right *(right one character)*

%RT

Window/Viewport left *(left one character)*

%LF

Paging Keys

Page up

%PGUP

Page down

%PGDN

Page right

%PGRT

Page left

%PGLT

Tab Keys

Back Tab

%BTAB

Forward Tab

%TAB

Additional Functions - Opening TE 2000 Configuration Menus

TE 2000 configuration menus

%TECFG

Auto-Login Restart

Auto-Login Restart

%ALRS

3278 SNA Keys

Cursor Home (Home)

%Hm

Delete (Del)

%DEL

EOF

%EOF

Erase Input (Clr)

%EINP

Insert

%INS

New Line (Return)

%NL

Reset

%RST

AID-Generating (3270, 5250, Native) or Top-Row Function Keys (VT/ANSI)

Clear - 3270, 5250, Native

%CLR

Enter - 3270, 5250, Native

%CR

F1 - 3270, 5250, Native, VT220/320

%F1

F2 - 3270, 5250, Native, VT220/320

%F2

F3 - 3270, 5250, Native, VT220/320

%F3

F4 - 3270, 5250, Native, VT220/320

%F4

F5 - 3270, 5250, Native, VT220/320

%F5

F6 - 3270, 5250, Native, VT220/320

%F6

F7 - 3270, 5250, Native, VT220/320

%F7

F8 - 3270, 5250, Native, VT220/320

%F8

F9 - 3270, 5250, Native, VT220/320

%F9

F10 - 3270, 5250, Native, VT220/320

%F10

F11 - 3270, 5250, Native, VT100, VT220/320

%F11

F12 - 3270, 5250, Native, VT100, VT220/320

%F12

F13 - 3270, 5250, Native, VT100, VT220/320

%F13

F14 - 3270, 5250, Native, VT220/320

%F14

F15 - 3270, 5250, Native, VT220/320

%F15

F16 - 3270, 5250, Native, VT220/320

%F16

F17 - 3270, 5250, Native, VT220/320

%F17

F18 - 3270, 5250, Native, VT220/320

%F18

F19 - 3270, 5250, Native, VT220/320

%F19

F20 - 3270, 5250, Native, VT220/320

%F20

Note: *For VT/ANSI*, scan the following F21 bar code label to toggle between Line Edit (block) mode and Character mode:

F21 - 3270, 5250, Native, VT/ANSI

%F21

F22 - 3270, 5250, Native

%F22

F23 - 3270, 5250, Native

%F23

F24 - 3270, 5250, Native

%F24

PA1 - 3270

%PA1

PA2 - 3270

%PA2

PA3 - 3270

%PA3

Help (nonerror state) - 5250

%HELP

Print - 5250

%PRINT

Record Backspace (Home) - 5250

%Hm

Roll Down - 5250

%RODN

Roll Up - 5250

%ROUP

Preamble and Postamble - 3270, 5250

When you set the preamble or postamble for a bar code symbology, there are special values that the TE 2000 application interprets to generate the expected key stroke. These special values are listed in the following table:

Preamble and Postamble

3270 Emulation		Wedge Amble	API Amble
F10	VK_F10	0x7a	0xea
5250 Emulation			
Field Exit	VK_PLAY	0xfa	0x9a
Enter	VK_RETURN	0x0d	0x0d
BackTab	VK_BTAB	0x0a	0xdc
F10	VK_F10	0x7a	0xea

Symbols - 3270, 5250

Cent (¢)

%CENT

Not (¬)

%NOT

Field Exit Key - 5250

Field Exit

%FLDX

Signal Keys - 5250

Attn

%ATTN

Help (from error state)

%HELP

Special Control Keys - 5250

Delete (Del)

%DEL

Erase Input

%EINP

Error Reset

%ERR

Hex

%HEX

Home

%Hm

Insert

%INS

Special Host Key - 5250

Sys Req

%SYSR

5250 Additional Functions

Dup (duplicate enabled fields only)

%DUP

Field-

%FLD-

Field+

%FLD+

Field Mark

%FM

New Line

%NL

Special Function Keys - VT/ANSI

Backspace

%BKSP

Delete (Del)

%DEL

Editing Keys - VT/220/320

Find

%FIND

Insert

%INS

Next Screen

%NEXT

Previous Screen

%PREV

Remove

%REM

Select

%SEL

Encoded Code 39

The following table lists escape characters and key press sequences for Encoded Code 39. The “(t)” in the table indicates a terminating key. Any bar code data following this key code is ignored. The “t” sequences, therefore, should be located only at the end of the bar code. If you attempt to use an invalid sequence (termed "reserved" in the table) the computer beeps and the data stream is flushed.

Note: For 5250 emulation, when Encoded Code 39 is enabled, a Field Exit is generated when the last scanned character is a data character (0 >= char <= 255).

Key Press Sequences for Encoded Code 39

Sequence	Key	Sequence	Key
$space *(VT/ANSI)*	Find (t)	+space *(VT/ANSI)*	09 hexadecimal (t)
$- *(VT/ANSI)*	Insert here (t)	+- *(VT/ANSI)*	0A hexadecimal (t)
$. *(VT/ANSI)*	Remove (t)	+. *(VT/ANSI)*	0B hexadecimal (t)
$0 *(VT/ANSI)*	Keypad 0 (t)	+0 *(VT/ANSI)*	0C hexadecimal (t)
$1 *(VT/ANSI)*	Keypad 1 (t)	+1 *(VT/ANSI)*	0D hexadecimal (t)
$2 *(VT/ANSI)*	Keypad 2 (t)	+2 *(VT/ANSI)*	0E hexadecimal (t)
$3 *(VT/ANSI)*	Keypad 3 (t)	+3 *(VT/ANSI)*	0F hexadecimal (t)
$4 *(VT/ANSI)*	Keypad 4 (t)	+4 *(VT/ANSI)*	10 hexadecimal (t)
$5 *(VT/ANSI)*	Keypad 5 (t)	+5 *(VT/ANSI)*	11 hexadecimal (t)
$6 *(VT/ANSI)*	Keypad 6 (t)	+6 *(VT/ANSI)*	12 hexadecimal (t)
$7 *(VT/ANSI)*	Keypad 7 (t)	+7 *(VT/ANSI)*	13 hexadecimal (t)
$8 *(VT/ANSI)*	Keypad 8 (t)	+8 *(VT/ANSI)*	14 hexadecimal (t)
$9 *(VT/ANSI)*	Keypad 9 (t)	+9 *(VT/ANSI)*	15 hexadecimal (t)
$A	New Line *(3270, 5250, VT/ANSI)*	+A	a
$B	Delete (t)	+B	b
$C	Forward Tab (t)	+C	c
$D	Forward Tab (t)	+D	d
$E	Back Tab *(3270, 5250, Native)*	+E	e
$F	Roll Up/Page Down *(5250)*	+F	f
$G	Roll Down/Page Up *(5250)*	+G	g
$H	Backspace (t)	+H	h
$I	Field + *(5250)*	+I	i
$J	Field - *(5250)*	+J	j
$K	Insert *(3270, 5250, Native)*	+K	k
$L	Home *(3270, 5250, Native)*	+L	l
$M	Enter (t) *(3270, Native, VT/ANSI)* or Enter/Rec Adv (t) *(5250)*	+M	m
$N	Erase *(3270)*, Field Exit *(5250)*, or End of Field *(Native)*	+N	n

Key Press Sequences for Encoded Code 39 (continued)

Sequence	Key	Sequence	Key
$O	Clear *(3270, Native)* or Erase Input *(5250)*	+O	o
$P	Attn *(5250)*	+P	p
$Q	PF1 (t) *(3270, Native, VT/ANSI)* or F1 (t) *(5250)*	+Q	q
$R	PF2 (t) *(3270, Native, VT/ANSI)* or F2 (t) *(5250)*	+R	r
$S	PF3 (t) *(3270, Native, VT/ANSI)* or F3 (t) *(5250)*	+S	s
$T	PF4 (t) *(3270, Native, VT/ANSI)* or F4 (t) *(5250)*	+T	t
$U	F5 (t) *(5250, VT/ANSI)* or PF5 *(3270, Native)*	+U	u
$V	F6 (t) *(5250, VT/ANSI)* or PF6 *(3270, Native)*	+V	v
$W	F7 (t) *(5250, VT/ANSI)* or PF7 *(3270, Native)*	+W	w
$X	F8 (t) *(5250, VT/ANSI)* or PF8 *(3270, Native)*	+X	x
$Y	F9 (t) *(5250, VT/ANSI)* or PF9 *(3270, Native)*	+Y	y
$Z	F10 (t) *(5250, VT/ANSI)* or PF10 *(3270, Native)*	+Z	z
%space *(VT/ANSI)*	Select (t)	/space *(VT/ANSI)*	16 hexadecimal (t)
%- *(VT/ANSI)*	Previous screen (t)	/- *(VT/ANSI)*	17 hexadecimal (t)
%. *(VT/ANSI)*	Next screen (t)	/. *(VT/ANSI)*	18 hexadecimal (t)
%0 *(VT/ANSI)*	Enter (t)	/0 *(VT/ANSI)*	19 hexadecimal (t)
%1 *(VT/ANSI)*	00 hexadecimal (t)	/1 *(VT/ANSI)*	1A hexadecimal (t)
%2 *(VT/ANSI)*	01 hexadecimal (t)	/2 *(VT/ANSI)*	1B hexadecimal (t)
%3 *(VT/ANSI)*	02 hexadecimal (t)	/3 *(VT/ANSI)*	1C hexadecimal (t)
%4 *(VT/ANSI)*	03 hexadecimal (t)	/4 *(VT/ANSI)*	1D hexadecimal (t)
%5 *(VT/ANSI)*	04 hexadecimal (t)	/5 *(VT/ANSI)*	1E hexadecimal (t)
%6 *(VT/ANSI)*	05 hexadecimal (t)	/6 *(VT/ANSI)*	1F hexadecimal (t)
%7 *(VT/ANSI)*	06 hexadecimal (t)	/7 *(VT/ANSI)*	Reserved
%8 *(VT/ANSI)*	07 hexadecimal (t)	/8 *(VT/ANSI)*	Reserved
%9 *(VT/ANSI)*	08 hexadecimal (t)	/9 *(VT/ANSI)*	Reserved
%A	Clear AID (t) *(3270, Native)* or Clear *(5250)*	/A	! (exclamation mark)
%B	F11 (t) *(5250, VT/ANSI)* or PF11 *(3270, Native)*	/B	" (double quote)
%C	F12 (t) *(5250, VT/ANSI)* or PF12 *(3270, Native)*	/C	# (pound)
%D	PA1 *(3270)* or Error Reset *(5250)*	/D	$ (dollar)

Key Press Sequences for Encoded Code 39 (continued)

Sequence	Key	Sequence	Key
%E	PA2 *(3270)* or Help *(5250)*	/E	% (percent)
(None)	PA3 *(3270)*		
%F	; (semicolon)	/F	& (ampersand)
%G	< (less than)	/G	' (single quote)
%H	= (equal)	/H	((left parenthesis)
%I	> (greater than)	/I	) (right parenthesis)
%J	? (question mark)	/J	* (asterisk)
%K	[(left brace) *(3270, Native, VT/ANSI)* or ¯(not symbol) *(5250)*	/K	+ (plus)
%L	(backslash)	/L	, (comma)
%M	] (right brace) *(3270, Native, VT/ANSI)* or ¢ (cent) *(5250)*	/M	- (hyphen)
%N	^ (circumflex) *(3270, Native, VT/ANSI)* or | (piping symbol) *(5250)*	/N	F14 (t)
%O	_ (underscore)	/O	/ (forward slash)
%P	{ (left brace)	/P	F15 (t)
%Q	| (vertical bar)	/Q	F16 (t)
%R	} (right brace)	/R	F17 (t)
%S	~ (tilde)	/S	F18 (t)
%T	Keyboard delete (t)	/T	F19 (t)
%U	Dup *(5250)*	/U	F20 (t)
%V	@ (at)	/V	F21 *(3270, 5250, Native)*
%W	ι (grave accent)	/W	F22 *(3270, 5250, Native)*
%X	Sys Req *(5250)*	/X	F23 *(3270, 5250, Native)*
%Y	Print *(5250)*	/Y	F24 *(3270, 5250, Native)*
%Z	F13 (t) *(5250, VT/ANSI)* or PF13 *(3270)*	/Z	: (colon)

Terminating Keys

Terminating keys are the nonprintable ASCII sequences and action keys. When the computer encounters them in a bar code, an action is taken, and the computer sends the data in the buffer to the host computer. Terminating keys should appear only at the end of the bar code. If they are located in the middle of a bar code, they are executed normally, but the data following them in the bar code is ignored. Terminating keys cause a computer-to-base station transmission. The computer ignores data in the bar code buffer following these keys once a transmission takes place.

For example, the computer interprets the following sequence:

`123$V456` `as`

`123F6`

The computer will not send "456" to the host PC, because it follows terminating key F6.

ASCII sequences can be used any time before a terminating key. For example, the computer interprets

`+H+E+L+L+O$M` as

`hello<Enter>`

Escape Characters

The four escape characters in the previous table yield a VT220 data stream key press equivalent when followed by another character. The escape characters are:

$ (dollar sign)
% (percent)
+ (plus)
/ (forward slash)

For example:

- If a bar code contains the sequence "%U" somewhere within it, the computer converts this sequence to an [ENTER] key and processes it as soon as encountered in the scanning buffer.
- "+B" is converted to the lower case "b."
- "%B" is converted to an F11 key press.

If you want the Encoded Code 39 option but the bar codes to be scanned already contain the "$," "%," "+," or "/," character, then each place where these characters occur must be expanded to a special "/" sequence:

- Every bar code where the "$" is maintained must be expanded to a "/D" sequence.
- Percent signs (%) must be expanded to "/E".
- Forward slashes (/) must be expanded to the letter "/O".
- Plus signs (+) must be expanded to "/K".

Overriding Auto Tab Scan and Auto Enter Scan (3270, Native)

When the computer is in **Auto Entr Scan** or **Auto Tab Scan** mode, eight Encoded Code 39 functions override these modes when they are scanned.

- Forward Tab and Back Tab
- End of Field and Home
- Backspace and Insert
- Clear and Delete

These codes are all of the screen-editing type, where an automatic [Enter] is not desired. The listed encoded operations never allow an **Auto Entr Scan** to occur. For example, if **Auto Entr Scan** was enabled and a "$C" (forward tab) was scanned, the computer forward tabs to the next field, but does not perform an [Enter], even though the **Auto Entr Scan** feature is enabled. In this case, the encoded forward tab overrides the **Auto Entr Scan** mode. However, if a "+D" is scanned, the computer places the letter "d" at the current cursor location and the **Auto Entr Scan** mode then executes an [Enter].

B Tables

This Appendix contains the following tables:

- Computer Font Set
- Full ASCII
- Decimal-to-Hex conversion
- Binary-to-EBCDIC conversion

Computer Font Set

Computers use a font set that supports English and Western European languages, such as French, German, Italian, Portuguese, and Spanish. You can develop applications that display any character in the computer font set. Depending on the type of keypad, you can also enter many of the characters. For help, see "Using the Computer Keypad" on page 19.

The following table lists the characters you can display on the computer. It also lists the decimal and hexadecimal index values.

Computer Font Set

Character	Decimal	Hexadecimal
Space	32	20
!	33	21
=	34	22
#	35	23
$	36	24
%	37	25
&	38	26
'	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
, (comma)	44	2C
- (dash)	44	2D
. (period)	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
: (colon)	58	3A
; (semicolon)	59	3B
< (less than)	60	3C
= (equal sign)	61	3D

Computer Font Set (continued)

Character	Decimal	Hexadecimal
> (greater than)	62	3E
? (question mark)	63	3F
@ (ampersand)	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
[(left bracket)	91	5B
\ (backslash)	92	5C
] (right bracket)	93	5D
^	94	5E
_	95	5F
`	96	60
a	97	61
b	98	62
c	99	63
d	100	64
e	101	65

Computer Font Set (continued)

Character	Decimal	Hexadecimal
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B
l	108	6C
m	109	6D
n	110	6E
o	111	6F
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A
{ (left curly bracket)	123	7B
¦	124	7C
} (right curly bracket)	125	7D
~	126	7E
⌂	127	7F
Ç	128	80
ü	129	81
é	130	82
â	131	83
ä	132	84
à	133	85
å	134	86
ç	135	87
ê	136	88
ë	137	89
è	138	8A
ï	139	8B
î	140	8C
ì	141	8D

Computer Font Set (continued)

Character	Decimal	Hexadecimal
Ä	142	8E
Å	143	8F
É	144	90
æ	145	91
Æ	146	92
ô	147	93
ö	148	94
ò	149	95
û	150	96
ù	151	97
ÿ	152	98
Ö	153	99
Ü	154	9A
¢	155	9B
£	156	9C
¥	157	9D
ℜ	158	9E
ƒ	159	9F
á	160	A0
í	161	A1
ó	162	A2
ú	163	A3
ñ	164	A4
Ñ	165	A5
ª	166	A6
º	167	A7
¿	168	A8
¬ (Not symbol)	169	A9
⌐	170	AA
½	171	AB
¼	172	AC
¡	173	AD
«	174	AE
»	175	AF
░	176	B0
▒	177	B1
▓	178	B2
│	179	B3
┤	180	B4
╡	181	B5

Computer Font Set (continued)

Character	Decimal	Hexadecimal
╢	182	B6
╖	183	B7
╕	184	B8
╣	185	B9
	186	BA
╗	187	BB
╝	188	BC
╜	189	BD
╛	190	BE
┐	191	BF
b	192	C0
┴	193	C1
>	194	C2
├	195	C3
–	196	C4
†	197	C5
╞	198	C6
	199	C7
╚	200	C8
╔	210	C9
╩	202	CA
╠	203	CB
╠	204	CC
=	205	CD
╬	206	CE
╧	207	CF
╨	208	D0
╤	209	D1
╥	210	D2
╙	211	D3
╘	212	D4
╒	213	D5
╓	214	D6
╫	215	D7
‡	216	D8
┘	217	D9
┌	218	DA
H	219	DB
▄	220	DC

Computer Font Set (continued)

Character	Decimal	Hexadecimal
▌	221	DD
▐	222	DE
▬	223	DF
∝	224	E0
β	225	E1
	226	E2
∏	227	E3
∑	228	E4
σ	229	E5
μ	230	E6
γ	231	E7
σ	232	E8
θ	233	E9
Ω	234	EA
δ	235	EB
∞	236	EC
ø	237	ED
∈	238	EE
∩	239	EF
≡	240	F0
±	241	F1
≥	242	F2
≤	243	F3
⌠	244	F4
⌡	245	F5
÷	246	F6
≈	247	F7
°	248	F8
○	249	F9
H	250	FA
√	251	FB
^{n}	252	FC
^{2}	253	FD
■	254	FE
Space	255	FF

Full ASCII

The following table lists the ASCII characters and their binary, hexadecimal, and Code 39 equivalents.

Full ASCII Characters

Binary[0]	Hex[1]	Dec[2]	C39[3]	Char[4]	Binary[0]	Hex[1]	Dec[2]	C39[3]	Char[4]
00000000	00	00	%U	NUL	00100000	20	32	SP	SP[5]
00000001	01	01	$A	SOH	00100001	21	33	/A	!
00000010	02	02	$B	STX	00100010	22	34	/B	“
00000011	03	03	$C	ETX	00100011	23	35	/C	#
00000100	04	04	$D	EOT	00100100	24	36	/D	$
00000101	05	05	$E	ENQ	00100101	25	37	/E	%
00000110	06	06	$F	ACK	00100110	26	38	/F	&
00000111	07	07	$G	BEL	00100111	27	39	/G	‘
00001000	08	08	$H	BS	00101000	28	40	/H	(
00001001	09	09	$I	HT	00101001	29	41	/I	)
00001010	0A	10	$J	LF	00101010	2A	42	/J	*
00001011	0B	11	$K	VT	00101011	2B	43	/K	+
00001100	0C	12	$L	FF	00101100	2C	44	/L	,
00001101	0D	13	$M	CR	00101101	2D	45	/M	-
00001110	0E	14	$N	SO	00101110	2E	46	/N	.
00001111	0F	15	$O	SI	00101111	2F	47	/O	/
00010000	10	16	$P	DLE	00110000	30	48	/P[6]	0
00010001	11	17	$Q	DC1	00110001	31	49	/Q	1
00010010	12	18	$R	DC2	00110010	32	50	/R	2
00010011	13	19	$S	DC3	00110011	33	51	/S	3
00010100	14	20	$T	DC4	00110100	34	52	/T	4
00010101	15	21	$U	NAK	00110101	35	53	/U	5
00010110	16	22	$V	SYN	00110110	36	54	/V	6
00010111	17	23	$W	ETB	00110111	37	55	/W	7
00011000	18	24	$X	CAN	00111000	38	56	/X	8
00011001	19	25	$Y	EM	00111001	39	57	/Y	9
00011010	1A	26	$Z	SUB	00111010	3A	58	/Z	:
00011011	1B	27	%A	ESC	00111011	3B	59	%F	;
00011100	1C	28	%B	FS	00111100	3C	60	%G	<
00011101	1D	29	%C	GS	00111101	3D	61	%H	=
00011110	1E	30	%D	RS	00111110	3E	62	%I	>
00011111	1F	31	%E	US	00111111	3F	63	%J	?

Full ASCII Characters (continued)

Binary[0]	Hex[1]	Dec[2]	C39[3]	Char[4]	Binary[0]	Hex[1]	Dec[2]	C39[3]	Char[4]
01000000	40	64	%V	@	01100000	60	96	%W	‘
01000001	41	65	A	A	01100001	61	97	+A	a
01000010	42	66	B	B	01100010	62	98	+B	b
01000011	43	67	C	C	01100011	63	99	+C	c
01000100	44	68	D	D	01100100	64	100	+D	d
01000101	45	69	E	E	01100101	65	101	+E	e
01000110	46	70	F	F	01100110	66	102	+F	f
01000111	47	71	G	G	01100111	67	103	+G	g
01001000	48	72	H	H	01101000	68	104	+H	h
01001001	49	73	I	I	01101001	69	105	+I	i
01001010	4A	74	J	J	01101010	6A	106	+J	j
01001011	4B	75	K	K	01101011	6B	107	+K	k
01001100	4C	76	L	L	01101100	6C	108	+L	l
01001101	4D	77	M	M	01101101	6D	109	+M	m
01001110	4E	78	N	N	01101110	6E	110	+N	n
01001111	4F	79	O	O	01101111	6F	111	+O	o
01010000	50	80	P	P	01110000	70	112	+P	p
01010001	51	81	Q	Q	01110001	71	113	+Q	q
01010010	52	82	R	R	01110010	72	114	+R	r
01010011	53	83	S	S	01110011	73	115	+S	s
01010100	54	84	T	T	01110100	74	116	+T	t
01010101	55	85	U	U	01110101	75	117	+U	u
01010110	56	86	V	V	01110110	78	118	+V	v
01010111	57	87	W	W	01110111	77	119	+W	w
01011000	58	88	X	X	01111000	78	120	+X	x
01011001	59	89	Y	Y	01111001	79	121	+Y	y
01011010	5A	90	Z	Z	01111010	7A	122	+Z	z
01011011	5B	91	%K	[	01111011	7B	123	%P	{
01011100	5C	92	%L	\	01111100	7C	124	%Q	|
01011101	5D	93	%M	]	01111101	7D	125	%R	}
01011110	5E	94	%N	^	01111110	7E	126	%S	~
01011111	5F	95	%O	-	01111111	7F	127	%T[7]	n[8]

Notes for the Full ASCII Table:

0 Bit positions are 76543210.
1 Hexadecimal value
2 Decimal value
3 Code 39 character(s)
4 SCII character
5 SP is the SPACE character.
6 The Code 39 characters /P through /Y may be interchanged with the numbers 0 through 9.
7 May be interchanged with %X or %Y or %Z.
8 n is the DELETE character.

Decimal to Hexadecimal

Following are decimal and hexadecimal values for nondisplayable ASCII and displayable graphic characters.

Decimal to Hexadecimal Values

Decimal	Hex	Character	Decimal	Hex	Character	Decimal	Hex	Character
0	00	NUL	38	26	&	76	4C	L
1	01	SOH	39	27	'	77	4D	M
2	02	STX	40	28	(	78	4E	N
3	03	ETX	41	29	)	79	4F	O
4	04	EOT	42	2A	*	80	50	P
5	05	ENQ	43	2B	+	81	51	Q
6	06	ACK	44	2C	,	82	52	R
7	07	BEL	45	2D	-	83	53	S
8	08	BS	46	2E	.	84	54	T
9	09	HT	47	2F	/	85	55	U
10	0A	LF	48	30	0	86	56	V
11	0B	VT	49	311	1	87	57	W
12	0C	FF	50	32	2	88	58	X
13	0D	CR	51	33	3	89	59	Y
14	0E	SO	52	34	4	90	5A	Z
15	0F	SI	53	35	5	91	5B	[
16	10	DLE	54	36	6	92	5C	
17	11	DC1	55	37	7	93	5D	]
18	12	DC2	56	38	8	94	5E	^
19	13	DC3	57	39	9	95	5F	_
20	14	DC4	58	3A	:	96	60	`
21	15	NAK	59	3B	;	97	61	a
22	16	SYN	60	3C	<	98	62	b
23	17	ETB	61	3D	=	99	63	c
24	18	CAN	62	3E	>	100	64	d
25	19	EM	63	3F	?	101	65	e
26	1A	SUB	64	40	@	102	66	f
27	1B	ESC	65	41	A	103	67	g
28	1C	FS	66	42	B	104	68	h
29	1D	GS	67	43	C	105	69	i
30	1E	RS	68	44	D	106	6A	j
31	1F	US	69	45	E	107	6B	k
32	20	DP	70	46	F	108	6C	l
33	21	!	71	47	G	109	6D	m

Decimal to Hexadecimal Values (continued)

Decimal	Hex	Character	Decimal	Hex	Character	Decimal	Hex	Character
34	22	“	72	48	H	110	6E	n
35	23	#	73	49	I	111	6F	o
36	24	$	74	4A	J	112	70	p
37	25	%	75	4B	K	113	71	q
114	72	r	173	AD		232	E8	è
115	74	s	174	AE		233	E9	é
116	74	t	175	AF		234	EA	ê
117	75	u	176	B0	d	235	EB	ë
118	76	v	177	B1	+	236	EC	ì
119	77	w	178	B2	2	237	ED	í
120	78	x	179	B3	3	238	EE	î
121	79	y	180	B4		239	EF	ï
122	7A	z	181	B5	m	240	F0	
123	7B	{	182	B6	¶	241	F1	ñ
124	7C	|	183	B7	·	242	F2	ò
125	7D	}	184	B8		243	F3	ó
126	7E	~	185	B9	1	244	F4	ô
127	7F	DEL	186	BA	º	245	F5	õ
128	80		187	BB	»	246	F6	ö
129	81		188	BC	¼	247	F7	
130	82		189	BD	½	248	F8	ø
131	83		190	BE		249	F9	ù
132	84	IND	191	BF	¿	250	FA	ú
133	85	NEL	192	C0	À	251	FB	û
134	86	SSA	193	C1	Á	252	FC	ü
135	87	ESA	194	C2	Â	253	FD	ÿ
136	88	HTS	195	C3	Ã	254	FE	
137	89	HTJ	196	C4	Ä	255	FF	
138	8A	VTS	197	C5	Å			
139	8B	PLD	198	C6	Æ			
140	8C	PLU	199	C7	Ç			
141	8D	RI	200	C8	È			
142	8E	SS2	201	C9	É			
143	8F	SS3	202	CA	Ê			
144	90	DCS	203	CB	Ë			
145	91	PU1	204	CC	ì			
146	92	PU2	205	CD	í			
147	93	STS	206	CE	î			
148	94	CCH	207	CF	ï			
149	95	MW	208	D0				

Decimal to Hexadecimal Values (continued)

Decimal	Hex	Character	Decimal	Hex	Character	Decimal	Hex	Character
150	96	SPA	209	D1	Ñ			
151	97	EPA	210	D2	Ò			
152	98		211	D3	Œ			
153	99		212	D4	Ø			
154	9A		213	D5	ú			
155	9B	CSI	214	D6	ù			
156	9C	ST	215	D7	û			
157	9D	OSC	216	D8	Ø			
158	9E	PM	217	D9	ù			
159	9F	APC	218	DA	ú			
160	A0		219	DB	û			
161	A1	¡	220	DC	Ü			
162	A2	¢	221	DD	ÿ			
163	A3	£	222	DE				
164	A4		223	DF	ß			
165	A5	¥	224	E0	à			
166	A6	¦	225	E1	á			
167	A7	§	226	E2	â			
168	A8	¤	227	E3	ã			
169	A9	E	228	E4	ä			
170	AA	ª	229	E5	å			
171	AB	«	230	E6	æ			
172	AC		231	E7	ç			

Binary to EBCDIC

The following table includes binary to EBCDIC conversion values.

Binary to EBCDIC Values

Bits 0-7	EBCDIC Code	Display Graphic
00 0000	40	<space>
00 0001	C1	A
00 0010	C2	B
00 0011	C3	C
00 0100	C4	D
00 0101	C5	E
00 0110	C6	F
00 0111	C7	G
00 1000	C8	H
00 1001	C9	I
00 1010	4A	¢
00 1011	4B	.
00 1100	4C	<
00 1101	4D	(
00 1110	4E	+
00 1111	4F	|
01 0000	50	&
01 0001	D1	J
01 0010	D2	K
01 0011	D3	L
01 0100	D4	M
01 0101	D5	N
01 0110	D6	O
01 0111	D7	P
01 1000	D8	Q
01 1001	D9	R
01 1010	5A	!
01 1011	5B	$
01 1100	5C	*
01 1101	5D	)
01 1110	5E	;
01 1111	5F	`
10 0000	60	–
10 0001	61	/
10 0010	E2	S
10 0011	E3	T

Binary to EBCDIC Values (continued)

Bits 0-7	EBCDIC Code	Display Graphic
10 0100	E4	U
10 0101	E5	V
10 0110	E6	W
10 0111	E7	X
10 1000	E8	Y
10 1001	E9	Z
10 1010	6A	EBCDIC
10 1011	6B	,
10 1100	6C	%
10 1101	6D	-
10 1110	6E	>
10 1111	6F	?
11 0000	F0	0
11 0001	F1	1
11 0010	F2	2
11 0011	F3	3
11 0100	F4	4
11 0101	F5	5
11 0110	F6	6
11 0111	F7	7
11 1000	F8	8
11 1001	F9	9
11 1010	7A	:
11 1011	7B	#
11 1100	7C	@
11 1101	7D	'
11 1110	7E	=
11 1111	7F	“
1000 0000	80	
1000 0001	81	a
1000 0010	82	b
1000 0011	83	c
1000 0100	84	d
1000 0101	85	e
1000 0110	86	f
1000 0111	87	g
1000 1000	88	h
1000 1001	89	i
1000 1010	8A	
1000 1011	8B	

Binary to EBCDIC Values (continued)

Bits 0-7	EBCDIC Code	Display Graphic
1000 1100	8C	
1000 1101	8D	
1000 1110	8E	
1000 1111	8F	
1001 0000	90	
1001 0001	91	j
1001 0010	92	k
1001 0011	93	l
1001 0100	94	m
1001 0101	95	n
1001 0110	96	o
1001 0111	97	p
1001 1000	98	q
1001 1001	99	r
1001 1010	9A	
1001 1011	9B	
1001 1110	9C	
1001 1101	9D	
1001 1110	9E	
1001 1111	9F	
1010 0000	A0	
1010 0001	A1	
1010 0010	A2	s
1010 0011	A3	t
1010 0100	A4	u
1010 0101	A5	v
1010 0110	A6	w
1010 0111	A7	x
1010 1000	A8	y
1010 1000	A9	z

C Native Sockets

The Intermec® Application Server or gateway provides two-way message services for the wireless network but it does not generate the messages. These come from a host program running independently of the server or gateway. In fact, the host application runs on a separate physical host.

Norand® Native is a simple command language for controlling computers from a host application. Native sockets enables a host computer running a Native application to communicate with Native computers through the server or gateway via a TCP/IP socket connection.

The server or gateway supports Native sockets applications for Enterprise Wireless LAN® systems. Contact your Intermec Representative for information about the version that provides Native sockets support.

If you are converting to the Intermec Application Server from the 6950 Enterprise Gateway Server, the Wireless Network Access Server, or a serial controller, see the *Intermec Application Server User's Guide* (P/N 072242). The manual contains information about modifying your host application to use a socket interface on the Intermec Application Server.

Communication Basics

You can write Native sockets programs for any host or workstation supporting Ethernet and TCP/IP. Initiate communication between the host and the server or gateway by configuring both with a common port number. The server or gateway uses the designated TCP port number to communicate with the host.

You must meet two requirements to set up communication:

1. Configure the computers and the server or gateway with a common host name, which defines the communication sessions.
2. Configure each computer for the Native data stream and assign it a unique computer number.

As each computer powers on, the host application opens a unique socket between the server or gateway and the host for the computer. The server or gateway keeps an internal table mapping computers to socket numbers. However, computer numbers are *not* available to the application. It must use socket numbers to identify each computer session.

Sockets remain open until the host application closes the session or someone powers the computer off and then on again, initiating a new session.

Host Programming

Software developers must be familiar with sockets programming to use the Native sockets capability. Intermec does not provide programming support for internet stream sockets. A good reference on sockets programming is *UNIX Network Programming* by W. Richard Stevens, ISBN 0-13-94876-1.

Coding the Application

See “Native Programming” on page 240 for details about the Native command syntax. The application reads and writes directly to the computers through the server or gateway.

The Native protocol was originally written for early model INTERMEC controllers that supported serial host communications. Addition of the sockets communication capabilities to the server or gateway required use of a modified Native syntax. The following table provides a cross reference. Letter conventions are excerpted from “Native Programming” on page 240.

All commands terminate with a carriage return <cr>. The server or gateway implementation maintains the 1024-character maximum message length limitation.

Abbreviated Commands

Command	Server or Gateway Native Syntax	Original Syntax
Set Terminal Parameters	eD*o*/*c*	S*t*D*o*/*c*
Sound Audio Annunciator	B*d*	W*t*B*d*
Write Display *	D/	W*t*D/
	D/	W*t*D/
Reset Terminal	g	G*t*
Terminal Firmware Version	v	D*t*V
Echo Back Diagnostic	Not supported	D*t*E/
** All "D/" and "D/ command extensions are supported except Send Multiple Write Displays, "D//".*		

Note: The Intermec Application Server does not support the DtE (Echo-Back Diagnostic) command in a Native socket host.

Response Formats

Response formats for each command are listed in "Native Programming" on page 240. However, with Native sockets, the computer numbers are not returned in the response string.

For example, a keyboard response to a "D*o*/*x*"<cr> command is not in the form "*t*K*rd*<cr>". A response with data from a bar code scanner is "S*rbcm*<cr>". For descriptions of the parameters, see the "Write to Display (WTD)" on page 224.

Error Messages

Native command syntax is processed partly by the server or gateway, and partly by the computer. For example, the WtD/ (Write Display) command has an options list *o* and message content *x* in the form "D*o*/*x*<cr>". The server or gateway processes content preceding the forward slash, and the computer processes content after the slash.

Error responses from the server or gateway take the form "?*x*<cr>", where *x* is a numeric error code from 0 to 11. For more information, see "Write Display Errors" on page 256.

The computer's error responses depend on the specific command. For the command that produced the error, see "Native Programming" on page 240.

Normal Communication

Each computer is identified by its number, and a separate host session is initiated for communication with it. This works regardless of the operating mode of the computer (terminal emulation or Native).

Coding the Application

Code does not need to be designed to handle multiple computers. If two operators select the same application number, a copy of the application runs for each computer.

Syntax checking is not done on messages transmitted through the server or gateway to the computer. Check the response to an asynchronous command for error notifications from the server or gateway.

Syntax for Computers

For details about Norand Native syntax, see “Native Programming” on page 240.

WtD/ (Write Display) Command

You may use any WtD/ command specified in “Native Programming” on page 240. Intermec recommends that you familiarize yourself with these commands before writing any code.

For WtD/ commands, the communication driver in the server or gateway adds the “W.” If the “W” is included here, the message is rejected.

The server or gateway allows three additional abbreviated Native commands. The following table shows the commands you can send in Native mode, and provides a description of the normal computer responses.

Abbreviated Computer Commands

Feature	Description
“e” or “s” = StD	Sends the “St” command. You need to send from the “D” on when issuing this. For example, if you were transmitting to computer 002, sending “eD” would send “S002D”. The next response would be “CO”.
“v” = DtV	Sending a “v” requests the version number of the computer. That is, sending a “v” issues a DtV command to the computer. The next message from that computer should be its software version and computer number. Internet socket applications return only a version number.
“g” = G or Gt	Sending a “g” resets the computer. The next message from that computer should be a power-up message.

Responses received from the computers follow the responses listed in “Native Programming” on page 240. In addition, the server or gateway can return codes in the computer's number field. They are explained under “Native Mode Responses” on the next page.

Native Mode Responses

The server or gateway sends the following codes in the computer number field returned to the application.

Codes in Computer Number Field

Command	Server or Gateway Native Syntax
`800`	No messages are available for the application. The application can perform housekeeping tasks or go back to waiting for a message (same as `NUI_TIMEOUT`).
`998`	The computer has not passed a message for a predetermined length of time. The application can either ignore the message or act on it. For example, this code could be returned if the computer is left unattended with the application running (same as `NUI_DEAD`).
`999`	The server or gateway wants it to terminate. If the application does not terminate within 15 seconds, it is terminated automatically. The application receives this message when the server or gateway shuts down or is reset, or if the server or gateway detects an error (same as `NUI_STOP`).
`NUI_OK`	The server or gateway completion was normal.
`NUI_STOP`	The server or gateway wants to terminate (same as 999 code).
`NUI_DEAD`	The associated computer has not passed a message for a predetermined length of time (same as 998 code).
`NUI_TIMEOUT`	No data available from the application to be read from the server or gateway (same as 800 code).
`NUI_BAD`	A fatal error occurred, which prevents continued execution.
`NUI_BOOT`	The associated computer is powered on and can be used to reload data into the computer. Or, it can be ignored.

Program interfaces return the computer number and the server, gateway, or computer response. If one of these codes is not set, the field contains the number of the computer.

Configuration

To use Native sockets, you must configure IP, RF communication, and computer parameters for the server, gateway, and computer. To configure these parameters, refer to the device's user manual for instructions.

When everything is configured correctly and the computer is powered on, a unique connection (socket) is established between the computer and your sockets-based Native emulation program.

Commands to the computers must be in the appropriate syntax for their Native mode. Ignore the parameter and diagnostic commands.

The following WtD/ (Write Display) command is a sample startup menu for a computer with viewing screen size set to 16.

`DBCDNL1H2P15/01 - STDIO TEST 02 - DIRECT TEST`

Note that this message does not have the “W” at the beginning, nor does it contain the computer number. The communication driver in the server or gateway adds the “W” and the computer number.

Demonstration Program

Following is a “C” demonstration host program for Native sockets. The program uses internet sockets to send Native mode syntax to an Intermec® computer via the server or gateway. The program sends key presses made by a computer user back to their display.

```
/****************************************************************
*****************************************************************
**                                                             **
**  nui_sock.c                                                 **
**                                                             **
** This is a demo program that uses sockets to send/receive data**
** to/from a Native Mode terminal via the nui_tlnt redirector   **
** application.                                                **
**                                                             **
**  Usage:  ./nui_sock <portno>                                **
**                                                             **
*****************************************************************
****************************************************************/
#include <stdio.h>
#include <errno.h>
#ifdef WIN32
#include <windows.h>
#define  FD_SETSIZE  256   /* set to number of sockets to open */
#include <winsock.h>
#else
#include <sys/time.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
                    #endif
#define  VERSION  "1.00"
unsigned int   listen_fd;
unsigned short ipport;
char work_buf[1024];
char tdata[4096];
unsigned char param[]="eD/@  OG00";
int shut_down=0;   /* true if program shutdown requested */
struct sockaddr_in  cli_addr;
struct hostent      host_rec;
fd_set read_fd_set;
fd_set main_fd_set;
/*
**  linked list management stuff
*/
struct LIST {
        unsigned int    fdin;
        unsigned int    fdout;
        char            host[32];
        char            ip[32];
        char            sic[8];
        unsigned short  port;
        unsigned        msgcount;
```

```
        struct LIST     *next;
};
struct  LIST    *head;  /* first pointer in linked list */
struct  LIST    *tail;  /* last pointer in linked list */
struct  LIST    *curr;  /* always points to current socket pointer */
int  main(int argc,char **argv );
int  process_port(void);
int  process_socket(void);
int  add_fd(unsigned int fd,char *host,char *ip);
int  delete_fd(unsigned int fd);
/*****/
main(argc, argv)
int argc;
char *argv[];
    {
    struct sockaddr_in  serv_addr;
    int x=0;
#ifdef WIN32
    WSADATA WsaData;
#endif
/*
**  check the command line arguments
*/
    if(argc < 2)
        {
        printf("nui_sock: Wrong number of arguments.");
        printf("nui_sock: Usage: nui_sock <portno>");
        fflush(stdout);
        exit(1);
        }
/*
**  get the port number
*/
    strcpy(work_buf,argv[1]);
    if(strlen(work_buf) < 1)
        {
        printf("nui_sock: Invalid portno! Must be numeric and > 0.");
        printf("nui_sock: Usage: nui_sock <portno>");
        fflush(stdout);
        exit(1);
        }
    ipport=atoi(work_buf);
/*
** set up curses
*/
    for(x=0;x<53;x++)
        {
        printf("");
        }
    fflush(stdout);
    printf("nui_sock: Native Socket Demo - Version %s ",VERSION);
    fflush(stdout);
/*
**  startup sockets
*/
#ifdef WIN32
    x = WSAStartup (0x0101, &WsaData);
    if(x==SOCKET_ERROR)
        {
```

```
        printf("nui_sock: can't init winsock interface %d.",errno);
        fflush(stdout);
        exit(1);
        }
#endif
/*
* Open a TCP socket (an Internet stream socket).
*/
    FD_ZERO(&main_fd_set);
    if((listen_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
        {
        perror("nui_sock: can't open local socket");
        exit(1);
        }
/*
* Fill in the structure "serv_addr" with the address of the
* server that we want to connect with.
*/
    memset(&serv_addr,0x00,sizeof(serv_addr));
    serv_addr.sin_family      = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port        = htons(ipport);
    if (bind(listen_fd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
        {
        perror("nui_sock: can't bind to local socket");
        exit(1);
        }
    listen(listen_fd, 5);
/*
**  add this fd to our work list so we know when clients connect
*/
    FD_SET(listen_fd, &main_fd_set);
/*
**  we are connected, lets see if this thing can work
*/
    printf("nui_sock: Listen on Port %d",ipport);
    fflush(stdout);
    process_port();
/*
**  shut down the system
**  (traverse the "connected" list to shutdown each terminal)
*/
    curr = head;
    while(curr != NULL)
        {
        /* this will shutdown and reset the terminal */
        delete_fd(curr->fdin);
        curr = curr->next;
        }
    shutdown(listen_fd,2);
    close(listen_fd);
    printf("nui_sock: System Shutdown.");
    fflush(stdout);
    exit(0);
    }
/*****/
/*
**  main function loop for handling the socket calls from the
```

```
**  cradle server
*/
int process_port()
    {
    int hdrcnt=0;
    int x=0;
    int clilen=0;
    struct timeval seltime;
    int gotdata=0;
    struct hostent *host_rec;
    char host[32];
    char *ip;
    int sockfd;
    int incount=0;
/*
**  use the select to see if anything is waiting
*/
    for(;;)
        {
        if (shut_down)
            {
            break;  /* user requested shutdown, so goodbye */
            }
        memcpy((void *)&read_fd_set, (void *)&main_fd_set, sizeof(fd_set));
        seltime.tv_usec = 0;
        seltime.tv_sec = 15;  /* 15 second timeout */
        gotdata=select(FD_SETSIZE,&read_fd_set,0,0,&seltime);
        if(gotdata > 0)
            {
/*
**  it is a new connection, connect it
*/
            if(FD_ISSET(listen_fd,&read_fd_set))
                {
                 clilen = sizeof(cli_addr);
                 sockfd = accept(listen_fd,(struct sockaddr
*)&cli_addr,&clilen);
                if(sockfd < 0)
                    {
                    printf("nui_sock: Accept error. Port=%d
(%d)",ipport,errno);
                    fflush(stdout);
                    }
                 ip=inet_ntoa(cli_addr.sin_addr);
                 host_rec=gethostbyaddr((char *)&cli_addr.sin_addr,4,AF_INET);
                strcpy(host,host_rec->h_name);

                printf("nui_sock: HEARD <%s:%s>",host,ip);
                fflush(stdout);
                FD_CLR(listen_fd,&read_fd_set);
                 add_fd(sockfd,host,ip);  /* add the connection to our fd list
*/
                /* send terminal control parameters */
                strcpy(tdata,param);
                write(sockfd,tdata,strlen(tdata));
                sleep(1);  /* Wait, don't send the next one too fast */
                /* send first entry prompt */
                sprintf(work_buf,"DBSCQDEL0H16P16/%-16.16s%-16.16s","Enter
data now:"," ");
```

```
                strcpy(tdata,work_buf);
                write(sockfd,tdata,strlen(tdata));
                }
            else
                {
/*
**  look for an active descriptor and process accordingly
*/
                curr = head;
                while(curr != NULL)
                    {
                    if(FD_ISSET(curr->fdin, &read_fd_set))
                        {
                        if(!process_socket())
                            {
                            /* this will shutdown/reset the terminal */
                            delete_fd(curr->fdin);
                            curr = NULL;
                            break;
                            }
                        }
                    curr = curr->next;
                    }
                }
            }
        }
    return(1);
    }
/*
**  This function reads data from the socket (response from the
**  terminal) and writes to the socket (sends commands to the
**  terminal).
*/
int process_socket()
    {
    int datacnt=0;
/*
**  read the data from the socket
*/
    memset(tdata,0x00,sizeof(tdata));
    datacnt=read(curr->fdin,tdata,sizeof(tdata));
    if (datacnt < 1)
        {
        printf("nui_sock: Bad message. fd=%d,
datacnt=%d",curr->fdin,datacnt);
        fflush(stdout);
        return(0);
        }
/*
**  we have a data message, process it
*/
    if(tdata[0]=='K' && tdata[1]=='Q')   /* stop and reset this terminal */
        {
        printf("nui_sock: Terminal requested stop.");
        fflush(stdout);
        return(0);
        }
    if(tdata[0]=='K' && tdata[1]=='X')
        {
```

```
        /* Stop this application and reset all connected terminals */
        printf("nui_sock: Program stop requested.");
        fflush(stdout);
        shut_down = 1;
        return(0);
        }
    if(tdata[strlen(tdata) - 1] == 13)  /* don't forget to strip cr */
        tdata[strlen(tdata) - 1] = 0;
/*
** I'm not doing anything with the incomming data but sending it ** back, thus
I build my new output message from the input
** message
*/
   sprintf(work_buf,"DBSCQDEL0H16P16/%-16.16s%-16.16s%-16.16s%-16.16s","Enter
data now:"," ","Previous data:",&tdata[1]);
    strcpy(tdata,work_buf);
/*
**  write the data to the socket
*/
    write(curr->fdin,tdata,strlen(tdata));
    return(1);
    }
/*
**  This function adds a new socket File Descriptor to our
**  linked list of connected FDs (which are actually terminals).
*/
int add_fd(unsigned int fd, char *host, char *ip)
    {
    struct LIST *ptr;
    ptr = (struct LIST *)malloc(sizeof(struct LIST));
    if(ptr == NULL)
        {
        printf("nui_sock: add_fd Cannot allocate buf ptr.");
        fflush(stdout);
        return(-1);
        }
    ptr->fdin = fd;
    strcpy(ptr->ip,ip);
    strcpy(ptr->host,host);
    ptr->next = NULL;
    if(tail != NULL)
        {
        tail->next = ptr;
        tail=ptr;
        }
    else
        {
        head = ptr;
        tail = ptr;
        }
    FD_SET(fd, &main_fd_set);
    return(1);
    }
/*
**  This function removes a socket File Descriptor from our
**  linked list and sends a reset to the terminal to shut
**  it down properly through the nui_tlnt program.
*/
```

```
int delete_fd(unsigned int fd)
    {
    struct LIST *before;
    struct LIST *ptr;
    int foundflag = 0;
    ptr = head;
    before = NULL;
    while(ptr != NULL)
        {
        if(ptr->fdin == fd)
            {
            foundflag = 1;
            break;
            }
        before = ptr;
        ptr = ptr->next;
        }
    if(!foundflag)
        {
        printf("nui_sock: delete_fd Tried to delete an fd not in list!");
        fflush(stdout);
        return(-1);
        }
     FD_CLR(ptr->fdin, &main_fd_set);
     if(ptr->fdin > 0)
        {
/*
**  sending a 'g' will reset the terminal and close connection
**  with nui_tlnt
*/
        tdata[0]='g';   /*  reset this radio only  */
        tdata[1]=0;
        write(ptr->fdin,tdata,strlen(tdata));
        sleep(1);  /* don't shutdown the socket too fast */
        shutdown(ptr->fdin,2);  /* shutdown this socket connection*/
        close(ptr->fdin);
        }
    if(before != NULL)
        {
        before->next = ptr->next;
        if(ptr == tail)
            {
            tail = before;
            }
        }
    else
        {
        head = ptr->next;
        if(head == NULL)
            tail = NULL;
        }
    free(ptr);
    return(1);
    }
/*
*****  last line of program  */
```

I Index

Numerics

A

B

Worldwide Headquarters
6001 36th Avenue West
Everett, Washington 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

TE 2000 Terminal Emulation Programmer's Guide

P/N 977-055-009