Intermec

Programmer's Guide

Native Terminal Emulation

Intermec Technologies Corporation

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

Technical Communications Department
550 Second Street SE
Cedar Rapids, IA 52401
U.S.A.

www.intermec.com

The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and service Intermec-manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermec.

Information and specifications contained in this document are subject to change without prior notice and do not represent a commitment on the part of Intermec Technologies Corporation.

The word Intermec, the Intermec logo, Norand, ArciTech, CrossBar, Data Collection Browser, dcBrowser, Duratherm, EasyCoder, EasyLAN, Enterprise Wireless LAN, EZBuilder, Fingerprint, i-gistics, INCA (under license), InterDriver, Intermec Printer Network Manager, IRL, JANUS, LabelShop, Mobile Framework, MobileLAN, Nor*Ware, Pen*Key, Precision Print, PrintSet, RoutePower, TE 2000, Trakker Antares, UAP, Universal Access Point, and Virtual Wedge are either trademarks or registered trademarks of Intermec Technologies Corporation.

Throughout this manual, trademarked names may be used. Rather than put a trademark (™ or ®) symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement.

There are U.S. and foreign patents pending.

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Bluetooth is a trademark of Bluetooth SIG, Inc., U.S.A.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.opensssl.org/).

This product includes cryptographic software written by Eric Young (EAY@cryptsoft.com).

Document Change Record

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

Revision	Date	Description of Change
C	04/2003	Added Direct Connect port numbre, added Define Origin option, revised remap key information, revised error tone information, added 2435A 48-key keypad information, added 2435A 31/32 column bar code information, removed paging keys information.
D	10/2003	Revised Radio Comm information for Trakker Antares terminals, revised Security Opts menu option.

Contents

A Bar Code Scanning

Before You Begin

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

Safety Summary

Your safety is extremely important. Read and follow all warnings and cautions in this document before handling and operating Intermec equipment. You can be seriously injured, and equipment and data can be damaged if you do not follow the safety warnings and cautions.

Do not repair or adjust alone

Do not repair or adjust energized equipment alone under any circumstances. Someone capable of providing first aid must always be present for your safety.

First aid

Always obtain first aid or medical attention immediately after an injury. Never neglect an injury, no matter how slight it seems.

Resuscitation

Begin resuscitation immediately if someone is injured and stops breathing. Any delay could result in death. To work on or near high voltage, you should be familiar with approved industrial first aid methods.

Energized equipment

Never work on energized equipment unless authorized by a responsible authority. Energized electrical equipment is dangerous. Electrical shock from energized equipment can cause death. If you must perform authorized emergency work on energized equipment, be sure that you comply strictly with approved safety regulations.

Safety Icons

This section explains how to identify and understand dangers, warnings, cautions, and notes that are in this manual. You may also see icons that tell you when to follow ESD procedures and when to take special precautions for handling optical parts.

A warning alerts you of an operating procedure, practice, condition, or statement that must be strictly observed to avoid death or serious injury to the persons working on the equipment.

Avertissement: Un avertissement vous avertit d’une procédure de fonctionnement, d’une méthode, d’un état ou d’un rapport qui doit être strictement respecté pour éviter l’occurrence de mort ou de blessures graves aux personnes manupulant l’équipement.

A caution alerts you to an operating procedure, practice, condition, or statement that must be strictly observed to prevent equipment damage or destruction, or corruption or loss of data.

Attention: Une précaution vous avertit d’une procédure de fonctionnement, d’une méthode, d’un état ou d’un rapport qui doit être strictement respecté pour empêcher l’endommagement ou la destruction de l’équipement, ou l’altération ou la perte de données.

Note: Notes either 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 http://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 http://www.intermec.com to download our current manuals in PDF format. To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

Visit the Intermec technical knowledge base (Knowledge Central) at http://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
Factory Repair and On-site Repair	Request a return authorization number for authorized service center repair, or request an on-site repair technician.	1
Technical Support	Get technical support on your Intermec product.	2
Service Contract Status	Inquire about an existing contract, renew a contract, or ask invoicing questions.	3
Schedule Site Surveys or Installations	Schedule a site survey, or request a product or system installation.	4
Ordering Products	Talk to sales administration, place an order, or check the status of your order.	5

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 Guide?

This guide provides you with information about the Native terminal emulation application, and how to install, configure, operate, maintain, and troubleshoot the application.

Related Documents

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

Manual	Part #
1100 Series Data Terminal User's Guide	961-047-069
5020 Data Collection PC User's Manual	068975-002
5055 Data Collection PC User's Guide	961-054-017
5900 Series User's Guide	961-047-121
The Bar Code Book	051241
DCS 300 System Manual	067296
DCS 300 Terminal Reference Manual	067717
EZBuilder Getting Started Guide	06645
EZBuilder Tutorial	066449
MobileLAN ™access 21xx System Manual	067150
*PEN*KEY Model 6400 User's Guide*	961-047-098
RT1700 Radio Terminal User's Guide	961-047-068
TE 2000 3270 Terminal Emulation Programmer's Guide	977-055-003
TE 2000 5250 Terminal Emulation Programmer's Guide	977-055-004
TE 2000 VT/ANSI Terminal Emulation Programmer's Guide	977-055-005
TRAKKER Antares 241X Hand-Held Terminal User's Manual	069538
TRAKKER Antares 2420 and 2425 Hand-Held Terminal User's Manual	064024
TRAKKER Antares 243X Hand-Held Terminal User's Manual	071791-001
TRAKKER Antares 2455 Vehicle Mount Terminal User's Manual	067358
TRAKKER Antares 2475 Vehicle-Mount Terminal User's Manual	072383
TRAKKER Antares 248X Stationary Terminal User's Manual	066960
TRAKKER Antares Application Development Tools System Manual	064433
TRAKKER Antares Optical Link Adapter Quick Reference Guide	065826
TRAKKER Antares TD2400 Communications Dock Quick Reference Guide	065555
TRAKKER Antares TD2410 Communications Dock Quick Reference Guide	069552

The Intermec web site at http://www.intermec.com contains many of our documents that you can download in PDF format.

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

1 Getting Started

This chapter introduces the Native terminal emulation application.

Understanding Network Protocol Options

Native Terminal Emulation applications for the Enterprise Wireless LAN™ system use one of the following network protocol options:

- **UDP Plus**
  The terminal communicates with the host computer through the INTERMEC® Data Collection Server (DCS) 300 and an access point.
- **TCP/IP**
  The terminal communicates through an INTERMEC access point, which is directly connected to the host computer on an Ethernet or a token ring network.
- **WTP**
  The terminal communicates with the host computer through the DCS 300, INTERMEC access point, or other INTERMEC gateways.

For network configuration options, refer to your terminal's user manual.

Setting Up the Terminal and the Network

Before you can start using the Native Terminal Emulation application on your terminal, you need to do the following:

1. **Set up your terminal.**
   Set-up includes charging and installing the battery pack and turning on the terminal for the first time. For instructions, refer to your terminal's user manual.

   Note: Battery packs do not apply to all terminals. Vehicle mount and stationary terminals are powered via an external source.

2. **Configure your terminal and the network.**
   To use RF communications on the terminal, you need to:

   a. Configure the DCS 300 (UDP Plus or WTP), other INTERMEC gateway (WTP), or host (TCP/IP).

   b. Configure the access point. *This does not apply to the 2480/2481 Terminal, which contains an Ethernet NIC instead of a radio.*

   c. Configure the network parameters on the terminal.

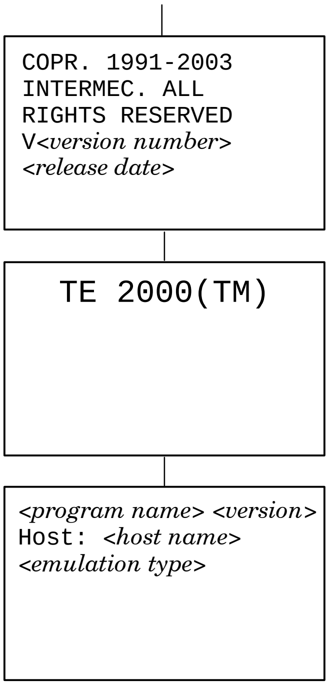
   For instructions, refer to your terminal's user manual.

3. **Verify that your terminal is communicating correctly with the access point, gateway, or host.**
   To verify that your terminal is communicating correctly, refer to the terminal's user manual for instructions.

Starting the Native Terminal Emulation Application

You are ready to start your application once the terminal has been set up, the terminal and the network have been configured, and communications have been established with the gateway and access point or host.

To start your application:
Turn on the terminal. Wait a few seconds while the initialization screens (below) clear and the application starts.



```
COPR. 1991-2003
INTERMEC. ALL
RIGHTS RESERVED
V<version number>
<release date>
```

```
   TE 2000(TM)
```

```
<program name> <version>
Host: <host name>
<emulation type>
```

Note: If your application does not start after a few seconds, you may not have configured the terminal correctly. For help, refer to your terminal's user manual.

You can now do one of the following:

- Become familiar with Native TE if you have not previously used it.
- Perform a quick configuration.
- Configure your Native TE application.
- Customize your Native TE application.

Becoming Familiar With Native Terminal Emulation

If you have not previously used Native TE, see Chapter 3, "*Using the Terminal's Keyboard*," to become familiar with your terminal's keyboard and the keys you need to press to perform Native commands. See Chapter 6, "*Terminal Commands*," to understand Native commands.

Performing a Quick Configuration

Note: For Terminal Emulation Version 6.60 or greater, the default data stream is “VT/ANSI.”

1 Change the data stream to Native. The default data stream is “Native” for the 6400 (WTP), 5055 (WTP), 11XX, 17XX, and 59XX Terminals. The default data stream for 2415, 2425, 2435A, 2455, 2475, 248X, and IP terminals is “3270.”

a Access the terminal emulation configuration menus by pressing the following keys, then choose **1) Set-up Parms** from the Main Menu.

Terminal	**Key Sequence**
2415	f F1 *(55-key keyboard)* f 5 *(37-key keyboard)*
2425	f F1
2435A	◨ ◧ *(57-key keyboard)* [Green] □ [Orange] □ *(48-key keyboard)* ◨ ◧ *(39-key function numeric keyboard)*
2455, 2475, 248X	f F1
6400	[Gold] [Blue]
5055	[Blue] [M] *or* [ALT] [M]
59XX	[Brown] [Space]
17XX, 11XX	[Gold] [Black]

Note: Press a number to select a menu option, then press [**Enter**] to return to a previous menu.

b At the **Enter Password** prompt, enter **cr52401**.
On 17XX (37-key) Terminals, the password is: ***F12 F11 5 2 4 0 1***

c From the **Set-up Parms** menu, select **3) Protocol Opts**, 2) **Data Stream**, then 7) **Native**.

2 From the **Main Menu**, select 7) **More**, then **2) Save Parms**.

3 At the **Enter Password** prompt, type **cr52401**, then press enter.
On 17XX (37-key) Terminals, the password is: ***F12 F11 5 2 4 0 1***

4 From the **Main Menu**, select **6) Exit Menus**.

Note: For WTP devices, you should also set a terminal number.

5 Log in to a TE session, then use the terminal to collect/transmit data.

Configuring the Native Application

You can use the terminal's TE configuration menus to configure site-specific operational parameters, including UDP Plus, WTP, or TCP/IP communications, terminal emulation options, and the **Main Menu** password. For information about configuring the terminal, see Chapter 3, "*Using the Terminal Emulation Menus.*"

Using Advanced Features

You can customize the standard Native TE program to do the following: For more information, see Chapter 4, "*Customizing Your Configuration.*"

- Use the auto-login feature to send the same login information each time you login to the host.
- Create a custom parameter set-up file.
- Change the text of TE configuration menus or system messages.
- Preinitialize the Native TE program

Unsupported Commands and Functions

Native TE for the 2415, 2425, 2435A, 2455, 2475, and 248X Terminals does not support the following commands and functions, which were supported in previous versions.

- **End** (viewport)
  Moved the window/viewport to the end of the last line displayed on the TE screen.
- **Erase Field**
  Erased all characters in the field where the cursor was positioned.
- **Erase Last Word**
  Erased all characters from the cursor back to the start of a word or a space on the screen.
- **Fast Cursor Right or Left**
  Moved the cursor two positions to the right or left rather than one.
- **Home** (viewport)
  Moved the window or viewport to the top left corner of the TE screen.
- **Reshow**
  Resent a screen image from the locally maintained host buffer to refresh the terminal screen.
- **Status**
  Toggled the terminal's screen between the status line display and the normal field input display.
- **Status line messages**
  Reported the operating status of the terminal and host system.

- **Dual sessions**
  Not supported in Trakker Antares UDP Plus or TCP terminals.

If you scan the bar code for an unsupported command, the bar code data is read into the terminal. If you press the key sequence for the command, the sequence is ignored.

Program Names

The following chart lists TE options and program names.

Note: Native TE does not support SST (Spread Spectrum Transmission) or regular UHF. "S-UHF" is synthesized UHF.

Model	Option	Program Name	
2415, 2425, 2435A, 2455, 2475, 248X	TE/WTP/2.4 GHz OpenAir	FWP240H0	
	TE/WTP/802.11		
	TE/UDP Plus/2.4 GHz OpenAir		
	TE/UDP Plus/802.11		
	TE/IP/2.4 GHz OpenAir		
	TE/IP/802.11		
		51-key keyboard	*41-key keyboard*
6400	TE/WTP/2.4 GHz OpenAir	FWP640H0	FWP640H4
	TE/WTP/802.11	FWP640H0	FWP640H4
	TE/WTP/900 MHz Falcon	FWP640H0	FWP640H4
	TE/IP/2.4 GHz OpenAir	FWP64TH0	FWP64TH4
	TE/IP/802.11	FWP64TH0	FWP64TH4
5055	TE/WTP/2.4 GHz OpenAir	FWP650H0	
	TE/WTP/802.11	FWP650H0	
	TE/IP/2.4 GHz OpenAir	FWP65TH0	
	TE/IP/802.11	FWP65TH0	
59XX	TE/WTP	FWP592H0	
	TE/WTP	FWP594H0	
	TE/WTP/2.4 GHz OpenAir	FWP598H0	
	TE/WTP/900 MHz Falcon	FWP596H0	
	TE/WTP/S-UHF	FWP591H1	
		57-key keyboard	*37-key keyboard*
17XX	TE/WTP	FWP170H0	FWP170H3
	TE/WTP	FWP174H0	FWP174H3
	TE/WTP/2.4 GHz OpenAir	FWP178H0	FWP178H3
	TE/WTP/900 MHz Falcon	FWP176H0	FWP176H3
	TE/WTP/S-UHF	FWP171H0	FWP171H3
11XX	TE/WTP	FWP110H0	
	TE/WTP	FWP114H0	
	TE/WTP/2.4 GHz OpenAir	FWP118H0	
	TE/WTP/900 MHz Falcon	FWP116H0	
	TE/WTP/S-UHF	FWP111H0	

2 Using the Terminal's Keyboard

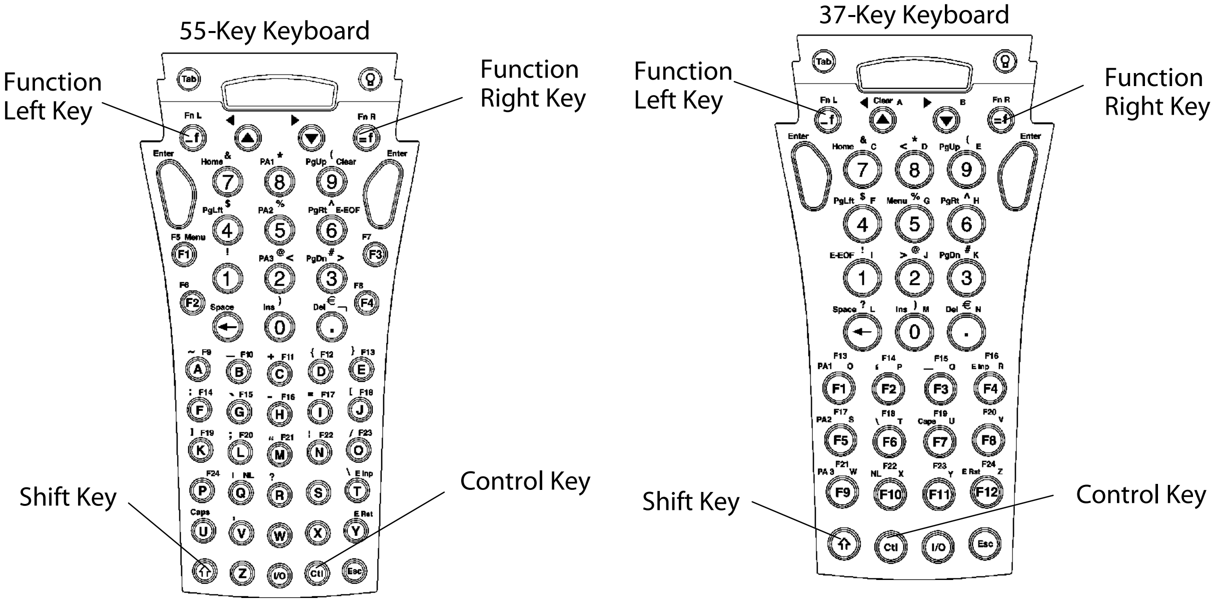
Your terminal has a special keyboard that contains most of the keys available on your Native terminal keyboard. Use the keyboard to Enter data in the TE screens.

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

You can access TE commands and functions printed above the keys by pressing key combinations.

2415 Terminal

For help with using the keyboard, refer to the *TRAKKER Antares 241X Hand-Held Terminal User's Manual* (P/N 069538).



Your 2415 Terminal has either a 55-key (left) or a 37-key (right) keyboard as shown in this illustration.

2415 Cursor Keys

To Enter	Press the Keys
Window/viewport up	▲
Window/viewport down	▼
Window/viewport right	_f ▼
Window/viewport left	_f ▲

2415 Tab Keys

To Enter	Press the Keys
Back Tab	_f Tab
Forward Tab	Tab

2415 AID-Generating Keys

	Press the Keys	
To Enter	**55-Key Keyboard**	**37-Key Keyboard**
Clear	f 9	⇧ ▲
F1	F1	F1
F2	F2	F2
F3	F3	F3
F4	F4	F4
F5	f F1	F5
F6	f F2	F6
F7	f F3	F7
F8	f F4	F8
F9	f A	F9
F10	f B	F10
F11	f C	F11
F12	f D	F12
F13	f E	⇧ F1
F14	f F	⇧ F2
F15	f G	⇧ F3
F16	f H	⇧ F4
F17	f I	⇧ F5
F18	f J	⇧ F6
F19	f K	⇧ F7
F20	f L	⇧ F8
F21	f M	⇧ F9
F22	f N	⇧ F10
F23	f O	⇧ F11
F24	f P	⇧ F12

2415 Auto-Login Restart

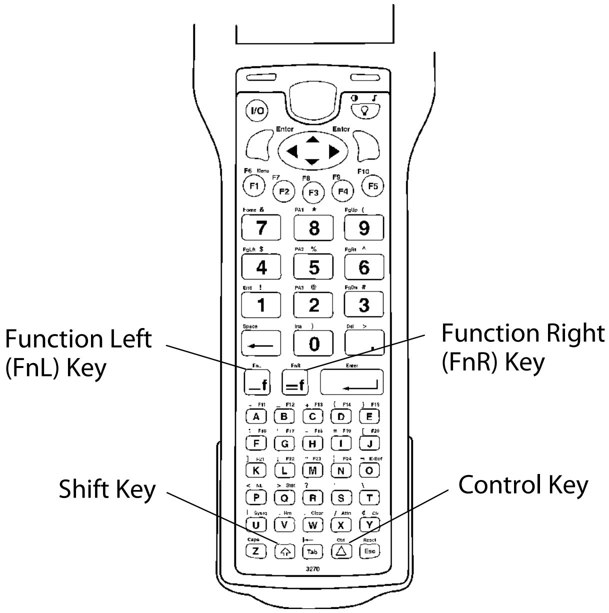
To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

2425 Terminal

For help with using the keyboard, refer to the *TRAKKER Antares 2420 and 2425 Hand-Held Terminal User's Manual* (P/N 064024).



This illustration shows the keyboard for the 2425 Terminal.

2425 Cursor Keys

To Enter	Press the Keys
Window/viewport up	_f ▲
Window/viewport down	_f ▼
Window/viewport right	_f ▶
Window/viewport left	_f ◀

2425 Tab Keys

To Enter	Press the Keys
Back Tab	_f Tab
Forward Tab	Tab

2425 AID-Generating Keys

To Enter	Press the Keys
Clear	f W
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	f F1
F7	f F2
F8	f F3
F9	f F4
F10	f F5
F11	f A
F12	f B
F13	f C
F14	f D
F15	f E
F16	f F
F17	f G
F18	f H
F19	f I
F20	f J
F21	f K
F22	f L
F23	f M
F24	f N

2425 Auto-Login Restart

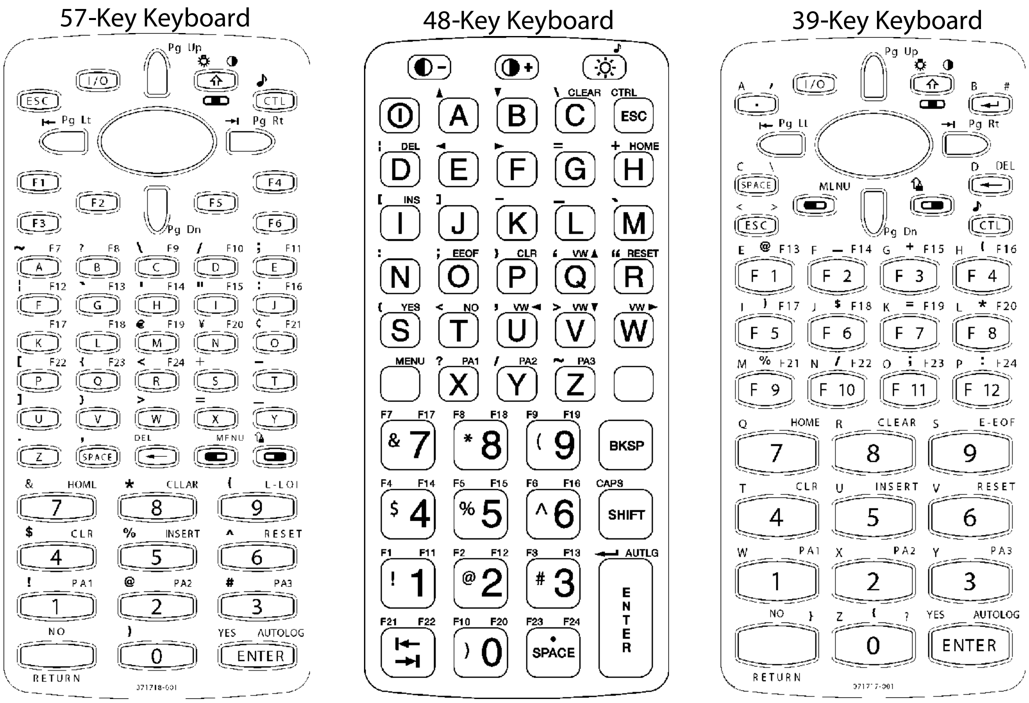
To Enter Auto-Login Restart, press f F5 or scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

2435A Terminal

For help with using the keyboard, refer to the *TRAKKER Antares 243X Hand-Held Terminal User's Manual* (P/N 071791-001).



Your 2435A Terminal has either a 57-key (left), a 48-Key (middle), or a 39-key (right) keyboard as shown in this illustration.

2435A Cursor Keys

	Press the Keys		
To Enter	**57-Key Keyboard**	**48-Key Keyboard**	**39-Key Function Numeric Keyboard**
Window/viewport up	▬ ▯	[Orange] □ A	▬ ▯
Window/viewport down	▬ ▯	[Orange] □ B	▬ ▯
Window/viewport right	▬ ▭	[Orange] □ F	▬ ▭
Window/viewport left	▬ ▭	[Orange] □ E	▬ ▭

2435A Standard Keys

	Press the Keys		
To Enter	**57-Key Keyboard**	**48-Key Keyboard**	**39-Key Function Numeric Keyboard**
0–9	0 – 9	0 – 9	0 – 9
Symbols	⇧ plus corresponding key	⇧ plus corresponding key	⇧ plus corresponding key

2435A Function Keys

	Press the Keys		
To Enter	**57-Key Keyboard**	**48-Key Keyboard**	**39-Key Function Numeric Keyboard**
Backspace	←	BKSP	←
Caps Lock	◧ ◨	[Orange] □ SHIFT	◧ ◨
Ctrl	Ctl	Not supported.	Ctl
Forward Tab	◨ □	→	◨ □
Return	ENTER	↵	ENTER
Shift	⇧	SHIFT	⇧
Space bar	SPACE	SPACE	SPACE

2435A AID-Generating Keys

	Press the Keys		
To Enter	**57-Key Keyboard**	**48-Key Keyboard**	**39-Key Function Numeric Keyboard**
F1	F1	[Orange] □ 1	F1
F2	F2	[Orange] □ 2	F2
F3	F3	[Orange] □ 3	F3
F4	F4	[Orange] □ 4	F4
F5	F5	[Orange] □ 5	F5
F6	F6	[Orange] □ 6	F6
F7	▬ A	[Orange] □ 7	F7
F8	▬ B	[Orange] □ 8	F8
F9	▬ C	[Orange] □ 9	F9
F10	▬ D	[Orange] □ 0	F10
F11	▬ E	[Green] □ 1	F11
F12	▬ F	[Green] □ 2	F12
F13	▬ G	[Green] □ 3	▬ F1
F14	▬ H	[Green] □ 4	▬ F2
F15	▬ I	[Green] □ 5	▬ F3
F16	▬ J	[Green] □ 6	▬ F4
F17	▬ K	[Green] □ 7	▬ F5
F18	▬ L	[Green] □ 8	▬ F6
F19	▬ M	[Green] □ 9	▬ F7
F20	▬ N	[Green] □ 0	▬ F8
F21	▬ O	[Orange] □ →	▬ F9
F22	▬ P	[Green] □ →	▬ F10
F23	▬ Q	[Orange] □ SPACE	▬ F11
F24	▬ R	[Green] □ SPACE	▬ F12
Clear	▬ 8	[Green] □ P	▬ 8

2435A Auto-Login Restart

To Enter Auto-Login Restart, press ▬ ENTER on the 57-key keyboard and 39-key function numeric keyboards; press [Green] □ ↵ on the 48-key keyboard, or scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

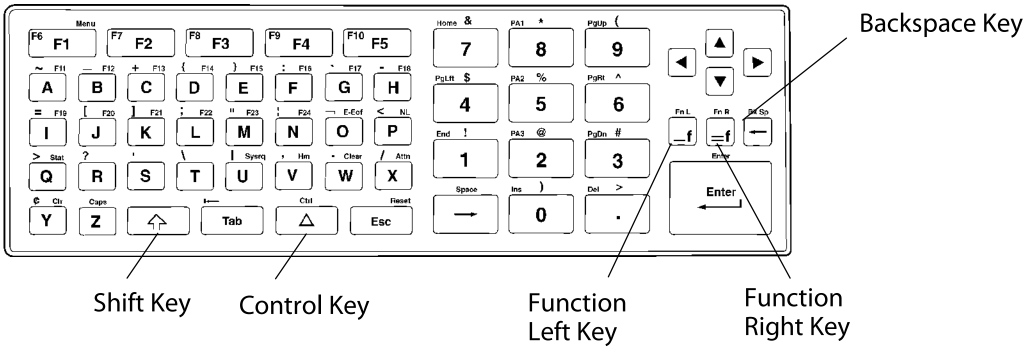
2435A 3270 Additional Functions

To access the TE configuration menus, press ◨ ◧ on the 57-key keyboard and 39-key function numeric keyboards, or press [Green] ▢ [Orange] ▢ on the 48-key keyboard.

2455 Terminal

For help with using the keyboard, refer to the *TRAKKER Antares 2455 Vehicle-Mount Terminal User's Manual* (P/N 067358).

Note: You must use the 2455 keyboard (P/N 067028) with the TE applications.



This illustration shows the keyboard for the 2455 Terminal.

2455 Cursor Keys

To Enter	Press the Keys
Window/viewport up	_f ▲
Window/viewport down	_f ▼
Window/viewport right	_f ▶
Window/viewport left	_f ◀

2455 Tab Keys

To Enter	Press the Keys
Back Tab	_f Tab
Forward Tab	Tab

2455 AID-Generating Keys

To Enter	Press the Keys
Clear	=f W
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	_f F1
F7	_f F2
F8	_f F3
F9	_f F4
F10	_f F5
F11	=f A
F12	=f B
F13	=f C
F14	=f D
F15	=f E
F16	=f F
F17	=f G
F18	=f H
F19	=f I
F20	=f J
F21	=f K
F22	=f L
F23	=f M
F24	=f N

2455 Auto-Login Restart

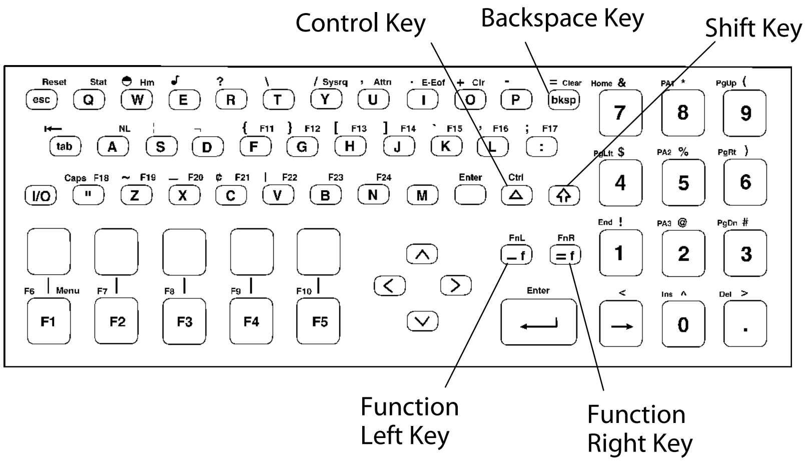
To Enter Auto-Login Restart, press =f F5 or scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

2475 and 248X Terminals

For help with using the keyboard, refer either to the *Trakker Antares 2475 Vehicle-Mount Terminal User's Manual* (P/N: 072383) or the *TRAKKER Antares 248X Stationary Terminal User's Manual* (P/N 066960).



This illustration shows the keyboard for the 2475 and 248X Terminals.

2475 and 248X Cursor Keys

To Enter	Press the Keys
Window/viewport up	_f ∧
Window/viewport down	_f ∨
Window/viewport right	_f >
Window/viewport left	_f <

2475 and 248X Tab Keys

To Enter	Press the Keys
Back Tab	_f tab
Forward Tab	tab

2475 and 248X AID-Generating Keys

To Enter	Press the Keys
Clear	f ⇦
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	f F1
F7	f F2
F8	f F3
F9	f F4
F10	f F5
F11	f F
F12	f G
F13	f H
F14	f J
F15	f K
F16	f L
F17	f :
F18	f [quotation mark]
F19	f Z
F20	f X
F21	f C
F22	f V
F23	f B
F24	f N

2475 and 248X Auto-Login Restart

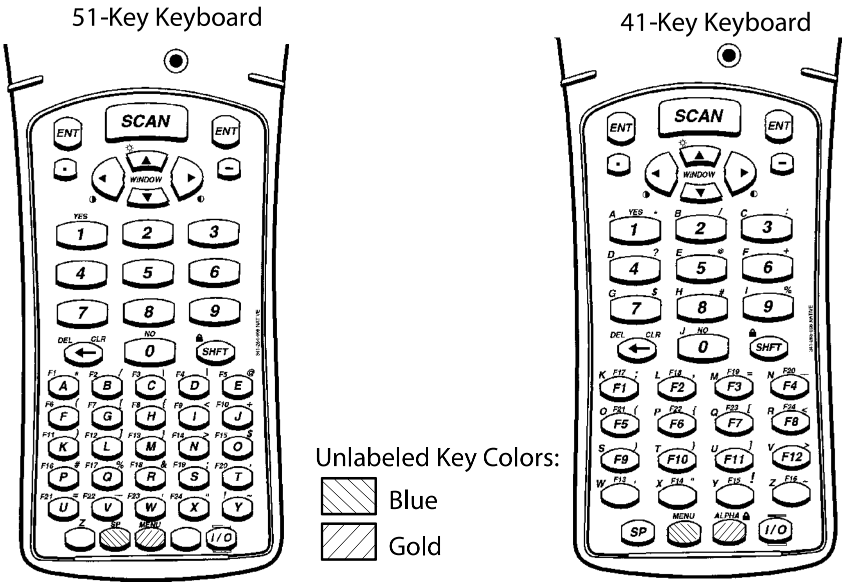
To Enter Auto-Login Restart, press f F5 or scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

6400 Computer

For help with the keyboard, see the *PEN*KEY Model 6400 User's Guide* (P/N 961-047-093).



Your 6400 Computer has either a 51-key (left) or a 41-key (right) keyboard as shown.

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

- **[Green]**
  The green [SHFT] puts the keyboard into green shift mode.
- **[Gold]** *(gold shift mode)*
  Press [Gold] plus a key to type a character or do an operation printed in gold on the overlay.
- **[Blue]** *(blue shift mode)*
  Press [Blue] plus a key to do operations printed in blue on the overlay.

The 51-key keyboard has two gray, unlabeled keys in the bottom row of the keyboard. Following are the keys' functions when they are unshifted (in the primary plane):

- Gray key with "Z" printed above it is the letter Z.
- Gray key with "SP" printed above it is the Space key ([SP]), which types one space.

Using the 41-Key Keyboard

The 41-key keyboard has numeric keys in its primary plane. It has alphabetic keys, special characters, and function keys in its secondary plane.

Because a 6400 Computer with a 41-key keyboard does not have alphabetic keys, follow these procedures to change passwords and cold-start the computer:

- Press [Blue] [3] or [F12] [3] to Enter a "C," [Blue] [F8] or [F11] [3] to Enter an "R," then press "52401" to Enter the password for the Set-up Parms in the terminal emulations menu. *Note that these are not case-sensitive.*

You can use one of two methods to type letters on the 41-key keyboard: standard mode or alpha lock mode. When engaged, alpha lock mode switches the alphabetic keys with the function keys. That is, it moves lowercase alphabetic keys from their normal [Blue] plane to the primary plane. It moves uppercase alphabetic keys from their standard [SHFT] [Blue] plane to the [SHFT] plane. Alpha lock provides a faster way to type a series of letters because it reduces the number of key combinations. *Alpha lock mode moves only the alpha keys to the primary plane.*

To engage alpha lock mode press [Blue] [Gold]. Then, to type a series of letters, press the correct key combination. The keyboard stays in alpha lock mode until you press [Blue] [Gold] to unlock it.

6400 Tab Keys

To Enter	Press the Keys
Back Tab	[Gold] [◀]
Forward Tab	[Gold] [▶]

6400 Function Keys

To Enter	Press the Keys
Clear	[Blue] [7]
F1	[Blue] [A]
F2	[Blue] [B]
F3	[Blue] [C]
F4	[Blue] [D]
F5	[Blue] [E]
F6	[Blue] [F]
F7	[Blue] [G]
F8	[Blue] [H]
F9	[Blue] [I]
F10	[Blue] [J]
F11	[Blue] [K]
F12	[Blue] [L]
F13	[Blue] [M]
F14	[Blue] [N]
F15	[Blue] [O]
F16	[Blue] [P]
F17	[Blue] [Q]
F18	[Blue] [R]
F19	[Blue] [S]
F20	[Blue] [T]
F21	[Blue] [U]
F22	[Blue] [V]
F23	[Blue] [W]
F24	[Blue] [X]

When alpha lock mode is engaged on the 41-key keyboard, it switches the function keys with the alphabetic keys. That is, function keys normally in the primary plane ([**F1**] through [**F12**]) move to the [**Blue**] plane. Function keys normally in the [**SHFT**] plane ([**F13**] through [**F24**]) move to the [**SHFT**] [**Blue**] plane.

The following chart describes how to do function operations when the 41-key keyboard is in standard mode or alpha lock mode.

Note: [**F13**] through [**F16**] are the four gray keys in the second row from the bottom of the overlay.

	Press the Keys (41-Key)	
To Enter	**Standard Mode**	**Alpha Lock Mode**
F1	[F1]	[Blue] [F1]
F2	[F2]	[Blue] [F2]
F3	[F3]	[Blue] [F3]
F4	[F4]	[Blue] [F4]
F5	[F5]	[Blue] [F5]
F6	[F6]	[Blue] [F6]
F7	[F7]	[Blue] [F7]
F8	[F8]	[Blue] [F8]
F9	[F9]	[Blue] [F9]
F10	[F10]	[Blue] [F10]
F11	[F11]	[Blue] [F11]
F12	[F12]	[Blue] [F12]
F13	[SHFT] Gray key with "F13" printed above	[SHFT] [Gold] "F13" key
F14	[SHFT] Gray key with "F14" printed above	[SHFT] [Gold] "F14" key
F15	[SHFT] Gray key with "F15" printed above	[SHFT] [Gold] "F15" key
F16	[SHFT] Gray key with "F16" printed above	[SHFT] [Gold] "F16" key
F17	[SHFT] [F1]	[SHFT] [Blue] [F1]
F18	[SHFT] [F2]	[SHFT] [Blue] [F2]
F19	[SHFT] [F3]	[SHFT] [Blue] [F3]
F20	[SHFT] [F4]	[SHFT] [Blue] [F4]
F21	[SHFT] [F5]	[SHFT] [Blue] [F5]
F22	[SHFT] [F6]	[SHFT] [Blue] [F6]
F23	[SHFT] [F7]	[SHFT] [Blue] [F7]
F24	[SHFT] [F8]	[SHFT] [Blue] [F8]

To engage alpha lock mode, press [**Blue**] [**Gold**]. Then press the correct key combinations in the chart. The keyboard stays in alpha lock mode until you press [**Blue**] [**Gold**] again to unlock it.

6400 Auto-Login Restart

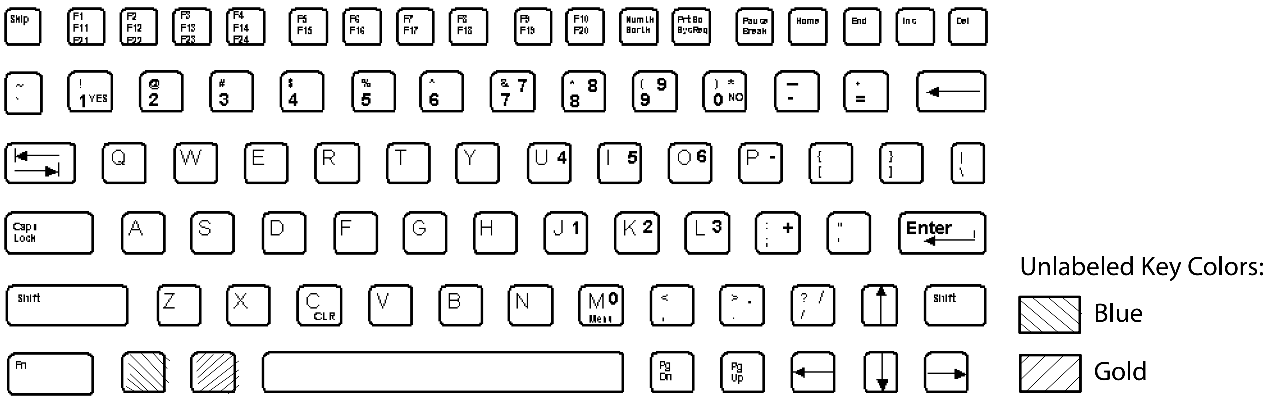
To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

5055 Data Collection PC

For help with using the keyboard, refer to the *5055 Data Collection PC User's Guide* (P/N 961-054-017).


Skip F1 F11 F21 F2 F12 F22 F3 F13 F23 F4 F14 F24 F5 F15 F6 F16 F7 F17 F8 F18 F9 F19 F10 F20 NumLk ScrLk PrtSc SysReq Pause Break Home End Ins Del
~ ` ! 1 YES @ 2 # 3 $ 4 % 5 ^ 6 & 7 7 * 8 8 (9 9) * 0 NO _ - + =
Q W E R T Y U 4 I 5 O 6 P - { [}] | \
Caps Lock A S D F G H J 1 K 2 L 3 : ; + " ' Enter
Shift Z X C CLR V B N M 0 Menu < , > . ? / / Shift
Fn Pg Dn Pg Up
Unlabeled Key Colors:
Blue
Gold

This illustration shows the keyboard for the 5055 Data Collection PC.

The special characters and functions printed on the overlay are color-coded to correspond with the matching shift keys. The shift keys are as follows.

- [**Shift**]
  Press [Shift] plus a letter to type the letter in uppercase.
- [**Blue**]
  The blue (Ctrl) key puts the keyboard into blue shift mode. Press the [Blue] key plus a keyboard key to do an operation printed in blue, or to send a control character.
- [**Gold**]
  The gold (Alt) key puts the keyboard into gold shift mode. Press the [Gold] key plus a keyboard key to do an operation printed in gold.
- [**NumLock**]
  The green (number lock) key puts the keyboard into [NumLock] mode. Press [NumLock] plus a keyboard key to type a number or character printed in green.

5055 Tab Keys

To Enter	Press the Keys
Back Tab	|←
Forward Tab	→ |

5055 Function Keys

To Enter	Press the Keys
Clear	[Gold] [C]
F1	[F1]
F2	[F2]
F3	[F3]
F4	[F4]
F5	[F5]
F6	[F6]
F7	[F7]
F8	[F8]
F9	[F9]
F10	[F10]
F11	[Blue] [F1]
F12	[Blue] [F2]
F13	[Blue] [F3]
F14	[Blue] [F4]
F15	[Blue] [F5]
F16	[Blue] [F6]
F17	[Blue] [F7]
F18	[Blue] [F8]
F19	[Blue] [F9]
F20	[Blue] [F10]
F21	[Gold] [F1]
F22	[Gold] [F2]
F23	[Gold] [F3]
F24	[Gold] [F4]

5055 Auto-Login Restart

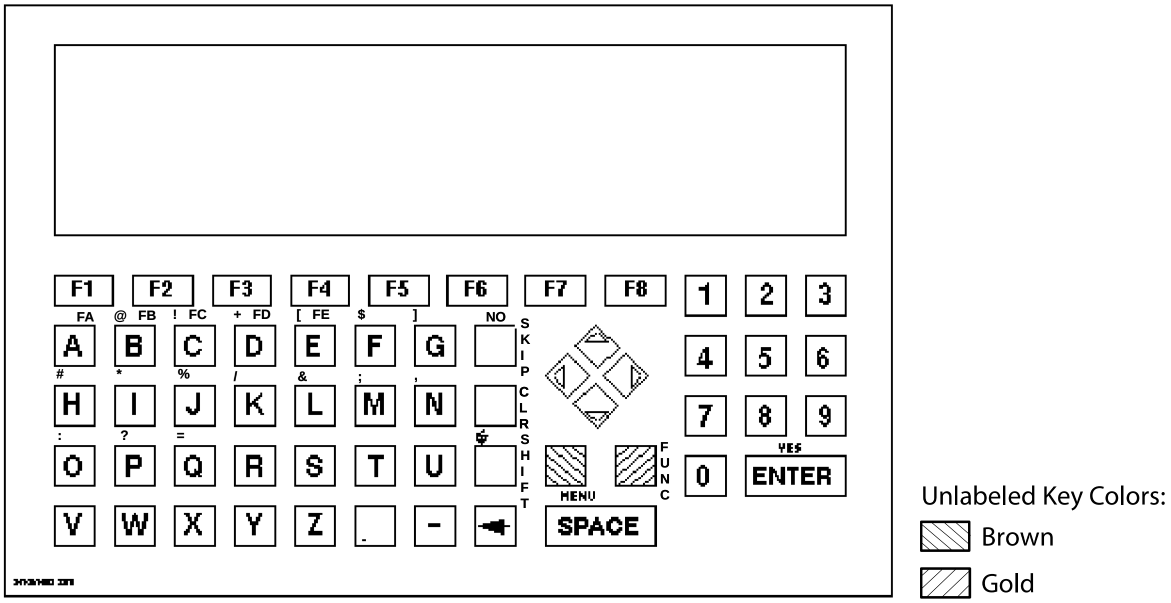
To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

59XX Terminal

For help with using the keyboard, refer to the *5900 Series User's Guide* (P/N 961-047-121).



This illustration shows the keyboard for the 59XX Terminal.

The special characters and functions printed on the overlay are color-coded to correspond with the matching shift keys. The shift keys are as follows.

- [**Shift**]
  Press this key plus a letter to type the letter in uppercase.
- [**Brown**]
  The brown key puts the keyboard into brown shift mode. Press the [**Brown**] key plus a keyboard key to do an operation printed in brown on the overlay.
- [**Gold**]
  The gold key puts the keyboard into gold shift mode. Press the [**Gold**] key plus a keyboard key to type a special character or do an operation printed in gold on the overlay.

The [**Insert**], [**CLR**], and [**Shift**] keys have their operations printed to the right of them. Operations printed to the right are the *unshifted* values. Operations printed above the keys are the *shifted* values.

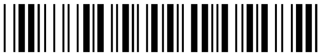
59XX Function Keys

To Enter	Press the Keys
Clear	[Gold] [.] (period)
F1	[F1]
F2	[F2]
F3	[F3]
F4	[F4]
F5	[F5]
F6	[F6]
F7	[F7]
F8	[F8]
F9	[Brown] [F1]
F10	[Brown] [F2]
F11	[Brown] [F3]
F12	[Brown] [F4]
F13	[Brown] [F5]
F14	[Brown] [F6]
F15	[Brown] [F7]
F16	[Brown] [F8]
F17	[Gold] [F1]
F18	[Gold] [F2]
F19	[Gold] [F3]
F20	[Gold] [F4]
F21	[Gold] [F5]
F22	[Gold] [F6]
F23	[Gold] [F7]
F24	[Gold] [F8]

59XX Auto-Login Restart

To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

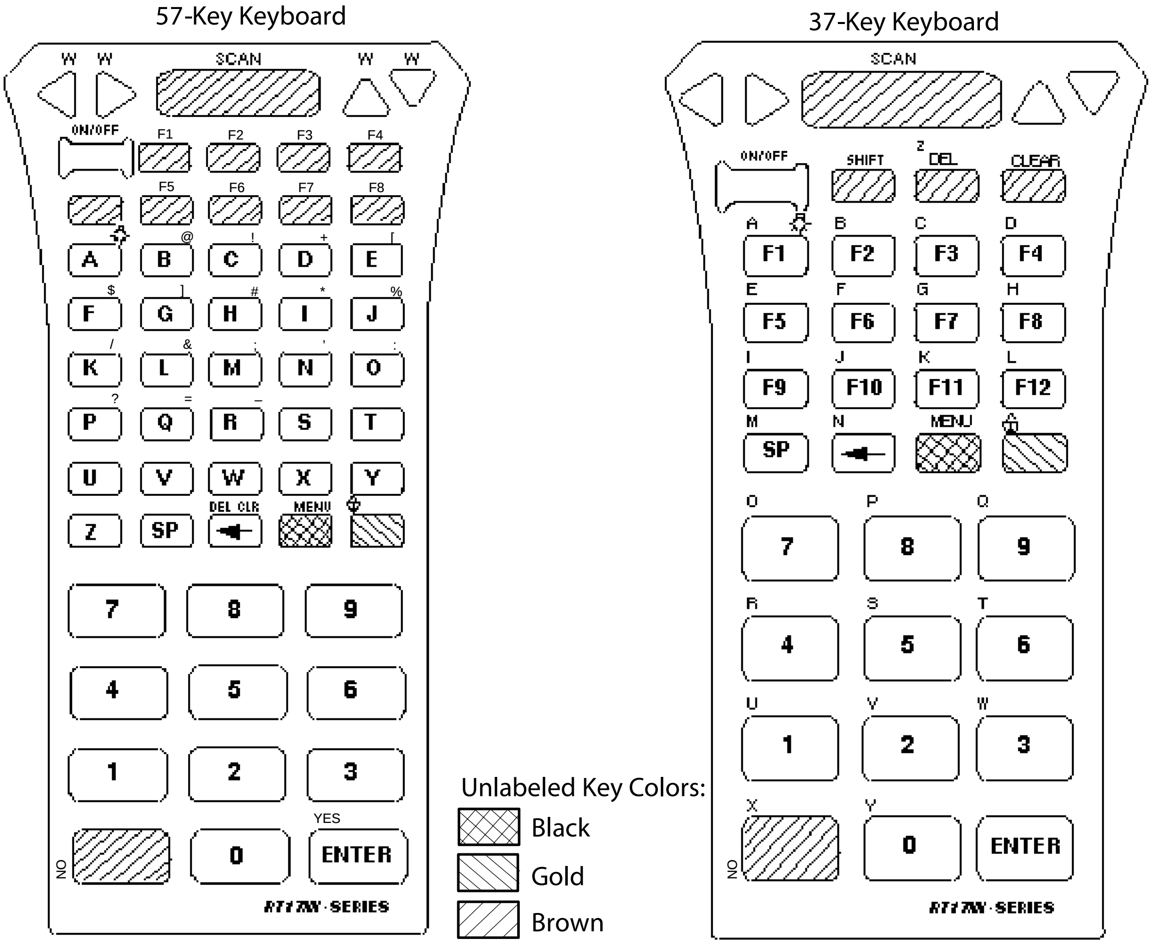
Auto-Login Restart



%ALRS

17XX Terminal

For help with using the keyboard, refer to the *RT17XX Radio Data Terminal User's Guide* (P/N: 961-047-068).



Your 17XX Terminal has either a 57-key (left) or a 37-key (right) keyboard as shown in this illustration.

The special characters and functions printed on the overlay are color-coded to correspond with the matching shift keys. The shift keys are as follows.

- **[Shift]**
  Press the [Shift] key plus a letter to type the letter in uppercase. The [Shift] key is located near the top of the keyboard.
- **[Gold]**
  This puts the keyboard into gold shift mode. Press [Gold] plus a keyboard key to type a character or do an operation printed in gold on the overlay.
- **[Black]**
  The Black key puts the keyboard into Black shift mode. Press [Black] plus a keyboard key to do an operation printed in Black on the overlay.

37-Key Keyboard

The 37-key keyboard has standard numeric keys, an [**Enter**] key, and user-defined function keys in its primary plane. It has alphabetic keys and special characters in its secondary plane.

Because a terminal with a 37-key keyboard does not have alphabetic keys in its primary plane, follow these procedures when using its firmware and downloading software to it. The following chart describes how to type letters on the 37-key keyboard.

- To access password-protected menus, press [**Gold**], [**Black**], [**F12**], [**F11**], then type "52401" for the password; or press [**Black**], [**F3**], [**Black**], then type "52401" for the password.
- To initiate the COLD START? menu option, press [**F10**] to answer "yes."
- To download software, hold down the [**F1**] key as you power up the terminal to go into download mode. This is similar to holding down the [**I**] key on the standard 57-key keyboard.

The following chart describes how to type letters on the 37-key keyboard.

To Enter	Press the Keys	To Enter	Press the Keys
a	[Black] [F1]	A	[Shift] [F1]
b	[Black] [F2]	B	[Shift] [F2]
c	[Black] [F3]	C	[Shift] [F3]
d	[Black] [F4]	D	[Shift] [F4]
e	[Black] [F5]	E	[Shift] [F5]
f	[Black] [F6]	F	[Shift] [F6]
g	[Black] [F7]	G	[Shift] [F7]
h	[Black] [F8]	H	[Shift] [F8]
i	[Black] [F9]	I	[Shift] [F9]
j	[Black] [F10]	J	[Shift] [F10]
k	[Black] [F11]	K	[Shift] [F11]
l	[Black] [F12]	L	[Shift] [F12]
m	[Black] [SP]	M	[Shift] [SP]
n	[Black] [🡐]	N	[Shift] [🡐]
o	[Black] [7]	O	[Shift] [7]
p	[Black] [8]	P	[Shift] [8]
q	[Black] [9]	Q	[Shift] [9]
r	[Black] [4]	R	[Shift] [4]
s	[Black] [5]	S	[Shift] [5]
t	[Black] [6]	T	[Shift] [6]
u	[Black] [1]	U	[Shift] [1]
v	[Black] [2]	V	[Shift] [2]
w	[Black] [3]	W	[Shift] [3]
x	[Black] [Return]	X	[Shift] [Return]
y	[Black] [0]	Y	[Shift] [0]
z	[Black] [.] (period)	Z	[Shift] [.] (period)

17XX Tab Keys

To Enter	Press the Keys
Back Tab	[◀]
Forward Tab	[▶]

17XX Function Keys

To Enter	**Press the Keys** (57-Key)
F1	[F1]
F2	[F2]
F3	[F3]
F4	[F4]
F5	[F5]
F6	[F6]
F7	[F7]
F8	[F8]
F9	[Black] [F1]
F10	[Black] [F2]
F11	[Black] [F3]
F12	[Black] [F4]
F13	[Black] [F5]
F14	[Black] [F6]
F15	[Black] [F7]
F16	[Black] [F8]
F17	[Brown] [F1]
F18	[Brown] [F2]
F19	[Brown] [F3]
F20	[Brown] [F4]
F21	[Brown] [F5]
F22	[Brown] [F6]
F23	[Brown] [F7]
F24	[Brown] [F8]

17XX Auto-Login Restart

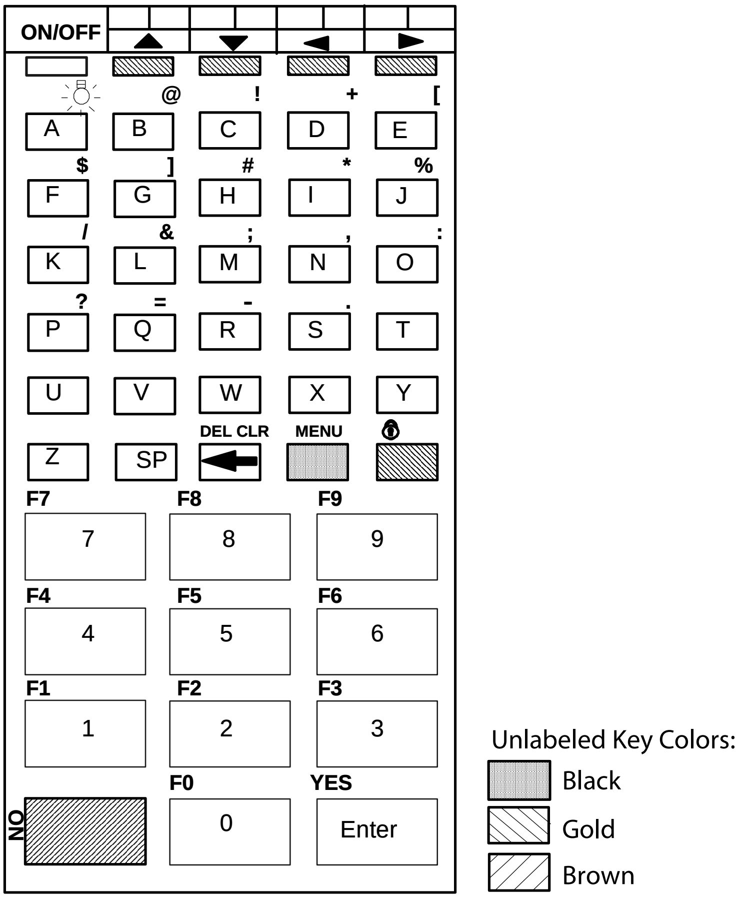
To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

11XX Terminal

For help with using the keyboard, see the *1100 Series Data Terminal User’s Guide* (P/N 961-047-069).



This illustration shows the keyboard for the 11XX Terminal.

The special characters and functions printed on the overlay are color-coded to correspond with the matching shift keys. The shift keys are as follows:

- [**Shift**] Press the brown [Shift] key plus a letter to type the letter in uppercase. The [Shift] key is located near the top of the keyboard.
- [**Gold**] The gold key puts the keyboard into gold shift mode. Press [Gold] plus a keyboard key to type a special character or do an operation printed in gold on the overlay.
- [**Black**] The Black key puts the keyboard into Black shift mode. Press [Black] plus a keyboard key to do an operation printed in Black on the overlay. To lock the keyboard into shift mode, press [**Black**] [**Gold**]. To unlock the keyboard, press [**Black**] [**Gold**] again.

The [**Return**] key has its operation printed to the left of it. The operation printed to the left is its *unshifted* value. The operation printed above [**Return**] is the *shifted* value.

11XX Tab Keys

To Enter	Press the Keys
Back Tab	|←
Forward Tab	→ |

11XX Function Keys

To Enter	Press the Keys
F0	[Black] [0]
F1	[Black] [1]
F2	[Black] [2]
F3	[Black] [3]
F4	[Black] [4]
F5	[Black] [5]
F6	[Black] [6]
F7	[Black] [7]
F8	[Black] [8]
F9	[Black] [9]
F10	[Brown] [0]
F11	[Brown] [1]
F12	[Brown] [2]
F13	[Brown] [3]
F14	[Brown] [4]
F15	[Brown] [5]
F16	[Brown] [6]
F17	[Brown] [7]
F18	[Brown] [8]
F19	[Brown] [9]
F20	[Brown] [U]
F21	[Brown] [V]
F22	[Brown] [W]
F23	[Brown] [X]
F24	[Brown] [Y]

11XX Auto-Login Restart

To Enter Auto-Login Restart, scan the following bar code (also in Appendix A, "*Bar Code Scanning*").

Auto-Login Restart

%ALRS

3 Using the Terminal Emulation Menus

This chapter lists ALL TE parameters. If a certain parameter does not apply to your terminal, the parameter will *not* appear in the TE configuration menus.

The CFGLIT.DAT file specifies the text of the TE configuration menus. This chapter assumes you are using the default settings in CFGLIT.DAT. To customize CFGLIT.DAT, see Chapter 4, "*Customizing Your Configuration*."

Function Keys

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

Enter Key

Press the terminal's [**Enter**] key to return to a previous TE configuration menu. Press [**Enter**] several times to return to the **Main Menu** from a submenu. This key also accepts the displayed or keyed input.

Terminals	Related [Enter] Keys
2415, 2425, 2455, 2475, 248X	⬯ or ⬯ keys
2435A	ENTER key *(57-key keyboard)* enter ↵ key *(48-key keyboard)* ENTER key *(39-key function numeric keyboard)*
6400	[**ENT**] keys
5055	**<Enter>** via external keyboard
59XX	[**ENTER**] key
17XX, 11XX	[**ENTER**] keys

Shift Keys

Use shift keys to put the keyboard in the desired shift mode. These shifted key functions are shown on the keyboard overlays in Chapter 2, "*Using the Terminal's Keyboard.*"

- *For 6400, 5055*
  [**Yellow**] and [**Blue**] shift keys are required. For the 5055 PC, the [**Yellow**] shift key substitutes the **<Alt>** key and the [**Blue**] shift key substitutes the **<Ctrl>** key on the external keyboard.
- *For 17XX, 11XX*
  [**Gold**] and [**Black**] shift keys are required.

Y ("Yes") Key

Several displays provide a warning that a certain action can cause your terminal 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.

Up and Down Arrows

For 6400, 5055, 11XX, the up and down arrow keys are defined by the host computer.

For 59XX, the up and down arrow keys can be made to function more efficiently, in many cases, by pressing the **FUNC** or **ALT** key, then pressing the desired arrow key.

Use these keys to:

- Adjust the length and frequency of the audible buzzer, key click, and error tone.
- Set the display screen size (the number of rows displayed and characters per row) and contrast.

Number Keys [0] through [9]

Many menus have numbered options. Press the corresponding numeric key 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 terminal. 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**] will cause the terminal to exit from a submenu or to reboot.

Display Annunciators

For 6400, 5055, 59XX, 17XX, 11XX, annunciators show the current status or operation in progress. Some of the more common annunciators are as follows:

Note: The following symbols appear larger than actual size on your terminal display. All individual letter annunciators are five pixels in height, except **T** (temperature) and **B** (bad battery) which are seven pixels.

Display Position 0

Radio *(6400, 5055)*
The radio is transmitting data to the base station.

Display Position 1

If the radio icon is displayed in position 0, then position 1 shows the status of the radio transmission (either receiving, sending, communication loss, or message waiting). If there is no radio icon shown in position 0, then this position is blank and position 1 shows the active session number 1 or 2.

Message
A message is received and has not been read.

Specific for 6400, 5055, 59XX Terminals

Sending
Radio is sending data to the base station.

Receiving
Radio is receiving data from the base station. This annunciator appears only when the information is for this particular computer's address.

Communication Loss
No communication with the host computer has occurred for at least 60 seconds. The radio computer may be out of radio range, the base station may not have power, or communications from the host computer to the base station may not be properly set up.

Specific for 17XX, 11XX Terminals

TX **Transmitting data**
The terminal is transmitting data to the host computer.

RX **Receiving data**
The terminal is receiving data.

CL **Communication Loss**
The terminal cannot communicate with the host computer. The terminal may be out of radio range, the base radio may not have power, or communication from the host to the base may not be properly set up.

Display Position 2

x **Input Inhibited**
The keyboard has accepted enough information for the current input field. If the “key ahead” feature is on, the terminal stores the keystrokes made after “input inhibited” appears. The host may also inhibit (lockout) the keyboard upon certain errors or when sending additional information to your computer.

^ **Insert Mode**
Characters are inserted, not overwritten.

Display Position 3

▲ **Alphabet Character Shift Mode** *(59XX, 17XX, 11XX, Green for 6400, 5055)*
The keyboard is in the alphabet character shift mode — any alphabet character keystrokes are entered as uppercase characters.

◀ **Shift Mode** *(Blue for 6400, 5055; Black for 11XX, 17XX; varies for 59XX)*
The keyboard is in a colored shift mode — keystrokes enter the symbol or perform the function shown on the overlay to the upper *left* of the key.

▶ **Shift Mode** *(Yellow or Gold for 6400, 5055, 11XX, 17XX; varies for 59XX)*
The keyboard is in a colored shift mode — keystrokes enter the symbol or perform the function shown on the overlay to the upper *right* of the key.

Display Position 4

A **Alpha Lock** *(6400)*
This symbol is used only for the 6400 Computer 41-key keyboard model.

C **Battery is charging** *(17XX, 11XX)*
The terminal is connected to a charging source. This does *not* necessarily mean the terminal is charging since this is dependent on battery state, temperature, and other conditions.

Display Position 5

E **Error** *(17XX)*
The terminal is connected to a charging source, but charging is *not* possible.

Display Position 6

B **Bad Battery** *(17XX)*
The battery pack you are currently using is bad.

T **Temperature** *(17XX)*
The terminal is currently out of the battery charging temperature range, which is 5° to 40° C.

Display Positions 1 through 3

S
C
A
N **Laser scanner in use** *(17XX, 11XX)*
Observe caution labels on laser scanners.

Display Positions 4 and 5

CL

Caps Lock *(5055)*
If [**Caps Lock**] is pressed (activated) on the 5055 external keyboard, the terminal will type all capital letters regardless of how the text is entered.

NL

Num Lock *(5055)*
If [**Num Lock**] is activated on the 5055 external keyboard, the terminal will enter numbers as pressed via the right-hand pad on the keyboard.

Display Positions 4 through 6

+ -

Low battery condition *(17XX, 11XX)*
A Low Battery indication is very important to the operation of the terminal, letting you know that your terminal will stop operating shortly. Your terminal should continue to operate 30 minutes to 2 hours depending on how you use it. Data stored in RAM will be saved for approximately 30 days. However, you should avoid testing this limit.

Your terminal will retain its RAM data storage for up to 5 minutes with the battery pack removed. It is always a good idea to replace the battery pack with a fresh recharged pack immediately after removing the discharged pack. Without battery power your terminal will not operate, and your display will be blank.

Display Positions 4 through 8

XX/YY

Current Row/Column Position *(59XX)*
Shows the current location of the cursor, with "XX" as the row position and "YY" as the column position. For example, if the cursor is at row 5, column 12, the annunciators will show "05/12" in positions 4–8.

Display Positions 5 through 8

▮▮▮▮

Battery Pack Gas Gauge Icons *(6400)*
These battery icons represent the charge level in your main battery pack.

- Four icons (positions 5–8) indicates more than 76% charge level
- Three icons (positions 5, 6, and 7) indicates between 51–75%
- Two icons (positions 5 and 6) indicates between 26–50%
- One icon (position 5) indicates between 21–25%

When the terminal gets below 20%, the icons are replaced by Ls and after the charge level is less than 10% your terminal emits a series of beeps.

- Four Ls indicates 16–20% charge level
- Three Ls indicates 11–15%. At the 15% charge level, your terminal beeps once. The sound duration and frequency are at levels you have set for your terminal.
- Two Ls indicates 6–10%. At the 10% and 7% charge levels, your terminal beeps once. The sound duration and frequency are at levels you have set for your terminal.
- One L indicates 0–5%. At this charge level, your terminal beeps three times at its current setting level for both duration and frequency.

Note: If you have the beeper turned **off**, you will not hear any of the low battery beep warnings.

The battery pack icons on your terminal display reflect the same status as the LEDs on the battery pack. However, there may be a one minute delay for updating the icons on the display when the battery pack is replaced.

Configuring TE Parameters

This chapter contains the TE configuration menus for the following terminals: 2415, 2425, 2435A, 2455, 2475, 248X (Trakker Antares), 6400 Computer, 59XX, 17XX, and 11XX Terminals, and 5055 Data Collection PCs unless otherwise noted. Additional information is available for each of the following:

• Trakker Antares (2415, 2425, 2435A, 2455, 2475, 248X)	page 99
• 6400 Computer	page 100
• 5055 Data Collection PC	page 101

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

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

Key Sequence to Open Main Menu

To configure TE parameters on your respective terminal, at the initialization screens or anywhere in a TE session, access the **Main Menu** by pressing the following key sequence:

To configure TE parameters:
At the initialization screens or anywhere in a TE session, access the **Main Menu** by pressing the key sequence.

Terminal	Key Sequence
2415	f F1 *(55-key keyboard)* f 5 *(37-key keyboard)*
2425	f F1
2435A	◨ ◧ *(57-key keyboard)* [Green] □ [Orange] □ *(48-key keyboard)* ◨ ◧ *(39-key function numeric keyboard)*
2455, 2475, 248X	f F1
6400	[Gold] [Blue]
5055	[Blue] [M] *or* [ALT] [M]
59XX	[Brown] [Black]
17XX, 11XX	[Gold] [Black]

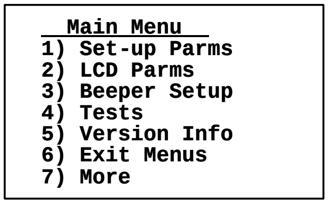
Note: You can also scan this bar code label to access the **Main Menu**.

TE configuration menus

%TECFG

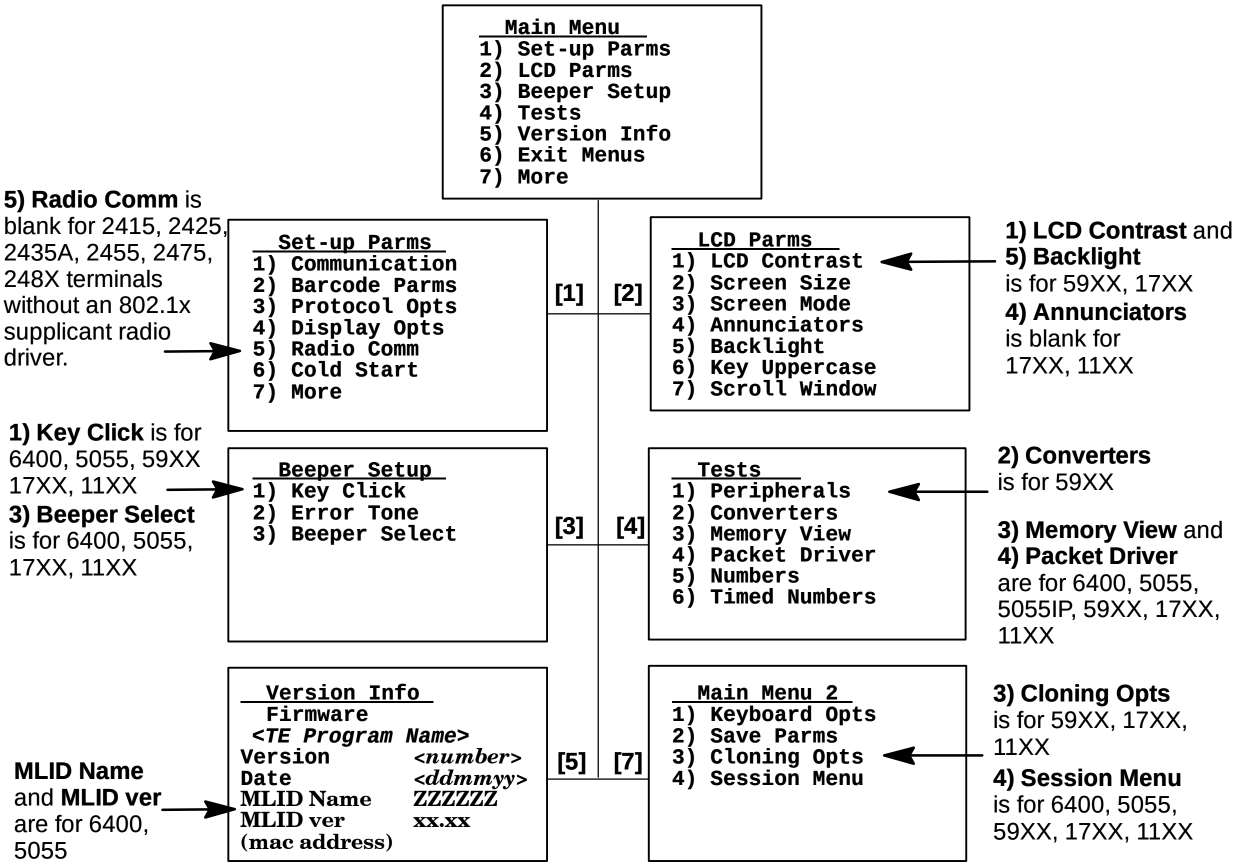
Opening the Main Menu

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 session, press the appropriate keys to access the **Main Menu** *(defined within each terminal's specific information)*. After the **Main Menu** appears, enter a number (1–7) to make a selection.



```
  Main Menu
1) Set-up Parms
2) LCD Parms
3) Beeper Setup
4) Tests
5) Version Info
6) Exit Menus
7) More
```

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



5) Radio Comm is blank for 2415, 2425, 2435A, 2455, 2475, 248X terminals without an 802.1x supplicant radio driver.

1) LCD Contrast and **5) Backlight** is for 59XX, 17XX

4) Annunciators is blank for 17XX, 11XX

1) Key Click is for 6400, 5055, 59XX 17XX, 11XX

3) Beeper Select is for 6400, 5055, 17XX, 11XX

2) Converters is for 59XX

3) Memory View and **4) Packet Driver** are for 6400, 5055, 5055IP, 59XX, 17XX, 11XX

MLID Name and **MLID ver** are for 6400, 5055

3) Cloning Opts is for 59XX, 17XX, 11XX

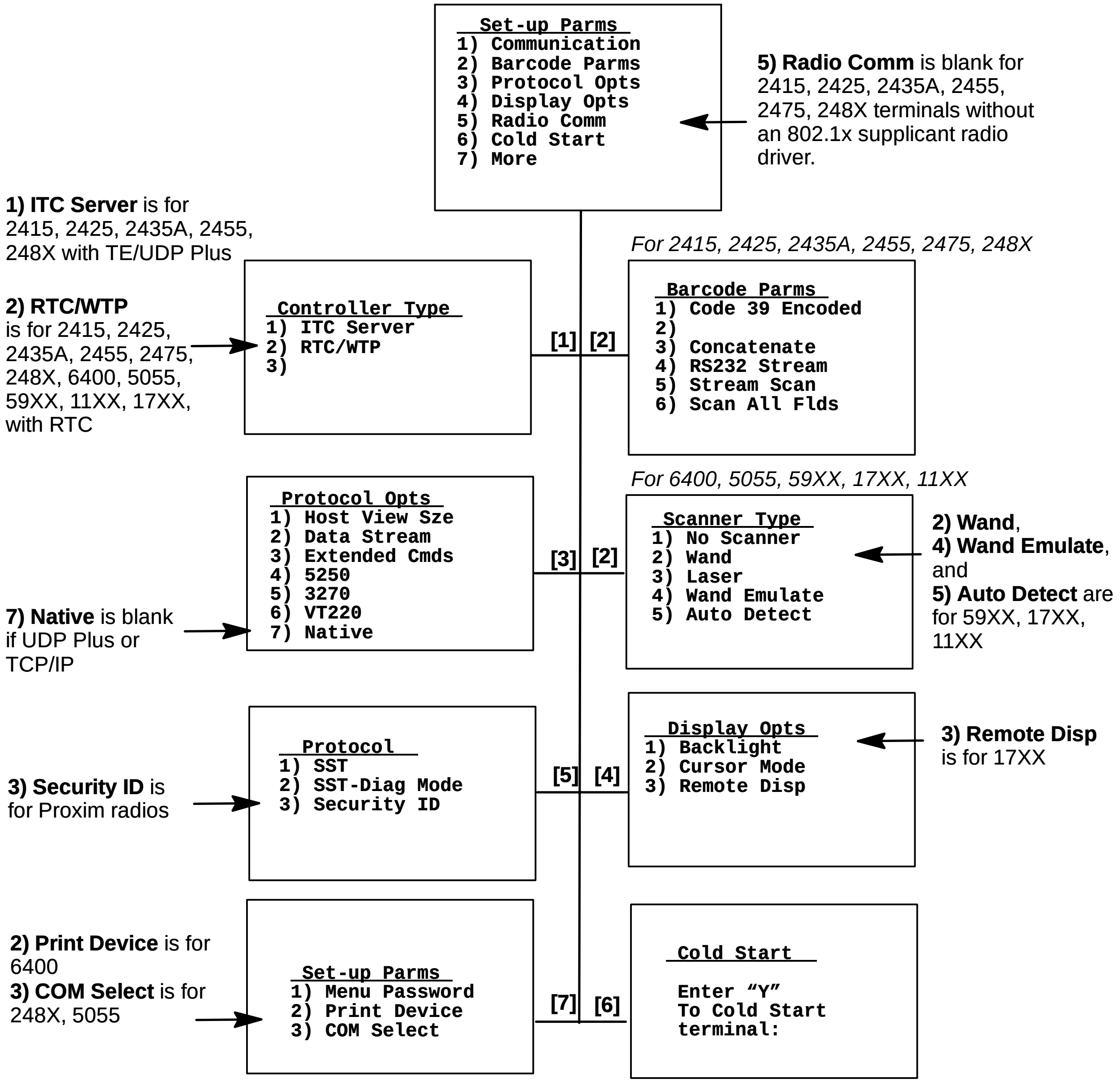
4) Session Menu is for 6400, 5055, 59XX, 17XX, 11XX

1) Set-Up Parms

1) **Set-Up Parms** is password-protected to prevent unauthorized users from changing parameters. You can change the password by customizing the parameter's set-up file (CONFIG.DAT). For more information about changing the password, see Chapter 4, "*Customizing Your Configuration.*"

The parameters set only 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* changing parameter settings.

To open the 1) **Set-up Parms** menu, press [1], [Enter], then type **CR52401** at the prompt to access the following menus.



```
   Set-up Parms
1) Communication
2) Barcode Parms
3) Protocol Opts
4) Display Opts
5) Radio Comm
6) Cold Start
7) More
```

5) Radio Comm is blank for 2415, 2425, 2435A, 2455, 2475, 248X terminals without an 802.1x supplicant radio driver.

1) ITC Server is for 2415, 2425, 2435A, 2455, 248X with TE/UDP Plus

2) RTC/WTP is for 2415, 2425, 2435A, 2455, 2475, 248X, 6400, 5055, 59XX, 11XX, 17XX, with RTC

```
 Controller Type
1) ITC Server
2) RTC/WTP
3)
```

For 2415, 2425, 2435A, 2455, 2475, 248X

```
  Barcode Parms
1) Code 39 Encoded
2)
3) Concatenate
4) RS232 Stream
5) Stream Scan
6) Scan All Flds
```

```
 Protocol Opts
1) Host View Sze
2) Data Stream
3) Extended Cmds
4) 5250
5) 3270
6) VT220
7) Native
```

7) Native is blank if UDP Plus or TCP/IP

For 6400, 5055, 59XX, 17XX, 11XX

```
 Scanner Type
1) No Scanner
2) Wand
3) Laser
4) Wand Emulate
5) Auto Detect
```

2) Wand, **4) Wand Emulate**, and **5) Auto Detect** are for 59XX, 17XX, 11XX

```
   Protocol
1) SST
2) SST-Diag Mode
3) Security ID
```

3) Security ID is for Proxim radios

```
  Display Opts
1) Backlight
2) Cursor Mode
3) Remote Disp
```

3) Remote Disp is for 17XX

2) Print Device is for 6400
3) COM Select is for 248X, 5055

```
  Set-up Parms
1) Menu Password
2) Print Device
3) COM Select
```

```
 Cold Start

Enter "Y"
To Cold Start
terminal:
```

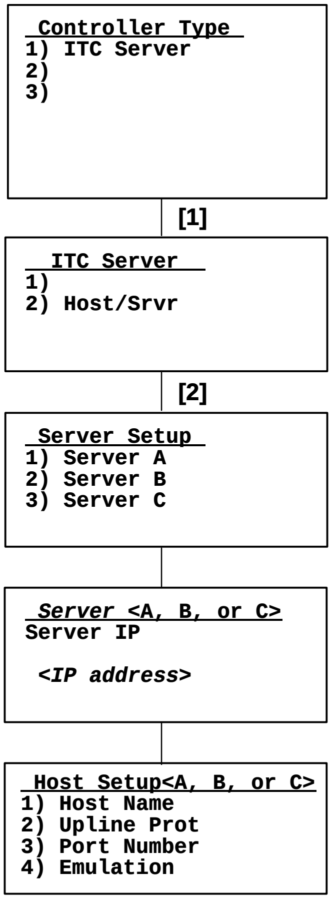
1) Communication

The communication option for the terminals are broken down as follows:

- *2415, 2425, 2435A, 2455, 2475, 248X Terminals with UDP Plus*
  1) **ITC Server** via the next paragraph.
- *2415, 2425, 2435A, 2455, 2475, 248X Terminals with RTC*
  2) **RTC/WTP** on page 46.
- *6400, 5055, 59XX, 17XX, 11XX Terminals with RTC*
  2) **RTC/WTP** on page 48.

1) ITC Server (2415, 2425, 2435A, 2455, 2475, 248X with TE/UDP Plus)

ITC (Intermec Technologies Corporation) 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).



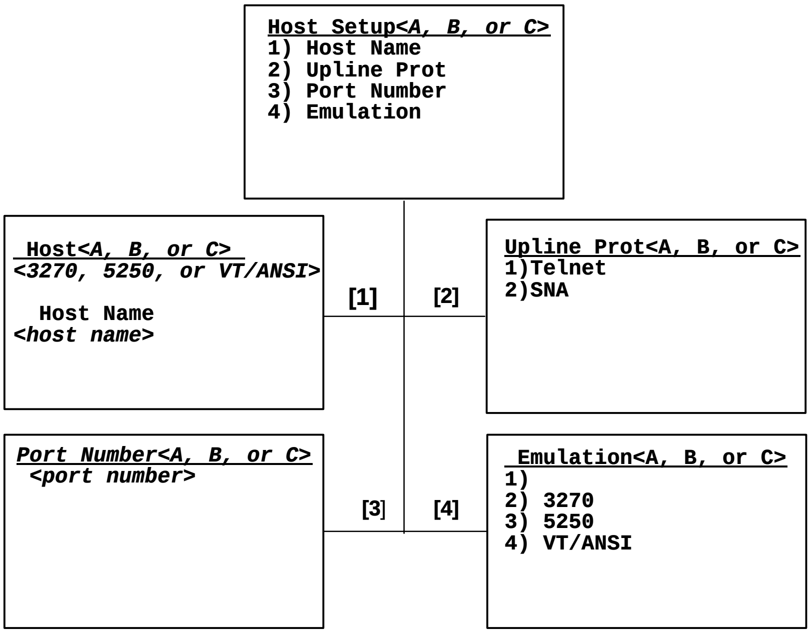
Server IP <A, B, or C>

The Server <A> IP address is read from either the CONFIG.DAT (if one is saved) or the terminal firmware. When an IP address is entered, it gets written to the terminal's firmware for the next boot.

If you are using the Server <B> or Server <C> option as a fallback DCS 300, enter the server's IP address or DNS name. The boot sequence will continue 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 16 or fewer characters in length (with no spaces). It is case-sensitive and must match a host name in the list of available hosts defined on the DCS 300, 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 terminal was linked with a host name on the DCS 300, or a default host was configured on this DCS 300, you do not need to enter a host name on this screen. The terminal displays a list of available hosts if these conditions are met:

  - The DCS 300 contains multiple hosts.
  - The terminal is not linked with a host name on the DCS 300.
  - The host name is blank.

  The user can then select the host from the list.

- 2) **Upline Prot**(ocol)
  When enabled, upline protocol options do the following:
  - **Telnet** *(Default)*
    Forces the DCS 300 to create a Telnet connection to the host.
  - **SNA**
    Forces the DCS 300 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 terminal each host computer type. *Default is 3270.*

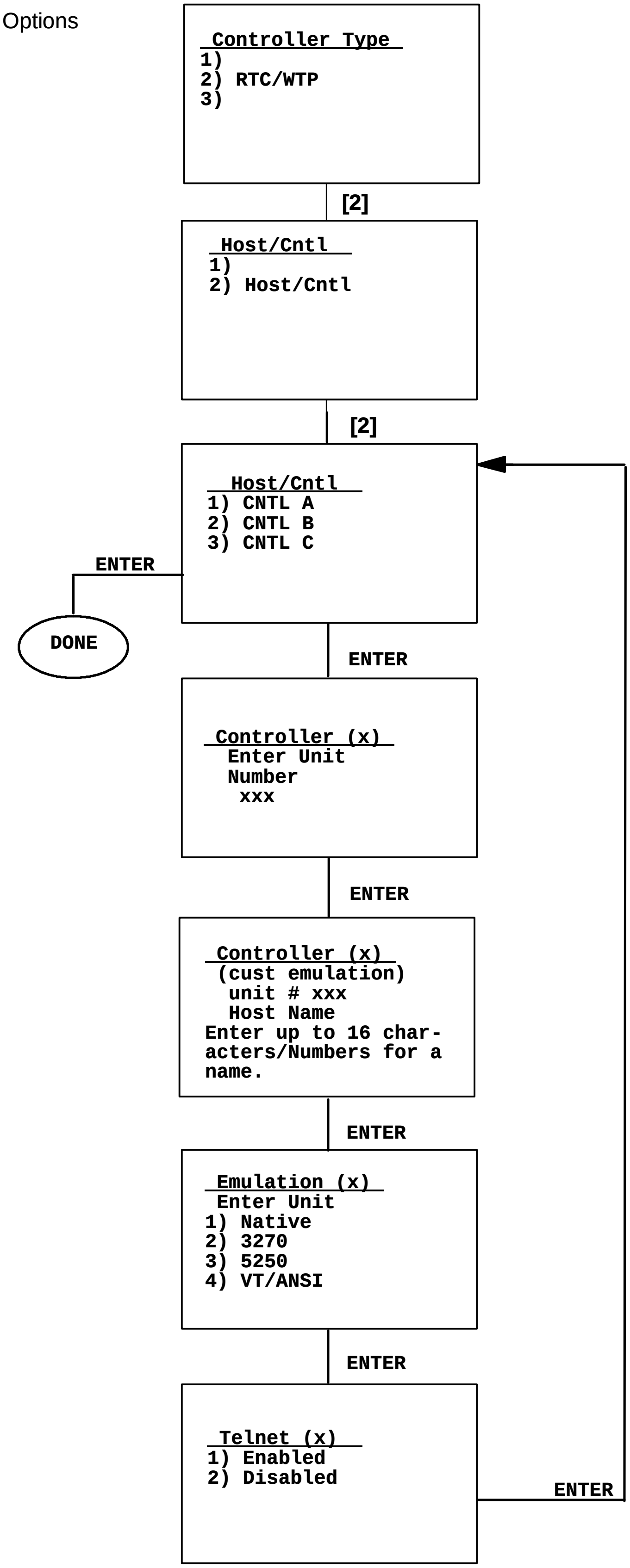
Note: You can also set the type through the 2) **Data Stream** option on the 3) **Protocol Opts** menu.

2) RTC/WTP (2415, 2425, 2435A, 2455, 2475, 248X)

Use 2) **RTC/WTP** to set, view, or change the terminal ID. This number allows the host computer to identify individual terminals in the radio data network. Each terminal must have a unique number in the same network. Use the numeric keys on the terminal keyboard to modify its ID. Enter a number between 0–126, then press [Enter] to go to the 2) **Host/Cntl** menus.

The 1) **Radio Setup** and 2) **Host/Cntl** menus are shown on the following pages. Pending on which option you are exercising you will either advance to the WTP Stack settings or the TCP/IP Stack settings. 1) **Radio Setup** is done in the firmware with the RTC stack.

WTP Stack Options



2) RTC/WTP (6400, 5055, 59XX, 17XX, 11XX)

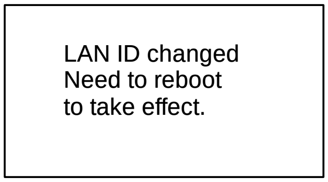
Use **2) RTC/WTP** to set, view, or change the terminal ID. This number allows the host computer to identify individual terminals in the radio data network. Each terminal must have a unique number in the same network.

The **1) Radio Setup** and **2) Host/Cntl** menus are shown on the following pages. Pending on which option you are exercising you will either advance to the WTP Stack settings or the TCP/IP Stack settings.

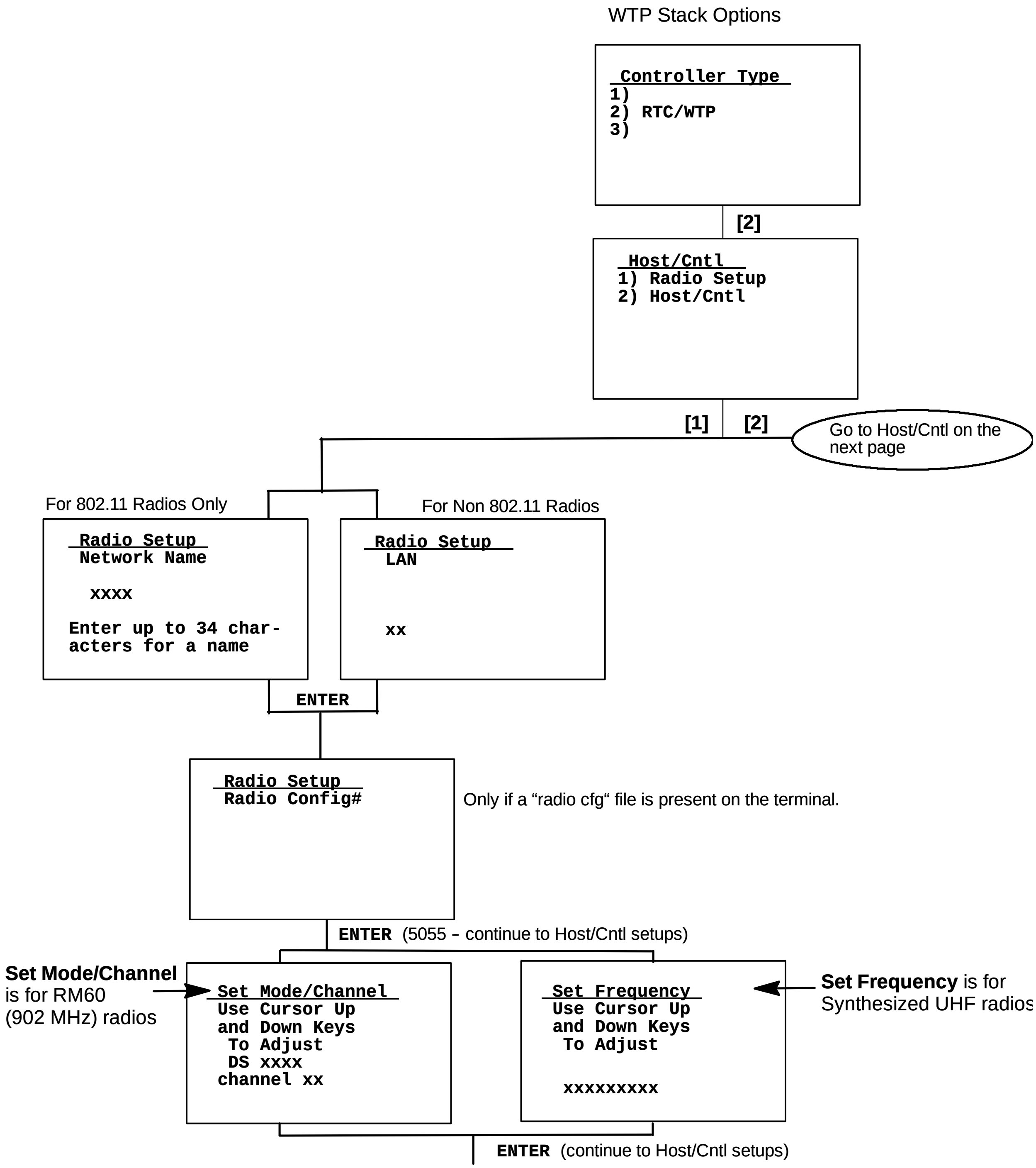
- **WTP Settings**
  Use the numeric keys on the terminal keyboard or 5055 external keyboard to modify its ID. Enter a number between 0–126, then press [Enter] to go to the **2) Host/Cntl** menus.

  The LAN ID number can be 0–255 depending on the radio being used. With the RM60/70 radio, the number can be 0–255; with the RM80/90 radio, the range is 0–15. Your terminal only communicates with equipment using the same ID number that you assigned to your LAN.

  If you enter a LAN ID number different from what is configured in the NET.CFG file you get the following message. If this happens, the system modifies NET.CFG, which is used by WTPPKT.EXE on initializing the terminal.



```
LAN ID changed
Need to reboot
to take effect.
```


WTP Stack Options
[2]
[1]
[2]
Go to Host/Cntl on the next page
For 802.11 Radios Only
For Non 802.11 Radios
ENTER
Only if a "radio cfg" file is present on the terminal.
ENTER (5055 – continue to Host/Cntl setups)
ENTER (continue to Host/Cntl setups)

```
 Controller Type
1)
2) RTC/WTP
3)
```

```
 Host/Cntl
1) Radio Setup
2) Host/Cntl
```

```
 Radio Setup
 Network Name

   xxxx

Enter up to 34 char-
acters for a name
```

```
 Radio Setup
  LAN



  xx
```

```
 Radio Setup
 Radio Config#
```

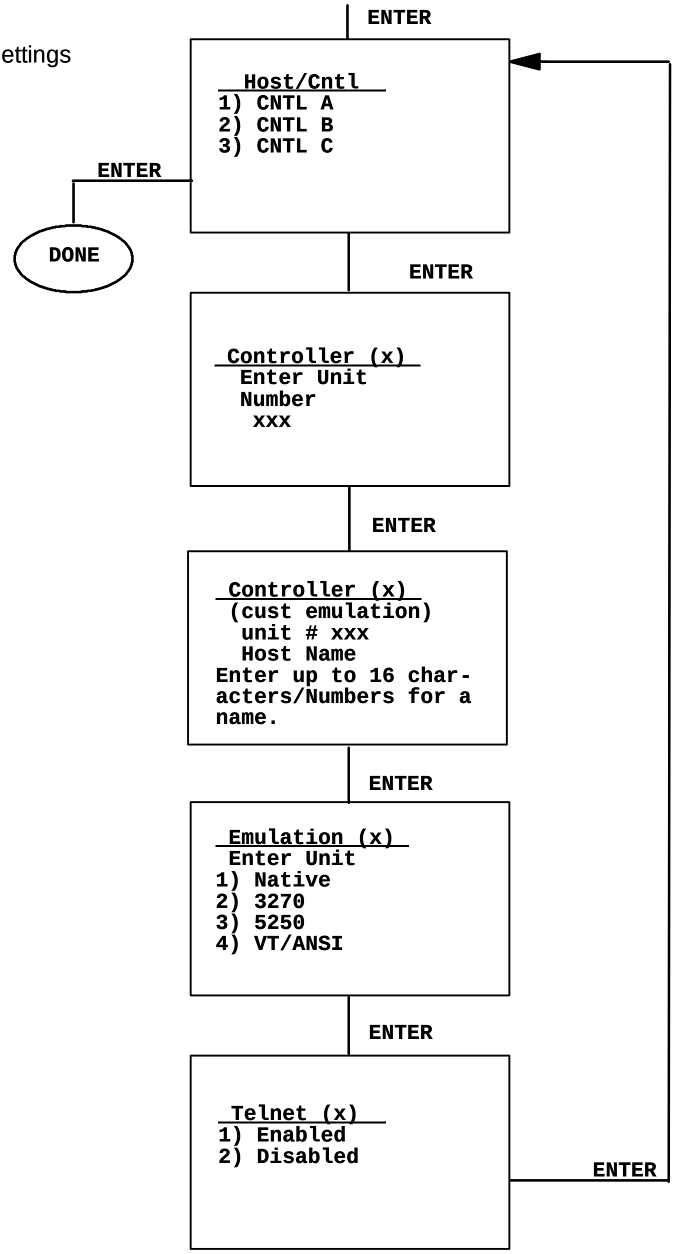
```
 Set Mode/Channel
 Use Cursor Up
 and Down Keys
  To Adjust
  DS xxxx
 channel xx
```

```
 Set Frequency
 Use Cursor Up
 and Down Keys
  To Adjust

  xxxxxxxxx
```

Set Mode/Channel is for RM60 (902 MHz) radios

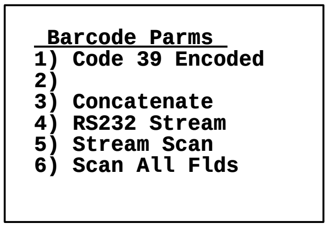
Set Frequency is for Synthesized UHF radios

Host/Cntl Settings



2) Barcode Parms (2415, 2425, 2435A, 2455, 2475, 248X)

The 2) **Barcode Parms** (parameters) menus designate whether you are using a bar code scanner and scan options.



```
 Barcode Parms
1) Code 39 Encoded
2)
3) Concatenate
4) RS232 Stream
5) Stream Scan
6) Scan All Flds
```

1) Code 39 Encoded

By default, **1) Code39 Encoded** is disabled. When you enable this option, the key press sequences for Encoded Code 39 are used. For more information, see Appendix A, "*Bar Code Scanning*."

3) Concatenate

Each bar code read is added to the end of the previous bar code read until the 2415, 2425, 2435A, 2455, 2475, or 248X Terminal 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.

Note: If you access the TE configuration menus when **4) RS232 Stream** is enabled, all RS-232 data received during this time is lost. Only the data arriving from the COM1 port on the hand-held terminal is received.

4) RS232 Stream

If this option is enabled, the terminal watches the RS-232 port for activity in which the terminal reads in the RS-232 data and sends it to the host. This enables support for fixed-station scanners or scales that send only RS-232 data without having the host send an extended command to enable the RS-232 port. *Default for **4) RS232 Stream** is disabled.*

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 (Fields)

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.

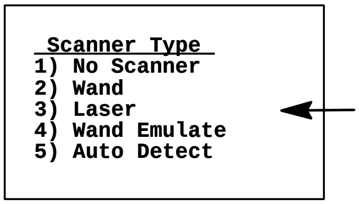
2) Barcode Parms (6400, 5055, 59XX, 17XX, 11XX)

Selecting 2) **Barcode Parms** (parameters) displays the **Scanner Type** menu.

Scanner Type

Use the **Scanner Type** menu to designate the type of bar code scanner. To make a selection, press the numeric key corresponding to the desired option, then press [Enter] to access the **Scan Options** menu.

Note: If 1) **No Scanner** is chosen, the remaining scanner and bar code options can be set. Then, if a scanner is used at a later time, only the scanner type needs to be designated. Selected options are highlighted on the display. Press a key corresponding to that option to deselect a selected option.

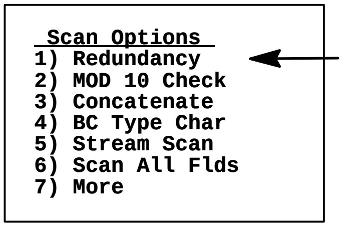


```
 Scanner Type
1) No Scanner
2) Wand
3) Laser
4) Wand Emulate
5) Auto Detect
```

2) Wand,
4) Wand Emulate, and
5) Auto Detect
are for 59XX, 17XX, 11XX

Scan Options

Use the **Scan Options** menu to assign scan options, enable bar code symbologies to use *(the terminal can only decode the enabled symbologies)*, and set options for each enabled bar code symbology, such as the bar code lengths. Press [Enter] to advance to the **Scan Options 2** menus.



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

1) Redundancy and
4) BC Type Char
are for 6400, 59XX, 17XX, 11XX

3) Concatenate is for 6400, 5055

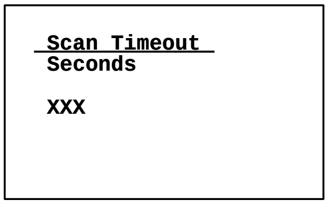
- 1) **Redundancy** *(6400, 59XX, 17XX, 11XX)*
  This option requires two identical scans of a bar code, one right after another, before the terminal accepts the scan as valid.

- 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 transmission 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.

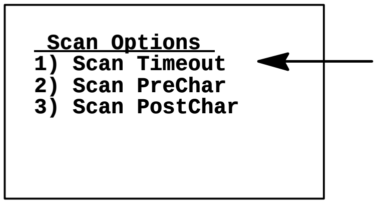
- 3) **Concatenate** *(6400, 5055)*
  Each bar code read is added to the end of the previous bar code read until the 6400 or 5055 terminal meets a condition forcing transmission to the host.

  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** *(6400, 59XX, 17XX, 11XX)*
  This option adds a character associated with the bar code type at the beginning of the scanned bar code. *See page 268 for bar code type characters.*
- 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** (Fields)
  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.
  - 1) **Scan Timeout** *(59XX, 17XX, 11XX)*
    Use this option to adjust the scanning timeout period from 1 to 200 seconds. When using a proximity-detect scanner, you can shorten the timeout period to speed up scanning while also reducing the duration of errant scans. With a long-range scanner, you may want to increase the timeout period to ensure sufficient aiming time.



```
 Scan Timeout
 Seconds

 XXX
```

  - 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.



```
 Scan Options
1) Scan Timeout
2) Scan PreChar
3) Scan PostChar
```

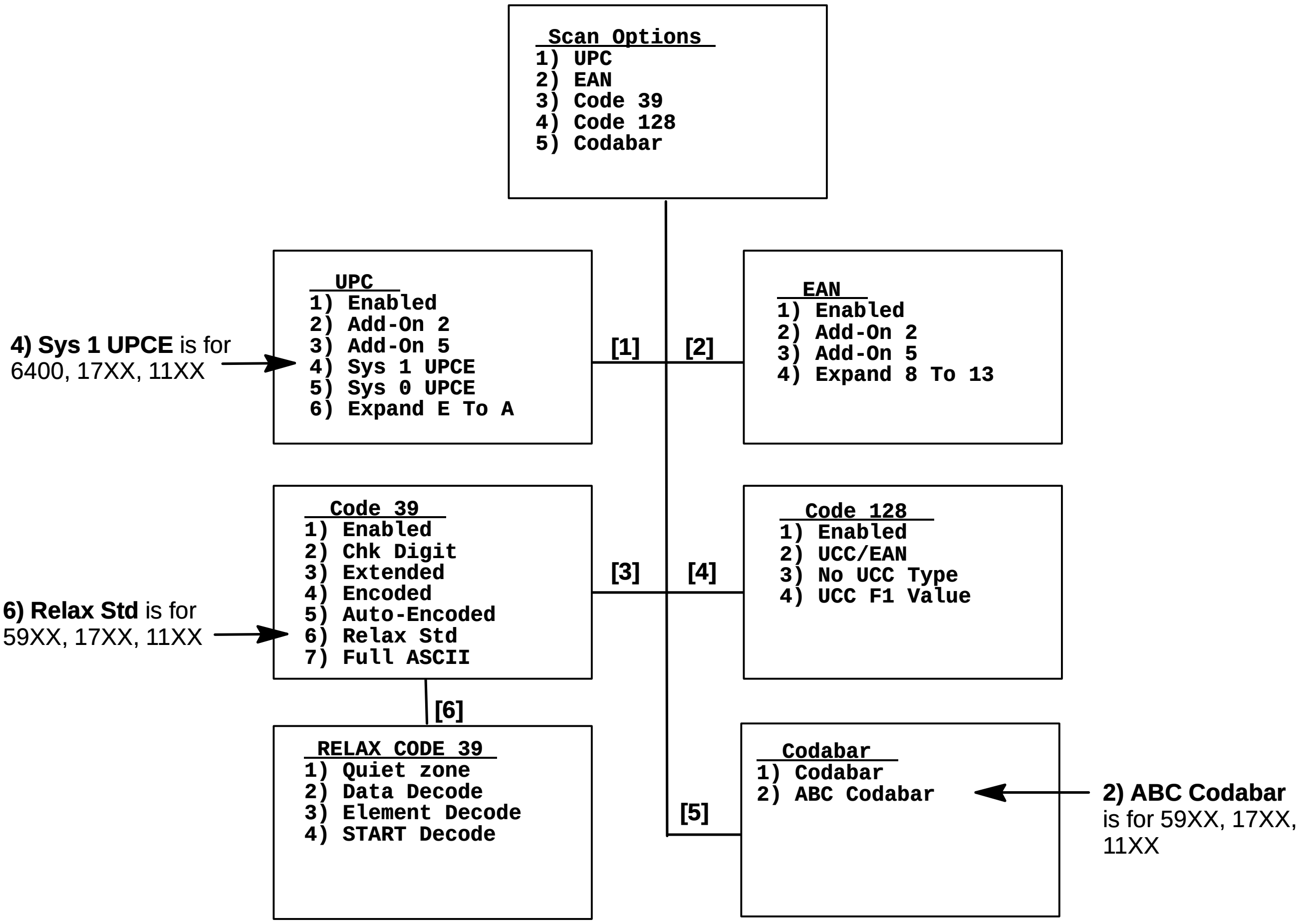
1) Scan Timeout is for 59XX, 17XX, 11XX

Scan Options to Enable Bar Codes

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 terminal returns to the same **Scan Options** menu. Choose another bar code symbology from that menu or press [Enter] to advance to the **Scan Options 2** menu.



```
 Scan Options
1) UPC
2) EAN
3) Code 39
4) Code 128
5) Codabar
```

```
  UPC
1) Enabled
2) Add-On 2
3) Add-On 5
4) Sys 1 UPCE
5) Sys 0 UPCE
6) Expand E To A
```

4) Sys 1 UPCE is for 6400, 17XX, 11XX

```
  EAN
1) Enabled
2) Add-On 2
3) Add-On 5
4) Expand 8 To 13
```

```
  Code 39
1) Enabled
2) Chk Digit
3) Extended
4) Encoded
5) Auto-Encoded
6) Relax Std
7) Full ASCII
```

6) Relax Std is for 59XX, 17XX, 11XX

```
  Code 128
1) Enabled
2) UCC/EAN
3) No UCC Type
4) UCC F1 Value
```

```
 RELAX CODE 39
1) Quiet zone
2) Data Decode
3) Element Decode
4) START Decode
```

```
  Codabar
1) Codabar
2) ABC Codabar
```

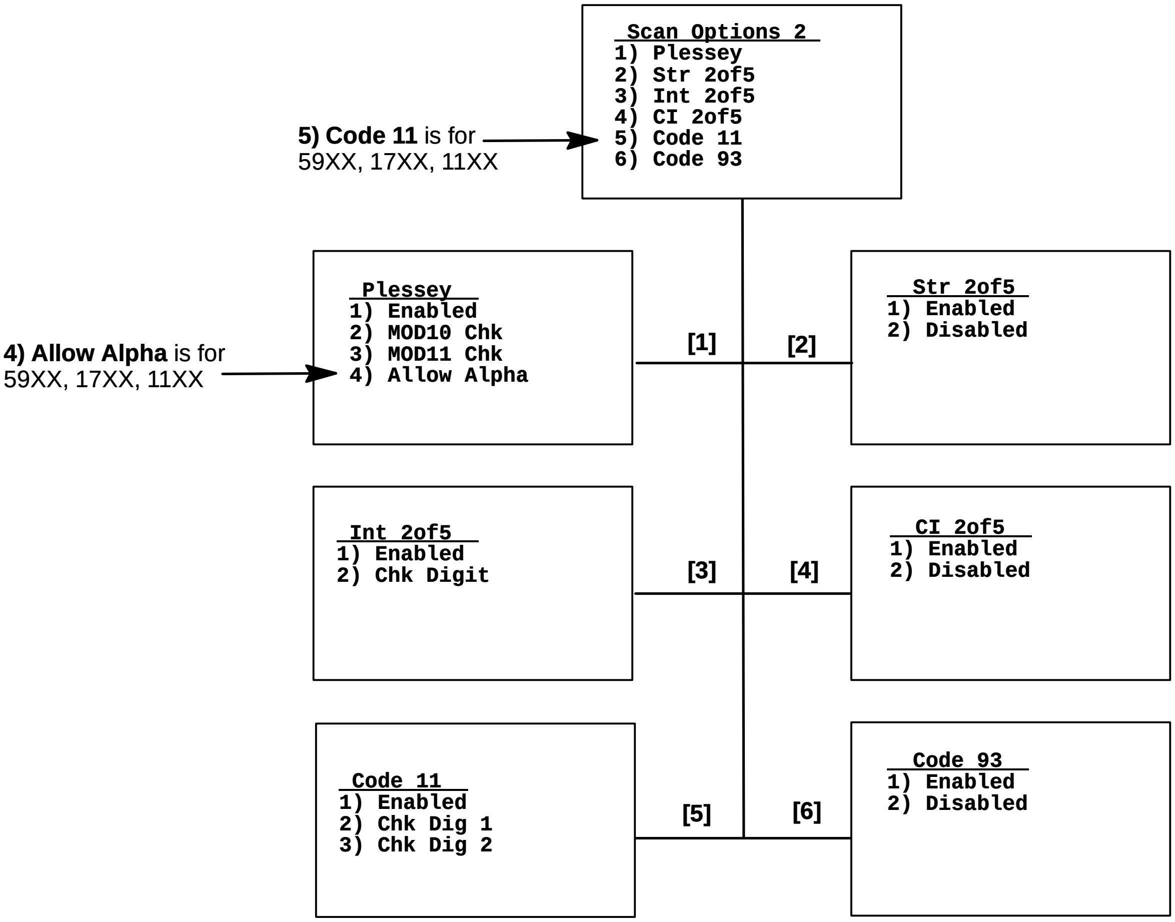
2) ABC Codabar is for 59XX, 17XX, 11XX

Scan Options 2 (Additional Bar Codes)

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 2** 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.



5) Code 11 is for 59XX, 17XX, 11XX

4) Allow Alpha is for 59XX, 17XX, 11XX

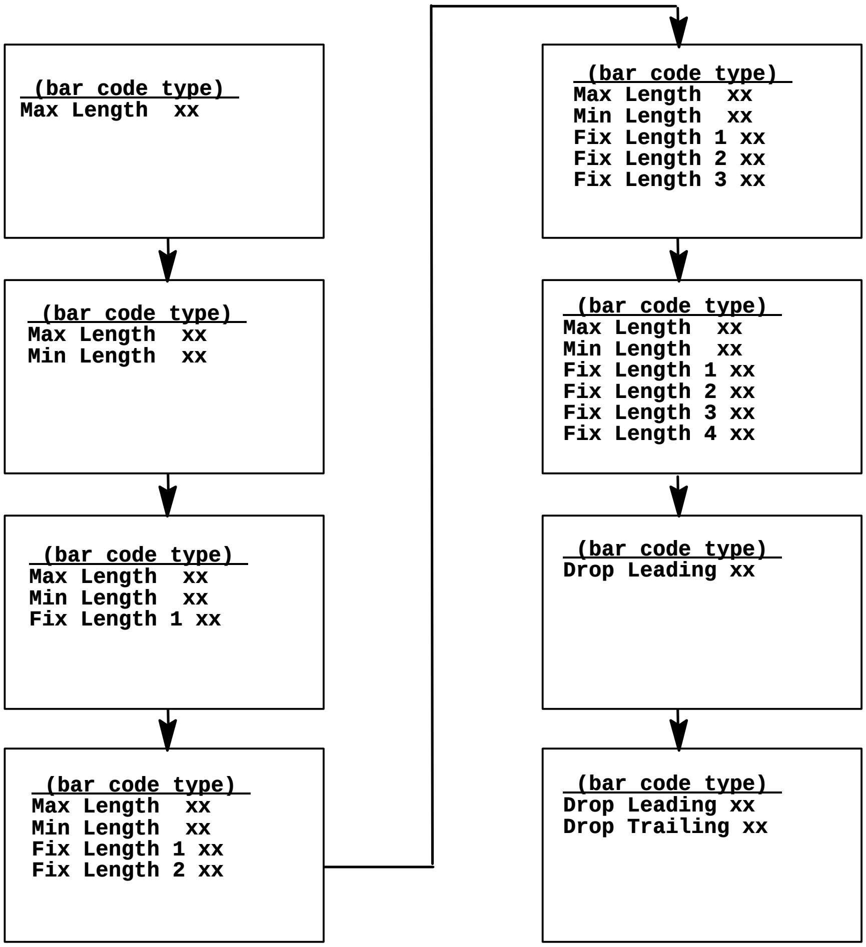
Lengths Options

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

1. Key in the maximum length (0–99), then press [Enter].
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 that Int 2of5 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].

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 *(pending on from which menu the bar code was enabled).*



```
 (bar code type) 
Max Length  xx
```

```
 (bar code type) 
Max Length  xx
Min Length  xx
```

```
 (bar code type) 
Max Length  xx
Min Length  xx
Fix Length 1 xx
```

```
 (bar code type) 
Max Length  xx
Min Length  xx
Fix Length 1 xx
Fix Length 2 xx
```

```
 (bar code type) 
Max Length  xx
Min Length  xx
Fix Length 1 xx
Fix Length 2 xx
Fix Length 3 xx
```

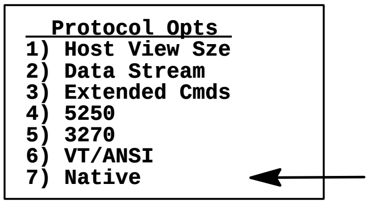
```
 (bar code type) 
Max Length  xx
Min Length  xx
Fix Length 1 xx
Fix Length 2 xx
Fix Length 3 xx
Fix Length 4 xx
```

```
 (bar code type) 
Drop Leading xx
```

```
 (bar code type) 
Drop Leading xx
Drop Trailing xx
```

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 or TCP/IP

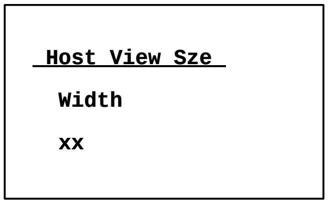
1) Host View Sze

The **1) Host View Sze** option tells the terminal 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.

The default setting for Native Terminal Emulation is 120 columns by 16 rows (equals 1920). The default for VT/ANSI Terminal Emulation is 80 columns by 24 rows (equals 1920). The default for 5250 Terminal Emulation is 80 columns by 24 rows (25 with error messages).

The default for 3270 Terminal Emulation depends on what the original setting was before changing to the 3270 Terminal Emulation. Remember that the total characters must equal 1920 (number of rows by the number of columns) unless using 5250 Terminal Emulation with error messages.

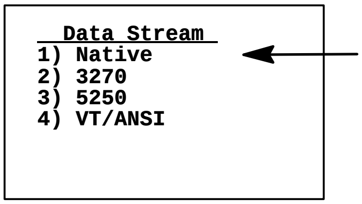
Enter a number between 1–80 to change the width. Press [Enter] to return to the **3) Protocol Opts** menu. "xx" is the width.



```
 Host View Sze
  Width
  xx
```

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.



```
   Data Stream
1) Native
2) 3270
3) 5250
4) VT/ANSI
```

1) Native is blank if UDP Plus or TCP/IP

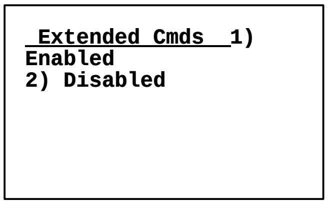
Note: For 2415, 2425, 2435A, 2455, 2475, and 248X, you can also set the emulation type through the **1) ITC Server** option.

For information about 3270, 5250, or VT/ANSI TE, see the following manuals:

- *TE 2000 3270 Terminal Emulation Programmer's Guide* (P/N: 977-055-003)
- *TE 2000 5250 Terminal Emulation Programmer's Guide* (P/N 977-055-004)
- *TE 2000 VT/ANSI Terminal Emulation Programmer's Guide* (P/N 977-055-005)

3) Extended Cmds (Commands)

Use 3) **Extended Cmds** to enable or disable extended commands. By default, extended commands are disabled. 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.

- 1) **Beep On Error**
  Causes the buzzer to emit when an error condition occurs; does not lock up the keyboard.
- 2) **Auto Tab Scan**
  Causes the cursor to automatically tab forward to the next input field when a good scan is obtained.
- 3) **Telnet**
  Handles the telnet option negotiations to establish a session with an appropriate telnet server.

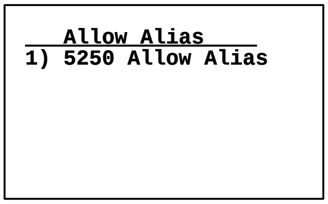
Note: 3) **Telnet** is not changeable if TCP/IP is used.

- 4) **Device Name**
  Enters a physical name for the device. This name is 1–10 characters with allowable values of A–Z, a–z, 0–9, and #, $, -, and @. *5250E RFC 1572 Support.*

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

The following screen displays after a physical name is entered.

- 1) **5250 Allow Alias**
  If this is enabled, and your current device name returns an error in use, then the terminal attempts to appeand 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 terminal, causing the terminal to reboot. This continues until the device name is no longer in use.



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

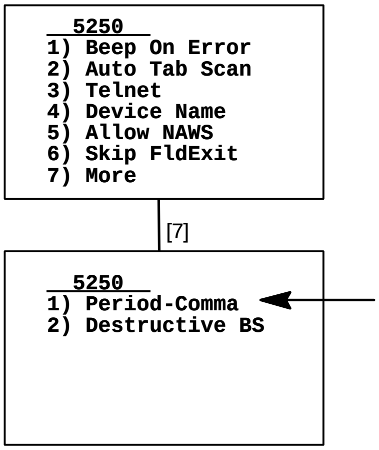
- 5) **Allow NAWS** (Negotiate About Window Size)
  When enabled, the terminal will support Telnet option 31 if prompted. When disabled, this option will return a “won’t do” message.
- 6) **Skip FldExit** (Field Exit)
  When this option is enabled, a field that would generate an error code of “0020” (field exit required) when [Enter] was pressed will now send the field data.

  The legacy TE application would send the entire field so a “7” entered into a 3-character blank-filled field would transmit as “7~~bb~~”, a “12” would transmit as “12~~b~~”. TE2000 strips the field fill characters so “7” will transmit as a “7”, “12” will transmit as a “12” and so forth.

Note A “~~b~~” indicates a one-byte space.

- 7) **More**
  Select 7) **More** to modify the function of the period or backspace key.
  - 1) **Period-Comma** *(2415, 2425, 2435A, 2455, 2475, 248X)*
    Select **1) Period-Comma** to change the period key (.) into a comma.

- 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.



1) Period-Comma is for 2415, 2425, 2435A, 2455, 2475, 248X

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.

- 1) **Keybrd Unlock**
  Unlocks the keyboard after the [**PA1**], [**PA2**], or [**CLEAR**] keys are pressed. (Normally the keyboard remains locked, until unlocked or the terminal 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 when a good scan is obtained.

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 that this cannot be enabled at the same time as **Auto Tab Scan**.*

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

- 4) **Emulate 3210** *(6400, 5055, 59XX, 17XX, 11XX)*
  The Intermec 3210 emulation option allows the 3270 keyboard to emulate an RT3210 Hand-Held Computer, performing the same functions. For full 3210 compatibility, disable buffering the keyboard, set cursor to lazy mode, and set the LCD Parms to Key Uppercase.

- 5) **Telnet**
  Handles telnet option negotiations to establish a session with an appropriate telnet server.

Note: 5) **Telnet** cannot be changed if you are using TCP/IP.

- 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.

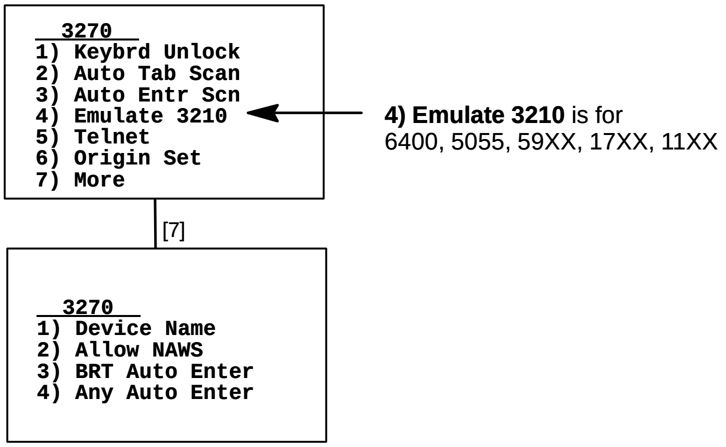
  - 1) **Device Name**
    Enters a physical name for the device. This name is 1–10 characters with allowable values of A–Z, a–z, 0–9, and #, $, -, and @. 3270E RFC 2355 Support.

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

  - 2) **Allow NAWS** (Negotiate About Window Size)
    When enabled, the terminal will support Telnet option 31 if prompted. When disabled, this option will return 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 terminal will automatically send the data for this screen back to the host with an “Enter AID” code. If disabled, the screen will wait for the user to hit the **AID** key prior to sending data back to the host. *Default is disabled.*

  - 4) **Any Auto Enter**
    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.*



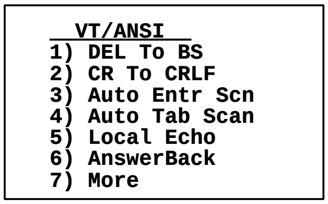
```
  3270
1) Keybrd Unlock
2) Auto Tab Scan
3) Auto Entr Scn
4) Emulate 3210
5) Telnet
6) Origin Set
7) More
```

4) Emulate 3210 is for 6400, 5055, 59XX, 17XX, 11XX

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

6) VT/ANSI

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 this option is not enabled, **<Enter>** performs a carriage return only. *Default is disabled.*
- 3) **Auto Entr Scn** (Auto Enter Scan)
  Actuates the **<Enter>** function whenever a good scan is obtained. *Default is enabled for version 6.60 or greater, otherwise the default is disabled.*

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

- 4) **Auto Tab Scan**
  Causes the terminal to automatically tab to the next input field when a good scan is obtained. *Default is disabled.*

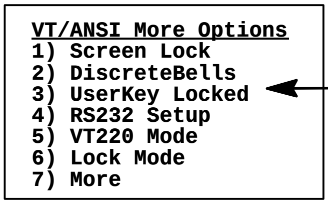
Note: 4) **Auto Tab Scan** cannot be enabled at the same time as 3) **Auto Entr Scan**.

- 5) **Local Echo**
  Allows characters to be displayed from terminal memory but not from host memory. *Default is disabled.*
- 6) **AnswerBack**
  Enables you to enter a character string sent to the host in response to an inquiry (hexadecimal 05). The range is 0–30, *default is null string.*

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

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

- 7) **More**
  Calls up the following **VT/ANSI More Options** screen.



```
VT/ANSI More Options
1) Screen Lock
2) DiscreteBells
3) UserKey Locked
4) RS232 Setup
5) VT220 Mode
6) Lock Mode
7) More
```

2) DiscreteBells and
4) RS232 Setup
are for 6400, 5055, 59XX, 17XX,
11XX

- 1) **Screen Lock**
  Sets and locks the screen to a specified size. Any characters outside this screen size are ignored by the hand-held unit. *Default is disabled.*
- 2) **DiscreteBells** *(6400, 5055, 59XX, 17XX, 11XX)*
  If enabled, the terminal beeps for each bell character the host sends. If disabled, the unit ignores the bell characters if it is already beeping. *Default is disabled.*
- 3) **UserKey Locked**
  If set, the host ignores a host command defining the Function keys. *Default is disabled.*

- 4) **RS232 Setup** *(6400, 5055, 59XX, 17XX, 11XX)*
  When using the media copy command, use this option to set up the communications port, then press [Enter] to return to **3) Protocol Opts**.

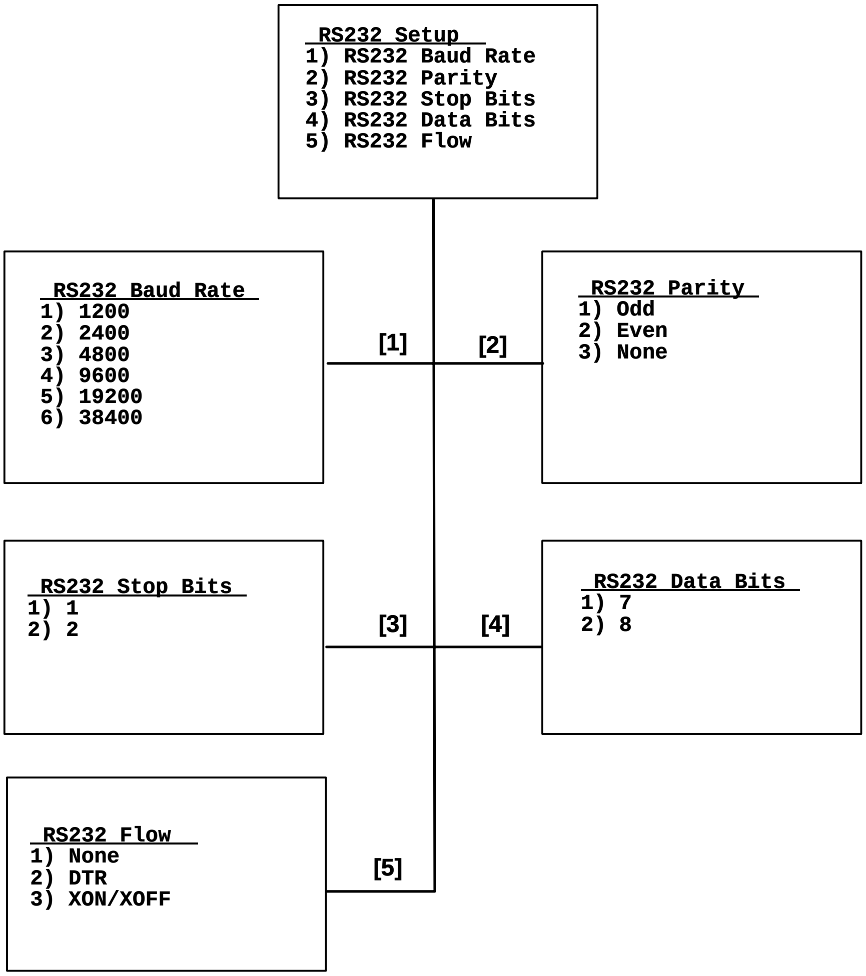
  - 1) **RS232 Baud Rate**
    Select 1200, 2400, 4800, 9600, 19200, or 38400. *Default is 9600.*

  - 2) **RS232 Parity**
    Select from Odd, Even, or None. *Default is None.*

  - 3) **RS232 Stop Bits**
    Select either 1- or 2-bits. *Default is 1.*

  - 4) **RS232 Data Bits**
    Select either 7- or 8-bit. *Default is 8.*

  - 5) **RS232 Flow**
    Select from None, DTR, or XON/XOFF. *Default is None.*

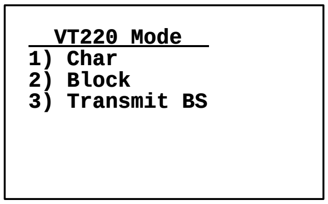


- 5) **VT220 Mode**
  Selects operating in character or block mode. Select a choice, then press [Enter] to return to the **3) Protocol Opts** menu. *Default is 1) Char.*

  - 1) **Char** *(Default)*
    Sets the mode to Character mode. The terminal sends each character as it is pressed.

  - 2) **Block**
    Sets the mode to Line Edit (block) mode. The terminal sends a block of characters when a terminating key is pressed.

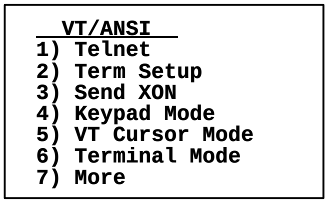
  - 3) **Transmit BS**
    When enabled, pressing the backspace key sends a backspace to the host for the host to echo back to the terminal. When disabled, the backspace key is handled locally on the terminal by doing a destructive backspace to the printed data characters on the display.



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

- 6) **Lock Mode**
  Use the Mode key on the terminal'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.*

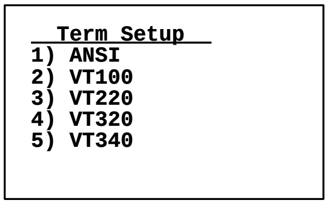
- 7) **More**
  Selects the telnet option for the VT/ANSI data stream.



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

  - 1) **Telnet**
    Handles telnet option negotiations to establish a session with an appropriate telnet server. *This cannot be changed if using TCP/IP.*

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



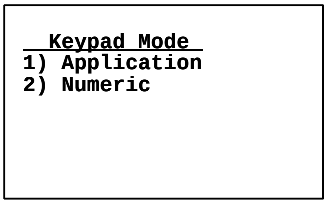
```
  Term Setup
1) ANSI
2) VT100
3) VT220
4) VT320
5) VT340
```

- 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****.*

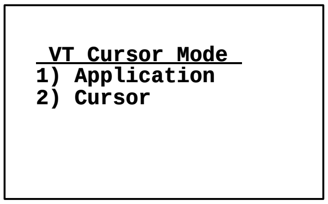
  - 1) **Application**
    Generates application ESC sequences for the key code. For help, refer to your VT manual.

  - 2) **Numeric**
    Generates ANSI cursor control ESC sequences that correspond to what appears on the face of the keys.



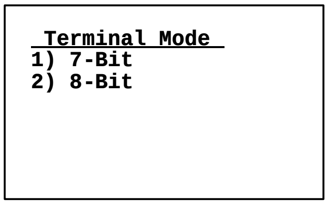
```
  Keypad Mode
1) Application
2) Numeric
```

- 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.



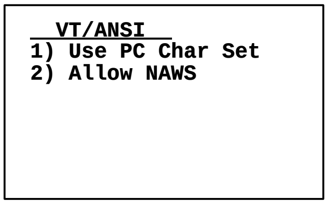
```
 VT Cursor Mode
1) Application
2) Cursor
```

- 6) **Terminal Mode** *(VT220/320 only)*
  Sets the terminal mode to 7-bit or 8-bit. This option sets the mode VT-series terminals 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 RS-232 stream and PC char set options:
  - 1) **Use PC Char Set**
    Select this option to default the font character to the terminal's character set instead of a DEC terminal character set.
  - 2) **Allow NAWS** (Negotiate About Window Size)
    When enabled, the terminal will support Telnet option 31 if prompted. When disabled, this option will return a "won't do" message.

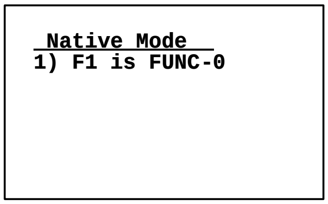


```
  VT/ANSI
1) Use PC Char Set
2) Allow NAWS
```

Note: 7) **Native** is enabled by default to preserve backwards compatibility.

7) Native (2415, 2425, 2435A, 2455, 2475, 248X, 6400, and 5055 with WTP, 59XX, 17XX, 11XX)

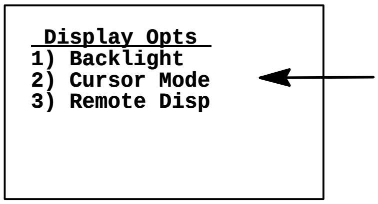
- *For 2415, 2425, 2435A, 2455, 2475, 248X Terminals with WTP:* When enabled, pressing [**F1**] is equivalent to pressing [**FnL**] [7] (Home), while pressing [**F2**] is equivalent to pressing [**F1**], etc. When disabled, [**F1**] is equivalent to [**F1**], [**F2**] is equivalent to [**F2**], etc.
- *For 2435A Terminals with 48-key keyboards:* When enabled, pressing [**F1**] is equivalent to pressing [Green] □ H .(Home), while pressing [**F2**] is equivalent to pressing [Orange] □ 1, etc. When disabled, [**F1**] is equivalent to pressing [Orange] □ 1, [**F2**] is equivalent to pressing [Orange] □ 2, etc.
- *For 6400 Computers with WTP:*
    - *On 51-key keyboards*, when enabled, pressing [**F1**] is equivalent to pressing [**Blue**] [**0**], while pressing [**F2**] is equivalent to pressing [**Blue**] [**1**], etc. When disabled, [**F1**] is equivalent to [**Blue**] [**1**], [**F2**] is equivalent to [**Blue**] [**2**], etc.
    - *On 41-key keyboards,* when enabled, pressing [**F1**] is equivalent to pressing [**Blue**] [**Space**].
- *For 5055 Terminals with WTP:* When enabled, pressing [**F1**] is equivalent to pressing [**Blue**] [**0**], while pressing [**F2**] is equivalent to pressing [**Blue**] [**1**], etc. When disabled, [**F1**] is equivalent to [**Blue**] [**1**], [**F2**] is equivalent to [**Blue**] [**2**], etc.
- *For 59XX, 17XX, 11XX,* When enabled, pressing [**F1**] is equal to [**Black**] [**0**] and [**F2**] is equal to [**Black**] [**1**], etc. When disabled, [**F1**] is equal to [**Black**] [**1**], [**F2**] is equal to [**Black**] [**2**], etc.



```
 Native Mode
1) F1 is FUNC-0
```

4) Display Opts

4) **Display Opts** adjusts the backlight timer, the cursor appearance on the display, or the remote display.



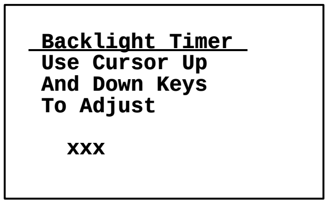
```
 Display Opts 
1) Backlight
2) Cursor Mode
3) Remote Disp
```

1) Backlight is for 2415, 2425, 2435A, 2455, 2475, 248X, 6400, 17XX, 11XX

3) Remote Disp is for 17XX

1) Backlight *(2415, 2425, 2435A, 2455, 2475, 248X, 6400, 17XX, 11XX)*

Use the up and down arrows to select “Off” or a number from “1–60” for 2415, 2425, 2435A, 2455, 2475, and 248X Terminals or from “1–255” for 6400, 17XX, and 11XX Terminals. This would dictate the number of seconds that the backlight stays on after a key is pressed.



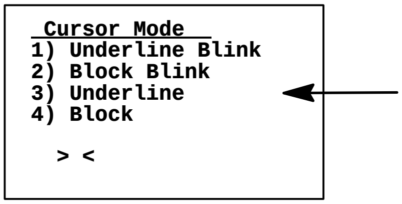
```
 Backlight Timer 
 Use Cursor Up
 And Down Keys
 To Adjust

   xxx
```

2) Cursor Mode

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** *(2415, 2425, 2435A, 2455, 2475, 248X, 6400, 59XX, 17XX, 11XX)*
  Causes the underline character to appear and disappear in its current location.
- 2) **Block Blink** *(2415, 2425, 2435A, 2455, 2475, 248X, 6400, 59XX, 17XX, 11XX)*
  Causes the block (■) character cursor to appear and disappear in its current location.
- 3) **Underline** *(6400, 59XX, 17XX, 11XX)*
  This option provides a solid underline character to display the cursor’s current location.

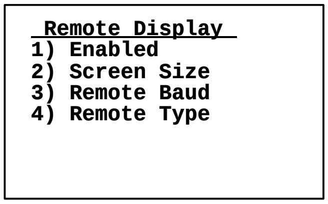
- 4) **Block** *(6400, 59XX, 17XX, 11XX)*
  This provides a reverse or highlighted block (■) character displaying the cursor's current location.



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

  > <
```

3) Underline and
4) Block
are for 6400, 59XX, 17XX,
11XX

- 3) **Remote Disp** *(17XX)*
  This menu allows communication to a remote display.
  - 1) **Enabled**
    Press this option to enable the remote display function.
  - 2) **Screen Size**
    Use this option to set the screen size of the remote display.
  - 3) **Remote Baud**
    Select the appropriate baud rate for the remote display: 9600, 19200, or 38400.
  - 4) **Remote Type**
    Use this option to identify the remote terminal: VT100, VT220, or ANSI.



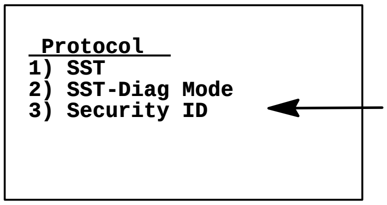
```
 Remote Display
1) Enabled
2) Screen Size
3) Remote Baud
4) Remote Type
```

5) Radio Comm (Blank for 2415, 2425, 2435A, 2455, 2475, 248X Terminals without an 802.1x Supplicant Radio Driver)

Note: Diagnostic modes disable data compression and are reserved for engineering tests.

Menus for 5) **Radio Comm** specify the communication protocol and speed in certain instances. Depending on circumstances, one of the two menus will appear.

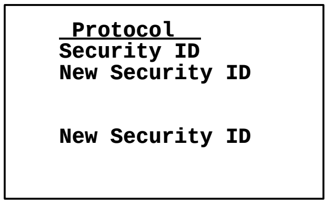
- 5) **Radio Comm** enables or disables SST diagnostic mode.



```
 Protocol
1) SST
2) SST-Diag Mode
3) Security ID
```

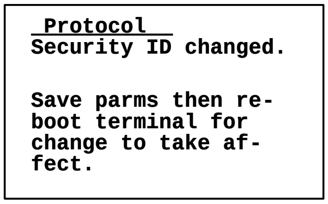
3) Security ID is for Proxim radios

- 3) Security ID *(Proxim radios)*
  This changes the radio security identification. Enter the new security ID (up to 16 characters) twice. If the old security ID is not correct or if the second new security ID was not entered correctly, you will see the "unchanged" message on the bottom line of the display.



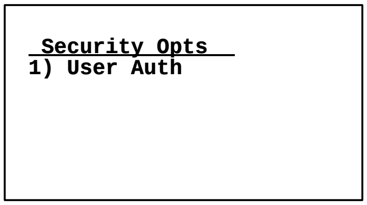
```
 Protocol
Security ID
New Security ID


New Security ID
```

If the new security ID is entered correctly, the following will appear.



```
 Protocol
Security ID changed.

Save parms then re-
boot terminal for
change to take af-
fect.
```

Security Opts
For 2415, 2425, 2435A, 2455, 2475, and 248X Terminals with an 802.1x supplicant radio driver installed, the following will appear:



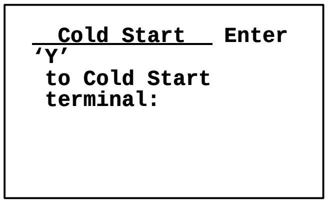
```
 Security Opts
1) User Auth
```

- **1) User Auth**
  When enabled, the terminal prompts for a user name and password before sending any radio data. When disabled, the terminal will use the machine authentication as set up in the firmware menus.

6) Cold Start

6) **Cold Start** resets all TE values to the stored configuration in CONFIG.DAT and performs a cold-start on the hand-held terminal. Press [Y] to reboot the terminal, or press any other key to return to the
1) **Set-up Parms** menu.

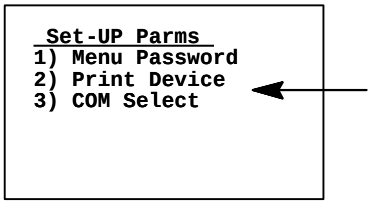
Depending on the setting in the firmware menu of the terminal, after rebooting, the terminal opens to the TRAKKER Antares 2400 Menu System.



```
  Cold Start    Enter
'Y'
 to Cold Start
 terminal:
```

7) More

7) **More** enables menu password protection for the TE 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 have this option disabled.



```
 Set-UP Parms
1) Menu Password
2) Print Device
3) COM Select
```

2) Print Device is for 6400

3) COM Select is for 248X, 5055

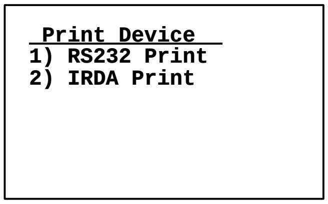
1) Menu Password

To enable the password, press [**1**], then type "3193693" for the fixed password set by Intermec Technologies Corporation. *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 terminal 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 that the menu password can be set via a custom configuration. See Chapter 4, "Customizing Your Configuration," for help on creating a custom configuration file.*

2) Print Device *(6400)*

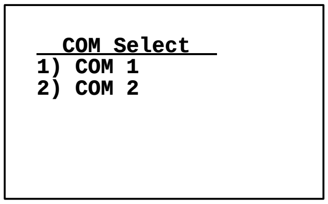
Establishes extended commands for the RS-232, IrDA, or RF printer drivers. Press [**1**] for the RS-232 driver *(default)* or [**2**] for the IrDA driver.



```
 Print Device
1) RS232 Print
2) IRDA Print
```

3) COM Select *(248X, 5055)*

Selects which communications port to use for extended commands, RS-232 communications, or media copy commands.

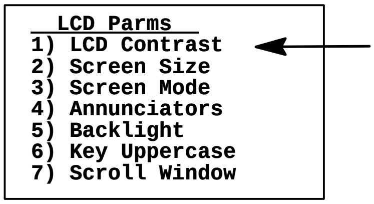


```
  COM Select
1) COM 1
2) COM 2
```

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.

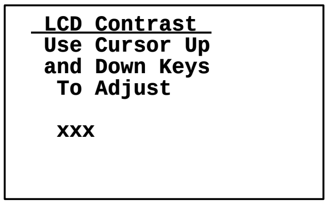


```
  LCD Parms
1) LCD Contrast
2) Screen Size
3) Screen Mode
4) Annunciators
5) Backlight
6) Key Uppercase
7) Scroll Window
```

1) LCD Contrast is for 59XX, 17XX
4) Annunciators is blank for 17XX
5) Backlight is for 59XX, 17XX

1) LCD Contrast (59XX, 17XX)

Use this option to adjust the contrast of the display to a more readable level (range 0–255). The terminal uses this setting as a baseline, automatically readjusting contrast as necessary, based on temperature changes. Use the up and down arrow keys to adjust the contrast, then press [Enter] to return to the **LCD Parms** screen.



```
 LCD Contrast
 Use Cursor Up
 and Down Keys
  To Adjust

  xxx
```

Note: 2) **Screen Size** is available for 2415, 2425, 2435A, 2455, 2475, and 248X Terminals on firmware 7.00 or greater and with Double-Byte Country Code equal to 00 (USA).

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:

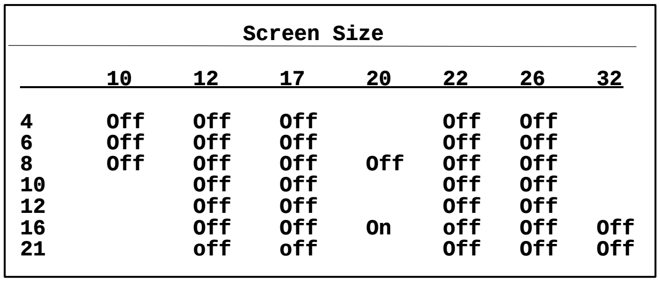
1. Press the arrow keys. Each key press moves the word **On** one position in the direction of the arrow key pressed. The word **On** indicates the current setting.
2. When the word **On** appears in the position corresponding to the desired screen size, press [Enter].

Note: For 2435A Terminals with Terminal Emulation version 6.60 or greater and firmware version 7.10 or greater, if icons are disabled in the firmware, then the 2435A Terminal uses the following screen sizes.

2415 and 2425 Terminal Screen Sizes

Choices for the 2415 and 2425 Terminals include the following. Only one option can be selected at a time. *(Default is 16 x 20)*

- 4 rows, with 10, 12, 17, 22, or 26 characters per row
- 6 rows, with 10, 12, 17, 22, or 26 characters per row
- 8 rows, with 10, 12, 17, 20, 22, or 26 characters per row
- 10 rows, with 12, 17, 22, or 26 characters per row
- 12 rows, with 12, 17, 22, or 26 characters per row
- 16 rows, with 12, 17, 20, 22, 26, or 32 characters per row
- 21 rows, with 12, 17, 22, 26, or 32 characters per row



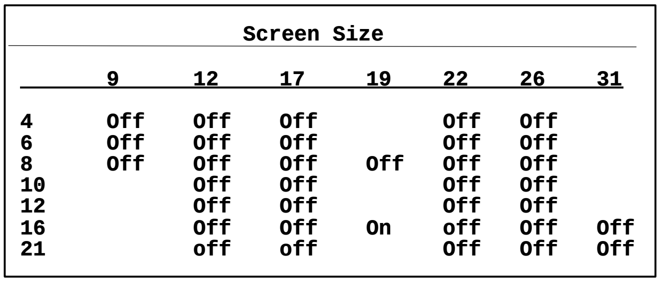
```
                 Screen Size
_____________________________________________
      10    12    17    20    22    26    32

4     Off   Off   Off         Off   Off
6     Off   Off   Off         Off   Off
8     Off   Off   Off   Off   Off   Off
10          Off   Off         Off   Off
12          Off   Off         Off   Off
16          Off   Off   On    off   Off   Off
21          off   off         Off   Off   Off
```

2435A Terminal Screen Sizes

For 2435A Terminals with Terminal Emulation version 6.60 or greater and firmware version 7.10 or greater, if icons are enabled in the firmware, then the 2435A Terminal uses the following screen sizes.

If the icons are disabled in the firmware *(default)*, then the 2435A Terminal uses the screen sizes as shown on page 76 for the 2415 and 2425 Terminals. Only one option can be selected at a time. *(Default is 16 x 19)*

- 4 rows, with 9, 12, 17, 22, or 26 characters per row
- 6 rows, with 9, 12, 17, 22, or 26 characters per row
- 8 rows, with 9, 12, 17, 19, 22, or 26 characters per row
- 10 rows, with 12, 17, 22, or 26 characters per row
- 12 rows, with 12, 17, 22, or 26 characters per row
- 16 rows, with 12, 17, 19, 22, 26, or 31 characters per row
- 21 rows, with 12, 17, 22, 26, or 31 characters per row

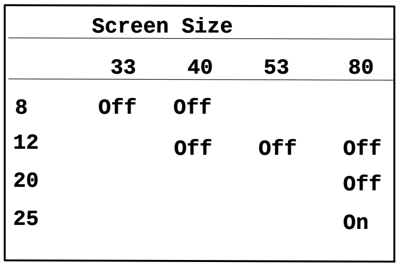


```
                  Screen Size
       9      12     17     19    22    26    31

4      Off    Off    Off          Off   Off
6      Off    Off    Off          Off   Off
8      Off    Off    Off    Off   Off   Off
10            Off    Off          Off   Off
12            Off    Off          Off   Off
16            Off    Off    On    off   Off   Off
21            off    off          Off   Off   Off
```

2455 Terminal Screen Sizes

Choices for the 2455 Terminal include the following. Only one option can be selected at a time. *(Default is 25 x 80)*

- 8 rows, with 33 or 40 characters per row
- 12 rows, with 40, 53, or 80 characters per row
- 20 rows, with 80 characters per row
- 25 rows, with 80 characters per row

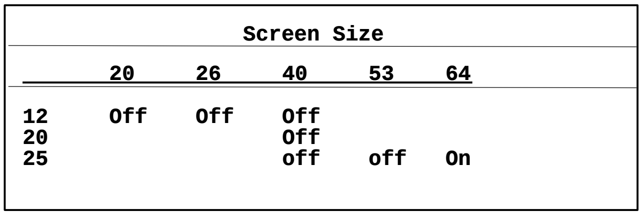


```
      Screen Size
        33    40    53     80
8      Off   Off
12           Off   Off    Off
20                        Off
25                        On
```

2475 and 248X Terminal Screen Sizes

Choices for the 2475 and 248X Terminals include the following. Only one option can be selected at a time. *(Default is 25 x 64)*

- 12 rows, with 20, 26, 40, 53, or 64 characters per row
- 20 rows, with 40 characters per row
- 25 rows, with 40, 55, or 64 characters per row

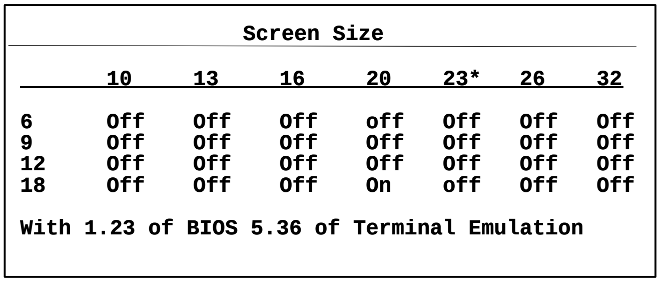


```
                    Screen Size
______________________________________________
       20     26     40     53    64

12     Off    Off    Off
20                   Off
25                   off    off   On
```

6400 Computer Screen Sizes

Choices for the 6400 Computer include the following. It is also possible to have selected 5, 8, 10, and 16 for number of rows. This depends on what size of display you have selected. Only one option can be selected at a time. *(Default is 18 x 20)*

- 6 rows, with 10, 13, 16, 20, 23, 26, or 32 characters per row
- 9 rows, with 10, 13, 16, 20, 23, 26, or 32 characters per row
- 12 rows, with 10, 13, 16, 20, 23, 26, or 32 characters per row
- 18 rows, with 10, 13, 16, 20, 23, 26, or 32 characters per row

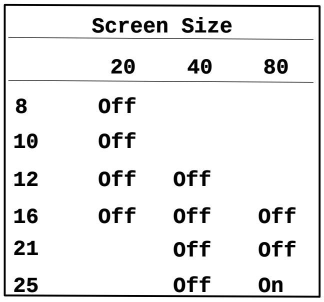


```
                    Screen Size
__________________________________________________
       10     13     16     20     23*    26     32

6      Off    Off    Off    off    Off    Off    Off
9      Off    Off    Off    Off    Off    Off    Off
12     Off    Off    Off    Off    Off    Off    Off
18     Off    Off    Off    On     off    Off    Off

With 1.23 of BIOS 5.36 of Terminal Emulation
```

5055 PC Screen Sizes

Choices for the 5055 PC include the following. It is also possible to select 5, 8, 10, and 16 for number of rows. This depends on what size of display you have selected. Only one option can be selected at a time.
(Default is 25 x 80)

- 8 rows, with 20 columns per row
- 10 rows, with 20 columns per row
- 12 rows, with 20 or 40 columns per row
- 16 rows, with 20, 40, or 80 columns per row
- 21 rows, with 40 or 80 columns per row
- 25 rows, with 40 or 80 columns per row

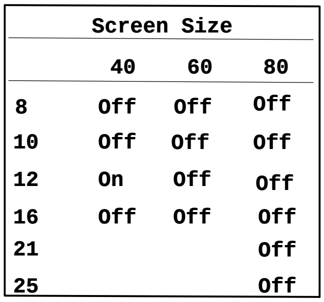


	Screen Size		
	20	40	80
8	Off		
10	Off		
12	Off	Off	
16	Off	Off	Off
21		Off	Off
25		Off	On

59XX Terminal Screen Sizes

Choices for the 59XX Terminal includes the following. Only one option can be selected at a time. *(Default is 12 x 40)*

- 8 rows, with 40, 60, or 80 columns per row
- 10 rows, with 40, 60, or 80 columns per row
- 12 rows, with 40, 60, or 80 columns per row
- 16 rows, with 40, 60, or 80 columns per row
- 21 rows, with 80 columns per row
- 25 rows, with 80 columns per row

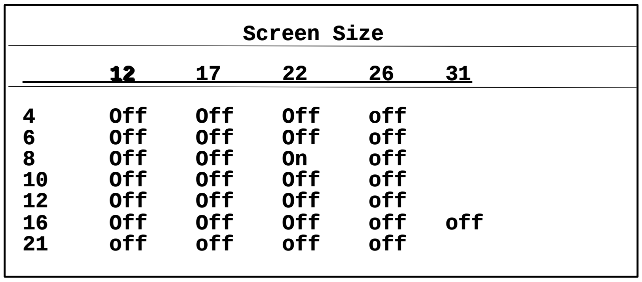


	Screen Size		
	40	60	80
8	Off	Off	Off
10	Off	Off	Off
12	On	Off	Off
16	Off	Off	Off
21			Off
25			Off

17XX Terminal Screen Sizes

Choices for the 17XX Terminal include the following. Only one option can be selected at a time. *(Default is 8 x 22)*

- 4 lines, with 12, 17, 22, or 26 characters per line
- 6 lines, with 12, 17, 22, or 26 characters per line
- 8 lines, with 12, 17, 22, or 26 characters per line
- 10 lines, with 12, 17, 22, or 26 characters per line
- 12 lines, with 12, 17, 22, or 26 characters per line
- 16 lines, with 12, 17, 22, 26, or 31 characters per line
- 21 lines, with 12, 17, 22, or 26 characters per line

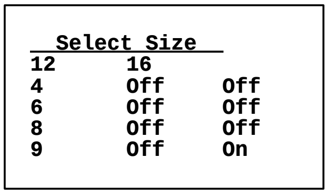


```
                 Screen Size
______________________________________________
      12    17    22    26    31
______________________________________________
4     Off   Off   Off   off
6     Off   Off   Off   off
8     Off   Off   On    off
10    Off   Off   Off   off
12    Off   Off   Off   off
16    Off   Off   Off   off   off
21    off   off   off   off
```

11XX Terminal Screen Sizes

Choices for the 11XX Terminal include the following. Only one option can be selected at a time. *(Default is 9 x 16)*

- 4 rows, with 12 or 16 characters per row
- 6 rows, with 12 or 16 characters per row
- 8 rows, with 12 or 16 characters per row
- 9 rows, with 12 or 16 characters per row

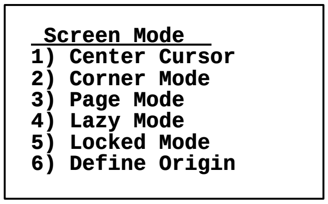


```
  Select Size
12     16
4      Off    Off
6      Off    Off
8      Off    Off
9      Off    On
```

3) Screen Mode

3) **Screen Mode** selects the cursor position and movement as you scroll through data in the display buffer. This buffer stores data in a standard CRT format (as sent from the host computer). Since the terminal display is smaller than a CRT, these 3) **Screen Mode** options optimize your view of information (data) on the display. *Default is 2) Corner Mode.*

Use arrow keys to scroll (or "move") the view port of the screen 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 terminal 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 terminal 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

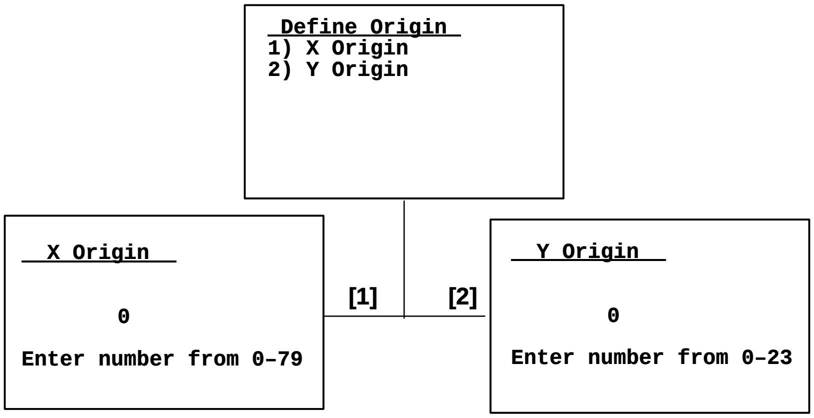
When this is selected the screen view is locked to the upper left-hand corner of the display. Any characters selected outside of the display window size are written to the screen but are not visible. The windowing keys do not move the visible window.

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 terminal 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*

.



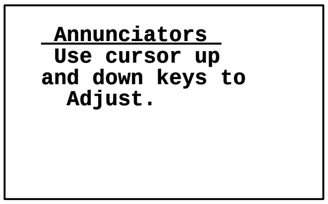
```
 Define Origin
1) X Origin
2) Y Origin
```

```
  X Origin

      0

Enter number from 0–79
```

```
  Y Origin

      0

Enter number from 0–23
```

4) Annunciators (Blank for 17XX, 11XX)

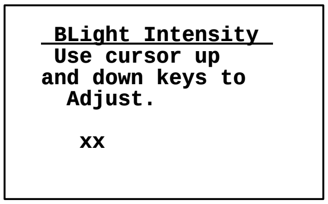
4) **Annunciators** selects and changes the location of the annunciators on the display screen. Press the up or down arrows to position the annunciators around each of the four display corners, once as a line and once as a box. You can also select "Stealth Mode" which displays the annunciators only when you make a change and then hides it when you press a key. *Default is "Stealth Mode."*



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

5) Backlight (59XX, 17XX)

This adjusts the intensity (brightness) of the display backlight. The intensity can be set from 0 (off) to 15 (fully on). Keep in mind that the backlight uses considerable battery power. To conserve battery power, keep the intensity and duration of the backlight as low as possible. Use the up and down arrow keys to adjust the intensity of the backlight. Press the up arrow key to increase the intensity, the down arrow key to decrease. Press [Enter] when finished to return to the **LCD Parms** screen.



```
  BLight Intensity
  Use cursor up
 and down keys to
   Adjust.

    xx
```

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) Scroll Window

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

1) Tab Size

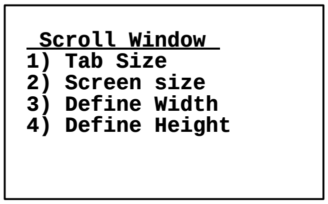
Moves the cursor by the amount configured for 3) **Define Width** and 4) **Define Height**.

2) Screen Size

Causes the cursor to move by the virtual screen size selected in 2) **LCD Parms** under the **Main Menu**.

3) Define Width and 4) Define Height

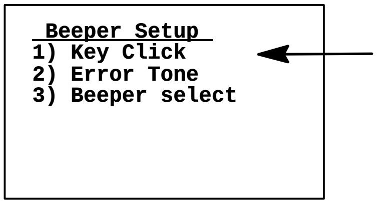
These options customize the x-axis and y-axis (up and down) movement of the cursor when 1) **Tab Size** is selected. *Default is 8.*



```
 Scroll Window
1) Tab Size
2) Screen size
3) Define Width
4) Define Height
```

3) Beeper Setup

The 3) **Beeper Setup** menus adjust the beeper tones for key clicks (presses) and error conditions. *For 17XX Terminals*, 3) **Beeper Select** directs the audio output to the internal buzzer or to a headset.



```
 Beeper Setup
1) Key Click
2) Error Tone
3) Beeper select
```

1) Key Click is for
6400, 5055, 59XX, 17XX, 11XX
3) Beeper Select
is for 6400, 5055, 17XX, 11XX

1) Key Click (6400, 5055, 59XX, 17XX, 11XX)

Use 1) **Key Click** to adjust the frequency and length (duration) of the sound made when you press a key.

To adjust the key click tone, press the appropriate option, then press the up or down arrow keys to make the desired adjustment. Each key press is accompanied by a tone at the new frequency or length. Press [Enter] to confirm the selection and return to the previous 1) **Key Click** screen.

1) Volume (59XX)

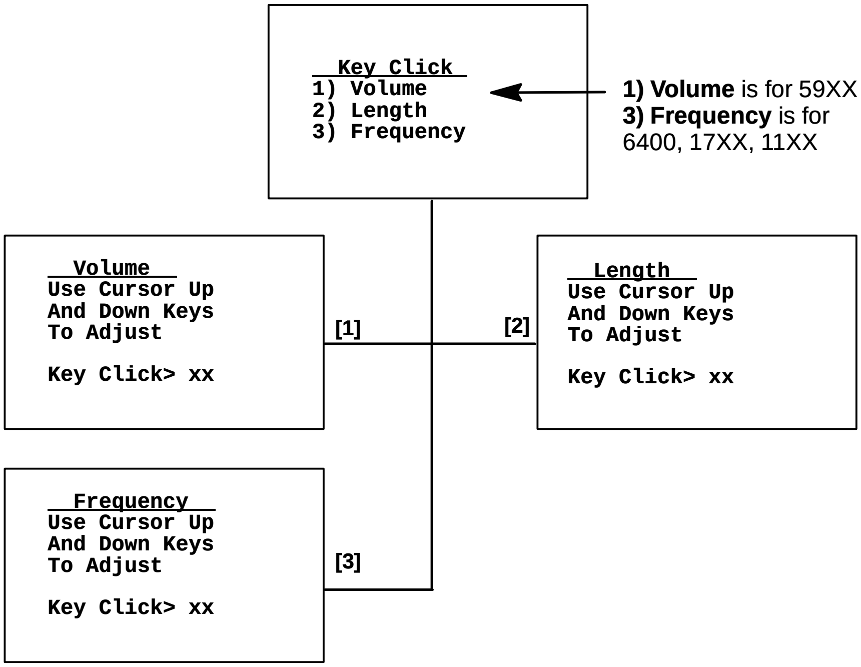
Press [1], then the up and down arrow keys to adjust the length in steps from 0 to 32.

2) Length

Press [2], then the up and down arrow keys to adjust the length in steps from 0 to 10.

3) Frequency (6400, 17XX, 11XX)

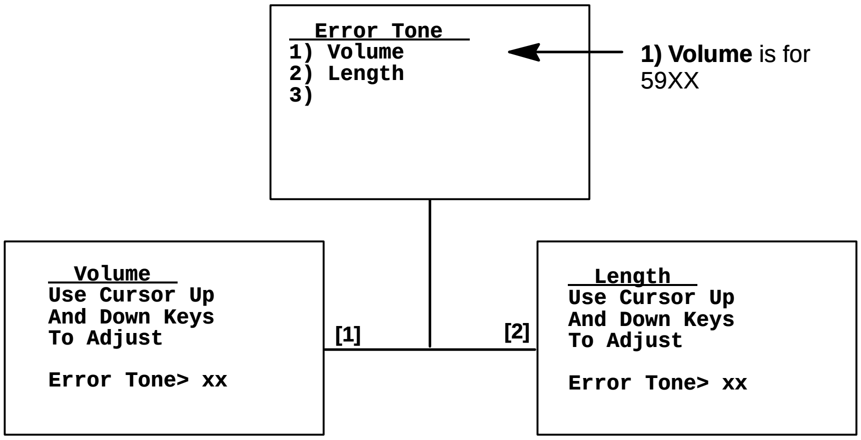
Press [3], then the up and down arrow keys to adjust the tone frequency in steps from 0 to 32.



1) Volume is for 59XX
3) Frequency is for
6400, 17XX, 11XX

2) Error Tone

2) **Error Tone** adjusts the length and frequency of the tone made when an error occurs (for example, pressing an invalid key). The **2) Error Tone** adjustments, and the procedures for making them are identical to the 1) **Key Click** adjustments. Press **2) Length** to set the length. The range is 1 through 10, which equates to a 500 to 5000 ms beep for all errors. *Default error tone length is 3 (or 1500 ms).*



1) Volume is for
59XX

3) Beeper Select (6400, 5055, 17XX, 11XX)

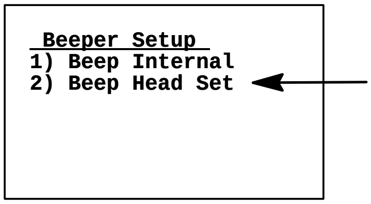
For 17XX, 11XX, both buzzers are not allowed to beep simultaneously when using a headset. If both options are enabled, the head set buzzer will beep first before the internal buzzer. If you are using the RS-232 port for communications, enable 1) **Beep Internal** and disable 2) **Beep Head Set**.

1) Beep Internal

Press [1] to direct the audio to the internal beeper. *Default is enabled.*

2) Beep Head Set (17XX, 11XX)

Press [2] to direct sound to the headset.

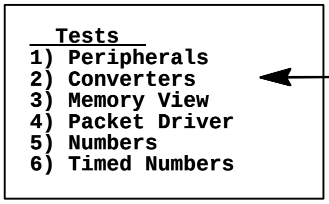


```
 Beeper Setup
1) Beep Internal
2) Beep Head Set
```

2) Beep Head Set is for 17XX, 11XX

4) Tests

Intermec Systems Engineers use the 4) **Tests** menus to verify terminal operation during environmental stress tests, peripherals, the display, RF communications, and memory.



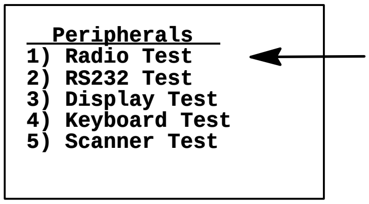
```
  Tests
1) Peripherals
2) Converters
3) Memory View
4) Packet Driver
5) Numbers
6) Timed Numbers
```

2) Converters is for 59XX

3) Memory View and
4) Packet Driver
are for 6400, 5055, 59XX, 17XX, 11XX

1) Peripherals

Select 1) **Peripherals** to test terminal peripheral devices, such as radio, RS-232, display, keyboard, and scanner. Detailed descriptions of each peripheral test follows.



```
  Peripherals
1) Radio Test
2) RS232 Test
3) Display Test
4) Keyboard Test
5) Scanner Test
```

1) Radio Test is for
6400, 5055 — 802.11 (Lucent) radios

2) RS232 Test,
3) Display Test, and
4) Keyboard Test are for
6400, 5055, 59XX, 17XX, 11XX

1) Radio Test (802.11 (Lucent) Radios — 6400, 5055)

Several factors influence the rate at which a frame is transmitted. First, the radio's transmission rate is configured in the **net.cfg** file. The keyword **Transmit_Rate** in **net.cfg** holds a numeric value for the transmission rate.

The transmission rate is also affected by the characteristics of the RF link. If the radio encounters problems sending a higher transmission rate because the RF link is weak, busy, or noisy, the radio may retry the transmission at a lower data rate. Thus, the **TxRate** reported may be lower than the configured transmission rate. Only the transmission rate of the most recent and successful transmission is reported.

Transmit_Rate values and their respective supported transmission rates available in the high speed Lucent radios are as follows:

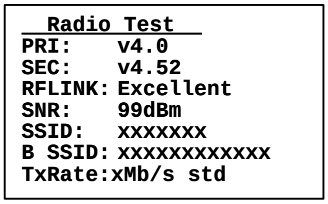
1	Fixed Low (1 Mb/s)
2	Fixed Standard (2 Mb/s)
3	ARS High (11 Mb/s *default*)
4	Fixed Medium (5.5 Mb/s)
5	Fixed High (11 Mb/s)
4	ARS Standard (2 Mb/s)
4	ARS Medium (5.5 Mb/s)

- **PRI**
  The primary version of low-level radio firmware.
- **SEC**
  The secondary version of the low-level radio firmware.
- **RFLINK**
  A general description of the RF connection overall quality between Lucent radios in the terminal and radios in the Access Point. The Lucent radio grades the RF link on a scale from 0–92, representing the RF channel Signal-to-Noise Ratio. The RF link quality descriptions are assigned as: 0–23 "Adequate," 24–47 "Good," or 48–92 "Excellent."
- **SNR** *(Signal-to-Noise Ratio)*
  Calculated by dividing the RF signal level by the RF noise level. The result is a number that indicates the overall communications quality of the RF link and is expressed in dBm.
- **SSID** *(Service Set ID)*
  A 32-character, alphanumeric string that identifies the service set, or infrastructure, with which the terminal is currently associated. The SSID is a user-configurable parameter and is configured by the keyword **WaveLAN_Network_Name** in the **net.cfg** file. Setting the keyword to ANY allows the station to associate with any service set.
- **B SSID** *(Basic Service Set ID)*
  The 6-byte MAC address of the Lucent radio in the Access Point with which the station is associated. The Lucent radio in the terminal may associate with a non-Lucent (but 802.11-compliant) radio in an Access Point. Looking at the first three bytes of the MAC address (OUI) can help identify the Access Point radio's manufacturer.

Example

An OUI of "00601D" points to a Lucent radio, while "0020A6" indicates a Proxim radio.

- **TxRate**
  The data rate (in Mb/s) of the last message transmitted by the Lucent radio. The default TXRate of 2 Mb/s and is displayed when the terminal starts up and has not transmitted a frame yet. High speed Lucent radios are capable of data rates of 1 Mb/s, 2 Mb/s, 5.5 Mb/s, and 11 Mb/s. The TxRate 5.5 Mb/s data rate is displayed as 6 Mb/s. This occurs because the Lucent radio reports an integer value for current transmission data rate, so the data rate of 5.5 Mb/s is rounded up to 6 Mb/s.



```
  Radio Test
PRI:    v4.0
SEC:    v4.52
RFLINK: Excellent
SNR:    99dBm
SSID:   xxxxxxx
B SSID: xxxxxxxxxxxx
TxRate:xMb/s std
```

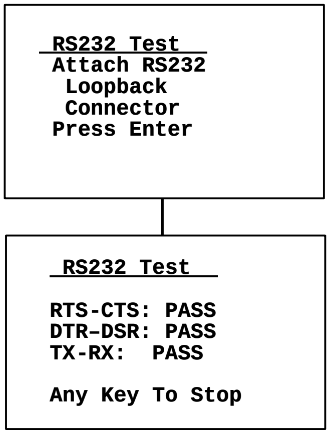
Primary version of Lucent driver
Secondary version of driver
RF link quality
Signal-to-Noise ratio
Network name in use
MAC address of access point
Data rate

2) RS232 Test (6400, 5055, 59XX, 17XX, 11XX)

Tests the data communication port on the terminal, requiring a special loop-back connector that links the following input and output pins on the data connector:

- TX output to the RX input
- DTR output to the DSR input
- CTS output to the RTS input

To run the test, select 2) **RS232 Test**, attach the loop-back connector to the port connector on the terminal, then press [Enter]. The display changes to the second **RS232 Test** menu shown below. The condition of each line-pair displays as passing or failing the test. If any test fails, return the terminal for service.



```
 RS232 Test
 Attach RS232
  Loopback
  Connector
 Press Enter
```

```
 RS232 Test

RTS-CTS: PASS
DTR–DSR: PASS
TX-RX:  PASS

Any Key To Stop
```

3) Display Test (6400, 5055, 59XX, 17XX, 11XX)

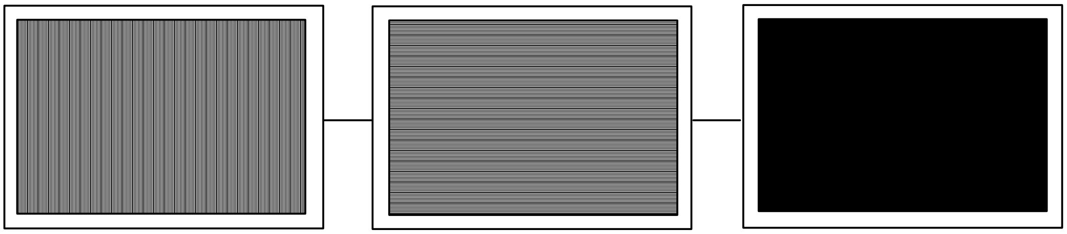
3) **Display Test** tests the operation of and activates each element in the terminal LCD panel. If a line appears broken, or there are gaps in the display, replace the LCD panel.

1 From the **Display Test** menu, press [Enter] to start the test. The display screen is painted with a series of vertical lines. Visually inspect the lines. All lines should appear clear and complete (no broken segments).

2 Press [Enter] to move to a screen drawn with horizontal lines. Visually inspect the lines using the same criteria.

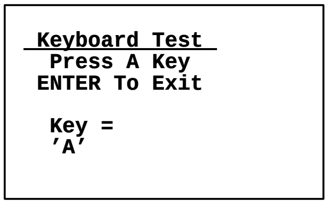
3 Press [Enter] to move to the final screen. Visually inspect the painted black lines which should be uniformly black for monochrome displays and uniformly white for color displays.

4 Press [Enter] to return to the **1**) **Peripherals** menu.



4) Keyboard Test (6400, 5055, 59XX, 17XX, 11XX)

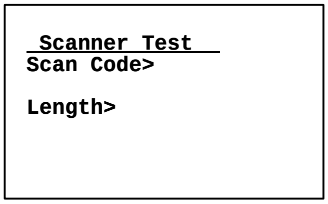
Tests each key on the terminal keypad or 5055 external keyboard. Press each key, except the [Enter] key, and a character corresponding to the pressed key should appear on the display.



```
 Keyboard Test 
  Press A Key
 ENTER To Exit

  Key =
  'A'
```

5) Scanner Test

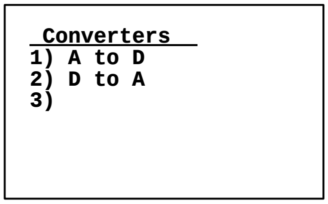
This option tests the operation of an attached bar code scanner. From this menu, scan an enabled bar code. The bar code and the bar code length should appear on the display to pass. Press any key to exit.



```
 Scanner Test 
Scan Code>

Length>
```

2) Converters (59XX)

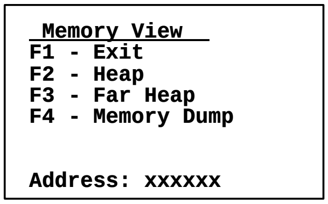
Use this option to test the analog-to-digital (A to D) and the digital-to-analog (D to A) converters. The **RSSI Test** (*not available on all radios*) offers a choice of graphics displays of the received radio signals. Press any key to exit.



```
 Converters
1) A to D
2) D to A
3)
```

3) Memory View (6400, 5055, 17XX, 11XX)

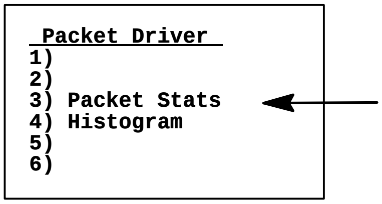
This menu is reserved for engineering test and evaluation. Press [**F1**] to exit out of this test, [**F2**] to view the heap, [**F3**] to view the far heap, or [**F4**] to do a memory dump.



```
 Memory View
F1 - Exit
F2 - Heap
F3 - Far Heap
F4 - Memory Dump


Address: xxxxxx
```

4) Packet Driver (6400, 5055, 59XX, 17XX, 11XX)

4) **Packet Driver** tests accuracy of data transmissions to and from the terminal.

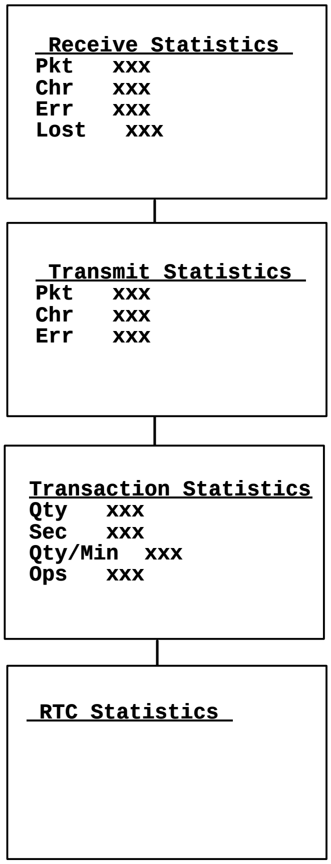


```
 Packet Driver
1)
2)
3) Packet Stats
4) Histogram
5)
6)
```

3) Packet Stats is for 6400, 5055, 59XX, 17XX, 11XX

3) Packet Stats (Statistics) (6400, 5055, 59XX, 17XX, 11XX)

Shows the number of packets sent and received, number of errors, and number of packets dropped.



```
 Receive Statistics
Pkt   xxx
Chr   xxx
Err   xxx
Lost   xxx
```

```
 Transmit Statistics
Pkt   xxx
Chr   xxx
Err   xxx
```

```
Transaction Statistics
Qty   xxx
Sec   xxx
Qty/Min  xxx
Ops   xxx
```

```
 RTC Statistics
```

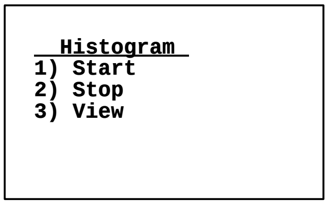
*** See Text**
'Histogram Options'

4) Histogram

An Intermec engineer may ask you to access this menu if your terminal has problems. From this menu you can provide the Intermec system engineer with vital information about your unit.

The actual menus are not shown in this manual, however some of the general terminology is shown below. Several of the menu options, such as "Clear. . ., " are password-protected and are only accessible if the engineer provides you with the correct password. An Intermec system engineer will work with you on these selections. Terminology you might see:

- avg tx time
- avg trans time
- ptt (push-to-talk) count
- addrs on list
- poll no data
- send retries
- rcv retries
- reset count
- interval time
- time bucket #



```
  Histogram
1) Start
2) Stop
3) View
```

5) Numbers

5) **Numbers** checks the operation of the terminal microprocessor. During the test, the terminal display is filled with hexadecimal numbers that move across the screen horizontally and scroll vertically. Character movement indicates that the processor is still running. To stop the test, press any key.

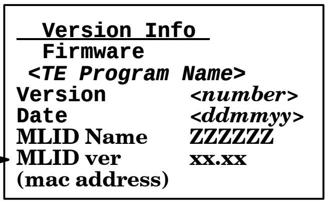
This test is helpful in detecting lockups during severe operating conditions.

6) Timed Numbers

6) **Timed Numbers** does the same thing as 5) **Numbers** except that it keeps track of the time in seconds that it took to run the test. The test terminates when a key is pressed or when 65,536 numbers are displayed.

5) Version Info (Information)

5) **Version Info** shows the name, version, and release date of the program you are using, the version of the radio driver, and the MAC address of the 6400, 5055, or 11XX Terminal. See Chapter 1, "*Getting Started*," for TE program names.

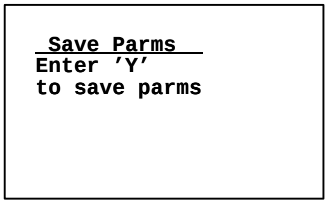


```
  Version Info
  Firmware
 <TE Program Name>
Version       <number>
Date          <ddmmyy>
MLID Name     ZZZZZZ
MLID ver      xx.xx
(mac address)
```

MLID Name and **MLID ver** are for 6400, 5055

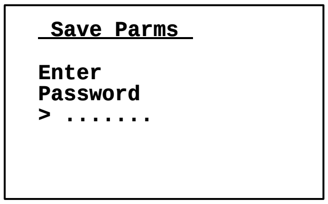
6) Exit Menus

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



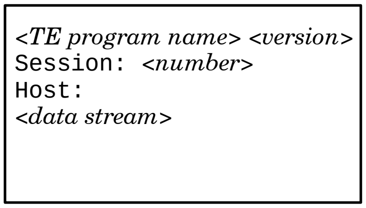
```
 Save Parms
Enter 'Y'
to save parms
```

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



```
 Save Parms

Enter
Password
> .......
```

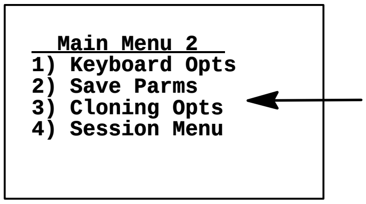
After you enter the correct password, your settings are saved to Flash, and the terminal may reboot. (Some changes automatically reboot the terminal.) If you press a key other than [**Y**], you exit the menus and the new settings are *not* saved. In this case, the new settings are lost when you reboot your terminal. When you exit the TE configuration menus, the following information appears:



```
<TE program name> <version>
Session: <number>
Host:
<data stream>
```

7) More (Main Menu 2)

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

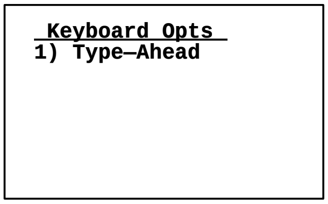


```
  Main Menu 2
1) Keyboard Opts
2) Save Parms
3) Cloning Opts
4) Session Menu
```

3) Cloning Opts is for 59XX, 17XX, 11XX
4) Session Menu is for 2415 (WTP), 2425 (WTP), 2435A (WTP), 2455 (WTP), 2475 (WTP), 248X (WTP), 6400, 5055, 17XX, 11XX

1) Keyboard Opts

Use 1) **Keyboard Opts** > 1) **Type-Ahead** to key in information when the terminal cannot immediately send data to the host computer.
1) **Type-Ahead** enters information when the terminal cannot immediately send data to the host.



```
 Keyboard Opts
1) Type–Ahead
```

This stores keystrokes after the Input Inhibited annunciator (below) appears on the status line, and then saves them for the next input field. Type-ahead is enabled by default.

X Input Inhibited annunciator

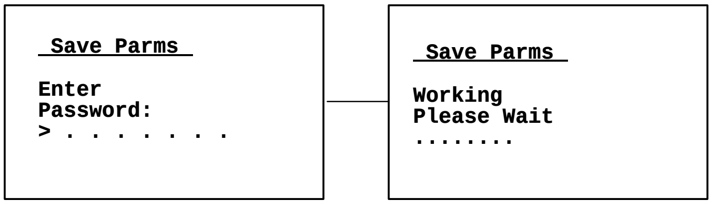
2) Save Parms (Parameters)

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

Note: Use this option sparingly. Each time it is used, additional memory space is occupied because previously saved changes are not erased. The memory cannot be recovered on 11XX, 17XX, and 59XX Terminals without reflashing.

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 can be altered 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 terminal will return to 2) **Save Parms** screen without saving your entries.



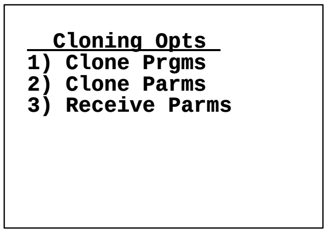
```
 Save Parms 

Enter
Password:
> . . . . . . .
```

```
 Save Parms 

Working
Please Wait
........
```

3) Cloning Opts (59XX, 17XX, 11XX)

Note: 3) **Cloning Opts** is disabled for 59XX Terminals.

These options transfer the application program or parameter settings from one terminal to another. All three choices are password-protected, with **CR52401** as the default password. To perform this function you need a cloning cable and both terminals set for cloning.



```
  Cloning Opts 
1) Clone Prgms
2) Clone Parms
3) Receive Parms
```

1 Connect the two terminals together using the cloning cable. Use cable P/N: 216-911-001, with radio terminals containing 12-volt Flash *(see illustration on page 96)* for connecting with the NC1100 Power Supply and Charger *(see illustration on page 97)*.

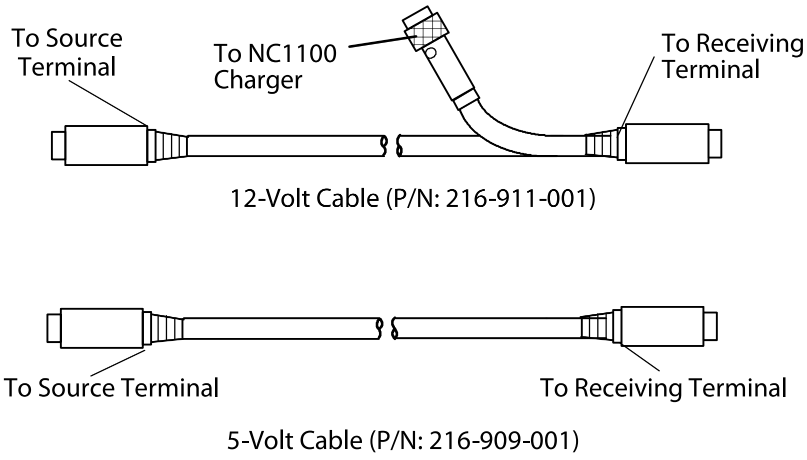
2 Plug the NC1100 Power Supply and Charger into an ac outlet. Use cable P/N: 216-909-001, with radio terminals containing 5-volt Flash. You may use the 12-volt Flash cable, but this is optional.

3 Turn on the source terminal from which to copy the application.

4 Press 3) **Cloning Opts**, then 1) **Clone Prgms**.

5 Key in the password.

6 Hold down the [**I**] key while turning on the terminal receiving the program. The application program will copy into the receiving terminal.



12-Volt Cable (P/N: 216-911-001)

5-Volt Cable (P/N: 216-909-001)

This illustrations shows the FLASH cloning cable connections as they pertain to step 1 on the previous page.

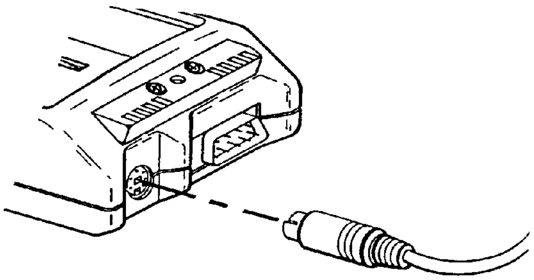
To clone parameters from one terminal to another, first ensure the desired parameters are already set in the terminal you wish to transfer from. Then:

1 Connect the terminals together using the cloning cable. Data Terminals with 12-volt FLASH use P/N: 216-911-001. Data Terminals with 5-volt FLASH use P/N: 216-909-001.

2 With both terminals turned ON, press 3) **Cloning Opts** on both terminals.

3 On the source terminal *(the terminal with the parameters from which to copy)*, press 2) **Clone Parms**.

4 On the receiving terminal, press 3) **Receive Parms**. If using the cable that requires the NC1100 Power Supply and Charger, make sure it is connected to the NC1100 Power Supply and Charger.



Follow the alignment from the NC1100 power supply to the cloning cable connection. This illustration pertains to step 1 on page 95.

4) Session Menu (2415, 2425, 2435A, 2455, 2475, and 248X with WTP, 6400, 5055, 59XX, 17XX, 11XX)

4) **Session Menu** defines different host communication sessions and designates a “hot key” that allows switching quickly between the different sessions.

1) Switch

Use this option to change the current session. The current (or intended) session must be identified (or designated) before setting Parameters.

2) Set Hot Key

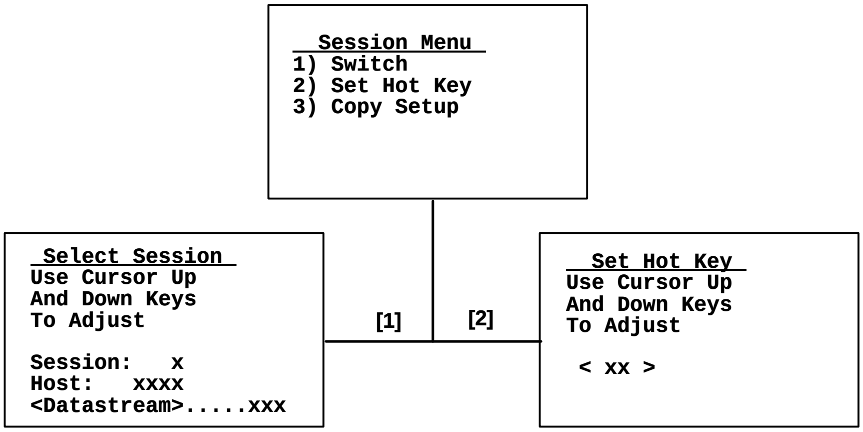
(Available on units that support session-switching)

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 background session to the current session.



```
 Session Menu
1) Switch
2) Set Hot Key
3) Copy Setup
```

```
 Select Session
Use Cursor Up
And Down Keys
To Adjust

Session:   x
Host:   xxxx
<Datastream>.....xxx
```

```
  Set Hot Key
Use Cursor Up
And Down Keys
To Adjust

 < xx >
```

Restarting Terminal Emulation

Note: The reset firmware bar code is not supported on the 59XX, 17XX, or 11XX Terminals. Instead, do the cold-start function to exit the TE application and return to the Windows CE main menu.

You need to restart your TE application if you are having problems or if you want to reconnect to the host with a new configuration. Restarting your TE application also clears the auto-login information, preventing another user from establishing a TE session using your login information.

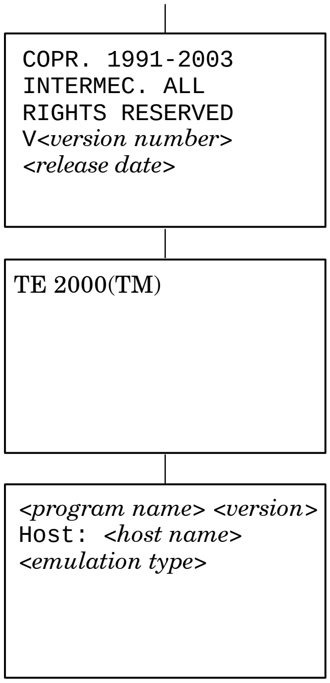
You can restart your TE application by either scanning the following **Reset Firmware** bar code or by using the **6) Cold Start** option in the TE configuration **Main Menu**. Both methods reset all terminal firmware and the application and run the application in a new session.

Reset Firmware

-.

The terminal restarts your TE application using the configuration saved in CONFIG.DAT, or follow this procedure:

1 At the third initialization screen or anywhere in a TE session, access the Main Menu by pressing the appropriate key combination (*see page 4*).



```
COPR. 1991-2003
INTERMEC. ALL
RIGHTS RESERVED
V<version number>
<release date>
```

```
TE 2000(TM)
```

```
<program name> <version>
Host: <host name>
<emulation type>
```

Note: If you are not connected to the DCS 300 or host, or have problems accessing the **Main Menu**, reset your terminal to go to the initialization screens.

2 Select 1) **Setup Parms**, then enter the “cr52401” password. *Do not press Enter.*

3 Select 6) **Cold Start**, press [Y] when prompted to restart, then start using the application or change the TE configuration.

See the terminal’s user guide or manual for information on how to recover from a lock-up condition.

2415, 2425, 2435A, 2455, 2475, 248X Terminals

The following instructions are specific for 2415, 2425, 2435A, 2455, 2475, and 248X Terminals.

You can configure several options for 3270 TE on your Trakker Antares® Terminal, including:

- UDP Plus or TCP/IP communications
- 3270 options
- Main Menu password

Note: You can also set the password for UDP Plus Terminals from the DCS 300.

You can access the TE configuration menus when the initialization screens appear or once you establish a TE session. The TE initialization screens appear each time you reboot the terminal or restart your application.

Note: Some parameters for 2415, 2425, 2435A, 2455, 2475, and 248X Terminals are available through the TRAKKER Antares 2400 Menu System. They are not reproduced in the TE configuration menus. See the terminal’s user manual for information about the menu system.

6400 Computer

The terminal emulation screens support terminals running Intermec Terminal Emulation or 6400 TCP/IP software. This describes the menus used to set operating and scanning parameters for the hand-held computer. Additional information can be found in the technical overview for your emulation program.

Note: Ensure there is a fully-charged battery pack in the 6400 Computer before setting parameters.

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**. To open the **Main Menu**, press the [Yellow] shift, then the [Blue] shift key. Briefly, these keys generally operate as:

- [Blue] shift, for functions labeled by blue legends above the keys.
- [Green] shift, for functions labeled by green legends above the keys.
- [Yellow] shift, for functions labeled by yellow legends above the keys.
- [Blue], then [Green] shift is CAPS LOCK.
- [Blue], then [Yellow] shift is Alphabetic Lock *(6400 Computer 41-key units).*
- [Yellow], then [Blue] shift is Terminal Emulation menus; your particular emulation software probably treat the keys for contrast control, backlighting and other functions differently.

To Exit Emulation Mode and Return to DOS

Do a cold-restart to exit the current emulation program and return back to the DOS keyboard layout. Press [**Y**], press [Enter] several times to put you at the DOS C:\ prompt. To exit to the beginning of the current emulation, press and hold the [Blue] and [Yellow] shift keys, and both [**ENT**] keys to do the hard reset.

5055 Data Collection PC

Caution

If you are running terminal emulations **and ever update the DOS load on the internal SanDisk IDE drive on the 5055 PC, be aware that when updating 50DS1000, avoid copying the CONFIG.SYS and AUTOEXEC.BAT files. The new versions of these two files may interfere with existing applications, such as Terminal Emulations. If affected, Terminal Emulations will automatically run on boot up.**

In all other situations, do **copy the CONFIG.SYS and AUTOEXEC.BAT files.**

This contains information about the terminal emulations with keyboards supported for the Intermec® 5055 Data Collection PC.

The 5055 PC supports terminal emulations with rugged keyboards. FWP650H0 is DOS-based and is not compatible with Windows DOS prompts.

An external PS/2-compatible keyboard is required for configuring and using terminal emulations.

Terminal emulation font files are needed to generate the various character fonts. See the related terminal emulation programmer's reference guide for additional information.

Programs Used to Create Terminal Emulation Menus

The Terminal Emulation Menu Screens are provided to support terminals running Intermec Terminal Emulation software. This describes the menus that set operating and scanning parameters for the 5055 PC. Additional information can be found in the programmer's guide or technical overview for your emulation program.

When you power on the 5055 PC, the AUTOEXEC.BAT file executes the following programs, in sequential order, to create the terminal emulation menus. Listed are their descriptions.

- **LSL**
  Creates the link layer for the protocol stack.
- **rl2pcm**
  Includes the Proxim radio driver for the Proxim RangeLAN2 radio, ***or***
  wvlan43
  Includes the WaveLAN/IEEE radio driver for the 802.11b HR radio.
- **6500ikps**
  Installs multitasking, beeper, and key remapping TSRs.
- **wtppkt**
  Includes the WTP Packet driver program if a WTP link accesses the host computer, ***or***
  ethdrv and **odipkt**
  Includes the Ethernet and ODI Packer driver programs if a TCP/IP link and a controller are used to gain access to the host computer.

- **65scn7b -c1**
  Activates the scanner driver.
- **6500dbcs**
  Intercepts INT10 and display characters on the display.
- **fwp650h0**
  Activates the terminal emulation program.

Press <Alt>, then [M] on the external keyboard to access the setup menus.

Note: On terminal emulation keyboards, use the yellow key in place of the <Alt> key. For example, to access the Setup Menus, press [Yellow], then [M].

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**.

To open the **Main Menu**, press the yellow shift, then the blue shift (MENU) key. Briefly, these keys generally operate as:

- [Blue] shift, for functions labeled by blue legends above the keys.
- [Green] shift, for functions labeled by green legends above the keys.
- [Yellow] shift, for functions labeled by yellow legends above the keys.
- [Blue], then [Green] shift is CAPS LOCK.
- [Yellow], then [Blue] shift is Terminal Emulation menus; your particular emulation software probably treat the keys for contrast control, backlighting and other functions differently.

To Exit Emulation Mode and Return to DOS

Do a cold-restart to exit the current emulation program and return back to the DOS keyboard layout. Press [Y], then press [Enter] several times to arrive at the DOS C:\ prompt. To exit to the beginning of the current emulation, press and hold the [Blue] and [Yellow] shift keys, and both [ENT] keys to do the hard reset.

4 Customizing Your Configuration

This chapter describes the procedures you can use to customize the standard Native TE program by Intermec Technologies Corporation. You customize the TE program by creating or modifying configuration files, and then downloading them to your terminal to do the following:

- Use the auto-login feature to send the same login information each time you login to the host *(page 104)*.
- Create a custom parameter set-up file to download a customized file to all terminals so they have the same setup information *(page 112)*.
- Change the text of TE configuration menus or system messages *(page 140)*.
- Preinitialize the Native TE program *(page 141)*.
- Remap national characters *(page 147)*.
- Substitute national characters *(page 148)*.

Using the Auto-Login Feature

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

To use the auto-login feature, you need to develop an auto-login script file and load the auto-login script file on the terminal. These steps are covered on the following pages along with a list of the necessary 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 terminals 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 will not be 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 109.

- **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 will be 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.
- **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 embedded IBM mnemonics in Native TE.

- **Pause “xxxxx”**
  Delays the terminal for x milliseconds, halts terminal 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. For additional information, see Note below.
- **Keyboard “1”**
  Enables the keyboard. Key presses are processed. The keyboard is enabled by default. For additional information, see Note below.
- # (pound symbol)
  Documents the script file. Text following a # (pound) symbol is considered a comment unless the # symbol is in a quoted string.

Note: Input the **Keyboard “0”** or **Keyboard “1”** command into the autologin script file after the **PromptSessionStart** command (if present) and the **HostName** command (if present). Also turn on the keyboard command before another **HostName** command is found in the file.

Search Strings

Some auto-login search string limitations are as follows. You can use line wrapping to look for unique strings.

- The searches are case sensitive.
- The maximum search string length is 20 characters.
- Each **WaitFor** command searches the entire screen from the top.

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 terminal 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. The following table lists control characters for Native TE.

Note: Some control characters may be represented by their hexadecimal values. For a description of control characters and hexadecimal equivalents, see the full tables in Appendix C, "*Terminal Font Set Table*."

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
<ERS_EOF>	Erase EOF	<SPACE>	Space

Loading the Auto-Login Script File

Follow these procedures to download an auto-login script file to your terminal. The method depends on the type of terminal you are using.

2415, 2425, 2435A, 2455, 2475, 248X Terminals

You can download more than one script file to a 2415, 2425, 2435A, 2455, 2475, or 248X Terminal, but immediately name the file to use as AUTOLOG.SCR and name any other script files with different names ending with .SCR. To learn more about transferring files, refer to your terminal's user manual.

Use one of the following download utilities to copy your completed AUTOLOG.SCR file to the terminal.

Using Serial Port to Download File

1 Connect the terminal to the development personal computer or host. For help, refer to your terminal's user manual or your accessory documentation.

2 Load the AUTOLOG.SCR file into drive C, where the TE application is stored. For help, refer to your terminal's user manual.

3 Use T24XCOPY.EXE (from the developer's kit) to copy the file directly to drive C: on the terminal; or use LOADER.EXE (from the flash upgrade) to copy the file to the terminal in flash mode using the following command:

```
loader AUTOLOG.SCR
```

Using RF Communications to Download File

Note: Loading the AUTOLOG.SCR file from the DCS 300 is supported.

1 Copy the AUTOLOG.SCR file to the DCS 300.

2 Configure the download server on the DCS 300 to send AUTOLOG.SCR to the terminals in groups of 10 or less.

3 **Start the download.**
When you reset your terminal, it clears the auto-login information, such as the password and user's name. You can cancel the auto-login process by pressing any key during the auto-login sequence. When a host session is broken, you can restart the auto-login sequence by rebooting your terminal.

6400, 5055 Terminals

Use INTERLNK/INTERSVR communications to connect your terminal to a personal computer. For help, see *"Downloading Files"* on page 151.

You must load AUTOLOG.SCR into drive C, where the TE application is stored.

59XX, 17XX, 11XX Terminals

The download process requires flash and utility files.

1 Using FLSHCONV.EXE from the TOOLS.EXE file from the flash program, append the AUTOLOG.SCR file to the end of the .HEX program. Below are sample command lines for each terminal type:

- 11XX Terminal with 256K flash *(this uses 1180, but applies to all radio types)*:

```
flshconv -eC000 -a fwp118c0.hex AUTOLOG.scr
-oNEW1180.HEX
```

- 11XX Terminal with 512K flash *(this uses 1180, but applies to all radio types)*:

```
flshconv -eFFC0 -a fwp118h0.hex AUTOLOG.scr
-oNEW1180.HEX
```

- 17XX Terminal *(example uses 1780, but applies to all radio types)*;

```
flshconv -eFFC0 -a fwp178h0.hex AUTOLOG.scr
-oNEW1780.HEX
```

- 59XX Terminal *(example uses 5980, but applies to all radio types)*:

```
flshconv -eE000 -a fwp598h0.hex AUTOLOG.scr
-oNEW5980.HEX
```

2 Using PROGDUX.EXE from the flash files, load the NEW.HEX file onto the terminal. Below is a sample command line using the 1780 Terminal:

```
PROGDUX -e NEW178H0.HEX
```

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 terminal you are using.

2415, 2425, 2435A, 2455, 2475, 248X Terminals

1 Press the appropriate key sequence or scan the following bar code to access the TRAKKER Antares® 2400 Menu System's Main Menu:

Enter Test and Service Mode

..-.

2 Choose System Menu and then choose File Manager.

3 Select drive C. Press ▲ or ▼ to highlight AUTOLOG.SCR.

4 Press [FnL .] to delete the file or press [F7] to rename the file.

5 Exit the TRAKKER Antares 2400 Menu System to return to your current TE session. Restart the TE application to use TE without the auto-login.

You can also rename the auto-login script file from your host. For help, refer to your terminal's user manual.

6400, 5055 Terminals

1 Reboot the terminal to a DOS prompt.

- *For the 6400 Computer*:
  press and hold the yellow, blue, and both green enter keys until the terminal resets. Watch the screen as the terminal boots and press 0 (zero) when the prompt appears.
- *For the 5055 Data Collection PC*:
  power-cycle the terminal and select the minimum DOS prompt from the boot menu.

2 Rename or delete the AUTOLOG.SCR file.

3 Reboot the terminal again and allow it to complete the boot into terminal emulation.

59XX, 17XX, 11XX Terminals

To disable auto-login on a 59XX, 17XX, or 11XX Terminal, reload the terminal with unmodified flash, then perform a cold start to flush the memory. The flash file does not allow selective erasing; files can be added, but not removed without removing all files.

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 Terminals Using the Same Account

```
HostName “*”                                #Use this to log into any host
WaitFor “login:”                            #Wait for the login prompt
Send “user_name<CR>”                        #Send the user name
WaitFor “Password:”                         #Wait for the password prompt
Send “letmein<CR>”                          #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** command 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 user_name                                      #Send the user name
Send “<CR>”                                         #Send a carriage return
WaitFor “Password:”                                 #Wait for password
prompt
Send password                                       #Send the users password
Send “<CR>”                                         #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 will display “*” in place of any characters the user types. All **Input** commands must be 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
WaitFor “<CR>”                                     #Send a carriage return
WaitFor “Password:”                                #Wait for password prompt
Send password                                      #Send the users password
Send “<CR>”                                        #Send a carriage return
WaitFor “Main Menu”                                #Wait for the main menu
Send “3”
Send “<CR>”                                        #Pick option 3 from menu
WaitFor “Wip Menu”                                 #Await work-in-process menu
Send “1”
Send “<CR>”                                        #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 “<CR>”                                           #Send a carriage return
WaitFor “Password:”                                #Wait for password prompt
Send password                                         #Send the users password
Send “<CR>”                                           #Send a carriage return
HostName “*”                                          #Match any host name
WaitFor “login:”                                      #Wait for login prompt
Send username                                         #Send the user name
WaitFor “<CR>”                                        #Send a carriage return
WaitFor “Password:”                                #Wait for password prompt
Send password                                         #Send the users password
Send “<CR>”                                           #Send a carriage return
WaitFor “Main Menu”                                   #Wait for the main menu
Send “3”
Send “<CR>”                                           #Pick option 3 from menu
WaitFor “Wip Menu”                                 #Await work-in-process menu
Send “1”
Send “<CR>”                                        #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

The Auto-Login Restart command starts 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 Chapter 2, "*Using the Terminal's Keyboard*." Or, scan the following bar code.

Auto-Login Restart

%ALRS

Creating a Custom Parameter Set-Up File

You can create a custom Native TE set-up file to preset almost any parameter you can set from the TE configuration menus. These parameter settings become the default (cold start) configuration for the terminal.

Syntax

The parameter set up file is an ASCII text file that you create which gets converted to a binary file by the CHECKCFG.EXE utility. New terminals do not come with this utility but it is available by request from the system engineer; or, if you ordered the upgrade version, it is included in the TOOLS.EXE self-extracting file.

You can name the input file as you choose. When you use the CHECKCFG utility later to verify the file, you must change the output file name to CONFIG.DAT. A field may have zero or more qualifiers.

The syntax of a line is `FieldName = Value;` *or* `Qualifier.FieldName = Value;`

- *FieldName* is the name of the parameter you want to modify.
- *Value* is the new value for the field.
- *Qualifier* and *FieldName* must be ASCII strings that match one of the configuration parameters.
- *Value* can be an ASCII string or a numeric value, depending on the type of the parameter *FieldName* specifies. The parameters along with their types and allowable values are listed under "*Parameters and Qualifiers*" later in this chapter.

CHECKCFG.EXE ignores the following:

- Blank lines
- Leading white space (spaces and tabs)
- White space on either side of any delimiter (a period, an equal sign, or a semicolon)

In addition, the set-up file converts consecutive white space characters within strings to one space. For example, this line:

```
Screen  Mode   =  Page Mode  ;
```

is the same as:

```
Screen Mode=Page Mode;
```

You can set the *Value* field of any configuration parameter to "?", which indicates that the TE configuration program should prompt the user for the appropriate value. For example, this parameter causes the terminal to prompt the user to select a screen mode from the list of values:

```
Screen Mode = ?;
```

Following is a sample set-up file.

```
Program Name = CUSTOM;

Screen Mode = Page Mode;

Host A.Port Number = 1;

Data Stream = Native;
Code 39.Encoded = Enabled;
Extended Cmds = Enabled;
```

Parameter Formats

The complete parameter list with value definitions can be directed to a file using this command:

```
CHECKCFG -P CFGLIT.DAT > parms.txt
```

Each parameter in the set-up 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.

  Example:

  ```
  [Session 1].Screen Mode
  Session 2.Screen Mode
     Center Cursor
     Corner Mode
     Page Mode
     Lazy Mode
     Locked Mode
  ```

  Screen Mode may be qualified by Session 1 or 2. 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.

  Example:
  [Session 1].[Host A].Port Number
  Session 2.Host A.Port Number
  [Session 1].Host B.Port Number
  Session 2.Host B.Port Number
  [Session 1].Host C.Port Number
  Session 2.Host C.Port Number
  Numeric, minimum = 0, maximum = 65535

  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.

  Example:
  Program Name
  String, minimum length = 8, maximum length = 8

  Program Name is unqualified. It must have eight characters. These configuration lines are valid:

```
Program Name = ABCDEFGH;
Program Name = FWP248H0;
```

Verifying Your Configuration

Use the CHECKCFG utility to verify that you properly configured your set-up file. CHECKCFG reads your configuration and reports any syntax errors. To get the utility, contact your Systems Engineer.

To verify your configuration:

Type: **`checkcfg <input config file> cfglit.dat config.dat`**

- *<input config file>* is the name of your parameter set-up file.
- CFGLIT.DAT contains the strings that appear in the configuration menus, parameter files, and system messages.
- CONFIG.DAT is the name of the output file you will download to the terminal. This file must be named CONFIG.DAT.

You are ready to download CONFIG.DAT to the terminal when CHECKCFG reports no errors. For download instructions, see "*Downloading Files*" on page 151.

If CHECKCFG reports an error, use a text editor to open your set-up file and correct it. The following chart lists the system messages CHECKCFG may display.

Default String	Description	ID #
Can’t open file	The input config (configuration) file could not be located.	F000
Bad option name	The right side of an expression in the input config file is an invalid name.	F001
Syntax error	The input config file contains a syntax error.	F002
Bad string length	The right side of an expression in the input config file is a string that is too long for the specified parameter.	F003
Unknown type	The parameter table in the configuration program contains a bad value. This error is for testing purposes only and should never occur.	F004
Value out of range	The right side of an expression in the input config file is a number that is out of range for the specified parameter.	F005
Bad value	The right side of an expression in the input config file is a string that is invalid for the specified parameter.	F006
Expected numeric	The right side of an expression in the input config file should be a numeric value but contains nonnumeric data.	F007
Missing ’=‘	A line in the input config file does not contain the required “=” (equal sign).	F008
Missing ’;‘	A line in the input config file is not terminated by a “;” (semicolon).	F009

Parameters and Qualifiers

Note: If parameters are duplicated, an error is not generated. Instead, the terminal will process the entire file and use 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. Default qualifiers are listed between square brackets “[ ]” in this chapter.

The CFGLIT.DAT file specifies the text of set-up menus or system messages. Parameters and qualifier strings in the set-up file assume you are using the default CFGLIT.DAT file. To customize CFGLIT.DAT, see *“Changing Text”* on page 140.

Trakker Antares terminals (2415, 2425, 2435A, 2455, 2475, and 248X), using WTP, support dual sessions. If the terminals use UDP Plus or TCP/IP, then dual sessions are *not* supported.

Note: Only WTP terminals support dual sessions (“Session 2”). For a list of terminals with WTP, see “*Program Names*” in Chapter 1, “*Getting Started*.”

Set-Up Parameters Options

- **Channel** *(900 MHz radio)*
  [Session 1].[Host A].Channel
  Session 2.Host A.Channel
  [Session 1].Host B.Channel
  Session 2.Host B.Channel
  [Session 1].Host C.Channel
  Session 2.Host C.Channel
    10
    25
    30
    25
    30
    35
    40

Note: For Terminal Emulation versions 6.60 and greater, “VT/ANSI” is the default data stream.

- **Data Stream**
  [Session 1].[Host A].Data Stream
  Session 2.Host A.Data Stream
  [Session 1].Host B.Data Stream
  Session 2.Host B.Data Stream
  [Session 1].Host C.Data Stream
  Session 2.Host C.Data Stream
    Native *(blank if UDP Plus or TCP/IP)*
    3270
    5250
    [VT/ANSI]

- **Frequency** *(S-UHF radio)*
  [Session 1].[Host A].Frequency
  Session 2.Host A.Frequency
  [Session 1].Host B.Frequency
  Session 2.Host B.Frequency
  [Session 1].Host C.Frequency
  Session 2.Host C.Frequency
    Numeric, minimum=0, maximum=0xFFFFFFFF

- **Host Name**
  [Session 1].[Host A].Host
  Session 2.Host A.Host
  [Session 1].Host B.Host
  Session 2.Host B.Host
  [Session 1].Host C.Host
  [Session 1].Host C.Host
    String, minimum length = 0, maximum length = 16 [null string]

- **Lan ID**
  [Session 1].[Host A].Lan
  Session 2.Host A.Lan
  [Session 1].Host B.Lan
  Session 2.Host B.Lan
  [Session 1].Host C.Lan
  Session 2.Host C.Lan
  Numeric, minimum = 0, maximum = 254 [0] *applies to 900 MHz*
  Numeric, minimum = 0, maximum = 15 [0] *2.4 GHz OpenAir*

Note: Values greater than 15 on terminals with OpenAir or 802.11 radios have a MOD 16 operation done on the terminals. For example, LAN IDs 0, 16, and 32 all equal 0 (zero).

- **Mode**
  [Session 1].[Host A].Mode
  Session 2.Host A.Mode
  [Session 1].Host B.Mode
  Session 2.Host B.Mode
  [Session 1].Host C.Mode
  Session 2.Host C.Mode
  For 900 MHz radio:
  DS 225K
  DS 090K
  DS 450K
  For S-UHF radio:
  Freq Agility
  Single Freq

- **Port Number**
  [Session 1].[Host A].Port Number
  Session 2.Host A.Port Number
  [Session 1].Host B.Port Number
  Session 2.Host B.Port Number
  [Session 1].Host C.Port Number
  Session 2.Host C.Port Number
  Numeric, minimum = 0, maximum = 65535 [23]

- **Radio Configuration Number**
  [Session 1].[Host A].Radio Config#
  Session 2.Host A.Radio Config#
  [Session 1].Host B.Radio Config#
  Session 2.Host B.Radio Config#
  [Session 1].Host C.Radio Config#
  Session 2.Host C.Radio Config#
  Numeric, minimum = 0, maximum = 255 [0]

- **Radio Number** (*same as unit number*)
  [Session 1].[Host A].Radio #
  Session 2.Host A.Radio #
  [Session 1].Host B.Radio #
  Session 2.Host B.Radio #
  [Session 1].Host C.Radio #
  Session 2.Host C.Radio #
  Numeric, minimum = 0, maximum = 127 [127]

- **Server IP**
  [Session 1].[Host A].Server IP
  Session 2.Host A.Server IP
  [Session 1].Host B.Server IP
  Session 2.Host B.Server IP
  [Session 1].Host C.Server IP
  Session 2.Host C.Server IP
  Numeric, minimum = 1, maximum = 16 [16]

- **SNA**
  [Session 1].SNA
  Session 2.SNA
  Enabled
  [Disabled]

- **Telnet**
  [Session 1].Telnet
  Session 2.Telnet
  [Enabled] *this default applies to the 6400 and 5055 TE over IP options, and to 2415, 2425, 2435A, 2455, 2475, 248*X
  [Disabled]

- **Terminal Type**
  [Session 1].[Host A].Terminal Type
  Session 2.Host A.Terminal Type
  [Session 1].Host B.Terminal Type
  Session 2.Host B.Terminal Type
  [Session 1].Host C.Terminal Type
  Session 2.Host C.Terminal Type
  Numeric, minimum = 0, maximum = 255 [255]

- **Unit Number** *(same as radio number)*
  [Session 1].[Host A].Unit #
  Session 2.Host A.Unit #
  [Session 1].Host B.Unit #
  Session 2.Host B.Unit #
  [Session 1].Host C.Unit #
  Session 2.Host C.Unit #
  Numeric, minimum = 0, maximum = 127 [127]

Bar Code Parameters

Note: Bar code parameters apply to 6400, 5055, 59XX, 17XX, and 11XX Terminals unless otherwise noted. 2415, 2425, 2435A, 2455, 2475, and 248X Terminals support the **Code 39 Encoded**, **Concatenate**, **RS232 Stream**, **Stream Scan**, and **Scan All Flds** parameters.

- **BC Type Character** *(not supported on the 5055)*
  [Session 1].BC Type Char
  Session 2.BC Type Char
  Enabled
  [Disabled]

- **Code 39 Encoded**
  [Session 1].Code 39 Encoded
  Session 2.Code 39 Encoded
  Enabled
  [Disabled]

- **Concatenate**
  [Session 1].Concatenate
  Session 2].Concatenate
  Enabled
  [Disabled]

- **MOD 10 Check**
  [Session 1].MOD 10 Check
  Session 2.MOD 10 Check
  Enabled
  [Disabled]

- **Redundancy** *(not supported on the 5055)*
  [Session 1].Redundancy
  Session 2.Redundancy
  Enabled
  [Disabled]

- **RS-232 Stream** *(2415, 2425, 2435A, 2455, 2475, 248X)*
  [Session 1].RS232 Stream
  Session 2.RS232 Stream
  Enabled
  [Disabled]

- **Scan All Fields**
  [Session 1].Scan All Flds
  Session 2.Scan All Flds
  Enabled
  [Disabled]

- **Scan Postamble Character**
  [Session 1].Scan PostChar
  Session 2.Scan PostChar
  Numeric, minimum = 0, maximum = 255 [0]

- **Scan Preamble Character**
  [Session 1].Scan PreChar
  Session 2.Scan PreChar
    Numeric, minimum = 0, maximum = 255 [0]
- **Scan Timeout** (*not supported on the 6400 and 5055*)
  [Session 1].Scan Timeout
  Session 2.Scan Timeout
    Numeric, minimum = 1, maximum = 200 [30]
- **Scanner Type** *(6400 and 5055 support Laser only)*
  [Session 1].Scanner Type
  Session 2.Scanner Type
    [No Scanner]
    Wand
    Laser
    Wand Emulate
    Auto Detect
- **Stream Scan**
  [Session 1].Stream Scan
  Session 2.Stream Scan
    Enabled
    [Disabled]

Bar Code Symbologies

- **ABC Codabar**
  [Session 1].ABC Codabar
  Session 2.ABC Codabar
    Enabled
    [Disabled]
- **Codabar**
  [Session 1].Codabar
  Session 2.Codabar
    Enabled
    [Disabled]
- **Code 11** *(not supported on 6400, 5055)*
  [Session 1].Code 11
  Session 2.Code 11
    Enabled
    [Disabled]
- **Code 39**
  [Session 1].Code 39
  Session 2.Code 39
    Enabled
    [Disabled]
- **Code 93**
  [Session 1].Code 93
  Session 2.Code 93
    Enabled
    [Disabled]

- **Code 128**
  [Session 1].Code 128
  Session 2.Code 128
  Enabled
  [Disabled]
- **Computer Identics 2 of 5**
  [Session 1].CI 2of5
  Session 2.CI 2of5
  Enabled
  [Disabled]
- **EAN**
  [Session 1].EAN
  Session 2.EAN
  Enabled
  [Disabled]
- **Interleaved 2 of 5**
  [Session 1].Int 2of5
  Session 2.Int 2of5
  Enabled
  [Disabled]
- **Plessey** *(6400 does not support Plessey alpha characters)*
  [Session 1].Plessey
  Session 2.Plessey
  Enabled
  [Disabled]
- **Straight 2 of 5**
  [Session 1].Str 2of5
  Session 2.Str 2of5
  Enabled
  [Disabled]
- **UPC**
  [Session 1].UPC
  Session 2.UPC
  Enabled
  [Disabled]

Generic Bar Code Options

- **Drop Leading**
  [Session 1].UPC.Drop Leading
  Session 2.UPC.Drop Leading
  [Session 1].EAN.Drop Leading
  Session 2.EAN.Drop Leading
  [Session 1].Code 128.Drop Leading
  Session 2.Code 128.Drop Leading
  [Session 1].Code 39.Drop Leading
  Session 2.Code 39.Drop Leading
  [Session 1].Codabar.Drop Leading
  Session 2.Codabar.Drop Leading
  [Session 1].ABC Codabar.Drop Leading
  Session 2.ABC Codabar.Drop Leading
  [Session 1].Str 2of5.Drop Leading
  Session 2.Str 2of5.Drop Leading
  [Session 1].Int 2of5.Drop Leading
  Session 2.Int 2of5.Drop Leading
  [Session 1].CI 2of5.Drop Leading
  Session 2.CI 2of5.Drop Leading
  [Session 1].Code 11.Drop Leading
  Session 2.Code 11.Drop Leading
  [Session 1].Code 93.Drop Leading
  Session 2.Code 93.Drop Leading
  [Session 1].Plessey.Drop Leading
  Session 2.Plessey.Drop Leading
  Numeric, minimum = 0, maximum = 15 [0]

- **Drop Trailing**
  [Session 1].UPC.Drop Trailing
  Session 2.UPC.Drop Trailing
  [Session 1].EAN.Drop Trailing
  Session 2.EAN.Drop Trailing
  [Session 1].Code 128.Drop Trailing
  Session 2.Code 128.Drop Trailing
  [Session 1].Code 39.Drop Trailing
  Session 2.Code 39.Drop Trailing
  [Session 1].Codabar.Drop Trailing
  Session 2.Codabar.Drop Trailing
  [Session 1].ABC Codabar.Drop Trailing
  Session 2.ABC Codabar.Drop Trailing
  [Session 1].Str 2of5.Drop Trailing
  Session 2.Str 2of5.Drop Trailing
  [Session 1].Int 2of5.Drop Trailing
  Session 2.Int 2of5.Drop Trailing
  [Session 1].CI 2of5.Drop Trailing
  Session 2.CI 2of5.Drop Trailing
  [Session 1].Code 11.Drop Trailing
  Session 2.Code 11.Drop Trailing
  [Session 1].Code 93.Drop Trailing
  Session 2.Code 93.Drop Trailing
  [Session 1].Plessey.Drop Trailing
  Session 2.Plessey.Drop Trailing
    Numeric, minimum = 0, maximum = 15 [0]

Note: Generic bar code options must be qualified by one of the bar code symbology strings. For example: “Session 1.UPC.Max Length = 13;”

- **Fix Length 1**
  [Session 1].UPC.Fix Length 1
  Session 2.UPC.Fix Length 1
  [Session 1].EAN.Fix Length 1
  Session 2.EAN.Fix Length 1
  [Session 1].Code 128.Fix Length 1
  Session 2.Code 128.Fix Length 1
  [Session 1].Code 39.Fix Length 1
  Session 2.Code 39.Fix Length 1
  [Session 1].Codabar.Fix Length 1
  Session 2.Codabar.Fix Length 1
  [Session 1].ABC Codabar.Fix Length 1
  Session 2.ABC Codabar.Fix Length 1
  [Session 1].Str 2of5.Fix Length 1
  Session 2.Str 2of5.Fix Length 1
  [Session 1].Int 2of5.Fix Length 1
  Session 2.Int 2of5.Fix Length 1
  [Session 1].CI 2of5.Fix Length 1
  Session 2.CI 2of5.Fix Length 1
  [Session 1].Code 11.Fix Length 1
  Session 2.Code 11.Fix Length 1
  [Session 1].Code 93.Fix Length 1
  Session 2.Code 93.Fix Length 1
  [Session 1].Plessey.Fix Length 1
  Session 2.Plessey.Fix Length 1
  Numeric, minimum = 0, maximum = 99 [0]

- **Fixed Length 2**
  [Session 1].UPC.Fix Length 2
  Session 2.UPC.Fix Length 2
  [Session 1].EAN.Fix Length 2
  Session 2.EAN.Fix Length 2
  [Session 1].Code 128.Fix Length 2
  Session 2.Code 128.Fix Length 2
  [Session 1].Code 39.Fix Length 2
  Session 2.Code 39.Fix Length 2
  [Session 1].Codabar.Fix Length 2
  Session 2.Codabar.Fix Length 2
  [Session 1].ABC Codabar.Fix Length 2
  Session 2.ABC Codabar.Fix Length 2
  [Session 1].Str 2of5.Fix Length 2
  Session 2.Str 2of5.Fix Length 2
  [Session 1].Int 2of5.Fix Length 2
  Session 2.Int 2of5.Fix Length 2
  [Session 1].CI 2of5.Fix Length 2
  Session 2.CI 2of5.Fix Length 2
  [Session 1].Code 11.Fix Length 2
  Session 2.Code 11.Fix Length 2
  [Session 1].Code 93.Fix Length 2
  Session 2.Code 93.Fix Length 2
  [Session 1].Plessey.Fix Length 2
  Session 2.Plessey.Fix Length 2
  Numeric, minimum = 0, maximum = 99 [0]

- **Fixed Length 3**
  [Session 1].UPC.Fix Length 3
  Session 2.UPC.Fix Length 3
  [Session 1].EAN.Fix Length 3
  Session 2.EAN.Fix Length 3
  [Session 1].Code 128.Fix Length 3
  Session 2.Code 128.Fix Length 3
  [Session 1].Code 39.Fix Length 3
  Session 2.Code 39.Fix Length 3
  [Session 1].Codabar.Fix Length 3
  Session 2.Codabar.Fix Length 3
  [Session 1].ABC Codabar.Fix Length 3
  Session 2.ABC Codabar.Fix Length 3
  [Session 1].Str 2of5.Fix Length 3
  Session 2.Str 2of5.Fix Length 3
  [Session 1].Int 2of5.Fix Length 3
  Session 2.Int 2of5.Fix Length 3
  [Session 1].CI 2of5.Fix Length 3
  Session 2.CI 2of5.Fix Length 3
  [Session 1].Code 11.Fix Length 3
  Session 2.Code 11.Fix Length 3
  [Session 1].Code 93.Fix Length 3
  Session 2.Code 93.Fix Length 3
  [Session 1].Plessey.Fix Length 3
  Session 2.Plessey.Fix Length 3
  Numeric, minimum = 0, maximum = 99 [0]

- **Fixed Length 4**
  [Session 1].UPC.Fix Length 4
  Session 2.UPC.Fix Length 4
  [Session 1].EAN.Fix Length 4
  Session 2.EAN.Fix Length 4
  [Session 1].Code 128.Fix Length 4
  Session 2.Code 128.Fix Length 4
  [Session 1].Code 39.Fix Length 4
  Session 2.Code 39.Fix Length 4
  [Session 1].Codabar.Fix Length 4
  Session 2.Codabar.Fix Length 4
  [Session 1].ABC Codabar.Fix Length 4
  Session 2.ABC Codabar.Fix Length 4
  [Session 1].Str 2of5.Fix Length 4
  Session 2.Str 2of5.Fix Length 4
  [Session 1].Int 2of5.Fix Length 4
  Session 2.Int 2of5.Fix Length 4
  [Session 1].CI 2of5.Fix Length 4
  Session 2.CI 2of5.Fix Length 4
  [Session 1].Code 11.Fix Length 4
  Session 2.Code 11.Fix Length 4
  [Session 1].Code 93.Fix Length 4
  Session 2.Code 93.Fix Length 4
  [Session 1].Plessey.Fix Length 4
  Session 2.Plessey.Fix Length 4
    Numeric, minimum = 0, maximum = 99 [0]

- **Maximum Length**
  [Session 1].UPC.Max Length
  Session 2.UPC.Max Length
  [Session 1].EAN.Max Length
  Session 2.EAN.Max Length
  [Session 1].Code 128.Max Length
  Session 2.Code 128.Max Length
  [Session 1].Code 39.Max Length
  Session 2.Code 39.Max Length
  [Session 1].Codabar.Max Length
  Session 2.Codabar.Max Length
  [Session 1].ABC Codabar.Max Length
  Session 2.ABC Codabar.Max Length
  [Session 1].Str 2of5.Max Length
  Session 2.Str 2of5.Max Length
  [Session 1].Int 2of5.Max Length
  Session 2.Int 2of5.Max Length
  [Session 1].CI 2of5.Max Length
  Session 2.CI 2of5.Max Length
  [Session 1].Code 11.Max Length
  Session 2.Code 11.Max Length
  [Session 1].Code 93.Max Length
  Session 2.Code 93.Max Length
  [Session 1].Plessey.Max Length
  Session 2.Plessey.Max Length
    Numeric, minimum = 0, maximum = 99 [0]

- **Minimum Length**
  [Session 1].UPC.Min Length
  Session 2.UPC.Min Length
  [Session 1].EAN.Min Length
  Session 2.EAN.Min Length
  [Session 1].Code 128.Min Length
  Session 2.Code 128.Min Length
  [Session 1].Code 39.Min Length
  Session 2.Code 39.Min Length
  [Session 1].Codabar.Min Length
  Session 2.Codabar.Min Length
  [Session 1].ABC Codabar.Min Length
  Session 2.ABC Codabar.Min Length
  [Session 1].Str 2of5.Min Length
  Session 2.Str 2of5.Min Length
  [Session 1].Int 2of5.Min Length
  Session 2.Int 2of5.Min Length
  [Session 1].CI 2of5.Min Length
  Session 2.CI 2of5.Min Length
  [Session 1].Code 11.Min Length
  Session 2.Code 11.Min Length
  [Session 1].Code 93.Min Length
  Session 2.Code 93.Min Length
  [Session 1].Plessey.Min Length
  Session 2.Plessey.Min Length
  Numeric, minimum = 0, maximum = 99 [0]

UPC Options

- **Add-On 2**
  [Session 1].UPC.Add-On 2
  Session 2.UPC.Add-On 2
  Enabled
  [Disabled]

- **Add-On 5**
  [Session 1].UPC.Add-On 5
  Session 2.UPC.Add-On 5
  Enabled
  [Disabled]

- **Expand E to A**
  [Session 1].UPC.Expand E to A
  Session 2.UPC.Expand E to A
  Enabled
  [Disabled]

- **System 0 UPCE**
  [Session 1].UPC.Sys 0 UPCE
  Session 2.UPC.Sys 0 UPCE
  Enabled
  [Disabled]

- **System 1 UPCE** *(not supported on 6400)*
  [Session 1].UPC.Sys 1 UPCE
  Session 2.UPC.Sys 1 UPCE
    Enabled
    [Disabled]

EAN Options

- **Add-On 2**
  [Session 1].EAN.Add-On 2
  Session 2.EAN.Add-On 2
    Enabled
    [Disabled]
- **Add-On 5**
  [Session 1].EAN.Add-On 5
  Session 2.EAN.Add-On 5
    Enabled
    [Disabled]
- **Expand 8 to 13**
  [Session 1].EAN.Expand 8to13
  Session 2.EAN.Expand 8to13
    Enabled
    [Disabled]

Code 39 Options

- **Auto-Encoded**
  [Session 1].Code 39.Auto-Encoded
  Session 2.Code 39.Auto-Encoded
    Enabled
    [Disabled]
- **Check Digit**
  [Session 1].Code 39.Chk Digit
  Session 2.Code 39.Chk Digit
    Enabled
    [Disabled]
- **Data Decode**
  [Session 1].Code 39.Data decode
  Session 2.Code 39.Data decode
    Enabled
    [Disabled]
- **Element Decode**
  [Session 1].Code 39.Element decod
  Session 2.Code 39.Element decod
    Enabled
    [Disabled]
- **Encoded**
  [Session 1].Code 39.Encoded
  Session 2.Code 39.Encoded
    Enabled
    [Disabled]

- **Extended**
  [Session 1].Code 39.Extended
  Session 2.Code 39.Extended
    Enabled
    [Disabled]

- **Full ASCII**
  [Session 1].Code 39.Full ASCII
  Session 2.Code 39.Full ASCII
    Enabled
    [Disabled]

- **Quiet Zone**
  [Session 1].Code 39.Quiet zone
  Session 2.Code 39.Quiet zone
    Enabled
    [Disabled]

- **START Decode**
  [Session 1].Code 39.START decode
  Session 2.Code 39.START decode
    Enabled
    [Disabled]

Interleaved 2 of 5 Option

- **Interleaved 2 of 5 Chk Digit**
  [Session 1].Int 2of5.Chk Digit
  Session 2.Int 2of5.Chk Digit
    Enabled
    [Disabled]

Code 11 Options

- **Check Digit 1** (*not supported on 6400*)
  [Session 1].Code 11.Chk Dig 1
  Session 2.Code 11.Chk Dig 1
    Enabled
    [Disabled]

- **Check Digit 2** *(not supported on 6400)*
  [Session 1].Code 11.Chk Dig 2
  Session 2.Code 11.Chk Dig 2
    Enabled
    [Disabled]

Plessey Options

- **Allow Alpha**
  [Session 1].Plessey.Allow Alpha
  Session 2.Plessey.Allow Alpha
    Enabled
    [Disabled]
- **MOD 10 Check**
  [Session 1].Plessey.MOD10 Chk
  Session 2.Plessey.MOD10 Chk
    Enabled
    [Disabled]
- **MOD 11 Check**
  [Session 1].Plessey.MOD11 Chk
  Session 2.Plessey.MOD11 Chk
    Enabled
    [Disabled]

Code 128 Options

- **No UCC Type**
  [Session 1].Code 128.NO UCC Type
  Session 2.Code 128.NO UCC Type
    Enabled
    [Disabled]
- **UCC/EAN**
  [Session 1].Code 128.UCC/EAN
  Session 2.Code 128.UCC/EAN
    Enabled
    [Disabled]

- **UCC F1 Value**
  [Session 1].UPC.UCC F1 Value
  Session 2.UPC.UCC F1 Value
  [Session 1].EAN.UCC F1 Value
  Session 2.EAN.UCC F1 Value
  [Session 1].Code 128.UCC F1 Value
  Session 2.Code 128.UCC F1 Value
  [Session 1].Code 39.UCC F1 Value
  Session 2.Code 39.UCC F1 Value
  [Session 1].Codabar.UCC F1 Value
  Session 2.Codabar.UCC F1 Value
  [Session 1].ABC Codabar.UCC F1 Value
  Session 2.ABC Codabar.UCC F1 Value
  [Session 1].Str 2of5.UCC F1 Value
  Session 2.Str 2of5.UCC F1 Value
  [Session 1].Int 2of5.UCC F1 Value
  Session 2.Int 2of5.UCC F1 Value
  [Session 1].CI 2of5.UCC F1 Value
  Session 2.CI 2of5.UCC F1 Value
  [Session 1].Code 11.UCC F1 Value
  Session 2.Code 11.UCC F1 Value
  [Session 1].Code 93.UCC F1 Value
  Session 2.Code 93.UCC F1 Value
  [Session 1].Plessey.UCC F1 Value
  Session 2.Plessey.UCC F1 Value
    Numeric, minimum = 0, maximum = 255

Native Protocol Options

- **F1 is Function –0**
  [Session 1].F1 is FUNC–0
  Session 2.F1 is FUNC–0
    Enabled
    [Disabled]

Display Options

- **Backlight State**
  [Session 1].Backlight State
  Session 2.Backlight State
    Enabled
    [Disabled]

- **Backlight Timer** *(not supported on 5055)*
  [Session 1].Backlight Timer
  Session 2.Backlight Timer
    Numeric, minimum = [0], maximum = 60 *(2415, 2425, 2435A, 2455, 2475, 248X)*
    Numeric, minimum = [0], maximum = 255 *(6400, 17XX, 11XX)*

- **Contrast Mode** *(59XX, 17XX, 11XX)*
  [Session 1].Contrast Mode
  Session 2.Contrast Mode
  [Compensated]
  Absolute
- **Cursor Mode**
  [Session 1].Cursor Mode
  Session 2.Cursor Mode
  Underln Blink *(2415, 2425, 2435A, 2455, 2475, 248X, 6400, 59XX, 17XX, 11XX*)
  Block Blink *(2415, 2425, 2435A, 2455, 2475, 248X, 6400, 59XX, 17XX, 11XX*)
  Underline *(6400, 5055, 59XX, 17XX, 11XX)*
  Block *(6400, 5055, 59XX, 17XX, 11XX)*
  [Underln Blink] is the default for 2415, 2425, 2435A, 2455, 2475, and 248X Terminals. [Block] is the default for 6400, 5055, 59XX, 17XX, and 11XX Terminals.
- **Remote Baud** *(17XX, 11XX)*
  [Session 1].Remote Baud
  Session 2.Remote Baud
  9600
  19200
  [38400]
- **Remote Display** *(17XX, 11XX*)
  [Session 1].Remote Disp
  Session 2.Remote Disp
  Enabled
  [Disabled]
- **Remote Display Remote Columns** (*17XX, 11XX*)
  Session 1.Remote Cols
  Session 2.Remote Cols
  Numeric, minimum = 1, maximum = 255 [80]
- **Remote Display Remote Rows** (*17XX, 11XX*)
  [Session 1].Remote Rows
  Session 2.Remote Rows
  Numeric, minimum = 1, maximum = 255 [24]
- **Remote Display Remote Type** (*17XX, 11XX*)
  [Session 1].Remote Type
  Session 2.Remote Type
  VT100
  VT220
  [ANSI]

Radio Communications Options

Note: Radio Communications options do not apply to 2415, 2425, 2435A, 2455, 2475, or 248X terminals that do not have an 802.1x supplicant radio driver installed. Set the Security ID for these terminals through the TRAKKER Antares 2400 Menu System.

- **Baud Rate** *(UHF radio)*
  Radio Baud Rate
    4800
    4800/9600
    9600
    The default depends on the type of radio in the terminal.
- **Protocol** *(6400, 5055, 59XX, 17XX, 11XX)*
  Radio Protocol
    The following qualifiers apply to UHF:
    RTC
    Adaptive Poll
    RTC-Diag Mode
    The following qualifiers apply to SST:
    SST
    SST-Diag Mode
    The default depends on the type of radio and is either RTC or SST.
- **Security ID**
  [Session 1].Security ID
  Session 2.Security ID
    String, minimum length=0, maximum length=16 [null string]
    Default is NORANDOWL for 6400, 5055, 59XX, 17XX, 11XX.
- **User Authentication** *(2415, 2425, 2435A, 2455, 2475, 248X with 802.1x supplicant radio driver installed)*
  [Session 1].User Auth
  Session 2.User Auth
    Enabled
    [Disabled]

More Options

- **Change Menu Password**
  The password must be ***enabled*** *and* ***set*** *for access to the Main Menu before you can change it. To set the password:*
  [Session 1].Change Menu Password
  Session 2.Change Menu Password
    String, minimum length = 1, maximum length = 10 [3193693]
  To enable the password, see "Main Menu password."
- **COM Select** *(248X, 5055)*
  [Session 1].Com Select
  Session 2.Com Select
    [COM 1]
    COM 2

- **Main Menu Password**
  The password must be ***enabled*** *and* ***set*** *for access to the Main Menu. To enable or disable the password:*
  [Session 1].Menu Password
  Session 2.Menu Password
    Enabled
    [Disabled]
  To set the password, see "Change menu password."

- **Print Device** *(6400)*
  [Session 1].Print Device
  Session 2.Print Device
    [RS232 Print]
    IRDA Print *(6400)*

LCD Options

- **Annunciator Format**
  [Session 1].Annun Format
  Session 2.Annun Format
    [Vertical] *2455, 2475, 248X, 5055, 59XX*
    [Horizontal] *2415, 2425, 2435A, 6400*

- **Annunciator Position**
  [Session 1].Annun Position
  Session 2.Annun Position
    Upper Right
    Upper Left
    [Lower Right] *6400, 5055, 59XX, 17XX, 11XX*
    Lower Left
    [Stealth] *2415, 2425, 2435A, 2455, 2475, 248X*

- **Backlight Intensity** *(59XX)*
  [Session 1].BLight Intensity
  Session 2.BLight Intensity
    Numeric, minimum = 0, maximum = 15

- **Define Height** (Scroll Window)
  [Session 1].Define Height
  Session 2.Define Height
    Numeric, minimum = 1, maximum = 24 [8]

- **Define Width** (Scroll Window)
  [Session 1].Define Width
  Session 2.Define Width
    Numeric, minimum = 1, maximum = 80 [8]

- **Key Uppercase**
  [Session 1].Key Uppercase
  Session 2.Key Uppercase
    [Enabled]
    Disabled

- **LCD Contrast** *(59XX, 17XX)*
  [Session 1].LCD Contrast
  Session 2.LCD Contrast
    Numeric, minimum = [0], maximum = 255

- **Screen Columns**
  [Session 1].Screen Cols
  Session 2.Screen Cols
    Numeric, minimum = 1, maximum = 255
    [16] *11XX*
    [17] *17XX*
    [19] *2435A*
    [20] *2415, 2425, 2455, 2475, 248X*
    [40] *59XX*
    [80] *5055*
    [16] *6400 wide display*

- **Screen Mode**
  [Session 1].Screen Mode
  Session 2.Screen Mode
    Center Cursor
    [Corner Mode]
    Page Mode
    Lazy Mode
    Locked Mode

- **Screen Rows**
  [Session 1].Screen Rows
  Session 2.Screen Rows
    Numeric, minimum = 1, maximum = 255
    [8] *11XX*
    [8] *17XX*
    [10] *59XX*
    [16] *2415, 2425, 2435A, 2455, 2475, 248X*
    [25] *5055*
    [16] *6400 wide display*

- **Scroll Window**
  [Session 1].Scroll Window
  Session 2.Scroll Window
    Screen Size
    Scroll Setting
    [Tab Size]

- **XOrigin**
  [Session 1].XOrigin
  [Session 2].XOrigin
    Numeric minimum = 0, maximum = 79 [0]

- **YOrigin**
  [Session 1].YOrigin
  [Session 2].YOrigin
    Numeric minimum = 0, maximum = 23 [0]

Beeper Setup Options

Note: Beeper setup options do not apply to the 2415, 2425, 2435A, 2455, 2475, and 248X Terminals.

- **Beep Head Set** *(17XX)*
  [Session 1].Beep Head Set
  Session 2.Beep Head Set
    Enabled
    [Disabled]

- **Beep Internal** *(not supported on 59XX)*
  [Session 1].Beep Internal
  Session 2.Beep Internal
    [Enabled]
    Disabled

- **Beeper (Error Tone) Frequency** *(not supported on 59XX)*
  [Session 1].Beep Frequency
  Session 2.Beep Frequency
    Numeric, minimum = 0, maximum = 30
    [13] *11XX and 17XX*
    [6] *6400*

- **Beeper (Error Tone) Length**
  [Session 1].Beep Length
  Session 2.Beep Length
    Numeric, minimum= 1 , maximum= 10 [3]

- **Beeper (Error Tone) Volume** *(59XX)*
  [Session 1].Beep Volume
  Session 2.Beep Volume
    Numeric, minimum = 0, maximum = 255 [95]

- **Key Click Frequency** *(not supported on 59XX)*
  [Session 1].KeyClick Freq
  Session 2.KeyClick Freq
    Numeric, minimum = 0, maximum = 30
    [13] *17XX and 11XX*
    [6] *6400*

- **Key Click Length**
  [Session 1].KeyClick Length
  Session 2.KeyClick Length
    Numeric, minimum = 1, maximum = 10
    [1] *59XX, 17XX, and 11XX*
    [6] *6400 and 5055*

- **Key Click Volume** *(59XX)*
  [Session 1].KeyClick Volume
  Session 2.KeyClick Volume
    Numeric, minimum = 0, maximum = 255 [95]

More (Main Menu 2) Option

- **Foreground Session**
  Foreground Sess
  [Session 1]
  Session 2 *(6400, 5055, 59XX, 17XX, 11XX)*

- **Set Hot Key** *(6400, 5055, 59XX, 17XX, 11XX)*
  [Session 1].Set Hot Key
  Session 2.Set Hot Key
  F1
  through
  F24
  [Disabled]

- **Type-Ahead**
  [Session 1].Type-Ahead
  Session 2.Type-Ahead
  [Enabled]
  Disabled

Additional Parameters

The following parameters do not have equivalent TE configuration menu options.

- **Alternate Screen Columns** (*6400, 5055, 17XX)*
  [Session 1].Alt Screen Cols
  Session 2.Alt Screen Cols
  Numeric, minimum = 1, maximum = 255
  [17] *17XX*
  [40] *5055*
  [16] *6400*

- **Alternate Screen Rows** (*6400, 5055, 17XX)*
  [Session 1].Alt Screen Rows
  Session 2.Alt Screen Rows
  Numeric, minimum = 1, maximum = 255
  [8] *17XX*
  [12] *5055*
  [16] *6400 wide display*

- **Key Repeat** *(59XX)*
  [Session 1].Key Repeat
  Session 2.Key Repeat
  Enabled
  [Disabled]

- **Program Name** *(appears on initialization and version screens)*
  Program Name
  String, minimum length = 8, maximum length = 8 [FWP???H?]

- **Program Version** *(appears on initialization and version screens)*
  Program Version
  String, minimum length = 4, maximum length = 4 [?.??]

- **Set-Up Parms Menu Password**
  [Session 1].Password
  Session 2.Password
  String, minimum length = 1, maximum length = 10 [CR52401]

Changing Text

Modify CFGLIT.DAT to change the text of TE configuration menus, configuration parameters, or system messages. This file contains the strings that appear in the configuration menus, parameter set-up files, and 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 terminal.

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 text 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 PORT\nIN USE\n\nPLEASE 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 above line to look like this:

```
0x2f10 “Printing\nPlease 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 151.

```
makelit cfglit.txt cfglit.dat
```

Preinitializing the Native TE Program

You can preinitialize the Native TE program. You must name the Native initialization file as NATIVE.INI. The file is processed when you reset or warm start the terminal. 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 151.

Data is encoded in binary format. To create NATIVE.INI, you may need a HEX editor or other special program.

NATIVE.INI contains Native data stream commands and orders. It contains ASCII data because Intermec controllers perform EBCDIC to ASCII translation for the data stream.

Following is the format for the Native data stream. The information assumes you have a working knowledge of the data stream command formats or escape sequences, or both.

The following Native commands are supported:

• 0x31	Write
• 0x32	Read buffer
• 0x34	Toggle SSCP command mode
• 0x35	Erase write
• 0x36	Read modified
• 0x3d	Erase write alternate
• 0x3e	Read modified all
• 0x3f	Erase all unprotected

The following Native orders are supported:

• 0x07	Beep (Intermec extension)
• 0x09	Program tab
• 0x11	Set buffer address
• 0x12	Erase unprotected to address
• 0x13	Insert cursor
• 0x14	Repeat to address
• 0x1d	Start of field
• 0x1f	Scanner (Intermec extension)

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 initialization files.

```
35 04 48 45 4c 4c 4f 20 57 4f 52 4c 44
 |  |  H  E  L  L  O     W  O  R  L  D
 |  |
 |  |
 |  |
-|- |- - - - - - - - Write control character with alarm bit
   -|- - - - - - - - - Native erase/write command
```

Remapping the Terminal’s Keys

You may need to remap the terminal’s keys if your users need to press a key in Native TE that is not on a standard 101-key keyboard. You can also remap a terminal key to transmit a text string or message to the personal computer.

To remap the terminal keys, you create the REMAP.CFG file and add a Remap command to remap a terminal 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 terminal key or two-key sequence that does not perform a specific function on the terminal. For example, you can remap the [B] key because it only types the lowercase letter B.

Note: On the 2425, 2455, 2475, and 248X, you cannot remap the two-key sequence [_f] ▲ because it moves the window/viewport up on the Trakker Antares Terminals.

Each terminal key or two-key sequence generates a 4-digit hexadecimal remap code as listed in the “*Key Code*” table starting on page 144. The key code table gives the 4-digit hexadecimal codes for ASCII characters for the terminals. 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

Remapping a Key or Two-Key Sequence

Note: For terminals made before 1997, refer to the “*Using FLSHCONV*” section for the proper -e parameter value.

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 “*Key Code*” table on the next page.

2 Connect the terminal 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. If ”<” is desired in the string, then precede it with another “<” like the following.

```
  remap=<xxxx>=“a less-than sign is a <<”
```

4 Save the new file as REMAP.CFG.

5 Download REMAP.CFG to drive C: on your terminal.

Example 1

Suppose you want to remap "+" on your terminal to send a message and then enter a carriage return. In the REMAP.CFG file, add the following 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 the **`remap=<0042>="<ESC>[17~"`** command to the REMAP.CFG file.

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 1<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 key or 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 using one of the following commands. Replace ??? with your type of terminal:

- 59XX Terminals

```
FLSHCONV -a –eE000 FWP59??H0.HEX REMAP.CFG –oNEW.HEX
```

- 17XX, 11XX Terminals

```
FLSHCONV -a –eFFC0 FWP1???H0.HEX REMAP.CFG –oNEW.HEX
```

5 Download REMAP.CFG to drive C on your terminal.

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: `remap=<002b>="Change the battery pack now.<CR>"`

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 will produce a “5” key. If nesting was allowed, the “5” key would be recognized as a macro that produces the “3” key and the “3” key would be recognized as a “5” key that produces the “7” key, and so forth.

Key Code Table

Note: Values that are 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=<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”   /* F15 key */
remap=<1066>=“string”   /* F16 key */
remap=<1067>=“string”   /* F17 key */
remap=<1068>=“string”   /* F18 key */
remap=<1069>=“string”   /* F19 key */
remap=<106a>=“string”   /* F20 key */
remap=<106b>=“string”   /* F21 key */
remap=<106c>=“string”   /* F22 key */
remap=<106d>=“string”   /* F23 key */
remap=<106e>=“string”   /* F24 key */

remap=<2041>=“string”   /* Auto-Login Restart key */
remap=<206c>=“string”   /* Menu 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 Native TE must be NATIVE.XLT. For instructions on how to download the file to the terminal, see "*Downloading Files*" on page 151.

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 be replaced.
- The second byte of a pair is the ASCII value that replaces the first.

These translations are only made when a character is written to a display device. If the character is sent to the host (such as keystroke or scan data) or sent to an external device (such as a printer), it is sent as the original, untranslated value.

Suppose you want a terminal running Native emulation to replace the uppercase B with the Greek letter beta, and replace the uppercase Z with the Greek letter omega. Create a file named Native.XLT 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.

Substituting National Characters

This topic tells how to create a file that remaps one character in the terminal's Terminal Font Set for another character in the set. Use this information to provide characters on the terminal in the local language.

Intermec provides a table for converting from characters used in English (U.S.) to German, Danish, Finnish, Italian, Spanish, French, Belgian, and English (UK). The procedure for converting the characters appears first. Then two slightly longer procedures show how to remap any character to another. One procedure describes the steps with the symbol's EBCDIC value. The other one describes the steps without a value. Finally, examples show the remap files for German characters and a few English symbols.

For all procedures, you need the terminal's Terminal Font Set table. The table lists characters and their decimal and hexadecimal values. For the 2415, 2425, 2435A, 2455, 2475, and 248X Terminals, refer to the terminal's user manual. For the 5055, 6400, 59XX, 17XX, or 11XX Terminal, see Appendix C, "*Terminal Font Set Table*."

Creating the File

Follow these steps to create a table that substitutes characters in the local language for characters normally used in U.S. English.

1. Find the character to replace in the "*National Replacement Character Set*" table on page 149.
2. Locate the character needed from the terminal's font set.
3. Create the NATIVE.XLT file.
4. Put the default value of the font character (found in step 1) as the first byte in the file.
5. Put the replacement value (found in step 2) as the second byte in the file.
6. Repeat steps 1 through 5 until all replacement characters are complete.
7. Download NATIVE.XLT to your terminal and restart the terminal emulation program for the changes to take effect. For downloading information, see "*Downloading Files*" on page 151.

National Replacement Character Sets

The terminal 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.

The following table lists characters in each NRC set that are different from the ASCII set. The terminal does not support “¾” (40 hexadecimal) in the Dutch NRC set.

National Replacement Character Set

	Hexadecimal											
Character Set	**23**	**40**	**5B**	**5C**	**5D**	**5E**	**5F**	**60**	**7B**	**7C**	**7D**	**7E**
ASCII	#	@	[	\	]	^	_	‘	{	|	}	~
British	£	@	[	\	]	^	_	‘	{	|	}	~
Dutch	£		ÿ	½	|	^	_	‘	¨	*f*	¼	’
Finnish	#	@	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
French	£	à	°	ç	§	^	_	‘	é	ù	è	¨
French Canadian	#	à	â	ç	ê	î	_	ô	é	ù	è	û
German	#	§	Ä	Ö	Ü	^	_	‘	ä	ö	ü	ß
Italian	£	§	°	ç	é	^	_	ù	à	ò	è	ì
Norwegian/Danish	#	@	Æ	Ø	Å	^	_	‘	æ	ø	å	~
Portuguese	#	@	Ã	Ç	Õ	^	_	‘	ã	ç	õ	~
Spanish	£	§	¡	Ñ	¿	^	_	‘	‘	°	ñ	ç
Swedish	#	É	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
Swiss	ù	à	é	ç	ê	î	è	ô	ä	ö	ü	û

Greek National Replacement Character Set

(6400, 5055, 59XX, 17XX, 11XX, Terminals)

row	column	0		1		2		3		4		5		6		7	
	b8 bits b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0		00		10		20	•	30	ΐ	40	Π	50	ΰ	60	π	70
1	0 0 0 1		01		11	‘	21	±	31	Α	41	Ρ	51	α	61	ϱ	71
2	0 0 1 0		02		12	’	22	2	32	Β	42		52	β	62	ς	72
3	0 0 1 1		03		13	£	23	β	33	Γ	43	Σ	53	Υ	63	σ	73
4	0 1 0 0		04		14		24	΄	34	Δ	44	Τ	54	δ	64	τ	74
5	0 1 0 1		05		15		25	΅	35	Ε	45	Υ	55	ε	65	υ	75
6	0 1 1 0		06		16	¦	26	Ά	36	Ζ	46	Φ	56	ς	66	φ	76
7	0 1 1 1		07		17	§	27	·	37	Η	47	Χ	57	η	67	χ	77
8	1 0 0 0		08		18	¨	28	Έ	38	Θ	48	Ψ	58	θ	68	ψ	78
9	1 0 0 1		09		19	©	29	Ή	39	Ι	49	Ω	59	ι	69	ω	79
A	1 0 1 0		0A		1A		2A	Ί	3A	Κ	4A	Ϊ	5A	ϰ	6A	ϊ	7A
B	1 0 1 1		0B		1B	«	2B	»	3B	Λ	4B	Ϋ	5B	λ	6B	ϋ	7B
C	1 1 0 0		0C		1C	¬	2C	Ό	3C	Μ	4C	ά	5C	μ	6C	ό	7C
D	1 1 0 1		0D		1D	SHY	2D	½	3D	Ν	4D	έ	5D	ν	6D	ύ	7D
E	1 1 1 0		0E		1E		2E	Ύ	3E	Ξ	4E	ή	5E	ξ	6E	ώ	7E
F	1 1 1 1		0F		1F	—	2F	Ώ	3F	Ο	4F	ι	5F	ο	6F		7F

Example

This example remaps three characters. It applies only to the 2415, 2425, 2435A, 2455, 2475, or 248X Terminal. It does not apply to the 6400, 5055, 59XX, 17XX, or 11XX Terminal because U.S. English values are correct in these terminals.

Example

ASCII hexadecimal file NATIVE.XLT remaps 3 characters
0x23 0x9C
0x40 0xA0
0x4B 0xF8

ASCII Hex Value	Original to be Replaced	Replacement for Display
23	#	£
40	@	Ê
5B	[	°

Downloading Files

2415, 2425, 2435A, 2455, 2475, or 248X Terminal

You can use one of several methods to download a file to a 2415, 2425, 2435A, 2455, 2475, or 248X Terminal, including:

- LOADER.EXE file loader utility. *Note that using this utility will automatically restart the TE 2000 application.*
- T24FCOPY.EXE through a serial connection from your Trakker Antares terminal to your personal computer
- The Download Server feature on the DCS 300 to a Trakker Antares terminal loaded with UDP Plus
- A TFTP application on a personal computer or host to a Trakker Antares terminal loaded with TCP/IP

To use these methods, refer to your terminal's user manual for help.

To restart your TE application, do one of the following:

- Scan the following bar code label. You can also send the Reset Firmware command over the network. For help, refer to your terminal's user manual.

  Reset Firmware

  -.

- Use the TRAKKER Antares 2400 Menu System to configure the Resume Execution command to resume "not allowed." Choose Terminal Menu from the Configuration Menu and then choose Power Management. Each time you press I/O to turn on the Trakker Antares terminal, it boots and restarts your application. See the terminal's user manual.
- Access the TRAKKER Antares 2400 Menu System and select the TE application through the File Manager option. For help, see your terminal's user manual.

The Trakker Antares terminal restarts your TE application using the configuration saved in CONFIG.DAT.

6400, 5055 Terminals

Before you can download a file to a 6400 Computer or , configure INTERLNK on a desktop or laptop PC. INTERLNK, part of MS-DOS, is a device driver that connects your 6400 Computer or 5055 PC and personal computer through their serial ports. This connection enables you to exchange files. It also enables you to edit the 6400 or 5055 configuration files, such as CONFIG.DAT.

INTERSVR is the INTERLNK server and is a communications option on your 6400 Computer or 5055 PC. INTERLNK and INTERSVR are provided with DOS and shipped with your 6400 or 5055 or toolkit. For complete installation instructions, refer to the README.TXT file provided with 6400 Computers and 5055 PCs.

Reprogramming Flash Memory

Flash upgrades can be ordered on diskette from Intermec (contact a representative), or downloaded from the Intermec Knowledge Central (see *Before You Begin*). Read all instructions before proceeding.

Note: Keep terminals on charge when setting up, reprogramming, or reflashing.

Prerequisites for INTERLNK Flash Update

- A working 6400 Computer or 5055 PC to do this procedure(if the flash is corrupted, perform a serial master mode boot).
- A RAM drive (D:) of at least 960 KB.
- A disk file contains the flash archive, 50BDXXXX.EXE. The last four numbers indicate the flash version (*0129* indicates flash version 1.29).
- A standard host PC for connecting to the 6400 Computer or 5055 PC. INTERLNK.EXE must be loaded by the CONFIG.SYS file. For more detailed information on running INTERLNK, refer to a DOS manual.
- A NULL modem cable to connect the host PC’s communications port to the 6400 Computer or 5055 PC with a single dock or communication adapter.

INTERLNK Installation

Use the following instructions if you have a version of flash that includes INTERSVR as a possible Comm option on your 6400 Computer or 5055 PC. If you do not have INTERSVR, you must update the flash using a serial master mode boot.

The following instructions assume that you have placed all files from the self-extracting archive in a directory (on your host personal computer) called C:\PENKEY\FLASH. If you choose to place these files in a different location, adjust the instructions accordingly.

Note: Delete the self-extracting file from this directory once the files are extracted.

Your host personal computer must be running INTERLNK, which is part of MS-DOS. Load INTERLNK as a device driver in your CONFIG.SYS file, using the following statement, at the end of the CONFIG.SYS file (after any other statement that creates a drive letter):

```
DEVICE=C:\DOS\INTERLNK.EXE /DRIVES:3
```

The previous statement assumes that MS-DOS is located in the host PC C:\DOS directory. The /DRIVES: 3 parameter allows mapping of three drives from the 6400 Computer or 5055 PC.

INTERLNK and INTERSVR

INTERLNK is a device driver that interconnects a 6400 Computer or 5055 PC and a host personal computer through serial ports. INTERSVR is the INTERLNK server, a communication option in the Norand Utilities program. These two resources are provided with ROM DOS 6.22 and are shipped with the 6400 Computer or 5055 PC toolkit. A standard null modem cable connects the personal computer to the 6400 Computer or 5055 PC. A TTY TCOM cable also works. A dock is needed for the 6400 Computer or 5055 PC or a communication adapter that plugs onto the bottom end of the terminal.

INTERLNK causes the 6400 Computer or 5055 PC drives to appear as virtual drives on the host personal computer, with drive letters immediately beyond the highest drive letter currently used on the host personal computer. Typing "INTERLNK" from the host personal computer command line displays the designations of the redirected drives. For details of INTERLNK and INTERSVR topics, refer to the DOS on-line help text.

INTERLNK is installed on a host PC, using the following statement in the CONFIG.SYS file:

```
device=c:\dos\interlnk.exe /drives:4
```

After installation, you can copy the application files to the 6400 Computer or 5055 PC. To terminate INTERSVR, press [ALT] + [F4].

59XX, 17XX, 11XX Terminals

For a 59XX, 17XX, or 11XX Terminal, use utility program CHECKCFG.EXE to verify the correctness of your configuration.

Using CHECKCFG to Compile and Decompile Custom Configurations

CHECKCFG reads your configuration and literal files, and reports any syntax errors. It also converts your data files to the proper format for the TE program.

The ASCII text of the configuration file converts to a compressed binary format to save space in the terminal. CHECKCFG can reverse the operation by converting a binary file into its ASCII source. CHECKCFG can also list set-up parameters, their types, and their allowable values.

To display the program version number and a short message that lists the different command line formats for the program, type `checkcfg`

Converting Files from ASCII to Binary

To convert an ASCII parameter file into binary format before downloading it to a terminal, use the following command line:

```
checkcfg <input config file>cfglit.dat<output config file>
```

- <*input config file*> is the name of your ASCII text parameter file
- <*output config file*> is the name of the file that you must append to your TE hex file. The output file must be named CONFIG.DAT.

Converting Files from Binary to ASCII

To convert a binary parameter file back into its ASCII equivalent, use the following command line:

```
checkcfg -r <config file> cfglit.dat <output file>
```

- <*config file*> is the name of your binary parameter file
- CFGLIT.DAT is the name of your terminal literal file
- <*output file*> is the name of the file that will contain the converted ASCII output

Listing Parameters and Values

To display a list of all possible set up parameters and their values, use the following command line: `checkcfg -p cfglit.dat`

Note: The output from this command is about seven hundred lines long. You may want to redirect it into another file for viewing.

Using FLSHCONV.EXE to Build Customized HEX Files

FLSHCONV.EXE is a utility program that creates Intel-hex files in the proper format for downloading to an 11XX, 17XX, or 59XX Terminal. FLSHCONV can do the following:

- Locate and append .EXE programs for download in an Intel-hex file
- Append data and configuration files to an existing Intel-hex file (FLSHCONV *cannot* locate and append .EXE programs to an existing hex file)

To display the program version number and a short message that lists command line formats, type: `flshconv`

The following lists FLSHCONV command line options, where <*hexnum*> represents a hexadecimal segment address and <*number*> represents a decimal number.

- **–f<*hexnum*>**
  <*hexnum*> is the starting address of Flash memory for the terminal. *Default: 8000 (absolute address 0x80000).*
- **–e<*hexnum*>**
  <*hexnum*> is the ending address of Flash. If you omit this parameter, FLSHCONV does not report an error if your Flash image is too big.
- **–d<*hexnum*>**
  <*hexnum*> is the starting address of memory space for .EXE programs. *Default: 200 (absolute address 0x2000).* You should ordinarily use 60 (absolute address 0x600).

- **–o<*filename*>**
  <*filename*> is the Intel-hex output file name.
- **–v<*number*>**
  <*number*> indicates “verbosity.” *Default: 0, meaning no information appears on the standard output device*. Intermec recommends a value of 1 to generate a report of where each file is located. You can use higher values, but they may not produce useful information for the end user.
- **–a**
  This option tells FLSNCONV to append data files to an existing Intel-hex file. The hex file must be the first file name on the command line.

The following chart lists FLSHCONV command line arguments.

Terminal	**-f<*hexnum*>**	**-e<*hexnum*>**	**-d<*hexnum*>**
11XX, 256K Flash	-f8000	-eC000	-d60
11XX, 512K Flash	-f8000	-eFFC0	-d60
17XX, 512K Flash	-f8000	-eFFC0	-d60
5928-5948, 384K Flash	-f9000	-eE000	-d60

Locating and Appending .EXE Files

As an example, suppose you are building a customized version of the 17XX Flash program FWP170H0.HEX. As a minimum, you must have the following files:

• 1700BIOS.EXE	17XX BIOS program
• KERNEL.EXE	Intermec multitasking services
• FWP170H0.EXE	17XX VT/ANSI TE program
• NORAND.FNT	Display character font file. This may be the standard font file or a customized user replacement.
• CFGLIT.DAT	Terminal literal file. This may be the standard literal file or a customized user replacement.

To bind these files into an Intel-hex file that is ready for download to a 17XX, use the following single command line to build a file called NEW.HEX:

```
flshconv -f8000 -effcO -d60 1700bios.exe kernel.exe
fwp170h0.exe norand.fnt cfglit.dat -onew.hex
```

Or, you can create a file that contains each file name on a separate line. Assuming that file was named 1700FILE.LST, use the following command line for the same results:

```
flshconv -f8000 -effc0 -d60 @1700file.lst
```

Appending Data Files To Intel-Hex Files

In the above example, you could have bound the .EXE files together by typing the following (single) command line:

```
flshconv -f8000 -effc0 -d60 1700bios.exe kernel.exe
fwp170h0.exe -o1700exe s.hex
```

You can then customize the 1700EXES.HEX file with different font files and literal files by typing the following (single) command line:

```
flshconv -effc0 -a 1700exes.hex norand.fntcfglit.dat
-onewlang.hex
```

You can provide keyboard translation file pairs to change the default key values. The (single) command line is:

```
flshconv -effc0 -a newlang.hex vt220eml.key vt220eml.xlt
-onewkeys.hex
```

You can specify more than one file pair. For example, you could provide all four file pairs (eight files) to modify the keyboard for each of the four data streams.

Downloading a Hex File

Use these to download a .HEX file to 59XX, 17XX, 11XX Terminals:

- PROGDUX.EXE *(provided with the terminal's Flash)*
- Personal computer
- INTERMEC® cable P/N: 216-806-001, which is a powered Flash cable that plugs into the personal computer's COM port *(note that power is not required for download)*

To download a .HEX file:

1. Ensure the terminal is powered ON and PROGDUX.EXE is loaded on the personal computer.
2. Use cable P/N: 216-806-001 to connect the 11XX or 17XX Terminal to the PC. Use cable P/N: 216-831-001 to connect the 59XX Terminal to the personal computer. Power the terminal off. For help, refer to the terminal's user manual. *Power is not required for download.*
3. To receive the .HEX file, the terminal must be in download mode. To enter this mode, press and hold down the terminal's [I] key as you power up the terminal. For the 17XX 37-key keyboard, press and hold down the [F1] key as you power up the terminal.
4. At the DOS prompt, type: `progdux -?` to display a list of command line options.
5. Erase the terminal's original Flash and download the new .HEX file by typing: `progdux -e <newfile>` where <*newfile*> is the name of the .HEX file.

A successful download will boot the terminal into the TE application. If communications is interrupted, PROGDUX.EXE will send out a negative acknowledgement (NAK) and reset itself to try sending again. No interaction is required except to restore communications.

5 Data Collection Server and Gateway Commands

A Data Collection Server (DCS) or gateway command is either a "parameter" or a "diagnostic." This chapter contains detailed information about each command.

CMT (Return Parameter Values)

This command returns the current value of all CMT parameters.

For compatibility, the DCS or gateway response contains place-holders for commands supported on Intermec® legacy controllers and multiplexers, but *not* supported on Enterprise Wireless LAN™ DCSs and gateways. For more information about compatibility issues, see “*Compatibility with Legacy Systems*” on page 184.

Host Command

```
CMT<cr>
```

DCS 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>
```

DCS 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>`

This example does the following. 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 DCS 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 default values.

Host Command

`CMT0,x<cr>`

DCS or Gateway Response

`OK<cr>`

Example 1

Command: `CMT0,5<cr>`
Action: DCS 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 DCS or gateway waits for data from a terminal after it receives an R (Read) command from the host. Data received within this time passes immediately to the host. The DCS 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 terminal after the time limit is returned after the next R (Read) command.

Host Command

`CMT1,`*`x`*`<cr>`

DCS or Gateway Response

`OK<cr>`

Variables

- **0–254** (one-second increments)
  If the value is "0," the DCS or gateway responds immediately with terminal 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 terminal 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 DCS or gateway responds immediately with terminal 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>
```

DCS or Gateway Response

```
OK<cr>
```

X Variable Values

- **0**
  Disables interactive mode. The DCS or gateway never sends an OK. The host sends commands to the DCS or gateway, which does not respond unless the command has an error. The DCS or gateway sends terminal responses as it receives them *(default)*.
- **1**
  Enables interactive mode. The DCS or gateway responds to every host command whether or not data is requested. If it has no terminal 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 DCS or gateway to the host.

Host Command

```
CMT4,x<cr>
```

DCS 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 DCS 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 DCS 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 DCS 301 is 10 seconds, regardless of the CMT5 setting.

Values for variable *x* are 0-255 (default: 0). “255” disables gap time-out (the DCS or gateway waits indefinitely for a carriage return).

When you enable CMT5, the DCS 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>
```

DCS or Gateway Response

```
OK<cr>
```

Example 1

Command: `CMT5,3<cr>`
Action: DCS 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>`

DCS 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 DCS 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 DCS 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>`

DCS 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 Terminal)

Use CMT13 to automatically enable terminals. When the DCS or gateway receives a power-up message from a terminal, the terminal is automatically included in the active terminal array.

Host Command

```
CMT13,x<cr>
```

DCS or Gateway Response

```
OK<cr>
```

X Variable Values

- **0**
  Disables the terminals. The E (Enable) command must explicitly enable the terminal. If the E command does *not* enable a terminal, and the DCS or gateway receives a power-up message from that terminal, the power-up is ignored and the power-up message (*t*P) is not sent to the host *(default)*.
- 1
  Automatically enables the terminals.

Example

The DCS or gateway enables all responding terminals:
Command: `CMT13,1<cr>`
Response: `OK<cr>`

CMT21 (One-Character Address Mode)

If the DCS or gateway is configured for one-byte addressing (also "RM2216" or "old" mode), it powers up and behaves like an RM2216 Multiplexer. Terminal 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 DCS or gateway is configured for one-byte addressing, terminal power-up messages have the form "1P,060" where "1" is the one-byte address and ",060" is the terminal power-up type. The terminal power-up type resembles the standard three-byte power-up because of the terminal type, but the one-byte address appears.

The default setting is "0," which disables one-byte addressing. If the DCS or gateway is configured for three-character ("normal") addressing, the CMT21 command does nothing.

Host Command

`CMT21,`*`x`*`<cr>`

DCS 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 always enabled on the DCS 301.

Multiple Buffering Mode enables the DCS or gateway to buffer commands the host sends. Up to 25 commands per terminal may be buffered before the DCS or gateway will reject the host command with a "?10" (out-of-buffers) response. This may happen if the host program issues commands to a terminal that is not powered up or is out of range. The terminal must act on the commands before the DCS or gateway allows the host to send more commands. *See "Write Display Errors" on page 185 for more information.*

If CMT22 is disabled, the DCS 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 DCS or gateway in succession without the device fully acting on the command, the commands may be overwritten and lost.

Host Command

```
CMT22,x<cr>
```

DCS 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 terminal every second. For a Native terminal, the command syntax may be W000DB/08:38:00<cr>, followed 1 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 DCS or gateway does not guarantee that the data will go to the terminal. 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 DCS or gateway. In this case, the second, third, and fourth commands were not sent to the terminal.

Host Command	DCS or Gateway Response	Terminal 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 DCS or gateway and sent to the terminal sequentially. (However, there is no guarantee that these commands will arrive at the terminal 1 second apart.) Following is an example.

Host Command	DCS or Gateway Response	Terminal 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 terminal 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 not recognized.
- If you type the wrong character, you cannot use the backspace key to correct it. You must wait 1 second or press [**Enter**] before you can try 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>
```

DCS 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 DCS or gateway transmit buffer. The device responds by returning a status code for each terminal in the network (up to 127 codes).

Host Command

```
?<cr>
```

DCS or Gateway Response

```
xxx...x<cr>
```

? (Get Transmit Message Status) Codes

- **A** Active terminal
- **C** Terminal has a pending message
- **I** Inactive terminal
- **N** Disabled terminal
- **P** Inactive terminal has a pending message

Example

Command: `?<cr>`
Response: `CCCCAAAAPPPPNNN0N<cr>`
This response has 127 characters and shows the following:
Terminals 000–003 have pending messages.
Terminals 004–007 are active.
Terminals 008–011 have pending messages, but are inactive.
Terminals 012–126 are disabled.

E (Enable Terminals)

The E command enables the terminals (communication channels) in a network. You can either enable all terminals with one command, or a specific terminal.

Host Command

`Es0s<cr>` *(enables terminals 000-126)*
`Ets` *(enables a specific terminal)*

DCS or Gateway Response

`OK<cr>`

Variables

- *s*
  N = No *(disables terminal)*
  Y = Yes *(enables terminal)*
  X = "Do not care" *(retains current status)*
- *t*
  Terminal number *(000–126)*

The first Y, N, or X setting applies to terminal 000, the second applies to terminal 001, the third to terminal 002, etc. You do not need to list all the terminals.

Before a terminal can be polled or sent a message, it must be enabled by the host or allowed to enable and identify itself to the DCS or gateway. Enabling a terminal opens a channel to that terminal.

Example 1

Command: `EYYYYYXNNNNNNYYYYYYYYYY<cr>`
Action: Enables terminals 000–004 and 012–021. Does not change terminal 005. Disables terminals 006–011. Does not change terminals 022–126.

Example 2

Command: `E001Y<cr>`
Action: Enables terminal 001.

R (Read)

The R command is the idle, polling command. Use it to enable the DCS or gateway to transmit terminal data to the host. This is the only command where the device can send the terminal responses. If the DCS or gateway has a terminal response queued for transmission to the host, it cannot send it in response to a Write Display (WtD) command. It can only send it in response to an R command.

Host Command

```
R<cr>
```

DCS or Gateway Response

```
tK<cr>
tS<cr>
tP,x<cr>
tD
OK<cr>
```

Variables

- *t* Terminal number *(000–126)*
- K Keyboard data
- S Scanned data
- P Terminal has reset or powered-up
- *x* Terminal power-up type
- D Echo data

In general, an R command must follow the commands the host sends to the DCS or gateway. This enables the DCS or gateway to respond to or forward any data it received from a terminal and stored.

The DCS or gateway sends an OK<cr> to the host when interactive mode is enabled, and the CMT1 time-out value has been exceeded and the device has no data to send to the host.

The following table lists the terminal power-up types. The logical screen size for all terminals is 80x24, of which the terminal displays a portion. Power-up type “000” means the terminal has not powered up.

Terminal Power-Up Types

Data Stream	Host View Size	Power-Up Type
Native	16 21 Any size other than 16 or 21	032 033 060
Non-Telnet 3270	16 21 80 Any size other than 16, 21, or 80	034 035 037 061
Telnet 3270	Any size	071
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 terminal 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 DCS 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>
```

DCS or Gateway Response

```
<cr>
```

Example

Command: `T<cr>`
Action: The DCS or gateway repeats the last message sent to the host, and adds a carriage return.

F (Return Terminal Power-Up Type)

The F command is useful for listing all terminals 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>. See the "Terminal Power-Up Types" table on page 177 for more information.

Host Command

```
F<cr>
```

DCS 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 terminals 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 DCS or gateway. The DCS 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>
```

DCS 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 DCS 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>
```

DCS 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 DCS 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 DCS 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 DCS or gateway's software version number to the host. The value for variable *x.xx* is the version number.

Host Command

```
DMV<cr>
```

DCS or Gateway Response

```
QXSASYNC V<x.xx><cr>
```

Example 1

The following sample response appears when the DCS 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 DCS 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 DCS or gateway
- Commands destined for the terminal

Assume the system is set to interactive mode (CMT2,1). In general, the DCS 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 DCS or gateway accepts the terminal commands (with an OK<cr>) and then sends them to the terminal. The DCS or gateway receives the terminal response. After the host issues an R Read command (page 176), the DCS or gateway can send the terminal response.

In general, a terminal response cannot be sent in response to a non-Read command. An exception to this is the T (Repeat Response) command (page 178), where the last DCS 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`

Compatibility With Legacy Systems

Legacy (pre-Enterprise Wireless LAN) systems with INTERMEC multiplexers and controllers include the PC1000 PCMUX, RM2216, RM3216, RM3242, RM1202, RM1204, RC2250, RC3250, RCB4000, and RCB4030. They support various commands in this manual, and some or all of these commands.

Most of the commands listed below are specific for crystal-UHF or SST (Spread Spectrum Transmission) systems. If you are replacing a legacy system with a DCS 301, 6910 IGAP, 6950 Enterprise Gateway Server, or Wireless Network Access Server, your DCS or gateway should accept the command and return an “OK” response (if interactive mode [CMT2] is enabled).

- CMT3
- CMT6
- CMT7
- CMT11
- CMT12
- CMT14
- CMT15
- CMT16
- CMT17
- CMT18
- CMT20
- CMT23
- CMT24
- CMT25
- CMT26
- CCS
- DMS
- DMC

Write Display Errors

Error responses from the DCS or gateway take the form "?*x*"<cr>, where *x* is the numeric error code listed in the following table.

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 terminal command. For the WtD (Write Display) terminal command, syntax checking stops at the '/'.
4	Length Error	Host message is too long. If the DCS 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 terminal (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 165) sets the time.
8	Check Error	Occurs when CMT8 Set Data Error Checking (page 166) 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 terminal 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 DCS or gateway.

6 Terminal Commands

This chapter describes each terminal command with detailed information.

G (Reset Terminal)

The G command returns the terminal to its first "state," a known condition. All terminals reset when variable *t* (terminal number) is omitted. Terminals respond to a poll with their power-up message.

In the command, the range of *t* is 000–126. If the data collection 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 terminal)*
`G<cr>` *(reset all terminals)*

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tP,x<cr>
```

R Response Variables

- *t* Terminal number *(000–126)*
- P Terminal has reset or powered-up
- *x* Terminal power-up type *(See the "Terminal Power-Up Types" table on page 177.)*

Example

Command: `G002<cr>`
Action: Data collection server or gateway resets terminal 002.
R Response: `002P,032<cr>`

StD/ (Set Terminal Parameters)

Note: This is for 6400, 5055, 59XX, 17XX, 11XX Terminals only.

S*t*D/ sets terminal operating control parameters once, when you power up the terminal.

Command

```
StDo/c<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tC0<cr>
```

Command and R Response Variables

- ***t*** Terminal number *(000–126)*
- ***o*** Option *(see the “StD/ Options” table below)*
- ***c*** Bar code parameter *(default: UPC/EAN)*. Lengths default to a maximum of 31 and a minimum of 1 *(see the “StD/ Bar Code Parameters” table below)*.

StD/ Options

o	Description	Default
B	Disable key-pressed buzzer	Enabled
D*x*	Duration of buzzer for Write Display command and input errors. *x*=0-9 in 30 ms increments.	120 msec (*x*=4)
F*x*	*x*=character that appears for the [FUNC] key (\ not allowed)	<*space*>
S*x*	*x*=character that appears for the [SKIP] key <*space*> (\ not allowed)	<*space*>
T*x*	Gross time-out duration (*x*=0-255 sec)	30 sec

StD/ Bar Code Parameters

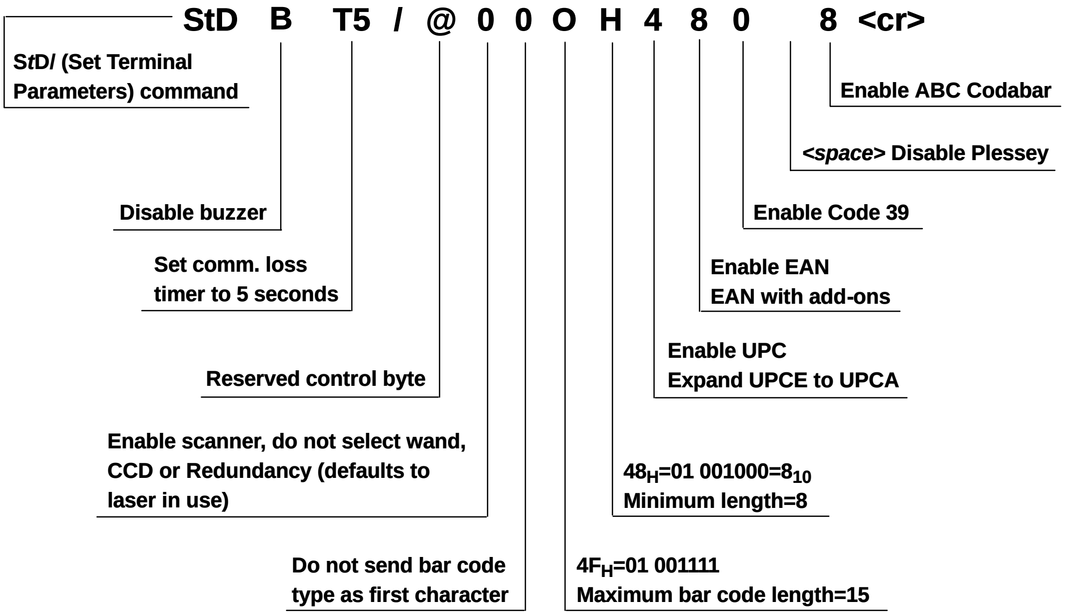
Byte Position	Description	Bit Position
1	Bar code block number (reserved).	01000000
2	Bit 4=Enable scanner Bit 3=CCD select Bit 2=Wand select Bit 1=CCD redundancy bit Bit 0=Laser scanner redundancy bit	001*XXXXX*
3	Bit 4=Disable bar code type character Bit 3=Unused Bit 2=Unused Bit 1=Must be 0 Bit 0=Enable MOD 10 check digit	001*X*00*XX*
4	Maximum bar code length. Bits 0-4=Length	010*XXXXX*
5	Minimum bar code length. Bits 0-4=Length	010*XXXXX*
6	Bit 4=Decode UPC Bit 3=Decode UPC with add-ons Bit 2=Expand UPC E to UPCA Bit 1=Enable UPC E type 1 only	001*XXXX*0
7	Bit 4=Decode EAN Bit 3=Decode EAN with add-ons	001*XX*000
8	Bit 4=Decode Code 39 Bit 3=Decode extended Code 39	001*XX*000
9	Plessey control algorithms: Bit 4=Decode Plessey Bit 3=0 (Discard 2nd check digit) Bit 3=1 (Keep 2nd check digit) Bit 2=0 (Validate 1st check digit) Bit 2=1 (Do not validate) Bit 1=0 (Discard 1st check digit) Bit 1=1 (Keep 1st check digit) Bit 0=0 (MOD 11 — 1st check digit) Bit 0=1 (MOD 10 — 1st check digit)	001XXXXX
10	Enable Codabar algorithms. Bit 4=Decode ABC Codabar Bit 3=Decode Codabar	001*XX*000
11	Enable Code 11 algorithm. Bit 4=Decode Code 11	001*X*0000
12	Enable Code 93 algorithm. Bit 4=Decode Code 93	001*X*0000
13	Enable Code 128 algorithm. Bit 4=Decode Code 128	001*X*0000
14	Enable Straight 2 of 5 and Computer Identics 2 of 5 algorithms. Bit 4=Decode Straight 2 of 5 Bit 3=Decode C-Identics 2 of 5	001*XX*000

StD/ Bar Code Parameters (continued)

Byte Position	Description	Bit Position
15	Maximum bar code length for Straight and Computer Identics 2 of 5. Bits 0-4=Length	010*XXXXX*
16	Minimum bar code length for Straight and Computer Identics 2 of 5. Bits 0-4=Length	010*XXXXX*
17	1st fixed bar code length for Straight and Computer Identics 2 of 5. Bits 0-4=Length	010*XXXXX*
18	2nd fixed bar code length for Straight and Computer Identics 2 of 5. Bits 0-4=Length	010*XXXXX*
19	Enable Interleaved 2 of 5 algorithm. Bit 4=Decode Interleaved 2 of 5	001*X*0000
20	Maximum bar code length for Interleaved 2 of 5. Bits 0-4=Length	010*XXXXX*
21	Minimum bar code length for Interleaved 2 of 5. Bits 0-4=Length	010*XXXXX*
22	1st fixed bar code length for Interleaved 2 of 5. Bits 0-4=Length	010*XXXXX*
23	2nd fixed bar code length for Interleaved 2 of 5. Bits 0-4=Length	010*XXXXX*

Enable a bar code by giving the parameters at least as many bytes as the position of the desired control byte. For example, to enable Plessey, parameters must have at least nine bytes. When bit positions have been determined, ASCII equivalents integrate into the following character.

01011101=5D (hex)="]" (ASCII)


StD B T5 / @ 0 0 O H 4 8 0 8 <cr>
StD/ (Set Terminal Parameters) command
Enable ABC Codabar
<*space*> Disable Plessey
Disable buzzer
Enable Code 39
Set comm. loss timer to 5 seconds
Enable EAN EAN with add-ons
Reserved control byte
Enable UPC Expand UPCE to UPCA
Enable scanner, do not select wand, CCD or Redundancy (defaults to laser in use)
48_H=01 001000=8_{10} Minimum length=8
Do not send bar code type as first character
$4F_H$=01 001111 Maximum bar code length=15

This illustration shows an example of an expanded StD/ command code.

The following is a bit pattern hexadecimal to ASCII chart.

Hexadecimal-to-ASCII Chart

Bit Pattern	Hex	ASCII	Bit Pattern	Hex	ASCII	Bit Pattern	Hex	ASCII
00100000	20	<space>	01000000	40	@	01100000	60	‘
00100001	21	!	01000001	41	A	01100001	61	a
00100010	22	”	01000010	42	B	01100010	62	b
00100011	23	#	01000011	43	C	01100011	63	c
00100100	24	$	01000100	44	D	01100100	64	d
00100101	25	%	01000101	45	E	01100101	65	e
00100110	26	&	01000110	46	F	01100110	66	f
00100111	27	’	01000111	47	G	01100111	67	g
00101000	28	(	01001000	48	H	01101000	68	h
00101001	29	)	01001001	49	I	01101001	69	i
00101010	2A	*	01001010	4A	J	01101010	6A	j
00101011	2B	+	01001011	4B	K	01101011	6B	k
00101100	2C	,	01001100	4C	L	01101100	6C	l
00101101	2D	-	01001101	4D	M	01101101	6D	m
00101110	2E	.	01001110	4E	N	01101110	6E	n
00101111	2F	/	01001111	4F	O	01101111	6F	o
00110000	30	0	01010000	50	P	01110000	70	p
00110001	31	1	01010001	51	Q	01110001	71	q
00110010	32	2	01010010	52	R	01110010	72	r
00110011	33	3	01010011	53	S	01110011	73	s
00110100	34	4	01010100	54	T	01110100	74	t
00110101	35	5	01010101	55	U	01110101	75	u
00110110	36	6	01010110	56	V	01110110	76	v
00110111	37	7	01010111	57	W	01110111	77	w
00111000	38	8	01011000	58	X	01111000	78	x
00111001	39	9	01011001	59	Y	01111001	79	y
00111010	3A	:	01011010	5A	Z	01111010	7A	z
00111011	3B	;	01011011	5B	[	01111011	7B	{
00111100	3C	<	01011100	5C	\	01111100	7C	|
00111101	3D	=	01011101	5D	]	01111101	7D	}
00111110	3E	>	01011110	5E	^	01111110	7E	~
00111111	3F	?	01011111	5F	_	01111111	7F	DEL

WtB (Audio Annunciator)

W*t*B turns on the terminal audio tone for a specified time, in increments of 1/4 second up to 2.25 seconds.

Command

```
WtBd<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tC0<cr>
```

Command and R Response Variables

- *t* Terminal number *(000–126)*
- *d* Audio tone duration times 250 milliseconds *(1–9 seconds)*

Example

Command: `W010B8<cr>`
Action: Terminal 010 sounds tone for 2 seconds.

WtD/ (Write Display)

W*t*D/ sends data to a terminal and enables its response. W*t*D/ downloads the prompt, which appears on the terminal, and sets up input parameters. The single input field is returned to the host.

Command

```
WtDo/x<cr>
```

DCS 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*** Terminal number *(000–126)*
- ***o*** Options *(see the "Write Display Options" table on the next page)*
- ***x*** Data

  |
  (0x7C) Allows all command extensions
  (see "Command Extensions" on page 196)

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 the lines are 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 terminal 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

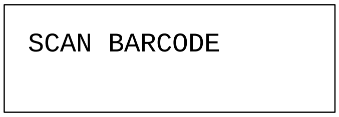
- *t* Terminal number *(000–126)*
- K Keyboard data
- *r* Optional ASCII display row number
- *d* Keyboard input data
- S Scanner data. If "F" option is enabled, response is always "S."
- *b* Bar code type
- *c* Bar code data
- *m* MOD 10 check digit, if enabled

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 be substituted for |D*xx* allows for a macro extension defined by the W*t*D/\B (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

|D*xx* *(00 ≤ xx ≤ 99)*

Blank

Syntax

|*xx* *(00 ≤ xx ≤ 99 blank spaces)*

Beep

The terminal 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 extension sends hexadecimal characters to the terminal. It is a pass-through mode that bypasses the data collection server or gateway's handling of this character.

For example, if you want to send a carriage return to a printer attached to a terminal, and if the host program sent the <cr> character, the data collection server or gateway would terminate the command. By sending a ^0D, the host sends these three characters through the data collection server or gateway to the terminal. The terminal converts the three characters into one byte.

Syntax

`^`*`xx`*

Example 1

This command sends the string "Hello World<cr><lf>" out the terminal's RS-232 port:

```
W000D/\P\Hello World^0D^0A<cr>
```

Example 2

This command displays "ABCD" on the terminal'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 terminal and enable terminal response mode. W*t*D// concatenates (appends) strings of data. It also permits setting up input parameters for the terminal's response. The single input field returns to the host upon data entry.

Command

```
WtD//Fx<cr>
WtD//Mx<cr>
WtDo//Lx<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tKrd<cr>
tSbcm<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Terminal number *(000–126)*
- **F** First in chain
- **M** Middle of chain
- **L** Last in chain
- ***o*** Options *(see the "Write Display Options" table on page 195)*
  x Data
  |
  (0x7C) Allows all command extensions
  (see "Command Extensions" on page 196)

R Response Variables

- ***t*** Terminal 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) — The terminal clears all chained data and replaces it with new data starting with the marked sequence.
- /L=(last in chain) — The terminal attaches the current field, and executes the Write Display command using options set for the /L command.
- /M=(middle of chain) — The terminal attaches the 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

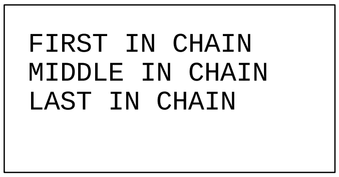
Command: `W003D//MMIDDLE IN CHAIN <cr>`
Response: `OK<cr>`

Example 4

Command: `R<cr>`
Response: `003K\\0<cr>`

Example 5

Command: `W003DBS//LLAST IN CHAIN<cr>`
Action: Terminal 003 sounds its tone and enables its scanner.



```
FIRST IN CHAIN
MIDDLE IN CHAIN
LAST IN CHAIN
```

This illustrates a sample terminal display for this command.

WtD/\yS (Download Table)

A *table* is a block of prompt and input field pairs downloaded to the terminal. The terminal can store a number of tables at the same time. Host commands activate individual tables.

While a table appears, the terminal operator can enter data to any or all its *input* fields before the input data is transmitted to the host.

W*t*D/\yS defines and sends a table to the terminal. The terminal stores the table definition until a W*t*D/\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 197.

Command

```
WtDo/\yS\p\fx\0\p\fx<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal 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*)
  (see "Input Field Control Byte" on page 201)
- *x* Input field

  |
  (0x7C) Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

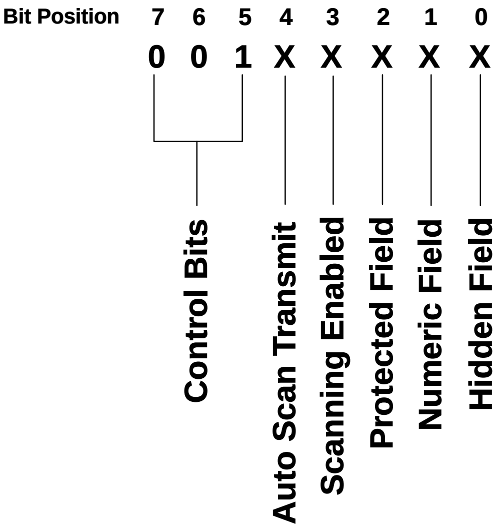
R Response Variables

- *t* Terminal 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 illustration defines the 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.



Control Bits – 001 identifies the input field control byte
Auto Scan Transmit – Automatically transmits table after a good scan
Scanning Enabled – Enables bar code scanner
Protected Field – No data entry allowed; data always returns to the host
Numeric Field – Numeric data only, no alphabetic, etc.
Hidden Field – Input data not displayed

This illustrates the definition input field control byte bits.

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 and input field pair must be smaller than the terminal'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
two times the number of pairs
must be
less than 1600 for tables 0 through 4
and be
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\QTY? \”|11<cr>
```

Response: `OK<cr>`

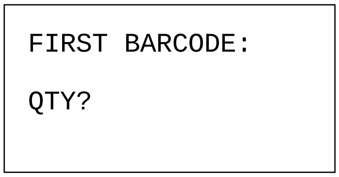
Table 0 in terminal 010 erases and a new set of prompt and input field pairs loads.

The first prompt is "1ST BARCODE:" followed by a blank extension of 4 spaces. This moves 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 terminal display.

WtD/ \y (Display Table)

W*t*D/*y* displays table *y* defined by the W*t*D/*y*S (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>
```

DCS 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*** Terminal 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

o	Description	Default
B	Sounds tone (S*t*D/ sets the duration).	Tone off
C	Displays terminal 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
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* Terminal 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 has been sent.)*

Example

Using the table from the W*t*D/\yS (Download Table) command:

Host command: `W010DQ/\0<cr>`
Response: `OK<cr>`

Terminal allows the operator to enter data in table 0 of terminal 010.

Scanning autoadvance is enabled. The terminal automatically advances to the next input field after getting a valid scan.

After the user presses [ENTER], the terminal responds to an R (Read) command from the host by uploading the stored input field data.

WtD/ \yT (Download and Display Table)

W*t*D/*y*T combines the W*t*D/*y*S (Download Table) and W*t*D/*y* (Display Table) commands. It downloads prompts and input fields into a buffer memory of the terminal 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 197.

Command

```
WtDo/\yT\p\fx\p\fx\p\fx\
0\p\fx<cr>
```

DCS 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*** Terminal 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)*
  (see "Input Field Control Byte" on page 201)
- ***x*** Input field

| *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

Note: See "*Rules for Designing Tables*" on page 202 for information about:

- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

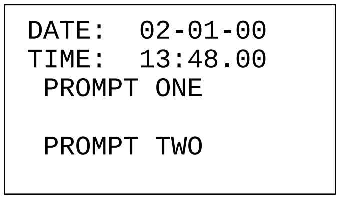
- *t* Terminal number *(000–126)*
- *y* Table number *(0–99)*
- *n* Input field number *(0–99)*
- *g* Input data type. If *g*=S and *x*=bar code type
    - *K=Keyboard*
    - *S=Scanner*
- *x* Input field
- M Indicates more data to come. Host continues to send an R (Read) command until it receives a 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
```

This is a sample terminal 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 the following 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 197.

Command

```
WtDo/\yU\i\{Fp\fx}
or
{Pp}
or
{Ifx}0<cr>
```

DCS 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* — Terminal 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)* *(see "Input Field Control Byte" on page 201)*
- *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 *(see "Command Extensions" on page 196)*.

Note: See "*Rules for Designing Tables*" on page 202 for information about:

- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

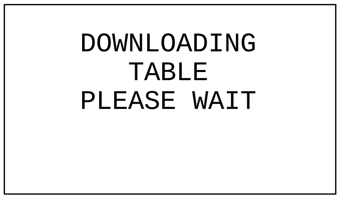
- *z* Response code
  - *0=Message Received*
  - *1=Bad Command*
  - *2=Field Length is 0*
  - *3=Data Overflow*
  - *4=Bad Table Number*
  - *5=Prompt Field, No Input Field*
  - *6=Field Does Not Exist*
  - *7=Missing Prompt and Input Field Pair*
  - *8=Field Pair Number Out of Range*
  - *9=Format 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>`

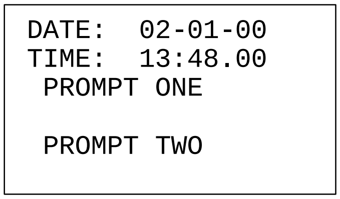


```
   DOWNLOADING
      TABLE
   PLEASE WAIT
```

This is a sample terminal display.

Example 2

Host command: `W000DBCUWQS/\0<cr>`
Response: `OK<cr>`
R response: `000K\\0<cr>`

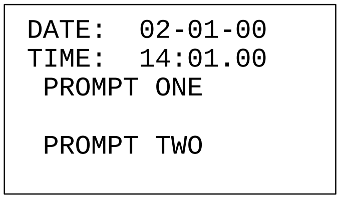


```
DATE:  02-01-00
TIME:  13:48.00
 PROMPT ONE

 PROMPT TWO
```

This is a sample terminal display.

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
```

This is a sample terminal display.

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 have been defined.
- 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 have been defined.
- 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 197.

Command

```
WtDo/\yV\i\{Fp\fx}
or
{Pp}
or
{Ifx}0<cr>
```

DCS 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* Terminal number *(000–126)*
- *o* Options *(see the "WtD/\y (Display Table) Options" table on page 203)*
- *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)* *(see "Input Field Control Byte" on page 201)*
- *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 *(see "Command Extensions" on page 196).*

Note: See "*Rules for Designing Tables*" on page 202 for information about:
- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

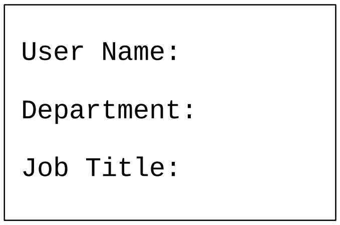
- *t* Terminal number *(000–126)*
- *y* Table number *(0–99; Length is 16725 bytes; # pairs is 25)*
- *n* Input field number *(0–99)*
- *g* Input data type *(K=Keyboard; S=Scanner)*
- *x* Input field
- M Indicates more data to come. Host continues to send an R (Read) command until it receives a response 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:
```

This is a sample terminal display.

Suppose the user enters the following data:

```
User Name:
John Doe
Department:
Applications
Job Title:
Programmer II
```

This is a sample terminal display.

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/*y*V (Update and Display Table) command allows the host to transmit to the terminal only the necessary changes. A command may be:

Host: **`W003DBCQ/\2V\0\FUser Number: |04\"|16<cr>`**
Response: **`OK<cr>`**

Note that:

- The terminal 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.
- The input field control byte changed to accept numeric data only.

The terminal 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 Terminal Function Keys)

If you want to map the [**F1**] key, do the following.

- *6400, 5055, 59XX, 17XX, or 11XX Terminal*:
  If you want [**F1**] to map to FUNC-0, you must enable the "F1 is FUNC-0" option in the terminal's terminal emulation configuration menus. This option ensures compatibility with INTERMEC legacy controllers and multiplexers.
- *2415, 2425, 2435A, 2455, 2475, and 248X Terminals running Native:*
  [**F1**] is mapped to FUNC-0 by default.

W*t*D/\F defines the terminal's 25 function keys. Buffers can contain a maximum of 16 characters.

Command

```
WtD/\F\\f1\f2\0\f25<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal number *(000–126)*
- *f1...f25* Operation of function keys 1–25

R Response Variables

- *t* Terminal 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 terminal.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "*Pass-Through Hex Character*" on page 197.

Command

```
WtD/\B\x\x\0\x<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal number *(000–126)*
- *x* Blank data (macro)

  |
(0x7C) Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

R Response Variables

- *t* Terminal 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*

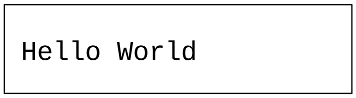
The following information applies to banks:

- 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/\B\Hello \World<cr>`
R response: `000K\\0<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 197.

Command

`WtD/\BU\nn\x<cr>`

DCS or Gateway Response

`OK<cr>`

R Response

`tK\\z<cr>`

Command Variables

- *t* Terminal number *(000–126)*
- *nn* Bank number *(00–99)*
- *x* Bank data (or macro)
  |
  (0x7C) Allows the Blank and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

R Response Variables

- *t* Terminal 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/\B\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 is longer than Data String)`

Example 3

Host: `W000D/\BU\00\Short<cr>`
R Response: `000K\\0<cr>`

Example 4

Display command: `W000D/|D00<cr>`

```
Short
```

WtD/ \BE (Erase Bank)

W*t*D/\BE 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>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- ***t*** Terminal number *(000–126)*
- ***nn*** Bank number *(00–99)*

R Response Variables

- ***t*** Terminal 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: `000K\\0<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>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal 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* Terminal 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 terminal'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/\PS (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 197.

Command

```
WtDo/\P\x<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- ***t*** Terminal number *(000–126)*
- ***o*** RS-232 port options *(see the following table)*
- ***x*** Data

| *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

RS-232 Port Options

o	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. *For values, see the "P Values" table on the next page.*	0 (1 stop bit, 8 data bits, even parity, parity disabled)
W	CTS busy	No wait for CTS

R Response Variables

- *t* Terminal 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 (for help, see the example below). No combination of values 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 terminal 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 terminal 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 197.

Command

```
WtD/\PMx<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal number *(000–126)*
- *x* Host-defined message to appear during RS-232 port operation

  | *(0x7C)* Allows the Bank and Blank command extensions *(see “Command Extensions” on page 196).*

R Response Variables

- *t* Terminal 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 you use a user-defined message, you can reset it to the default message by issuing W*t*D/\PM with a message length of 0. The minimum defined character string is one character. The maximum character string depends on screen size and font size.

Example 1

Host: `W000D/\PM Here is the message to appear during RS-232 port commands.<cr>`
Terminal response: `OK<cr>`

Host command: `R<cr>`
Terminal response: `000K\\0<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>`
Terminal response: `OK<cr>`

Host command: `R<cr>`
Terminal 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>`
Terminal response: `OK<cr>`

Host command: `R<cr>`
Terminal 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>`
Terminal response: `OK<cr>`

Host command: `R<cr>`
Terminal response: `000K\\3<cr>`

Status code "3" is returned because the blank extension caused the message to exceed the screen length. Note that although the host sent only 15 characters to the terminal, the total expanded character count is 316, which exceeds the screen size.

WtD/ \PS (RS-232 Port Options)

W*t*D/\PS establishes the baud rate, data bits, parity, and flow control at the terminal's RS-232 port. This command is sent before a W*t*D/*y*P (Print Table) command to initialize the RS-232 port. The port setup options consisting of P, L, H, W, D, and C variables (see the "RS-232 Port Options" table on page 219) control the RS-232 port.

Command

```
WtDo/\PS<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal number *(000–126)*
- *o* RS-232 port options *(see the "RS-232 Port Options" table on page 219)*

R Response Variables

- *t* Terminal 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: W*t*DP10L1C/\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. Use this command when the device sends a continuous data stream to the terminal.

Command

```
WtDo/\PGi<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\x<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Terminal number *(000–126)*
- ***o*** RS-232 port options *(see the "RS-232 Port Options" table on page 219)*
- ***i*** RS-232 input options *(see the following table)*

WtD/\PG** (Get Data from an External Device)* ***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	Start Character 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 (terminal waits *xx* sec for data)	*xx*=decimal number from 00-99. If *xx*=00, the terminal waits indefinitely for RS-232 input or a key press to terminate.	05 sec

R Response Variables

- *t* Terminal 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 terminal connects to an external scale. The scale continuously sends weight readings to the terminal 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>`

  The terminal stops receiving data after 10 characters. The problem is the terminal 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, the terminal 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 terminal 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 terminal starts saving data once the start character is encountered, and stops after the maximum number of characters has been 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 terminal's RS-232 port.

Individual hexadecimal characters can be escaped with the "^*xx*" extension. For more information, see "*Pass-Through Hex Character*" on page 197.

Command

```
WtDo/\PFi\y<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\x<cr>
tK\\z<cr>
```

Command Variables

- ***t*** Terminal number *(000–126)*
- ***o*** RS-232 port options *(see the "RS-232 Port Options" table on page 219)*
- ***i*** RS-232 input options *(see the "WtD/\PG RS-232 Input Options" table on page 224)*
- ***y*** Data transmitted to external device

| *(0x7C)* Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

R Response Variables

- ***t*** Terminal 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/ \yP (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>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z<cr>
```

Command Variables

- *t* Terminal number *(000–126)*
- *y* Table and data to print

R Response Variables

- *t* Terminal 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. (The 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. You can accomplish this by one of the following methods:

- Replace the prompt and input field pair together.
- Replace the prompt field only.
- Replace 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 197.

Command

```
WtDo/\yR\i\{Fp\fx}
or
{Pp}
or
{Ifx}0<cr>
```

DCS 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* Terminal 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*) *(see "Input Field Control Byte" on page 201)*
- *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 *(see "Command Extensions" on page 196)*.

Note: See "*Rules for Designing Tables*" on page 202 for information about:

- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- *t* Terminal 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 following table)*
- *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 "P Value" table page 220*	
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 terminal 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 "*Pass-Through Hex Character*" on page 197.

Command

```
WtDo/\yQ\i\{Fp\fx}
or
{Pp}
or
{Ifx}0<cr>
```

DCS 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*	Terminal number *(000–126)*
• *o*	Options *(see the "WtD/\yR Options" table on page 231)*
• *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*) *(see "Input Field Control Byte" on page 201)*
• *x*	Data
• P	Update prompt field only
• I	Update input field only
| *(0x7C)*	Allows the Bank, Blank, and Pass-Through Hex Stream command extensions *(see "Command Extensions" on page 196)*.

Note: See "*Rules for Designing Tables*" on page 202 for information about:

- Replacing the prompt and input fields together.
- Restrictions.
- Replacing the prompt field only.
- Replacing the input field only.

R Response Variables

- *t* Terminal 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)

W*t*D/\T sets the volume, frequency (pitch), and length (duration) of the tone.

Command

```
WtD/\Tvvvffflll
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tK\\z
```

Command Variables

- *t* Terminal number *(000–126)*
- *v* Volume *(000–255)*
- *f* Frequency or pitch *(000–030)*
- *l* Length or duration *(001–010)*

R Response Variables

- *t* Terminal 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 (Terminal Emulation Version)

Use D*t*V to return the terminal emulation version running in the terminal. The command variable is *t*, the terminal number (000-126). Following are R response variables and characters.

Command

```
DtV<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tDFWP<ab0H0> V<v.vv><cr>
```

Command and R Response Variables

- ***t*** Terminal number *(000–126)*
- **D** Diagnostic command or response
- **FWP** Terminal emulation program
- ***ab*** First two numbers in the terminal model name:
  - *24=2415, 2425, 2435A, 2455, 2475, or 248X Terminals*
  - *64=6400 Hand-Held Computer*
  - *65=5055 Data Collection PC*
  - *59=59XX Terminal*
  - *17=17XX Terminal*
  - *11=11XX Terminal*
- **H** Represents the combined Native, 3270, 5250, and VT/ANSI terminal emulation programs. *The 2415, 2425, 2435A, 2455, 2475, and 248X Terminals do not support the "FWP" program's Native option.*
- ***v*** Terminal emulation program's version number

Example

Command: `D003V<cr>`
R Response: `003DFWP240H0 V6.17<cr>`

Terminal 003 is a 2415, 2425, 2455, 2475, or 248X running terminal emulation program FWP240H0, version 6.17.

DtE (Echo-Back Diagnostic)

D*t*E tests the communication cycle. The host sends a data string to a terminal, which it immediately returns for comparison. The "D" in the R response indicates "diagnostic command or response."

Command

```
DtE/x<cr>
```

DCS or Gateway Response

```
OK<cr>
```

R Response

```
tDx<cr>
```

Command and R Response Variables

- *t* Terminal number *(000–126)*
- *x* Data *(1–112 characters)*

Example

Command: `D002E/ABCD1234<cr>`
Terminal response: `002DABCD1234<cr>`
Terminal 002 returns the same data in response to the next R (Read) command.

A Bar Code Scanning

This appendix lists bar code labels for Native TE commands and functions. It also contains Encoded Code 39 key press sequences.

Note: To scan the bar code labels, configure the terminal to use Code 39 in Full ASCII mode.

Note: For the 2415, 2425, 2435A, 2455, 2475, or 248X Terminal, configure the mode through the TRAKKER Antares® 2400 Menu System. See the terminal's user manual for help.

DK, Display Column Spacing

The spacing parameters define the number of pixels to be added between each row or column of characters. Note that font arrays have one or two lines built in for character separation.

```
DK<param1><param2><param3>
```

where:

- *param1* represents the font type and is one character
- *param2* represents column spacing and is two characters
- *param3* represents row spacing and is two characters

2415 and 2425 Terminal Displays

Below are the examples that apply to the 2415 and 2425 Terminal displays:

10 Columns

10 columns x 8 rows

$+DK20000

12 Columns

12 columns x 4 rows

$+DK90116

12 columns x 6 rows

$+DK90105

12 columns x 8 rows

$+DK90100

12 columns x 10 rows

$+DK80502

12 columns x 12 rows

$+DK80500

12 columns x 16 rows

$+DK70700

17 Columns

17 columns x 4 rows

*$+DK80122”

17 columns x 6 rows

$+DK80111

17 columns x 8 rows

$+DK80106

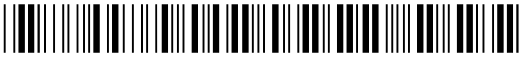
17 columns x 10 rows

$+DK80102

17 columns x 12 rows

$+DK80100

17 columns x 16 rows



$+DK70300

17 columns x 21 rows

$+DK60400

20 Columns

20 columns x 8 rows

$+DK10000

20 columns x 16 rows

$+DK00000

22 Columns

22 columns x 4 rows

$+DK70124

22 columns x 6 rows

$+DK70113

22 columns x 8 rows

$+DK70108

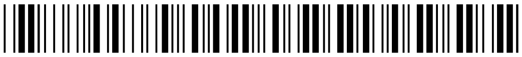
22 columns x 10 rows

$+DK70104

22 columns x 12 rows

$+DK70102

22 columns x 16 rows



$+DK70100

22 columns x 21 rows

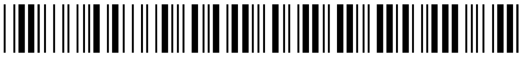
$+DK60200

26 Columns

26 columns x 4 rows

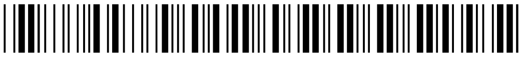
$+DK70024

26 columns x 6 rows



$+DK70013

26 columns x 8 rows



$+DK70008

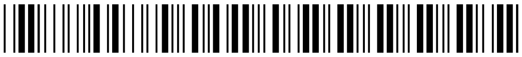
26 columns x 10 rows

$+DK70004

26 columns x 12 rows

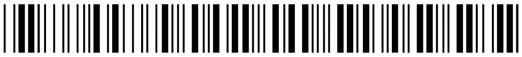
$+DK70002

26 columns x 16 rows



$+DK70000

26 columns x 21 rows



$+DK60100

32 Columns

32 columns x 21 rows

$+DK60000

2435A Terminal Display

Below are the examples that apply to the 2435A Terminal display:

9 Columns

9 columns x 8 rows

$+DK20000

12 Columns

12 columns x 4 rows

$+DK90116

12 columns x 6 rows

$+DK90105

12 columns x 8 rows

$+DK90100

12 columns x 10 rows

$+DK80502

12 columns x 12 rows

$+DK80500

12 columns x 16 rows

$+DK70700

17 Columns

17 columns x 4 rows

$+DK80122

17 columns x 6 rows

$+DK80111

17 columns x 8 rows

$+DK80106

17 columns x 10 rows

$+DK80102

17 columns x 12 rows

$+DK80100

17 columns x 16 rows

$+DK70300

17 columns x 21 rows

$+DK60400

19 or 20 Columns

Note: For 2435A Terminals with Terminal Emulation version 6.60 or greater and firmware version 7.10 or greater, if icons are enabled in the firmware, then the 2435A Terminal uses the 19 columns x 8 rows bar code. If the icons are disabled in the firmware, then the 2435A Terminal uses the 20 columns x 8 rows bar code.

19 Columns

19 columns x 8 rows

$+DK10000

19 columns x 16 rows

$+DK00000

20 Columns

20 columns x 8 rows

$+DK10000

20 columns x 16 rows

$+DK00000

22 Columns

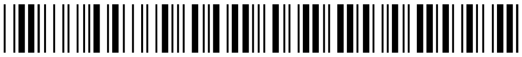
22 columns x 4 rows

$+DK70124

22 columns x 6 rows

$+DK70113

22 columns x 8 rows



$DK70108

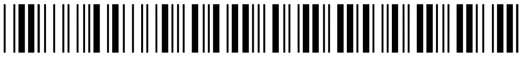
22 columns x 10 rows

$+DK70104

22 columns x 12 rows

$+DK70102

22 columns x 16 rows



$+DK70100

22 columns x 21 rows

$+DK60200

26 Columns

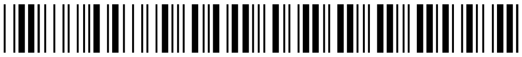
26 columns x 4 rows

$+DK70024

26 columns x 6 rows

$+DK70013

26 columns x 8 rows



$+DK70008

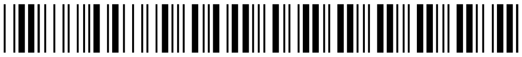
26 columns x 10 rows

$+DK70004

26 columns x 12 rows

$+DK70002

26 columns x 16 rows



$+DK70000

26 columns x 21 rows

$+DK60100

31 or 32 Columns

31 Columns

31 columns x 16 rows

$+DK60002

32 Columns

32 columns x 16 rows

$+DK60002

Note: For 2435A Terminals with Terminal Emulation version 6.60 or greater and firmware version 7.15 or greater, if icons are enabled in the firmware, then the 2435A Terminal uses the 31 columns x 16 rows bar code. If the icons are disabled in the firmware, then the 2435A Terminal uses the 32 columns x 16 rows bar code.

31 Columns

31 columns x 16 rows

$+DKB0000

32 Columns

32 columns x 16 rows

$+DKB0000

2455 Terminal Display

Below are the examples that apply to the 2455 Terminal display:

10 Columns

10 columns x 8 rows

$+DK20000

12 Columns

12 columns x 4 rows

$+DK90116

12 columns x 6 rows

$+DK90105

12 columns x 8 rows

$+DK90100

12 columns x 10 rows

$+DK80502

12 columns x 12 rows

$+DK80500

12 columns x 16 rows

$+DK70700

17 Columns

17 columns x 4 rows

$+DK80122

17 columns x 6 rows

$+DK80111

17 columns x 8 rows

$+DK80106

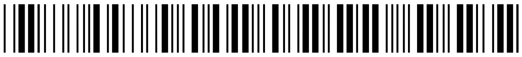
17 columns x 10 rows

$+DK80102

17 columns x 12 rows

$+DK80100

17 columns x 16 rows



$+DK70300

17 columns x 21 rows

$+DK60400

20 Columns

20 columns x 8 rows

$+DK10000

20 columns x 16 rows

$+DK00000

22 Columns

22 columns x 4 rows

$+DK70124

22 columns x 6 rows

$+DK70113

22 columns x 8 rows

$+DK70108

22 columns x 10 rows

$+DK70104

22 columns x 12 rows

$+DK70102

22 columns x 16 rows

$+DK70100

22 columns x 21 rows

$+DK60200

26 Columns

26 columns x 4 rows

$+DK70024

26 columns x 6 rows

$+DK70013

26 columns x 8 rows

$+DK70008

26 columns x 10 rows

$+DK70004

26 columns x 12 rows

$+DK70002

26 columns x 16 rows

$+DK70000

26 columns x 21 rows

$+DK60100

32 Columns

32 columns x 21 rows

$+DK60000

40 Columns

40 columns x 12 rows

$+DK20000

53 Columns

53 columns x 12 rows

$+DK90000

80 Columns

80 columns x 12 rows

$+DK10000

80 columns x 20 rows

$+DK80000

80 columns x 25 rows

$+DK00000

2475 and 248X Terminal Displays

Below are the examples that apply to the 2475 and 248X Terminal displays:

10 Columns

10 columns x 8 rows

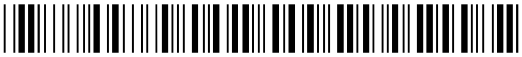
$+DK20000

12 Columns

12 columns x 4 rows

$+DK90116

12 columns x 6 rows



$+DK90105

12 columns x 8 rows

$+DK90100

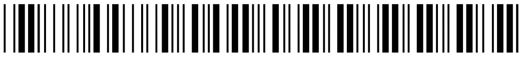
12 columns x 10 rows

$+DK80502

12 columns x 12 rows

$+DK80500

12 columns x 16 rows



$+DK70700

17 Columns

17 columns x 4 rows

$+DK80122

17 columns x 6 rows

$+DK80111

17 columns x 8 rows

$+DK80106

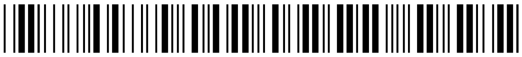
17 columns x 10 rows

$+DK80102

17 columns x 12 rows

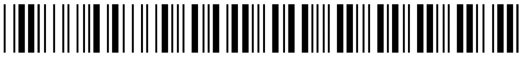
$+DK80100

17 columns x 16 rows



$+DK70300

17 columns x 21 rows



$+DK60400

20 Columns

20 columns x 8 rows

$+DK10000

20 columns x 12 rows

$+DK20000

20 columns x 16 rows

$+DK0000

22 Columns

22 columns x 4 rows

$+DK70124

22 columns x 6 rows

$+DK70113

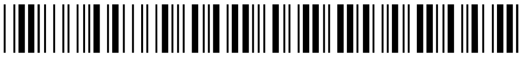
22 columns x 8 rows

$+DK70108

22 columns x 10 rows

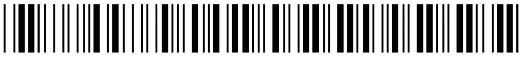
$+DK70104

22 columns x 12 rows



$+DK70102

22 columns x 16 rows



$+DK70100

22 columns x 21 rows

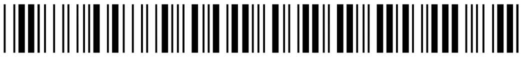
$+DK60200

26 Columns

26 columns x 4 rows

$+DK70024

26 columns x 6 rows



$+DK70013

26 columns x 8 rows

$+DK70008

26 columns x 10 rows

$+DK70004

26 columns x 12 rows

$+DK90000

26 columns x 16 rows

$+DK70000

26 columns x 21 rows

$+DK60100

32 Columns

32 columns x 21 rows

$+DK60000

40 Columns

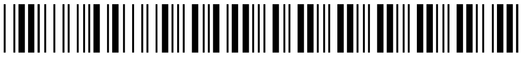
40 columns x 12 rows

$+DK10000

40 columns x 20 rows

$+DK80000

40 columns x 25 rows



$+DK00000

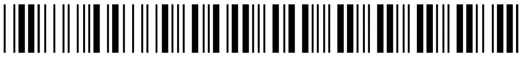
53 Columns

53 columns x 25 rows

$+DK70000

64 Columns

64 columns x 25 rows



$+DK60000

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

Tab Keys

Back Tab

%BTAB

Forward Tab

%TAB

AID-Generating Keys

Clear

%CLR

Enter

%CR

F1

%F1

F2

%F2

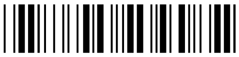
F3

%F3

F4

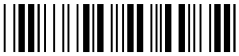
%F4

F5



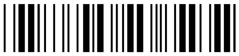
%F5

F6



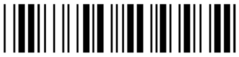
%F6

F7



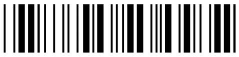
%F7

F8



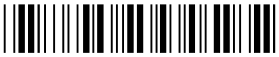
%F8

F9



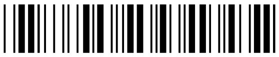
%F9

F10



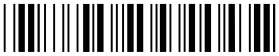
%F10

F11



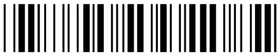
%F11

F12



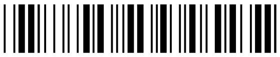
%F12

F13



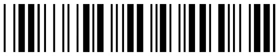
%F13

F14



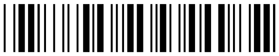
%F14

F15



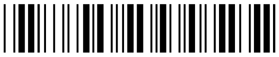
%F15

F16



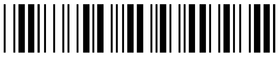
%F16

F17



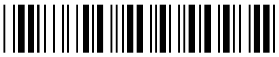
%F17

F18



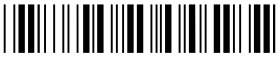
%F18

F19



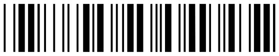
%F19

F20



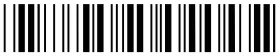
%F20

F21



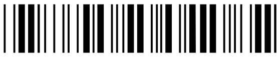
%F21

F22



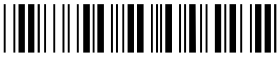
%F22

F23



%F23

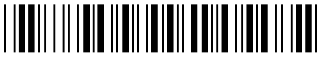
F24



%F24

Auto-Login Restart

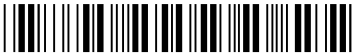
Auto-Login Restart



%ALRS

Native Additional Functions

TE configuration menus



%TECFG

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 terminal will beep and the data stream will be flushed.

Key Press Sequences for Encoded Code 39

Sequence	Key	Sequence	Key
$A	Reserved	+A	a
$B	Del	+B	b
$C	Forward Tab	+C	c
$D	Forward Tab	+D	d
$E	Back Tab	+E	e
$F	Reserved	+F	f
$G	Reserved	+G	g
$H	Backspace	+H	h
$I	Reserved	+I	i
$J	Reserved	+J	j
$K	Insert	+K	k
$L	Home	+L	l
$M	Enter (t)	+M	m
$N	End of Field	+N	n
$O	Clear	+O	o
$P	Reserved	+P	p
$Q	PF1 (t)	+Q	q
$R	PF2 (t)	+R	r
$S	PF3 (t)	+S	s
$T	PF4 (t)	+T	t
$U	PF5 (t)	+U	u
$V	PF6 (t)	+V	v
$W	PF7 (t)	+W	w
$X	PF8 (t)	+X	x
$Y	PF9 (t)	+Y	y
$Z	PF10 (t)	+Z	z

Key Press Sequences for Encoded Code 39 (continued)

Sequence	Key	Sequence	Key
%A	Clear AID (t)	/A	! (exclamation mark)
%B	PF11 (t)	/B	” (double quote)
%C	PF12 (t)	/C	# (pound)
%D	Reserved	/D	$ (dollar)
%E	Reserved	/E	% (percent)
%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)	/K	+ (plus)
%L	\ (backslash)	/L	, (comma)
%M	] (right brace)	/M	- (hyphen)
%N	^ (circumflex)	/N	F14
%O	_ (underscore)	/O	/ (forward slash)
%P	{ (left brace)	/P	F15
%Q	| (vertical bar)	/Q	F16
%R	} (right brace)	/R	F17
%S	~ (tilde)	/S	F18
%T	Delete	/T	F19
%U	Reserved	/U	F20
%V	@ (at)	/V	F21
%W	′ (grave accent)	/W	F22
%X	Reserved	/X	F23
%Y	Reserved	/Y	F24
%Z	Reserved	/Z	: (colon)

Terminating Keys

Terminating keys are the nonprintable ASCII sequences and action keys. When the terminal encounters them in a bar code, an action is taken, and the terminal 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 terminal-to-base station transmission. The terminal ignores data in the bar code buffer following these keys once a transmission takes place.

For example, the terminal interprets the following sequence:

```
123$V456
```

as

```
123F6
```

The terminal will not send "456" to the host computer, because it follows terminating key F6.

ASCII sequences can be used any time before a terminating key. For example, the terminal interprets

```
+H+E+L+L+O$M
```

as

```
hello<Enter>
```

Escape Characters

The four escape characters in the previous table yield a 3270 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 terminal 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

When the terminal is in **Auto Entr Scan** or **Auto Tab Scan** mode, eight Encoded Code 39 functions will override these modes when they are scanned. They are:

- Forward Tab
- Back Tab
- Home
- Clear
- End of Field
- Backspace
- Insert
- Delete

These codes are all of the screen editing type, where an automatic [Enter] would not be desired. The encoded operations listed will never allow an **Auto Entr Scan** to occur. For example, suppose **Auto Entr Scan** was enabled (through the TE configuration menus) and a "$C" (Forward Tab) is scanned. The terminal will forward tab to the next field but *not* cause an [Enter] operation, even though the **Auto Entr Scan** feature was enabled. The encoded forward tab overrides the **Auto Entr Scan** mode in this case. However, if a "+D" (d) is scanned, the terminal places a "d" at the current cursor location. The **Auto Entr Scan** mode then causes an [Enter] to be executed.

B Bar Code Symbologies

This appendix contains a brief explanation of each bar code symbology that the Native Terminal Emulation application decodes. It explains some of the general characteristics and uses of these bar code types.

Bar Code Algorithms

You can enable specific bar code algorithms using the setup menus or the host computer. Once the computer correctly decodes a bar code, the computer encodes data with descriptive information about the symbol. Response time is improved by limiting the computer to bar codes being used.

Bar Code Data String Formats

Bar Code Type Character	Symbology Type	Data Format	Data Length
"0"	UPC short (UPC-E)	nddddddc	8
"1"	EAN short (EAN-8)	fndddddc	8
"2"	UPC long (UPC-A)	nddddddddddc	12
"3"	EAN long (EAN-13)	fnddddddddddc	13
"4"	UPC short add-on 2	nddddddcaa	10
"5"	EAN short add-on 2	fndddddcaa	10
"6"	UPC long add-on 2	nddddddddddcaa	14
"7"	EAN long add-on 2	fnddddddddddcaa	15
"8"	UPC short add-on 5	nddddddcaaaaa	13
"9"	EAN short add-on 5	fndddddcaaaaa	13
":" (colon)	UPC long add-on 5	nddddddddddcaaaaa	17
";" (semicolon)	EAN long add-on 5	fnddddddddddcaaaaa	18
"<" (less than)	Interleaved 2 of 5	d......d	Scan device dependent
"=" (equal sign)	Standard 2 of 5	d......d	Scan device dependent
">" (greater than)	Plessey	d......dc	Scan device dependent
"@" (ampersand)	Codabar	sd....ds	Scan device dependent
"S"	Code 11	d......d	Scan device dependent
"P"	Code 39	d......d	Scan device dependent
"Q"	Extended Code 39	d......d	Scan device dependent
"R"	Code 93	d......d	Scan device dependent
"J"	Code 128	d......d	Scan device dependent

Note: These bar code data definitions apply to the Data Format column in the previous table:

a	Add-on code digits
c	Check digits
d	Bar code digits
f	EAN flag 1 characters
n	Number system digits
s	Start and stop digits

If MOD 10 or MOD 11 check digits are enabled, the digit falls at the end of a bar code data string. Each check digit enabled extends the bar code data string length by one character.

The Native Terminal Emulation application recognizes eleven of the most widely used bar code symbologies. With bar code symbologies, like languages, there are many different types. A bar code symbology provides the required flexibility for a particular inventory tracking system.

A symbology may be for particular industries, such as food and beverage, automotive, railroad, or aircraft. Some of these industries have established their own bar code symbology because other symbologies did not meet their needs.

Without going into great detail on the bar code structure, note that no two products use the same bar code. Each product gets a unique bar code.

Industries that use a particular type of bar code symbology have formed regulating committees or are members of national institutes that issue and keep track of bar codes. This ensures that each organization that contributes to a particular industry conforms to its standard. Without some form of governing body, bar coding would not work.

- UPC (Universal Product Code) with/without add-ons
- EAN (European Article Numbering Code) with/without add-ons
- Codabar
- C11 (Code 11)
- C39 (Code 39)
- C93 (Code 93)
- C128 (Code 128)
- I 2 of 5 (Interleaved 2 of 5 Code)
- S 2 of 5 (Standard 2 of 5)
- Plessey
- MSI (a variant of Plessey)

UPC

The UPC (Universal Product Code) is the symbology used throughout the grocery and retail industries. This bar code symbology contains two pieces of numerical information encoded on the bar code, producer identification, and product identification information.

The UPC symbol is 12 characters long. The first character of the UPC symbol is a number system character, such as “0” for grocery items and “3” for drug- and health-related items.

The UPC symbology is for retail environments such as grocery stores, convenience stores, and general merchandise stores.

Some retail items are so small that a standard UPC bar code cannot fit on the packaging. When this occurs there is a permitted shorter version of the UPC symbology, referred to as UPC-E. UPC-E is six characters long (eight including number system and check digit), approximately half the size of a standard UPC bar code.

EAN

EAN (European Article Numbering) symbology is similar to UPC symbology, except that it contains 13 characters and uses the first two to identify countries.

The EAN symbology is used throughout most of Europe in the retail environment. Although similar to UPC symbology, the two are not interchangeable.

Codabar

Codabar was for retail price-labeling systems. Today it is widely accepted by libraries, medical industries, and photo finishing services.

Codabar is a discrete, self-checking code with each character represented by a stand-alone group of four bars and three intervening spaces.

Four different start or stop characters get defined and designated “a”, “b”, “c”, and “d”. These start and stop characters are constructed using one wide bar and two wide spaces. A complete Codabar symbol begins with one of the start or stop characters followed by some number of data characters and ending in one of the start or stop characters.

Any of the start or stop characters may be used on either end of the symbol. It is possible to use the 16 unique start or stop combinations to identify label type or other information.

Since Codabar is variable-length, discrete, and self-checking, it is a versatile symbology. The width of space between characters is not critical and may vary significantly within the same symbol. The character set consists of “0” through “9”, “-”, “$”, “:”, “/”, “.”, and “+”.

The specific dimensions for bars and spaces in Codabar optimize performance of certain early printing and reading equipment. Codabar has 18 different dimensions for bar and space widths. So many different dimensions often result in labels printed out of specification and cause Codabar printing equipment to be more expensive.

Code 11

Code 11 satisfies the requirements for a very high density, discrete numeric bar code. The name Code 11 derives from 11 different data characters that can be represented, in addition to a start or stop character.

The character set includes the 10 digits and the dash symbol. Each character is represented by a stand-alone group of three bars and two intervening spaces. Although Code 11 is discrete, it is not self-checking. A single printing defect can transpose one character into another valid character. One or two check digits obtain data security.

The specifications for Code 11 suggest that this code should have a narrow element width of 7.5 mils. This results in an information density of 15 characters per inch.

Code 39

Code 39 (C39) is the most widely used symbology among the industrial bar codes. Most major companies, trade associations, and the federal government find this code to fit their needs. The main feature of this symbology is the ability to encode messages using the full alphanumeric character set, seven special characters, and ASCII characters.

Programming for this symbology can be for any length that the application requires. The application program handles symbology that is at least one character but no more than 32 characters in length.

When programming the computer for Code 39, it is important to set the symbology limit as close as possible (minimum and maximum bar code lengths being scanned). Doing so keeps the computer bar code processing time to a minimum and conserves battery power.

Bar code readers can respond to Uniform Symbology Specification symbols in non-standard ways for particular applications. These methods are not for general applications, because of the extra programming required. Code 39 Full ASCII is one example of non-standard code.

Encoded Code 39 (Concatenation)

If the first data character of a symbol is a space, the reader may be programmed to append the information contained in the remainder of the symbol to a storage buffer. This operation continues for all successive symbols that contain a leading space, with messages being added to the end of previously stored ones. When a message is read which does not contain a leading space, the contents are appended to the buffer, the entire buffer is transmitted, and the buffer is cleared.

Encoded Code 39 (Full ASCII)

If the bar code reader is programmed for the task, the entire ASCII character set (128 characters) could be coded. This is done using two character sequences made up of one of the symbols (“$”,“.”,“%”,“/”) followed by one of the 26 letters.

Code 93

The introduction of Code 93 provided a higher density alphanumeric symbology designed to supplement Code 39. The set of data characters in Code 93 is identical with that offered with Code 39. Each character consists of nine modules arranged into three bars and three spaces.

Code 93 uses 48 of the 56 possible combinations. One of these characters, represented by a square, is reserved for a start or stop character, four are used for control characters, and the remaining 43 data characters coincide with the Code 39 character set. An additional single module termination bar after the stop character concludes the final space.

Code 93 is a variable length, continuous code that is not self-checking. Bar and spaces widths may be one, two, three, or four modules wide. Its structure uses edge-to-similar-edge decoding. This makes the bar code immune to uniform ink spread, which allows liberal bar width tolerances.

Code 93 uses two check characters. Its supporters believes this makes it the highest density alphanumeric bar code. The dual check digit scheme provides for high data integrity. All substitution errors in a single character are detected for any message length.

Code 128

Code 128 (C128) is one of the newest symbologies used by the retail and manufacturing industries. It responds to the need for a compact alphanumeric bar code symbol that could encode complex product identification.

The fundamental requirement called for a symbology capable of being printed by existing data processing printers (primarily dot-matrix printers) that produce daily, work-in-progress, job, and product traceability documents. The ability to print identification messages between 10 and 32 characters long, on existing forms and labels deemed an important requirement.

Code 128 uniquely addresses this need as the most compact, complete, alphanumeric symbology available.

Additionally, the Code 128 design with geometric features, improves scanner read performance, does self-checking, and provides data message management function codes.

Code 128 encodes the complete set of 128 ASCII characters without adding extra symbol elements. Code 128 contains a variable-length symbology and the ability to link one message to another for composite message transmission. Code 128, being a double-density field, provides two numeric values in a single character.

Code 128 follows the general bar code format of start zone, data, check digit, stop code, and quiet zone. An absolute minimum bar or space dimension of nine mils (0.010 inch minimum nominal ± 0.001 inch tolerance) must be maintained.

Characters in Code 128 consist of three bars and three spaces so that the total character set includes three different start characters and a stop character.

UCC/EAN-128 Shipping Container Labeling is a versatile tool that can ease movement of products and information. The Shipping Container Labeling bar code can take any form and usually has meaning only within the company or facility where applied.

Because this *random* data can get mistaken later for an industry standard code format, the UCC and EAN chose a symbology uniquely identified from these other bar codes. This standard is for maximum flexibility, to handle the diversity of distribution in global markets by cost efficiency.

The UCC/EAN-128 Container Labeling specification calls for a FUNC1 to immediately follow the bar code's start character. FUNC1 also follows any variable-length application field. The specification also calls for the computer to send "]C1" for the first FUNC1. The specification requires that the computer send a "<GS>" (hex 1D) for subsequent FUNC1 codes in the bar code.

Because "<GS>" is not compatible with computer emulation data streams, the Uniform Code Council has been asked to change the specification. This change is made to send the same three character sequence "]C1" to identify the embedded FUNC1 codes.

This implementation should provide for clean application coding by identifying the same sequences for the same scanned codes. If the communication of Norand bar code types is enabled, the Shipping Container Label codes precede with a "J". These strings will appear on the computer display. The application may have to allow for strings longer than 48 characters (maximum length indicated in the specification). Actual length variance depends on the number of variable-length data fields. Allowing for 60 characters should be sufficient. Within the Code 128 specification, the computer can link bar codes together. If this is to happen, allow for more characters (computer limit is 100 characters).

The Application Identifier Standard, that is part of the UCC/EAN Shipping Label concept, complements, rather than replaces, other UCC/EAN standards. Most UCC/EAN standards primarily identify products.

Several industries expressed the need to standardize more than product identification. The UCC/EAN Code 128 Application Identifier Standard supplies this tool. The standard adds versatility for inter-enterprise exchanges of perishability dating, lot and batch identification, units of use measure, location codes, and several other information attributes.

For more detailed information on Code 128 UCC/EAN Shipping Label bar code and Application Identifier Standard, refer to the UCC/EAN-128 Application Identifier Standard specification.

I 2 of 5 (Interleaved)

I 2 of 5 (Interleaved 2 of 5 Code) is an all-numeric symbology, widely used for warehouse and heavy industrial applications. Its use has been particularly prevalent in the automobile industry. The I 2 of 5 symbology can be placed on smaller labels than what the standard UPC symbology requires.

I 2 of 5 also provides a little more flexibility on the type of material it can print on. Interleaved 2 of 5 Code has its name because of the way the bar code is configured.

I 2 of 5 bars and spaces both carry information. The bars represent the odd number position digits, while spaces represent the even number position digits. The two characters are interleaved as one. Messages encoded with this symbology have to use an even number of characters since two numeric characters always get interleaved together.

S 2 of 5 (Standard 2 of 5)

The code S 2 of 5 (Standard 2 of 5 Code) is designed primarily for:

- Warehouse inventory handling
- Identification of photo finishing envelopes
- Airline tickets
- Baggage and cargo handling

The code S 2 of 5 is simple and straightforward. All information is contained in the widths of the bars, with the spaces serving only to separate the individual bars.

Bars can either be wide or narrow, and the wide bars are usually three times the widths of the narrow bars. Spaces may be any reasonable width but are typically equal to the narrow bars. Narrow bars are identified as zero bits and wide bars as one bits.

Remember the code structure by associating the bar positions from left to right with weighting factors 1, 2, 4, 7, and parity. Exceptions to this rule are zero, start, and stop. This code is a discrete code, since the white spaces between the characters are not part of the code. Because the white spaces carry no information, their dimensions are not critical.

The S 2 of 5 code is self-checking, meaning a scanner passing through a printing void would detect the proper ratio of wide bars to total bars. When the scanner spots an error, a non-read will occur.

Plessey

Plessey finds its origin in the pulse width modulated (PWM) code developed in England. It is widely used for shelf markings in grocery stores. Pulse width modulated codes represent each bit of information by a bar and space pair. A zero bit consists of a narrow bar followed by a wide space, while a one bit consists of a wide bar followed by a narrow space. It is mainly a numeric symbology (0–9) with six extra characters available for assigning any symbol or letter desired.

Plessey codes are not self-checking and employ a variety of check characters. Plessey employs a polynomial-based Cyclic Redundancy Check (CRC). For start and stop characters, Plessey employs a 1101 and previously used a 0101.

This symbology is very limited about what information can be encoded. It is not considered for new applications.

MSI Code (Variant of Plessey)

In addition to Plessey characteristics, the MSI Code employs a Modulus 10 Check. For start and stop checks, MSI employs a single bit pair of 1 as a start symbol and a single bit pair of 0 as a stop symbol. MSI reverses the 1-2-4-8 BCD pattern for bit pair weighting to 8-6-2-1.

C Terminal Font Set Table

Note: The Terminal Font Set table in this appendix applies to 6400, 5055, 59XX, 17XX, and 11XX. For the Terminal Font Set table for the 2415, 2425, 2435A, 2455, 2475, or 248X Terminals, refer to the terminal's user manual.

6400, 5055, 59XX, 17XX, and 11XX Terminals use a font set that supports English and Western European languages, such as French, German, Italian, Portuguese, Spanish, and others. You can develop applications that display any character in the terminal font set. Depending on the type of keyboard, you can also enter many of the characters. For help, see Chapter 2, "*Using the Terminal's Keyboard.*"

The following table lists the characters you can display on the terminal. It also lists the decimal and hexadecimal index values.

Terminal Font Set
(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
Space	0	00
Ã	1	01
ã	2	02
Õ	3	03
õ	4	04
∅	5	05
¨	6	06
—	7	07
ↄ	8	08
⁊	9	09
∩	10	0A
≡	11	0B
ψ	12	0C
Δ	13	0D
|	14	0E
¤	15	0F
▶	16	10
◀	17	11
ϟ (“communications loss” annunciator)	18	12
X̄ (5250 dup character)	19	13
X (“input inhibited” annunciator)	20	14
§	21	15
⌐ (“message waiting” annunciator)	22	16
⊞ (“radio” annunciator)	23	17
Unused	24	18
Unused	25	19
→ (”transmit” annunciator)	26	1A
← (”receive” annunciator)	27	1B
▮ (“battery” annunciator)	28	1C
Space	29	1D
▲	30	1E
Ý	31	1F

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

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
> (greater than)	62	3E
? (question mark)	63	3F
@ (ampersand)	64	40
A	65	41
B	66	42
C	67	43

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
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
f	102	66
g	103	67

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
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

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
î	140	8C
ì	141	8D
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
₧	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

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
░	176	B0
▒	177	B1
▓	178	B2
│	179	B3
┤	180	B4
╡	181	B5
╢	182	B6
╖	183	B7
╕	184	B8
╣	185	B9
║	186	BA
╗	187	BB
╝	188	BC
┘	189	BD
╛	190	BE
┐	191	BF
└	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

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
╥	210	D2
╙	211	D3
╘	212	D4
╒	213	D5
╓	214	D6
╫	215	D7
‡	216	D8
┘	217	D9
┌	218	DA
█	219	DB
▄	220	DC
▌	221	DD
▐	222	DE
▀	223	DF
∝	224	E0
β	225	E1
Γ	226	E2
Π	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
ϵ	238	EE
∩	239	EF
≡	240	F0
±	241	F1
≥	242	F2
≤	243	F3
⌠	244	F4

Terminal Font Set (continued)

(6400, 5055, 59XX, 17XX, 11XX Terminals)

Character	Decimal	Hexadecimal
⌡	245	F5
÷	246	F6
≈	247	F7
°	248	F8
○	249	F9
▪	250	FA
√	251	FB
^{n}	252	FC
^{2}	253	FD
■	254	FE
Space	255	FF

D Native Sockets

The INTERMEC® Data Collection Server (DCS) 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 DCS or gateway. In fact, the host application runs on a separate physical host.

Norand® Native is a simple command language for controlling terminals from a host application. Native sockets enables a host computer running a Native application to communicate with Native terminals through the DCS or gateway via a TCP/IP socket connection.

The DCS 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 DCS 301 from the 6950 Enterprise Gateway Server, the Wireless Network Access Server, or a serial controller, refer to the *DCS 301 User's Guide* (P/N 070284). The manual contains information about modifying your host application to use a socket interface on the DCS 301.

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 DCS or gateway by configuring both with a common port number. The DCS 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 terminals and the DCS or gateway with a common host name, which defines the communication sessions.

2 Configure each terminal for the Native data stream and assign it a unique terminal number.

As each terminal powers on, the host application opens a unique socket between the DCS or gateway and the host for the terminal. The DCS or gateway keeps an internal table mapping terminals to socket numbers. However, terminal numbers are *not* available to the application. It must use socket numbers to identify each terminal session.

Sockets remain open until the host application closes the session or someone powers the terminal 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 Technologies 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 Chapter 6, "*Terminal Commands,*" for details about the Native command syntax. The application reads and writes directly to the terminals through the DCS 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 DCS or gateway required use of a modified Native syntax. The following table provides a cross reference. Letter conventions are excerpted from Chapter 6, "*Terminal Commands.*"

All commands terminate with a carriage return <cr>. The DCS or gateway implementation maintains the 1024-character maximum message length limitation documented in Chapter 5, "*Data Collection Server and Gateway Commands.*"

Abbreviated Commands

Command	DCS 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/ D/\	W*t*D/ W*t*D/\
Reset Terminal	g	G*t*
Terminal Firmware Version	v	D*t*V
Echo Back Diagnostic	Not supported	DtE/

** All "D/" and "D/\" command extensions are supported except Send Multiple Write Displays, "D//".*

Note: The DCS 301 does not support the DtE (Echo-Back Diagnostic) command in a Native socket host.

Response Formats

Chapter 6, "*Terminal Commands*," lists response formats for each command. However, with Native sockets, the terminal 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>". See the WtD/ (Write Display) command in Chapter 6. "*Terminal Commands*," for descriptions of the parameters.

Error Messages

Native command syntax is processed partly by the DCS or gateway, and partly by the terminal. For example, the WtD/ (Write Display) command has an options list *o* and message content *x* in the form "D*o*/*x*<cr>". The DCS or gateway processes content preceding the forward slash, and the terminal processes content after the slash.

Error responses from the DCS 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 185.

The terminal's error responses depend on the specific command. See Chapter 6, "*Terminal Commands*," for the command that produced the error.

Normal Communication

Each terminal 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 terminal (terminal emulation or Native).

Coding the Application

Code does not need to be designed to handle multiple terminals. If two operators select the same application number, a copy of the application runs for each terminal.

Syntax checking is not done on messages transmitted through the DCS or gateway to the terminal. Check the response to an asynchronous command for error notifications from the DCS or gateway.

Syntax for Terminals

See Chapter 6, "*Terminal Commands*," for details about Norand Native syntax.

WtD/ (Write Display) Command

You may use any WtD/ command specified in Chapter 6, "*Terminal Commands*." Intermec recommends that you familiarize yourself with these commands before writing any code.

For WtD/ commands, the communication driver in the DCS or gateway adds the "W." If the "W" is included here, the message is rejected.

The DCS 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 responses from the terminal.

Abbreviated Terminal 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 terminal 002, sending "eD" would send "S002D". The next response would be "CO".
"v" = DtV	Sending a "v" requests the version number of the terminal. That is, sending a "v" issues a DtV command to the terminal. The next message from that terminal should be its software version and terminal number. Internet socket applications return only a version number.
"g" = G or Gt	Sending a "g" resets the terminal. The next message from that terminal should be a power-up message.

Responses received from the terminals follow the responses listed in Chapter 6, "*Terminal Commands*." (See this chapter for more information about the format of incoming messages.) In addition, the DCS or gateway can return codes in the terminal's number field. They are explained under "*Native Mode Responses*" on the next page.

Native Mode Responses

The DCS or gateway sends the following codes in the terminal number field returned to the application.

Codes in Terminal Number Field

Command	DCS 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 terminal 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 terminal is left unattended with the application running (same as NUI_DEAD).
999	The DCS 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 DCS or gateway shuts down or is reset, or if the DCS or gateway detects an error (same as NUI_STOP).
NUI_OK	The DCS or gateway completion was normal.
NUI_STOP	The DCS or gateway wants to terminate (same as 999 code).
NUI_DEAD	The associated terminal has not been 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 DCS or gateway (same as 800 code).
NUI_BAD	A fatal error occurred, which prevents continued execution.
NUI_BOOT	The associated terminal is powered on and can be used to reload data into the terminal. Or, it can be ignored.

Program interfaces return the terminal number and the DCS, gateway, or terminal response. If one of these codes is not set, the field contains the number of the terminal.

Configuration

To use Native sockets, you must configure IP, RF communication, and terminal parameters for the DCS, gateway, and terminal. To configure these parameters, refer to the device's user manual for instructions.

When everything is configured correctly and the terminal is powered on, a unique connection (socket) is established between the terminal and your sockets-based Native emulation program.

Commands to the terminals must be in the appropriate syntax for their Native mode (see Chapter 6, "*Terminal Commands*"). The parameter and diagnostic commands in Chapter 5, "*Data Collection Server and Gateway Commands*," should be ignored.

The following WtD/ (Write Display) command is a sample startup menu for a terminal 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 terminal number. The communication driver in the DCS or gateway adds the "W" and the terminal 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 terminal via the DCS or gateway. The program sends key presses made by a terminal 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(”\nnui_sock: Wrong number of arguments.”);
        printf(”\nnui_sock: Usage: nui_sock <portno>\n\n”);
        fflush(stdout);
        exit(1);
        }
/*
**  get the port number
*/
    strcpy(work_buf,argv[1]);
    if(strlen(work_buf) < 1)
        {
        printf(”\nnui_sock: Invalid portno! Must be numeric and > 0.”);
        printf(”\nnui_sock: Usage: nui_sock <portno>\n\n”);
        fflush(stdout);
        exit(1);
        }
    ipport=atoi(work_buf);
/*
** set up curses
*/
    for(x=0;x<53;x++)
        {
        printf(”\n”);
        }
    fflush(stdout);
    printf(”\nnui_sock: Native Socket Demo - Version %s ”,VERSION);
    fflush(stdout);
/*
**  startup sockets
*/
#ifdef WIN32
    x = WSAStartup (0x0101, &WsaData);
    if(x==SOCKET_ERROR)
        {
```

```
        printf(”\nnui_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(”\nnui_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(”\nnui_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(”\nnui_sock: Listen on Port %d\n”,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(”\nnui_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(”\nnui_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(”\nnui_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(”\nnui_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(”\nnui_sock: Terminal requested stop.”);
    fflush(stdout);
    return(0);
    }
  if(tdata[0]==’K’ && tdata[1]==’X’)
    {
        /* Stop this application and reset all connected terminals */
        printf(”\nnui_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(”\nnui_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(”\nnui_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

The General Index covers all topics. Those in italics are figures, those in bold are tables.

The Files Index is to assist you in locating descriptions for device drivers, applications, utilities, batch files, or other files within this publication.

General Index

Files Index

Corporate Headquarters
6001 36th Avenue West
Everett, Washington 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

Native Terminal Emulation Programmer's Guide - October 2003

P/N 977-055-006 REV D