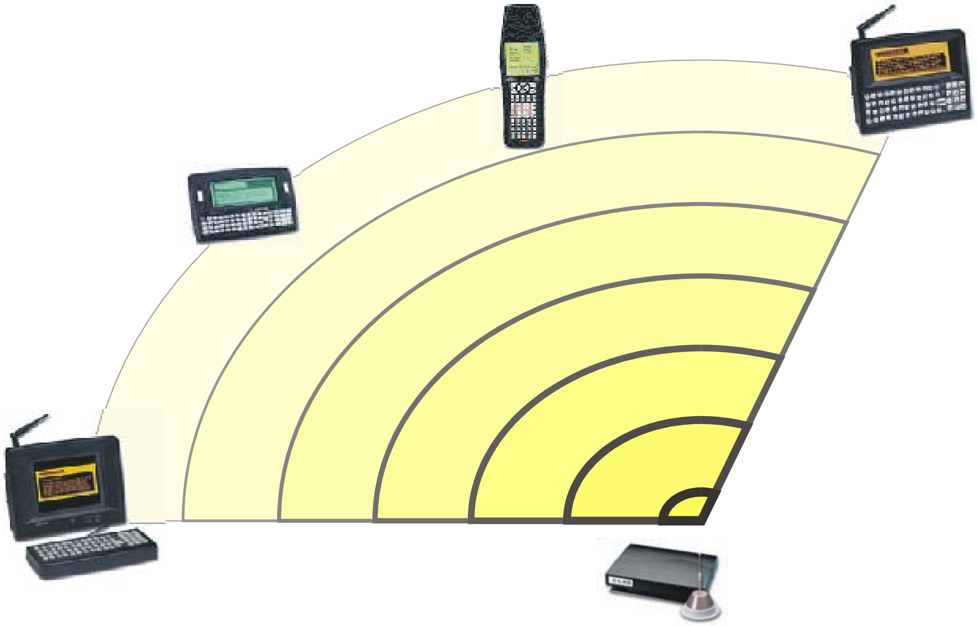
ANSI Plus Reference Guide



An EMS Technologies Company

ANSIPA136PRGREF
E-SW-ANSIPLUSRG-J

Notices

ANSI Plus is terminal emulation software developed by LXE. The software is installed in computer equipment. Any reference, whether direct or implied, to any LXE RF equipment requires the reader to refer to the specific RF equipment's User Manuals for cautions, warnings and federal notices (e.g. FCC, EMC, UL, CE, etc.).

Copyright Notice:

This manual is copyrighted. All rights are reserved. This document may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without prior consent, in writing, from LXE Inc.

Copyright © February, 2003 by LXE Inc. An EMS Technologies Company.
125 Technology Parkway, Norcross, GA 30092 U.S.A. (770) 447-4224

Trademarks:

LXE® and ANSI Plus are registered trademarks of LXE Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations. When this manual is in PDF format: "Acrobat ® Reader Copyright © 1987-2001 Adobe Systems Incorporated. All rights reserved. Adobe, the Adobe logo, Acrobat, and the Acrobat logo are trademarks of Adobe Systems Incorporated." applies.

Notice:

LXE Inc. reserves the right to make improvements or changes in the *software products* described in this manual at any time without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, LXE assumes no liability resulting from any errors or omissions in this document, or from the use of the information contained herein. Further, LXE Incorporated, reserves the right to revise this publication and to make changes to it from time to time without any obligation to notify any person or organization of such revision or changes.

Revision Notice

ANSI Plus Reference Guide

Upgrade From Revision H to Revision J

Section	**Explanation**
Notices	Update Notices, Copyrights and Trademarks.
Entire Manual	Delete references to the 2330. The 2330 is obsolete.
Chapter 1 – Introduction	Add VX4D Reference Guide to "ANSI Plus Related Manuals" section. Include information previously included in Chapter 2 – System Overview.
Chapter 2 – System Overview	Delete chapter. This information is now included in Chapter 1 – Introduction.

Chapter 3 – Getting Stated	Rename as Chapter 2 –Getting Started. Add note to “Autologin with a 2.4GHz Radio” section about multiple Autologin hosts. Add “Default Function Keys” and Exiting ANSI Plus” sections. Add local echo keys to “Hot Key Operation”. Revise “Starting ANSI Plus”, “Modifying the TE Selection Menu” and “ANSI Plus Related Manuals” sections.
Chapter 4 – Configuration Utility	Rename as Chapter 3 – Configuration Utility. Add parameters: TermType_A, AutoLoginOn_A, Host_A, Prompt1_A, Reply1_A, Prompt2_A, Reply2_A, Prompt3_A, Reply3_A, PortNum_A TermType_B, AutoLoginOn_B, Host_B, Prompt1_B, Reply1_B, Prompt2_B, Reply2_B, Prompt3_B, Reply3_B, PortNum_B, TermType_C, AutoLoginOn_C, Host_C, Prompt1_C, Reply1_C, Prompt2_C, Reply2_C, Prompt3_C, Reply3_C, PortNum_C, TermType_D, AutoLoginOn_D, Host_D, Prompt1_D, Reply1_D, Prompt2_D, Reply2_D, Prompt3_D, Reply3_D, PortNum_D, VT320NavKeys, Com1Power, Com2Power, CodeID, BCSymbologyID_1, Sym_1PreStrip, Sym_1PostStrip, BCSymbologyID_2, Sym_2PreStrip, Sym_2PostStrip, BCSymbologyID_3, Sym_3PreStrip, Sym_3PostStrip, BCSymbologyID_4, Sym_4PreStrip, Sym_4PostStrip, BCSymbologyID_5, Sym_5PreStrip, Sym_5PostStrip, BCSymbologyID_6, Sym_6PreStrip, Sym_6PostStrip, FNKeySize. Delete parameters: TermType, AutoLoginOn, Host, Prompt1, Reply1, Prompt2, Reply2, Prompt3, Reply3, PortNum, ViewingAngle, BacklightMode, BacklightTimer, NumPhysRows, NumPhysCols. Revise parameters: OperatingMode, CommModeCfg, InputFName, BigBufferCount, LockRow, KeyboardType, OutInDelay, ScreenSize, NormalIO, RFLogFS, Com1LogFS, Com2LogFS, SafeKeys.
Chapter 5 – Terminal Overview	Rename as Chapter 4 – Terminal Overview. Remove “Function Key Defaults” table. This information is now included in Appendix E, “Function Key Definitions”. Add “Local Echo” section to “Character Mode Communication”.
Chapter 6 – Control Codes	Rename as Chapter 5 – Control Codes.
Chapter 7 – ANSI Escape Sequences	Rename was Chapter 6 – ANSI Escape Sequences. Update parameters for DAQ to include 15, keyboard input only.
Chapter 8 – DEC Private Escape Sequences.	Rename as Chapter 7 – DEC Private Escape Sequences.
Chapter 9 – LXE Private Escape Sequences	Rename as Chapter 8 – LXE Private Escape Sequences.
Appendix C – Quick Reference Guide	Update parameters for DAQ to include 15, keyboard input only.
Appendix D – LXE 2325 and ANSI Plus	Replace with Appendix D – ANSI Plus Function Key Definitions. The information previously contained in this appendix has been incorporated into Chapter 3 – Configuration Utility.
Appendix E – Key Maps	Add VX4 Key Maps.

LXE®

An EMS Technologies Company

Table of Contents

CHAPTER 4 TERMINAL OVERVIEW 4-1

CHAPTER 8 LXE PRIVATE ESCAPE SEQUENCES 8-1

APPENDIX A MODE DEFINITIONS A-1

APPENDIX B ANSI PLUS MESSAGES B-1

APPENDIX C QUICK REFERENCE GUIDE C-1

APPENDIX D ANSI PLUS FUNCTION KEY DEFINITIONS D-1

APPENDIX E KEY MAPS E-1

Illustrations

Table of Contents

CHAPTER 1 INTRODUCTION 1-1

Illustrations

An EMS Technologies Company

Chapter 1
Introduction

Overview

The ANSI Plus Reference Guide provides a description of the use and operation of LXE's DOS based ANSI Terminal Emulator (TE) software. This manual includes installation instructions, user guidance, as well as information to help in the development of host applications that communicate with LXE devices.

This guide also includes the control codes, control sequences, and private control sequences applicable to LXE devices. Additionally, this guide provides information on LXE private remote setup commands.

As the reference for LXE's ANSI Plus TE, this guide provides detailed information on its features and functionality. Use this reference guide as you would any other source book: reading portions to learn about ANSI Plus, and then referring to it when you need more information about a particular subject. This guide takes you through all aspects of the installation and configuration of the LXE ANSI Plus TE.

System Requirements

ANSI Plus will run on LXE certified DOS devices. Minimum requirements are:

- 640 KB of RAM
- 700 KB of free drive space
- Standard VGA display
- DOS 6.2 or greater

Installation

ANSI Plus comes pre-installed on LXE certified DOS devices. Before using ANSI Plus, it is recommended that a software backup be performed.

IMPORTANT The ANSI Plus drivers must be loaded prior to starting ANSI Plus.

The following lines must appear in your AUTOEXEC.BAT file before running ANSI Plus:

```
[drive:\path]\radio_driver_name
[drive:\path\keyboard_driver_name]
[drive:\path]\plus_all_other_files_for_device
```

where:

[drive:\path] - is replaced with the drive and path location of the ANSI Plus files (Typically C:\APLUS).

[drive:\path\radio_driver_name] - is the radio driver for the installed radio in a specific LXE DOS device.

[drive:\path\keyboard_driver_name] - is the keyboard driver for a specific LXE DOS device.

Radios and Keyboards

The radio driver filename is different for the 900 MHz radio and the 2.4 GHz radio. Please refer to the specific equipment's user manual or documentation.

Each LXE DOS device has a specific keyboard driver file. Please refer to the reference guide for your specific LXE DOS device for the appropriate keyboard driver name.

Re-Installation

To re-install the ANSI Plus software, simply restore the application from backup. Refer to the reference guide for the specific LXE DOS device for backup/restore procedures.

In This Guide

Chapter 1, “Introduction”, describes this reference guide’s structure and an overview of the ANSI Plus TE system. Also describes the function and layout of an LXE Radio Frequency Data Communication system and describes RF backbone compatibility with LDS Plus.

Chapter 2, “Getting Started”, takes you through installation, initial setup of the LDS Plus software and system, and logging in. Instructions are included for using LDS Plus key sequences, manipulating the Compose and Receive buffers and working with forms.

Chapter 3, “Configuration Utility”, contains information and instructions relating to the configuration utility resident in each ANSI Plus device. Configuration parameter instructions are included in this chapter.

Chapter 4, “Terminal Overview”, provides details on the LXE ANSI Plus terminals and the operation of these terminals under ANSI Plus.

Chapter 5, “Control Codes”, contains information on control codes used to control or modify the LXE terminal and attached devices.

Chapter 6, “ANSI Escape Sequences”, lists the ANSI escape sequences that extend the functionality of the control codes.

Chapter 7, “DEC Private Escape Sequences”, discusses the DEC private escape sequences available for LXE equipment.

Chapter 8, “LXE Private Escape Sequences”, lists the LXE specific escape sequences and command strings.

Appendix A, “Mode Definitions”, details the differences in on certain escape sequences depending on the mode in which the LXE device is operating.

Appendix B, “ANSI Plus Messages”, provides information about general and fatal error messages ANSI Plus may display.

Appendix C, “Quick Reference Guide”, contains a summary of features available under ANSI Plus.

Appendix D, “ANSI Plus Function Key Definitions”, details the codes generated by an ANSI Plus function key keypress as compared to the VT100/VT220 standards.

Appendix E, “Key Maps”, describe the LDS specific keypresses to use when operating LXE DOS certified computers.

Document Conventions

This Reference Guide uses the following conventions:

ALL CAPS	All caps are used to represent disk directories, file names, and application names.
“Quotes”	Indicates the title of a chapter or a section within a chapter (for example, “Document Conventions”).
[Brackets]	Indicates a key on the DOS keyboard (for example, [Enter]).
📖	Indicates a reference to other documentation.
📡	Differences in ANSI Plus operation or commands due to type of radio.
Note:	Keyword that indicates immediately relevant information.
Caution: ⚠	Keyword and icon that indicates a cautionary warning to follow.
IMPORTANT	Keyword that indicates vital or pivotal information to follow.

Examples in this Guide

The examples provided throughout this guide are for illustration purposes. Do not use these examples for your operation because every operation is different. Additionally, the examples in this guide are only fragments of actual C programming code and are not intended for implementation.

LXE Legacy Terminals vs. LXE DOS Computers

Devices used in the LXE Legacy ANSI system (LXE 1280, 1290, 2280, 2285, 2286 and 2315 terminals and the 5460 wireless modem) are frequently called “LXE RF terminals.”

To distinguish between the Legacy ANSI terminals and LXE certified DOS computers required for ANSI Plus (e.g. 13XX vehicle mount and 23XX hand held computers) this reference guide refers to ANSI Plus devices as DOS terminals *or* DOS computers.

Getting Help

All LXE manuals are now available on one CD and they can also be viewed/downloaded from the LXE website. Contact your LXE representative to obtain the LXE Manuals CD (Product No. 9000A426LXEMANUALS).

You can also get help from LXE by calling the telephone numbers listed on the LXE Manuals CD, in the file titled "Contacting LXE". This information is also available on the LXE website www.lxe.com.

ANSI Plus Related Manuals

The following lists the manuals that are available on the LXE Documentation CD-ROM (9000A426LXEMANAUALS). These manuals provide references that may be required when using ANSI Plus with LXE certified DOS computers.

These manuals provide setup, operating instructions, software / accessory installation instructions:

- 1380 Reference Guide
- 1390 Reference Guide
- 2325 Reference Guide
- MX1 Reference Guide
- MX2 Reference Guide
- MX3 Reference Guide
- VX1 Reference Guide
- VX2 Reference Guide
- VX4D Reference Guide
- 6224 Session Manager Reference Guide
- DOS Autoconfigurator Instructions

The following manuals describe how to use SNMP to configure, monitor and update LXE computers with 2.4GHz radios:

- Client Configuration Manager Reference Guide
- SNMP Agent Reference Guide

The following manual describes how to use the network management workstation (NMW) to configure, monitor, and diagnose the LXE 6200 system:

- 6200 Network Management Guide

System Overview

This chapter provides a broad overview of the LXE ANSI Plus environment. The devices and their relationship to one another in an RF data communication system are presented.

Radio Frequency Data Communication

The following figure illustrates both a conventional, hardwired group of terminals and an LXE Radio Frequency Data Communication (RFDC) system showing both 900MHz and 2.4GHz. The restricted mobility of the hardwired terminals limit their effectiveness in situations where terminal operators must roam freely, as in a warehouse application. In contrast, terminals in the RFDC system use radios to link to a remote host, and are free to move wherever real-time data communication is required.

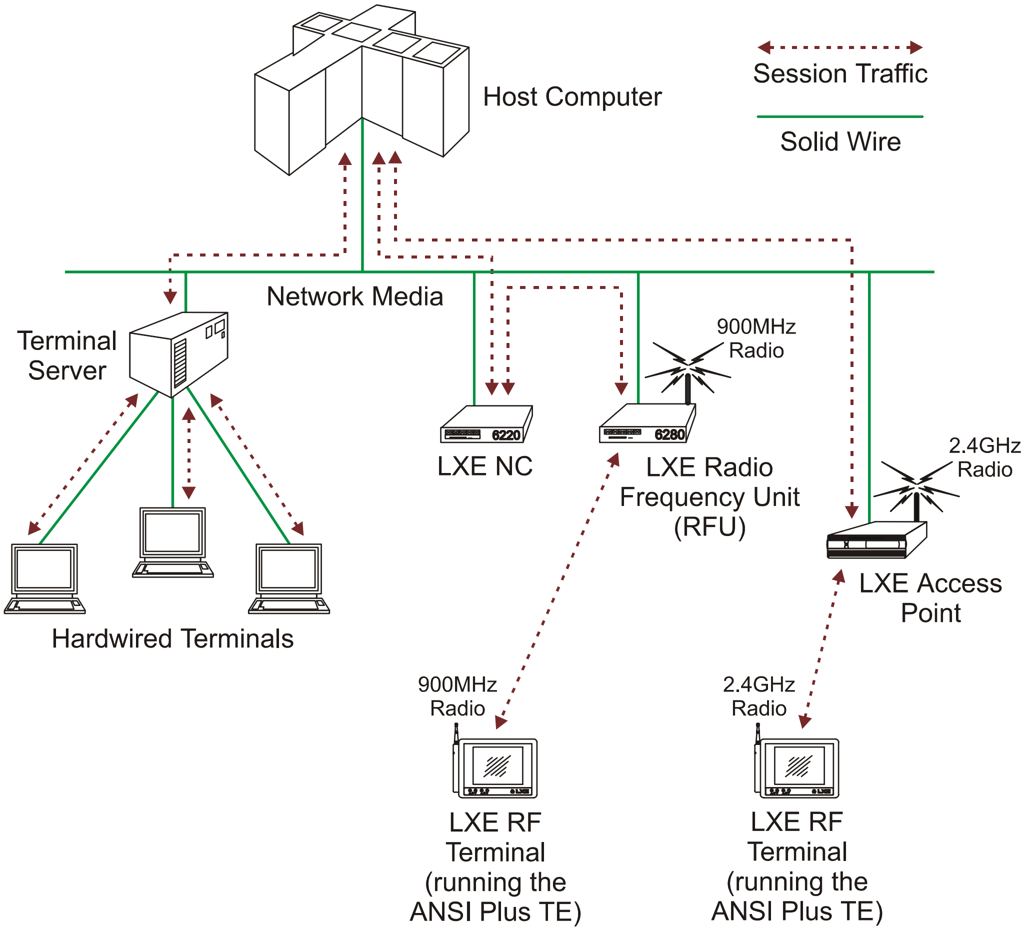


Figure 1-1 Hardwired versus Wireless Architecture

System Components

In a 900MHz System

The following diagram shows a basic 900MHz LXE-based radio frequency data communication system with the DOS terminals running ANSI Plus:

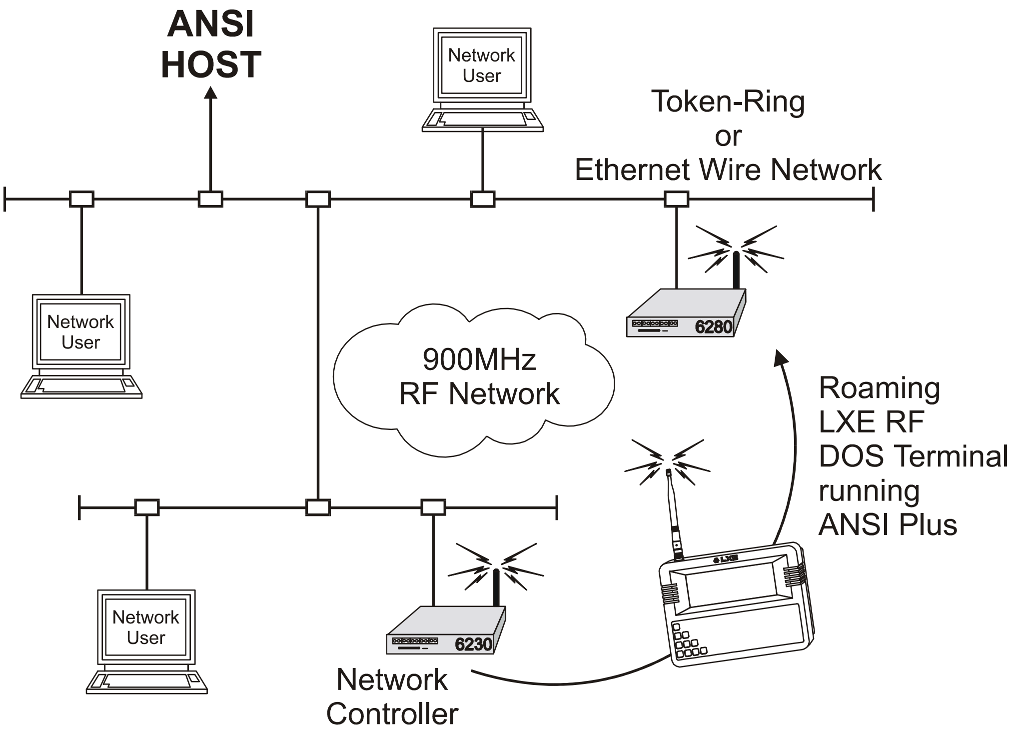


Figure 1-2 Basic LXE 900MHz Radio Frequency Data Communication System

In a 2.4GHz System

The following diagram shows a basic 2.4GHz LXE-based radio frequency data communication system with the DOS terminals running ANSI Plus:

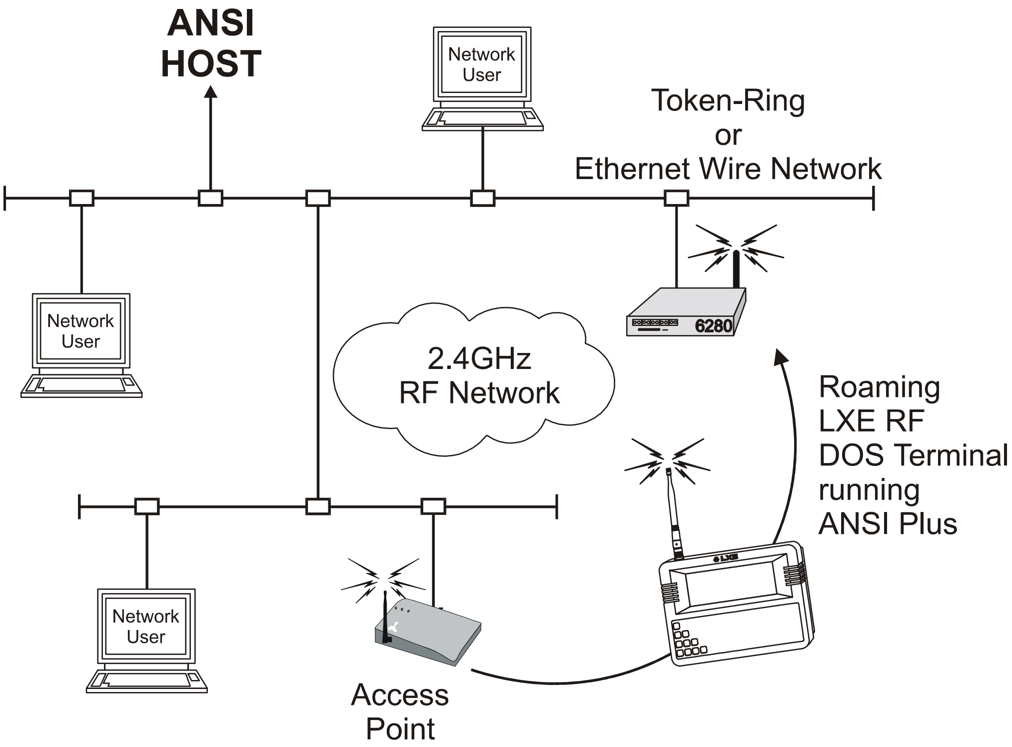


Figure 1-3 Basic LXE 2.4GHz Radio Frequency Data Communication System

RF Data Communication System Component Functions

ANSI Host Computer	A computer system that runs application programs and maintains databases.
Access Point (AP)	An LXE device that monitors and manages message flow between the host computer and the DOS RF ANSI Plus terminal using a 2.4GHz radio.
Network Controller (NC)	An LXE device that monitors and manages message flow between the host computer and the ANSI Plus RF terminals using a 900MHz radio.
Radio Frequency Unit (RFU)	An LXE device that links the NC to the ANSI Plus terminals using radio waves.
RF Terminal	An LXE device that an operator uses to communicate with the host computer. The RF terminal transmits data to and receives data from the RFU using radio waves.
Network Management Workstation (NMW)	A personal computer running LXE network management workstation software that enables the system manager to configure and monitor the RF system.

LXE®

An EMS Technologies Company

Table of Contents

CHAPTER 2 GETTING STARTED 2-1

Illustrations

An EMS Technologies Company

Chapter 2
Getting Started

Starting ANSI Plus

ANSI Plus is configured to start as follows:

- ANSI Plus is started automatically after the device boots if it is the only TE loaded on the computer and ANSI Plus is not configured for multiple hosts.
- If multiple Terminal Emulators (TE's) are installed, the TE Selection Menu is displayed after boot. To launch ANSI Plus, select the appropriate choice from the menu.
- If ANSI Plus is configured for multiple hosts, the TE Selection Menu is displayed after boot. To launch ANSI Plus, select the appropriate choice for the desired host from the menu.

ANSI Plus can be is started manually by returning to the TE Selection Menu with the following command at a DOS prompt:

```
c:\teselect
```

For 2.4GHz radio equipped computers:

- ANSI Plus is started automatically if it is the only TE installed (and ANSI Plus is not configured for multiple hosts).
- The TE Selection Menu is displayed if multiple TE's are installed or ANSI Plus is configured for multiple hosts. In either case, select the appropriate choice from the TE Selection Menu.

For 900MHz radio equipped computers:

- The computer reboots and ANSI Plus is started automatically if it is the only TE installed.
- The computer reboots and the TE Selection Menu is displayed if multiple TE's are installed.

This is necessary because 900MHz computers can change TE's only at a reboot.

Single TE

Power on the DOS computer and the ANSI Plus welcome screen is automatically displayed. When the TE is closed with an <Alt-X> keypress, the screen is cleared and the system returns to the DOS prompt.

Note: This is the default behavior for an ANSI Plus equipped computer. The TE can be configured to suppress the DOS prompt. Please refer to "SuppressPrompt" in Chapter 3, "Configuration Utility" for more details.

The TE Selection Menu

Multiple TEs

In the event there is more than one Terminal Emulator (TE) installed in the computer, the user is presented with the TE Selection Menu listing the available terminal emulators that can be accessed using the computer.

For example, a user could see:

```
[5] Launch 5250
[A] Launch ANSIPLUS
[E] Exit to DOS

Please select option: [5, A, E]?
```

Upon selecting one of the terminal emulator options, the desired TE is launched. When the user selects E, the screen is cleared and the DOS prompt is displayed. When the TE is closed with an <Alt-X> keypress, the screen is cleared and the menu is presented again.

Note: *The system administrator may change almost every facet of the menu display, including suppressing the "Exit to DOS" option, by modifying the variables found at the top of TESELECT.BAT. Details on modifying the TESELECT.BAT file are included in the "DOS Autoconfigurator Instructions", included on the LXE Manuals CD. The TE can be configured to suppress the DOS prompt. Please refer to "SuppressPrompt" in Chapter 3, "Configuration Utility" for more details.*

Multiple Hosts

ANSI Plus has been modified to support the selection of one of four possible auto-login hosts at launch. Up to four distinct ANSI Plus host options may be displayed in the TE Selection Menu. Any of the four ANSI Plus options can be enabled or suppressed by modifying the TE Selection Menu. An example menu is shown below:

```
[5] Launch 5250
[A] Launch ANSIPLUS (Host 1)
[B] Launch ANSIPLUS (Host 2)
[C] Launch ANSIPLUS (Host 3)
[D] Launch ANSIPLUS (Host 4)
[E] Exit to DOS

Please select option:  [5, A, B, C, D, E]?
```

Please refer to the "DOS Autoconfigurator Instructions", included on the LXE Manuals CD for complete details on editing the TE Selection Menu to include multiple ANSI Plus hosts.

Note: *There is no automatic connection between setting up these options and actually having the auto-login set up in the LXE.INI file by either the CONFIG.EXE application or remotely via SNMP sets (by using Client Configuration Manager, for example).*

Modifying the TE Selection Menu

Please refer to the "DOS Autoconfigurator Instructions", included on the LXE Manuals CD for complete details on modifying the TE Selection Menu.

Exiting ANSI Plus

The behavior of the LXE computer when exiting ANSI Plus can be customized as follows:

- The “exit to DOS” option in the TE Selection Menu can be suppressed. For more information on suppressing the prompt, please refer to the “DOS Autoconfigurator Instructions”, included on the LXE Manuals CD.
- ANSI Plus contains a “SuppressPrompt” parameter, which controls the <Alt-X> (exit) action in the TE. For more information on this parameter, please refer to Chapter 3, “Configuration Utility”.

The exit behavior is also dependent on the radio installed in the LXE computer. Please refer to the charts on the following pages for details.

2.4GHz Radios

Please refer to the following charts to determine the TE exit behavior for 2.4GHz ANSI Plus LXE computers:

If TE Selection Menu “Exit to DOS” Option is not Suppressed:

TE “Suppress Prompt” OFF	TE “Suppress Prompt” On
Single TE: <**Alt-X**> from the TE takes the system to a DOS prompt. **Dual TE:** <**Alt-X**> from the TE takes the system to the TE Selection Menu with the “Exit to DOS” option.	<**Alt-X**> from the TE is disabled. <**Control-D**> exits the TE and presents the user with the message “Disconnected. Press Enter to Continue.” <Enter> returns the user to the TE. **Single TE:** <**Control-DG**> takes the system to a DOS prompt. **Dual TE:** <**Control-DG**> takes the system to the TE Selection Menu with the “Exit to DOS” option.

If TE Selection Menu “Exit to DOS” Option is Suppressed:

TE “Suppress Prompt” OFF	TE “Suppress Prompt” On
Single TE: <**Alt-X**> from the TE takes the system to a DOS prompt. **Dual TE:** <**Alt-X**> from the TE takes the system to the TE Selection Menu without the “Exit to DOS” option.	<**Alt-X**> from the TE is disabled. <**Control-D**> exits the TE and presents the user with the message “Disconnected. Press Enter to Continue.” <Enter> returns the user to the TE. **Single TE:** <**Control-DG**> takes the system to a DOS prompt. **Dual TE:** <**Control-DG**> takes the system to the TE Selection Menu without the “Exit to DOS” option.

900MHz Radio

Please refer to the following charts to determine the TE exit behavior for 900MHz ANSI Plus LXE computers:

If TE Selection Menu "Exit to DOS" Option is not Suppressed:

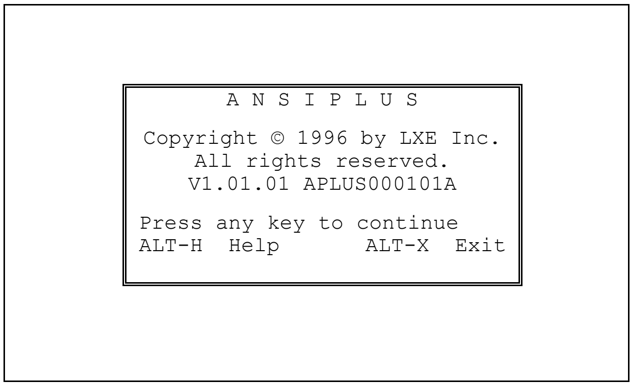
TE "Suppress Prompt" OFF	TE "Suppress Prompt" On
Single TE: <**Alt-X**> from the TE takes the system to a DOS prompt. **Dual TE:** <**Alt-X**> from the TE reboots the system to the TE Selection Menu with the "Exit to DOS" option.	<**Alt-X**> from the TE is disabled. <**Control-D**> exits the TE and presents the user with the message "Disconnected. Press Enter to Continue." <Enter> returns the user to the TE. **Single TE:** <**Control-DG**> takes the system to a DOS prompt. **Dual TE:** <**Control-DG**> reboots the system to the TE Selection Menu with the "Exit to DOS" option.

If TE Selection Menu "Exit to DOS" Option is Suppressed:

TE "Suppress Prompt" OFF	TE "Suppress Prompt" On
Single TE: <**Alt-X**> from the TE takes the system to a DOS prompt. **Dual TE:** <**Alt-X**> from the TE reboots the system to the TE Selection Menu without the "Exit to DOS" option.	<**Alt-X**> from the TE is disabled. <**Control-D**> exits the TE and presents the user with the message "Disconnected. Press Enter to Continue." <Enter> returns the user to the TE. **Single TE:** <**Control-DG**> takes the system to a DOS prompt. **Dual TE:** <**Control-DG**> reboots the system to the TE Selection Menu without the "Exit to DOS" option.

Title Screen

Once ANSI Plus has been started, the title screen, similar to the following, will appear:



```
        A N S I P L U S

 Copyright © 1996 by LXE Inc.
     All rights reserved.
    V1.01.01 APLUS000101A

Press any key to continue
ALT-H  Help       ALT-X  Exit
```

Figure 2-1 The ANSI Plus TE Title Screen

Note: The ANSI Plus title screen will automatically disappear if no key is pressed within a few seconds.

Once the ANSI Plus title screen has appeared, either wait until the title screen is cleared automatically or press any key to continue.

If your system is configured to suppress the TELNET prompt, the prompt will never be displayed. In that case, any reference to the TELNET prompt and commands in this reference guide do not apply.

If your system is NOT configured to suppress the TELNET prompt, simply press any key to get to the TELNET prompt screen.

Telnet Prompt

Once the ANSI Plus title screen has appeared, simply press any key to get to the TELNET prompt screen.

```
telnet>
```

Figure 2-2 ANSI Plus - TELNET Prompt

Telnet Prompt Suppressed

If your system is configured to suppress the Telnet prompt, the prompt will never be displayed. In that case, any reference to the Telnet prompt and commands in this reference guide do not apply.

Upon startup, the DOS terminal screen display may appear to be delayed while connection is made to the host.

ANSI Plus and the LXE 6224 Session Manager

If your system is equipped with an LXE 6224 Session Manager, the “open” command, from the DOS terminals, is to the IP address of the 6224, not to the desired host.

The 6224 will open a connection on behalf of the end user (DOS terminal operator). The 6224 will have been previously programmed with the necessary information to establish the connection with the correct host for each DOS terminal.

Exception Message Text

The amount and type of Exception Message text displayed is determined by setting the VerboseExcMsgs parameter to either Verbose or Terse.

Setting the parameter to `terse` displays only a brief explanation while `verbose` displays a more detailed description of the problem. For example, the following figure titled “ANSI Plus - Unable to Connect to NC” displays a terse message for a terminal with a 900MHz radio.

2.4GHz Radio Device

When the connecting device contains a 2.4GHz radio, the appearance of `telnet>` on the screen means you are now ready to enter the OPEN command to connect to the host.

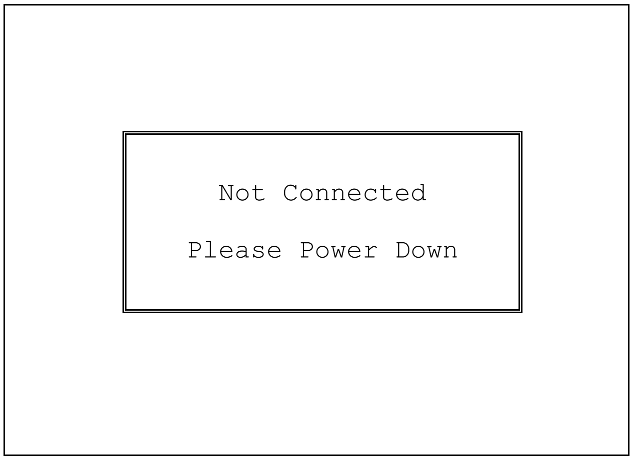
If a 2.4GHz radio device is unable to connect to a host, an error message is displayed. See the section titled “Unsuccessful Host Connection” later in this chapter.

900MHz Radio Device

When the connecting device (RFU) has a 900MHz radio, the appearance of `telnet>` on the screen means the RFU is connected to the LXE Network Controller. You are now ready to enter the OPEN command to connect to the host.

Note: There may be a short pause while ANSI Plus tries to contact the host.

If a 900MHz radio device is unable to connect to a Network Controller, the following screen will appear:



```
Not Connected

Please Power Down
```

Figure 2-3 ANSI Plus - Unable to Connect to NC

Opening a TELNET Session

The LXE ANSI Plus TE uses TELNET to communicate with the host computer. TELNET is a remote access terminal protocol that allows network users to log on and use a remote computer system as though they are directly connected to that system. The following procedure outlines opening a TELNET session with a host computer.

The Host Connection

Once the TELNET prompt appears, the terminal is in command mode. The terminal accepts and executes the commands while in command mode. In order to communicate with a TELNET host, you must open a connection between the terminal and the host. To open a connection:

1. Type the Open command and the Internet address (or name) of the host computer and press the [Enter] key.

   Example:
   ```
   telnet>open 186.25.18.100
   ```
   - or -
   ```
   telnet>open [hostname]
   ```

   Result: The host computer sends a log on and password screen to the terminal.

2. Type your log on and/or password at the host prompt and press the [Enter] key.

   Result: The TELNET protocol enters the input mode and the host prompt is displayed at the terminal.

3. Are you in the correct input mode?:

   If yes: proceed to step 5.

   If no: press [Ctrl]+[4] (the TELNET escape key sequence) to return the terminal to the TELNET prompt.

   Result: The TELNET prompt appears.

4. Type the Mode command at the TELNET prompt and press the [Enter] key. Mode choices are Character and Line. See “TELNET Commands” for a description of the Mode command.

   Example:
   ```
   telnet>mode line
   ```

   Result: The terminal is now in the Line mode.

5. To return to your host application when you are through communicating with TELNET, press the [Enter] key twice.

   Result: The terminal displays the host prompt.

TELNET Commands

A list of available TELNET commands may be displayed at the TELNET prompt. The following is a description of each TELNET command along with their syntax.

To display a list of TELNET commands:

1. Press [?] at the TELNET prompt.
2. Press [Enter].

Note: The following TELNET commands can be activated by typing enough of the command to separate it from other like commands:

Command	Abbreviation
Close	cl
Display	d
Mode	m
Open	o
Send	sen
Set	set
Status	st
Toggle	t

?

TELNET command help. With no arguments, TELNET prints a help summary. If you specify a command with **?**, TELNET prints the help information for that command, e.g. `? set echo`. TELNET host connection is not required.

```
Syntax: ? [command]
```

Close

Closes a TELNET session and returns the terminal to command mode. TELNET host connection is required.

```
Syntax: close
```

Display

Displays `set` and `toggle` values. TELNET host connection is not required.

```
Syntax: display [argument]
```

Please set “Set” and “Toggle” commands later in this section.

Mode

Enables you to specify Line or Character mode. Type `Mode` at the TELNET prompt, then type the desired mode (`Line` or `Character`). If the host is capable of entering that mode, Line or Character mode is enabled. Line mode uses the RF link more effectively than Character mode. TELNET host connection is not required.

```
Syntax: mode line
        mode character
```

Open

Opens a connection to the host that you specify. The host is specified by its Internet address that you enter using dot notation or by the host name in the lookup table.

```
Syntax:open [internet address]
       open [host name]
```

Send

Sends a special character sequence to the host. TELNET host connection is required.

```
Syntax:send [arguments]
```

The following arguments may be specified.

Argument	Function
escape	Sends the current TELNET Escape character.
synch	Sends the TELNET Synch sequence. This sequence causes the host to discard all previously typed (but not yet read) input. This sequence is sent as TCP urgent data. (It may not work if the host is a 4.2 BSD system. If synch doesn't work, lower case “r” may be echoed on the terminal.)
brk	Sends the TELNET Break sequence to the host.
ip	Sends the TELNET Interrupt Process sequence, which causes the host to abort the currently running process.
ao	Sends the TELNET Abort Output sequence, which causes the host to flush all output from the host to the user's terminal.
ayt	Sends the TELNET Are You There sequence to the host.
ec	Sends the TELNET Erase Character sequence, which causes the host to erase the last character entered.
el	Sends the TELNET Erase Line sequence, which causes the host to erase the line currently being entered.
ga	Sends the TELNET Go Ahead sequence to the host.
nop	Sends the TELNET No Operation sequence.
?	Prints out help information for the send command.

Set

Sets any one of a number of TELNET variables to a specific value. The special value *off* turns off the function associated with the variable. The values of variables may be interrogated with the `Display` command, e.g. `display echo`. TELNET host connection is not required.

`Syntax: set [argument]`

The arguments which may be specified are:

Argument	**Function**
echo	When in Line mode, this variable toggles between implementing local echo of entered characters (for normal processing) and suppressing the echo of entered characters (for privacy).
escape	Causes entry into TELNET command mode.
interrupt	If you type the interrupt variable while TELNET is in localchars mode, a TELNET IP sequence (send ip) is sent to the host. The initial value for the interrupt variable is interpreted as the terminal's interrupt character.
quit	If you type the quit variable and TELNET is in localchars mode, a TELNET BRK sequence (send brk) is sent to the host. The initial value for the quit variable is interpreted as the terminal's quit character.
erase	If you type the erase variable while TELNET is in localchars mode and if TELNET is operating in Character mode, a TELNET EC sequence (send ec) is sent to the host. The initial value for the erase variable is interpreted as the terminal's erase character.
kill	If you type the kill variable while TELNET is in localchars mode and if TELNET is operating in Character mode, a TELNET EL sequence (send el) is sent to the host. The initial value for the kill variable is interpreted as the terminal's kill character.

Status

Shows the current status of TELNET. This status includes the peer to which the terminal is connected, the TELNET escape character, and the current input mode. TELNET host connection is not required.

`Syntax: status`

Toggle

Toggles various flags that control how TELNET responds between true and false. TELNET host connection is not required. The values of variables may be interrogated with the `Display` command, e.g. `display toggle(localchars)`.

`Syntax: toggle [arguments]`

Valid arguments are:

Argument	**Function**
Valid Toggle arguments for 2.4GHz and 900MHz radios:	
localchars	If True, then the interrupt, quit, erase, and kill characters are recognized locally and transformed into appropriate TELNET control sequences (ip, brk, ec, and el; see Send). The initial value for this toggle is True in Line mode and False in Character mode.
crmod	Toggle carriage return mode. When this mode is enabled, most carriage return characters received from the host are mapped into a carriage return followed by a line feed. This mode does not affect those characters typed by the user, only those received from the host. This mode is not very useful unless the remote host only sends carriage return but never line feed. The initial value for this toggle is False.
?	Displays the legal toggle commands.
Valid Toggle arguments for 900MHz Radio Only:	
autoflush	If autoflush and localchars are both True, then when the intr or quit character is recognized (and transformed into appropriate TELNET control sequences detailed under Set above), TELNET refuses to display any data on the user's terminal until the remote system acknowledges (via a TELNET Timing Mark option) that it has processed those TELNET sequences. The initial value for this toggle is True.
autosynch	If autosynch and localchars are both True, then when either the intr or quit characters are typed, the TELNET sequence sent is followed by the TELNET Synch sequence. This procedure should cause the remote system to begin throwing away all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is False.
options	Toggles the display of some internal TELNET protocol processing (having to do with TELNET options). The initial value for this toggle is False.
netdata	Toggles the display of all network traffic (in hexadecimal format). The initial value for this toggle is False.
ttydata	Toggles the display of all terminal traffic (in hexadecimal format). The initial value for this toggle is False.

Unsuccessful Host Connection

If the terminal cannot connect with the host (or the address is invalid), error messages are displayed and terminal control returns to the terminal.

Note: There may be a short pause while ANSI Plus tries to contact the host.

Using 900MHz Radio

When the terminal is unable to connect to the Network Controller, a message appears and the terminal will need to be restarted. The terminal does not need to be powered off.

When the terminal has connected to the Network Controller, but unable to connect to the host after the OPEN command, an error message appears and another TELNET prompt is displayed.

Using 2.4GHz Radio

When the terminal is unable to connect to the host after an OPEN command, an error message is displayed on the screen explaining the inability to connect.

For Example:

Opening with a specific host name

```
telnet > open my_host
Resolving hostname ...
Couldn’t resolve host my_host
telnet >
```

Opening with a specific IP address

```
telnet > open 120.120.120.120
Resolving .....
Couldn’t resolve. FTP Error #999
telnet >
```

Refer to the FTP manual for an explanation of error messages.

Entering Messages

The LXE terminal accepts data entry from the keyboard, bar code scanner, and the auxiliary (RS-232) input port.

Keyboard Data Entry

Once ANSI Plus is started, you can enter data with the terminal keyboard. You could enter keyboard data into a stored form and transmit it to the host. You might respond to a prompt sent by the host application with a keyboard entry, such as a menu listing choices for your next action.

Bar Code Data Entry

The LXE terminal supports an accessory bar code reading device for reading preprinted labels. You can mix keyboard data entries with bar code data entries. Any scanner that decodes the bar code internally and outputs an RS-232 data stream may be used. The serial port parameters may need to be changed (using the configuration utility) to match the parameters of the scanner. See the hardware manual for more information on which COM ports are available for use with a scanner.

RS-232 Data Entry

The terminal accepts input from an RS-232 device connected to the RS-232 port of the terminal. The terminal processes data from the RS-232 port the same way it processes bar code data. The data is entered at the cursor position, and the data is subject to all of the bar code/RS-232 input menu parameters, such as truncate. You must activate the RS-232 input device before you can send data to the RS-232 port.

Hot Key Operation

ANSI Plus has several hot key sequences available. A “Hot Key” sequence is a combination of keystrokes that perform a specific action or function. The following is a list of the ANSI Plus hot keys available.

Hot Key	**Key Sequence**	**Reference**
Compose Key	[Alt]+[C]	“Compose Key” section of Chapter 4
Enable Local Echo	[Alt]+[E]	“Character Mode Communication” section of Chapter 4
CLR - clear field	[Alt]+[F]	“Scroll Mode Communication” section of Chapter 4
Disable Local Echo	[Alt]+[G]	“Character Mode Communication” section of Chapter 4
Hot Key Help screen	[Alt]+[H]	“Hot Key Help Screen” found in this section
CLS - clear screen	[Alt]+[L]	“Scroll Mode Communication” section of Chapter 4
INQ - end to end communications check *900MHz Radio Only*	[Alt]+[I]	“INQ” found in this section
Function key editor	[Alt]+[K]	“Function Key Editor” found in this section
Stored forms	[Alt]+[M]	“Stored Forms” found in this section
Unlock keyboard	[Alt]+[U]	“Unlock Keyboard” found in this section
Exit ANSI Plus	[Alt]+[X]	“Exiting ANSI Plus” found in this section
Answerback	[Ctrl]+[Enter]	“Answerback” found in this section
Window up	[Ctrl]+[UpArrow]	“Moving the Display Window” section of Chapter 4
Window down	[Ctrl]+[DownArrow]	“Moving the Display Window” section of Chapter 4
Window left	[Ctrl]+[LeftArrow]	“Moving the Display Window” section of Chapter 4
Window right	[Ctrl]+[RightArrow]	“Moving the Display Window” section of Chapter 4
Swap Network Controller *900MHz Radio Only*	[Ctrl]+[1]	“Switching to a Backup NC” found in this section

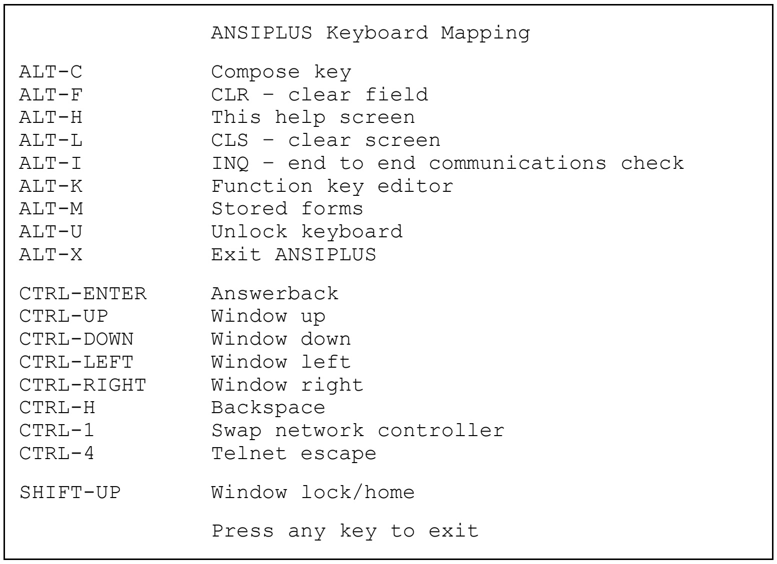
Hot Key	Key Sequence	Reference
Telnet escape	[Ctrl]+[4]	“TELNET Escape” found in this section
Window lock/home	[Shift]+[UpArrow] 2325 only: [Alt]+[UpArrow]	“Moving the Display Window” section of Chapter 4

Hot Key Help Screen

The hot key help screen displays a list of all the hot key sequences available from within ANSI Plus.

Note: *Help screens displayed on devices with screen displays with less than 20 columns may have more than one help screen. Nonetheless, all hot key descriptions are available for viewing.*

To display the hot key help screen press [Alt]+[H].



```
                ANSIPLUS Keyboard Mapping

ALT-C           Compose key
ALT-F           CLR - clear field
ALT-H           This help screen
ALT-L           CLS - clear screen
ALT-I           INQ - end to end communications check
ALT-K           Function key editor
ALT-M           Stored forms
ALT-U           Unlock keyboard
ALT-X           Exit ANSIPLUS

CTRL-ENTER      Answerback
CTRL-UP         Window up
CTRL-DOWN       Window down
CTRL-LEFT       Window left
CTRL-RIGHT      Window right
CTRL-H          Backspace
CTRL-1          Swap network controller
CTRL-4          Telnet escape

SHIFT-UP        Window lock/home

                Press any key to exit
```

Figure 2-4 ANSI Plus - Hot Key Help Screen

To exit the Help Screen, follow the instructions on the screen -- for example, [PageDown] for the next help screen, [PageUp] for the previous help screen. Press any other key to exit.

INQ

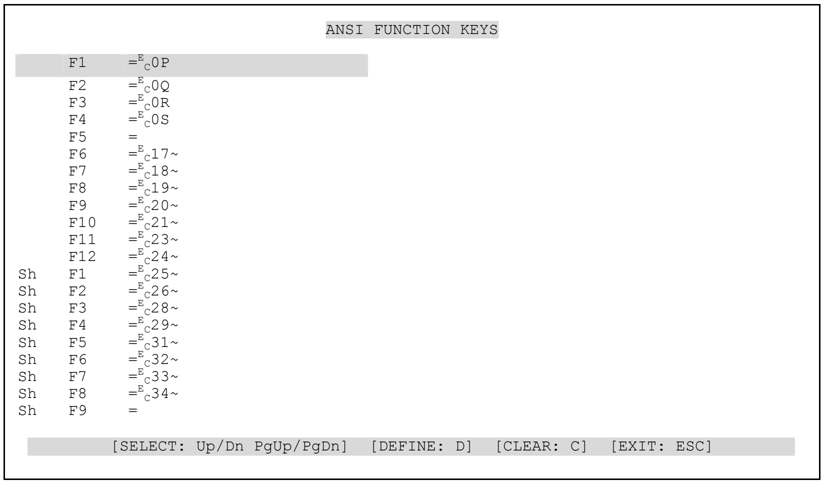
Valid hotkey for 900MHz radio devices only.

The INQ hot key ([Alt]+[I]) performs an end to end RF communications check between the terminal and the Network Controller. INQ determines whether the RF communication link between the terminal and Network Controller is active. If the RF communications test fails, a warning box will be displayed on the screen.

Function Key Editor

The ANSI Plus function key editor allows assigning multiple keystrokes to a single function key. The maximum number of keystrokes that can be assigned to a single function key is 32.

Note: All screens displayed on devices with displays smaller than approximately 7 inches by 6 inches may be required to display more than one screen per activity. Nonetheless, all information is available for viewing/or editing. Follow the on-screen (e.g [PageDown], [PageUp], [DownArrow], [UpArrow], [LeftArrow], [RightArrow]) instructions to access extra screens.



```
                              ANSI FUNCTION KEYS

      F1     =EC0P
      F2     =EC0Q
      F3     =EC0R
      F4     =EC0S
      F5     =
      F6     =EC17~
      F7     =EC18~
      F8     =EC19~
      F9     =EC20~
      F10    =EC21~
      F11    =EC23~
      F12    =EC24~
Sh    F1     =EC25~
Sh    F2     =EC26~
Sh    F3     =EC28~
Sh    F4     =EC29~
Sh    F5     =EC31~
Sh    F6     =EC32~
Sh    F7     =EC33~
Sh    F8     =EC34~
Sh    F9     =

          [SELECT: Up/Dn PgUp/PgDn]  [DEFINE: D]  [CLEAR: C]  [EXIT: ESC]
```

Figure 2-5 ANSI Plus - Function Key Editor

There are 48 function keys to choose from:

Available Function Keys

[F1]	through	[F12]
[Shift]+[F1]	through	[Shift]+[F12]
[Alt]+[F1]	through	[Alt]+[F12]
[Ctrl]+[F1]	through	[Ctrl]+[F12]

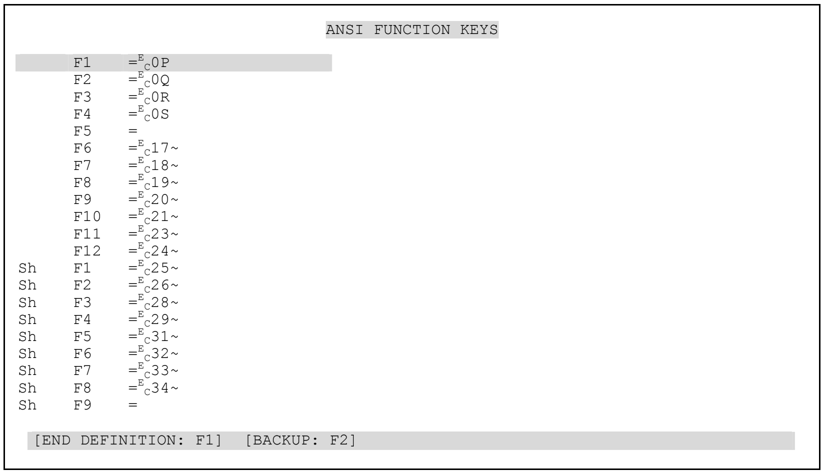
Default Function Keys

The default ANSI Plus Function Key code generations are included in Appendix D, “ANSI Plus Function Key Definitions”.

Assigning a Function Key

To assign multiple keystrokes to a function key, follow the procedure below.

1. Press [Alt]+[K] to access the function key editor.
2. Use the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys to highlight the desired function key.
3. Press [D] to define the function key.



```
                         ANSI FUNCTION KEYS

      F1    =␛0P
      F2    =␛0Q
      F3    =␛0R
      F4    =␛0S
      F5    =
      F6    =␛17~
      F7    =␛18~
      F8    =␛19~
      F9    =␛20~
      F10   =␛21~
      F11   =␛23~
      F12   =␛24~
Sh    F1    =␛25~
Sh    F2    =␛26~
Sh    F3    =␛28~
Sh    F4    =␛29~
Sh    F5    =␛31~
Sh    F6    =␛32~
Sh    F7    =␛33~
Sh    F8    =␛34~
Sh    F9    =

 [END DEFINITION: F1]  [BACKUP: F2]
```

Figure 2-6 ANSI Plus - Function Key Editor, Edit Mode

4. Type the keys you wish to assign to the function key.

   For example: to assign the number ten to a function key, type `10` .

   If an incorrect keystroke is assigned, simply press [F2] to backspace and erase the previous keystroke.
5. Press [F1] to exit edit mode when the assignment is complete.
6. Press [Esc] to exit the Function Key Editor.

Please refer to Appendix D, “ANSI Plus Function Key Definitions” for more information on ANSI Plus function keys.

Deleting Assigned Keystrokes

To delete the keystrokes assigned to a function key, follow the procedure below.

1. Press [Alt]+[K] to access the Function Key Editor.
2. Use the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys to highlight the desired function key.
3. Press [C] to clear the assigned keystrokes.
4. Press [Esc] to exit the Function Key Editor.

Exiting the Function Key Editor

To exit the Function Key Editor:

1. Press [Esc] to exit.

Stored Forms

A stored form is a combination of valid ANSI/DEC/LXE commands and text that, when executed by the terminal emulation and displayed on the screen, appears as a fill-in form. The commands and text that comprise the form are stored in a directory (e.g. C:\APLUS\STFORMS) on the DOS terminal's hard drive. The directory name and path are user configurable (refer to Chapter 3, "Configuration Utility", for more information).

Stored forms are written by the host application programmer and sent from the host to the DOS terminal.

Maximum Number of Stored Forms

The maximum number of stored forms supported by the ANSI Plus TE program is 100 and the maximum size of each stored form is 2K. Stored forms are placed in available internal Flash space in LXE certified DOS terminals.

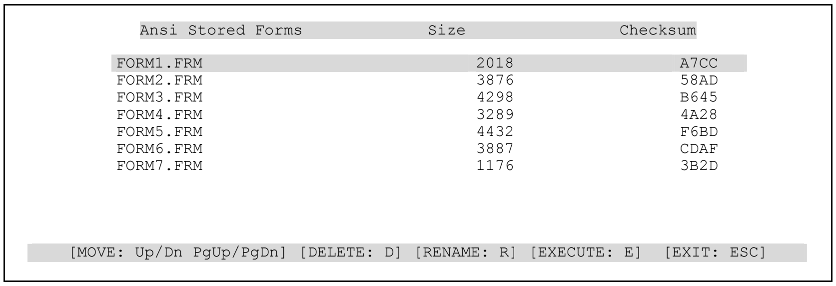
When more space is required for stored forms than is available in the terminal, the stored forms can be placed on an SRAM card. The stored forms directory location must be specified in the Configuration Utility in the StoredFormsDir parameter. The SRAM card is then inserted in the terminal and the forms are selected from the SRAM card's drive (e.g. E drive).

SRAM Cards

SRAM cards are commercially available. LXE has available SRAM or Flash PCMCIA Type II PC Cards (Various Sizes), Solid State Disk and ATA PCMCIA Type II or III Drive (Various Sizes). Each terminal has a limit to the size of SRAM card the terminal will accept. Please refer to the Technical Specifications section in the terminal's reference guides for this information. For example, at a minimum, most LXE certified DOS terminals accept 1Meg, 4Meg and 20Meg cards.

Procedure

The hot key sequence [Alt]+[M] enables you to delete, rename or execute a stored form from an alphabetical directory of up to 100 stored forms. You can scroll through the directory using the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys on the terminal.



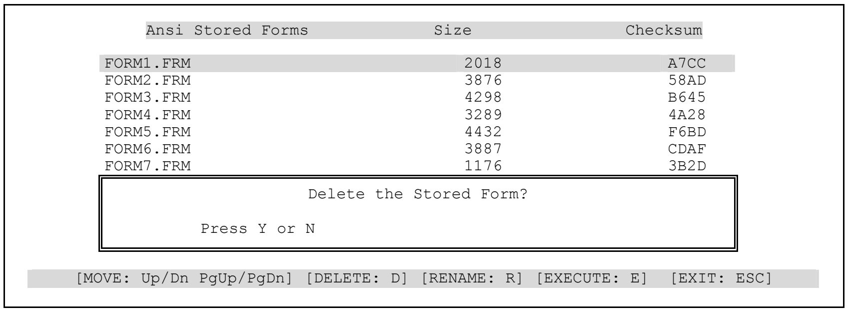
```
      Ansi Stored Forms                  Size                 Checksum

   FORM1.FRM                                 2018                   A7CC
   FORM2.FRM                                 3876                   58AD
   FORM3.FRM                                 4298                   B645
   FORM4.FRM                                 3289                   4A28
   FORM5.FRM                                 4432                   F6BD
   FORM6.FRM                                 3887                   CDAF
   FORM7.FRM                                 1176                   3B2D



     [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E]  [EXIT: ESC]
```

Figure 2-7 ANSI Plus - Stored Forms

Deleting a Form

Deleting a stored form will cause the form to be deleted from the terminal's hard drive. To delete a stored form, follow the procedure below.

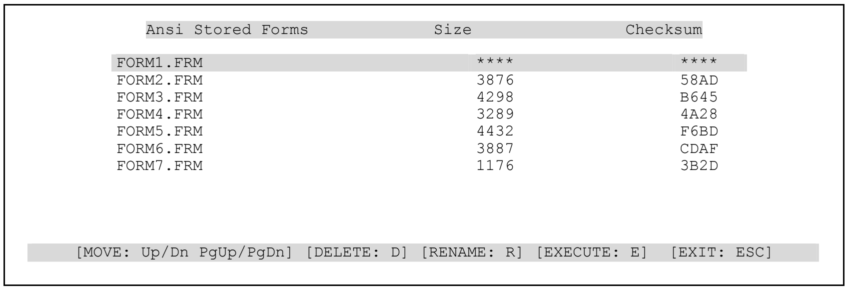
1. Press [Alt]+[M] to access the Stored Forms menu.
2. Use the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys to highlight the appropriate form name.
3. Press [D] (not case sensitive, may be D or d), the TE will ask for verification of whether or not the form is to be deleted.



```
              Ansi Stored Forms              Size                Checksum

          FORM1.FRM                             2018                  A7CC
          FORM2.FRM                             3876                  58AD
          FORM3.FRM                             4298                  B645
          FORM4.FRM                             3289                  4A28
          FORM5.FRM                             4432                  F6BD
          FORM6.FRM                             3887                  CDAF
          FORM7.FRM                             1176                  3B2D
                               Delete the Stored Form?

                    Press Y or N

     [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E]  [EXIT: ESC]
```

Figure 2-8 ANSI Plus - Stored Forms, Deleting a Form

If **Y** is selected, the form will be deleted. The form name will remain on the screen until the screen has been refreshed by pressing [Alt]+[M] again, or the Stored Forms menu has been exited and re-entered. The "****" string will be placed in the size and checksum fields of a deleted form that is still visible on the screen.



```
              Ansi Stored Forms              Size                Checksum

           FORM1.FRM                            ****                  ****
           FORM2.FRM                            3876                  58AD
           FORM3.FRM                            4298                  B645
           FORM4.FRM                            3289                  4A28
           FORM5.FRM                            4432                  F6BD
           FORM6.FRM                            3887                  CDAF
           FORM7.FRM                            1176                  3B2D



     [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E]  [EXIT: ESC]
```

Figure 2-9 ANSI Plus - Stored Forms, Deleted Form

If **N** is selected, the TE will return to the Stored Forms menu without deleting the selected form.

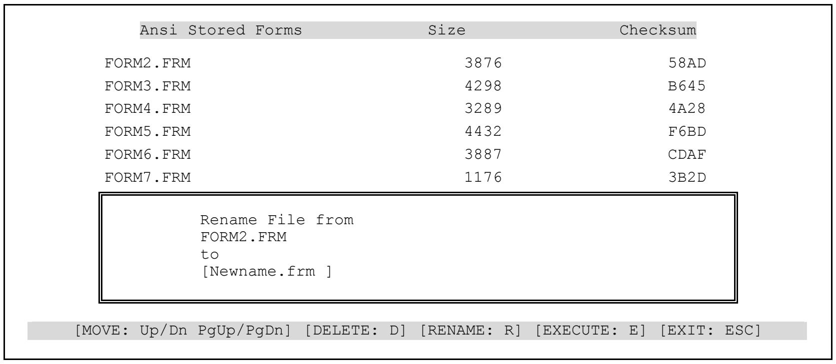
4. Press [Esc] to exit the Stored Forms menu.

Renaming a Form

Renaming a stored form will change the filename of the stored form.

To rename a stored form, follow the procedure below.

1. Press [Alt]+[M] to access the Stored Forms menu.
2. Use the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys to highlight the appropriate form name.
3. Press [R] (not case sensitive, may be R or r), the TE will display the old filename and ask for a new filename.

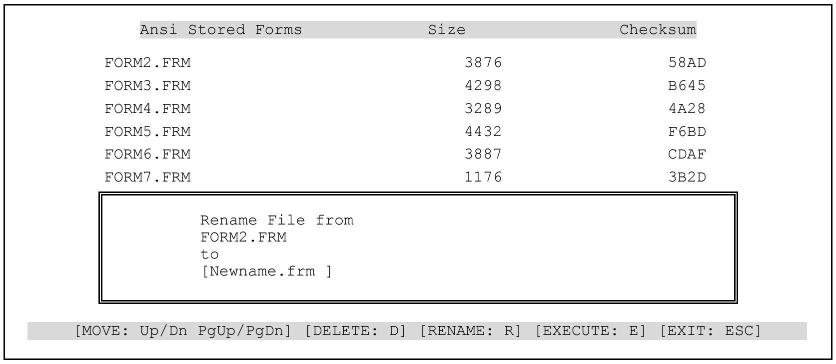


```
    Ansi Stored Forms             Size                Checksum

 FORM2.FRM                            3876                58AD
 FORM3.FRM                            4298                B645
 FORM4.FRM                            3289                4A28
 FORM5.FRM                            4432                F6BD
 FORM6.FRM                            3887                CDAF
 FORM7.FRM                            1176                3B2D

          Rename File from
          FORM2.FRM
          to
          [Newname.frm ]

 [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E] [EXIT: ESC]
```

Figure 2-10 ANSI Plus - Stored Forms, Renaming A Form

4. Type in the new filename. The string must follow normal DOS file naming conventions with a maximum of 12 characters (8 characters for the filename and 3 characters for the extension, separated by a period).

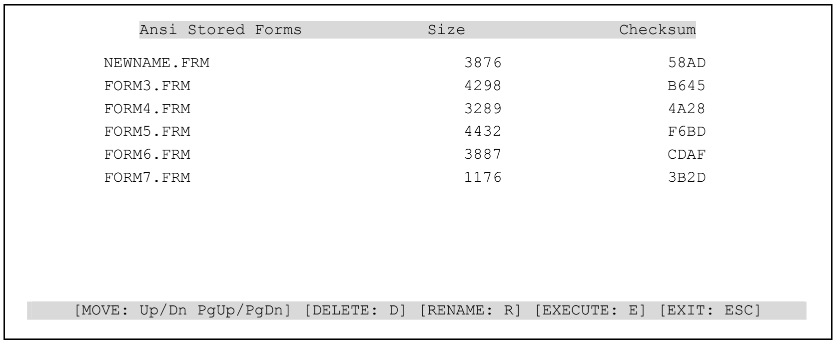
   To exit without changing the name press [Esc].



```
    Ansi Stored Forms             Size                Checksum

 FORM2.FRM                            3876                58AD
 FORM3.FRM                            4298                B645
 FORM4.FRM                            3289                4A28
 FORM5.FRM                            4432                F6BD
 FORM6.FRM                            3887                CDAF
 FORM7.FRM                            1176                3B2D

          Rename File from
          FORM2.FRM
          to
          [Newname.frm ]

 [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E] [EXIT: ESC]
```

Figure 2-11 ANSI Plus - Stored Forms, New Form Name

5. Press [Enter], the TE will return to the stored forms menu with the new filename displayed.



```
         Ansi Stored Forms               Size                Checksum

      NEWNAME.FRM                            3876                 58AD
      FORM3.FRM                              4298                 B645
      FORM4.FRM                              3289                 4A28
      FORM5.FRM                              4432                 F6BD
      FORM6.FRM                              3887                 CDAF
      FORM7.FRM                              1176                 3B2D




    [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E] [EXIT: ESC]
```

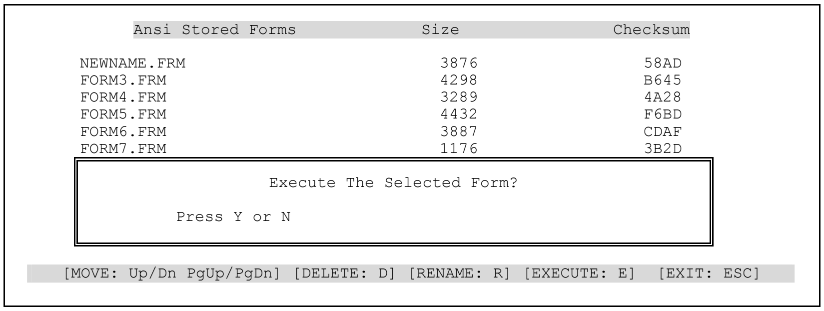
Figure 2-12 ANSI Plus - Stored Forms, Displaying Name

6. Press [Esc] to exit the Stored Forms menu.

Executing a Form

To execute a stored form, follow the procedure below.

1. Press [Alt]+[M] to access the Stored Forms menu.
2. Use the [UpArrow], [DownArrow], [PageUp], or [PageDown] keys to highlight the appropriate form name.
3. Press [E] (not case sensitive, may be E or e), the TE will ask for verification of whether or not the form is to be executed.



```
       Ansi Stored Forms              Size                Checksum

  NEWNAME.FRM                           3876                  58AD
  FORM3.FRM                             4298                  B645
  FORM4.FRM                             3289                  4A28
  FORM5.FRM                             4432                  F6BD
  FORM6.FRM                             3887                  CDAF
  FORM7.FRM                             1176                  3B2D

                        Execute The Selected Form?

              Press Y or N

    [MOVE: Up/Dn PgUp/PgDn] [DELETE: D] [RENAME: R] [EXECUTE: E]  [EXIT: ESC]
```

Figure 2-13 ANSI Plus - Stored Forms, Executing A Form

If **Y** is selected, the TE will immediately execute the selected form and the stored forms menu screen will be removed.

If **N** is selected, the TE will return to the Stored Forms menu without executing the selected form.

4. Press [Esc] to exit the Stored Forms menu.

Exiting the Stored Forms Menu

To exit the Stored Forms menu, press the [Esc] key.

Creating Stored Forms

Stored forms are written by the host application programmer and sent from the host to the terminal. You cannot create a form from the terminal. (Refer to the “Stored Forms” section of Chapter 8, “LXE Private Escape Sequences”, for more information on creating ANSI Plus stored forms).

Unlocking the Keyboard

When the operator initiates a transmission to the host, ANSI Plus will lock the keyboard until it receives any transmission from the host. While the keyboard is locked, any keypress causes a special beep, alerting the user that the keyboard is locked.

The unlock keyboard command ([Alt]+[U]) unlocks the keyboard and allows the operator to continue data entry before any transmission from the host is received.

Exiting ANSI Plus

To exit the ANSI Plus TE, press [Alt]+[X].

Answerback

The answerback key sequence ([Ctrl]+[Enter]) sends the C0 code ENQ (Inquiry) to the host.

If the host sends an ENQ to the terminal, ANSI Plus responds with a string containing information about the terminal. (Refer to Chapter 5, "Control Codes", for more information on the ENQ code.) The format of a default answerback message is:

```
LXE/x/number of lines/software-radio-platform-rev./terminal ID
```

where:

- **x** indicates the keyboard type.
- q = QWERTY keyboard
- a = ABCD keyboard
- **number of lines** is the number of lines on the display.
- **software-radio-platform-rev.** is the software, radio type, platform and software revision.
- terminal ID is the

900MHz terminal's RF channel identification number.

2.4GHz terminal's IP Address. If the terminal "secondary ID" parameter is enabled and a valid secondary ID value programmed, the secondary ID will replace the IP address.

Note: Some versions of ANSI Plus do not support sending the IP Address of the terminal. In those cases, the IP Address will be displayed as 000.000.000.000, regardless of the actual IP Address.

Example:

If extended addressing is ON:

```
LXE/q/25/APLUSxyy02D/255.255.255.255
```

If extended addressing is OFF:

```
LXE/q/25/APLUSxyy02D/0110
```

Switching to a Backup NC

This is a valid function for 900MHz radio devices only. 2.4GHz radio devices do not connect to Network Controllers.

If your RFDC system is configured to do so, the hot key sequence [Ctrl]+[1] enables you to switch from a primary to a secondary NC (Network Controller). This feature provides a way for the user to resume data processing if the primary NC fails.

The backup NC is not a “hot” backup, meaning the session with the primary NC is lost when you switch to the secondary NC. Transactions that were in progress on the primary NC are lost when you hot key to the secondary NC.

The usefulness of this feature depends on the physical architecture of your data collection system. In an ideal situation, the primary and secondary NCs draw power from separate sources and connect to separate host computers.

The following examples illustrate how this feature may or may not be useful:

Host went down.

If the secondary NC is attached to a different host, this feature will help; otherwise not.

Host to NC comm link went down.

If the primary and secondary NC do not share the same path to the host, this feature will help; otherwise they will both be affected by the common blockage.

Primary NC went down, no data flows through it.

Assuming the failure was not due to something the primary and secondary NCs shared in common (such as loss of power or the Ethernet), this feature should help.

Communication with any or all RFUs lost.

This feature will not help unless a very specific LAN routing failure caused the primary NC to lose contact with the RFUs, leaving the secondary NC unaffected.

Communication between terminals and RFUs lost.

This feature will not help.

Before You Begin

The primary and secondary Network Controller IDs should be configured.

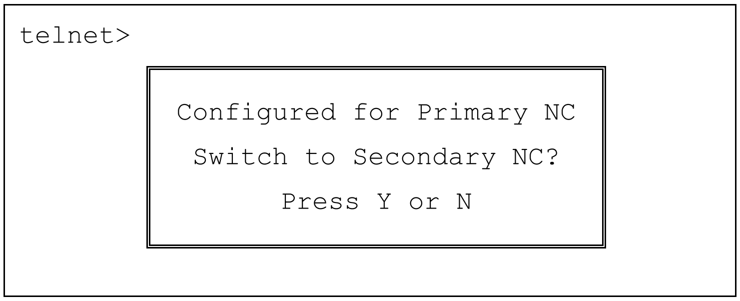
Refer to Chapter 3, “Configuration Utility”, of this reference manual for more details.

Procedure

Follow this procedure to switch to the backup NC:

1. Press the [Ctrl]+[1] key sequence.

   Result: The following screen will appear, depending on your current configuration:



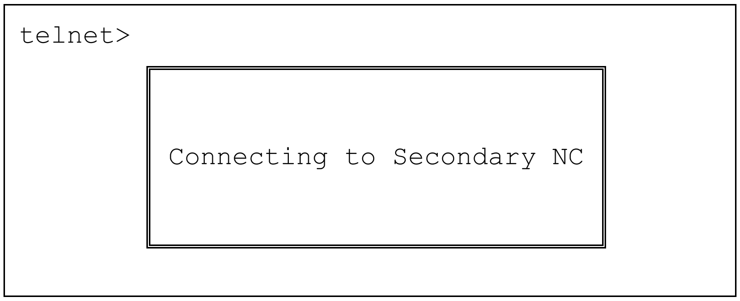
```
telnet>

        Configured for Primary NC
         Switch to Secondary NC?
               Press Y or N
```

Figure 2-14 Switching To Backup NC

2. If **N** is selected, the screen will return to what appeared on it before you pressed [Ctrl]+[1].

   If **Y** is selected, the terminal will restart the emulation you were in when you pressed [Ctrl]+[1].

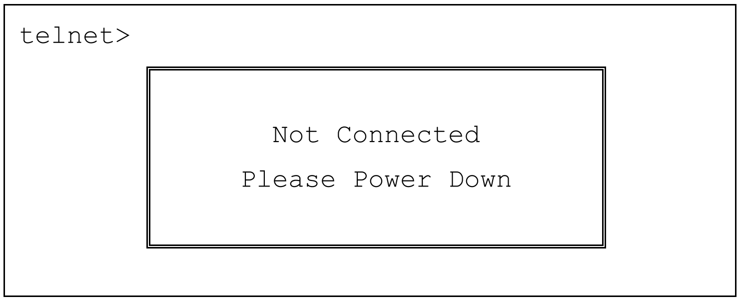
   The following messages will appear.



```
telnet>

       Connecting to Secondary NC
```

Figure 2-15 Connecting to Secondary NC

The following screen will appear if the secondary NC was not reachable:



```
telnet>

              Not Connected
            Please Power Down
```

Figure 2-16 Secondary NC Down

TELNET Escape

The TELNET escape key sequence ([Ctrl]+[4]) causes ANSI Plus TE to return to the TELNET Prompt.

If the escape code was changed at the TELNET prompt, [Ctrl]+[4] will send an ANSI Plus GS (Group Separator, see Chapter 5, “Control Codes”, for GS command details) to the network controller. See the TELNET Set Escape command.

If your system is configured to suppress the TELNET prompt, the TELNET Escape sequence has no function.

Autologin

The Autologin feature enables terminals to automatically log on to a TELNET host computer. Autologin relieves you from repeating the same log on information every time you power up a terminal.

Administering and troubleshooting the Autologin process is slightly different for LXE terminals with an installed 900MHz radio as compared to the Autologin process for a terminal with a 2.4GHz radio.

Autologin With A 900MHz Radio

Network Controller

The Network Controller provides a script file containing the host, user name, password, and the starting application command. The script file is created for each terminal and is programmed into the Network Controller's non-volatile RAM through the network management workstation.

Refer to the “6200 Network Management Guide” for details on configuring the Autologin feature.

Terminals

Using the Autologin feature, an LXE terminal will first connect to the Network Controller and then to the host each time the terminal is powered up. Host data is not displayed on the terminal screen until the Autologin process is complete.

Autologin Failure

An Autologin failure results in the “Connect Failed, please power down“ message appearing on the terminal’s screen display. Connection failure may be caused by:

- Specifying an unknown host.
- An error in the Network Controller’s autologin parameter settings or login script file.
- An offline host or network controller.

Autologin With A 2.4GHz Radio

Terminals

Autologin data is stored in each terminal and is modified using the CONFIG program, also resident in the terminal. Autologin can be enabled or disabled using the CONFIG program. Host data is not displayed on the terminal screen until the Autologin process is completed successfully.

Autologin can prompt for login, password, and the command the user would normally issue upon connection, such as a program name.

Note: Autologin parameters are edited in the terminal's CONFIG program, Main Menu option "Protocol", submenu "TCPIP Autologin."

Note: Autologin parameters can be configured for up to four hosts. When multiple Autologin hosts are available, use the CONFIG program to set the parameters for the desired hosts. Next edit the TE Selection Menu (as described in "DOS Autoconfigurator Instructions", included on the LXE Manuals CD). Each time the TE Selection Menu is displayed (at startup, upon exiting a TE, etc), the user is able to select among the configured Autologin hosts.

Autologin Failure

An Autologin failure results in the "Connect Failed (Autologin)" message appearing on the terminal's screen display and the TELNET prompt is again displayed.

Note: If your system is configured to suppress the TELNET prompt, the TELNET prompt will not be displayed.

Connection failure may be caused by:

- Specifying an unknown host.
- An error in the CONFIG file Autologin parameter settings.
- An offline host.

The user can, at the TELNET prompt, log on manually by typing the OPEN command with the desired host name,

- or -

at the TELNET prompt, exit the ANSI Plus program, edit the Autologin data using the CONFIG program, reboot the terminal and begin the Autologin process again.

Table of Contents

Illustrations

Chapter 3
Configuration Utility

Overview

The ANSI Plus configuration program provides the ability to modify the terminal emulation (TE) setup parameters, RF hardware and software parameters and platform hardware parameters. The configuration program is a DOS program that is executed from the DOS command line. The configuration utility allows the modification of the following parameters:

- System parameters.
- Terminal Emulation parameters.
- Host or network protocol parameters.
- Radio configuration parameters.
- Platform configuration parameters.

Configuration menu options displayed on the terminal screens, *and in this chapter*, are unique to the

- terminal being configured,
- the access level of the current user,
- terminal screen display size
- and installed radio type.

The illustrations contained in this chapter represent the menu options for each menu and submenu as shown on a screen that has 20 or more columns. Menus displayed on terminals with smaller screens will display all parameters relative to the terminal setup, although an extra keypress may be needed to move from parameter to parameter.

Parameters unique to a radio type are tagged with the radio type name, e.g. 900MHz or 2.4GHz. Parameters that are not tagged with a specific radio type are used for all terminals and radio types.

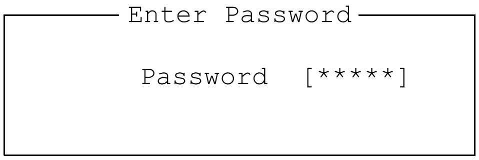
Note: Some parameters are unique to certain software revision levels. These parameters are identified with the revision level of the software.

Configuration Utility Access

To access the configuration utility:

1. Type `CD\APLUS` at the DOS prompt.
2. Press [Enter].
3. Type `CONFIG` at the prompt.
4. Press [Enter].

The configuration utility will then prompt the user for a password.



```
Enter Password

  Password   [*****]
```

Figure 3-1 Password Prompt

Pre-Configured Passwords

The ANSI Plus TE is shipped with passwords pre-configured. The pre-configured passwords are:

- Operator Access (Level 1) = [Tab] **[1] [2]**
- System Administrator Access (Level 2) = **[Tab] [3] [4] [5]**

Refer to the “System Passwords” section of this chapter for more information on changing the passwords.

If an incorrect password is entered, the user is placed back at the DOS prompt. If the correct password is entered, the Main Menu appears.

Using the Configuration Program

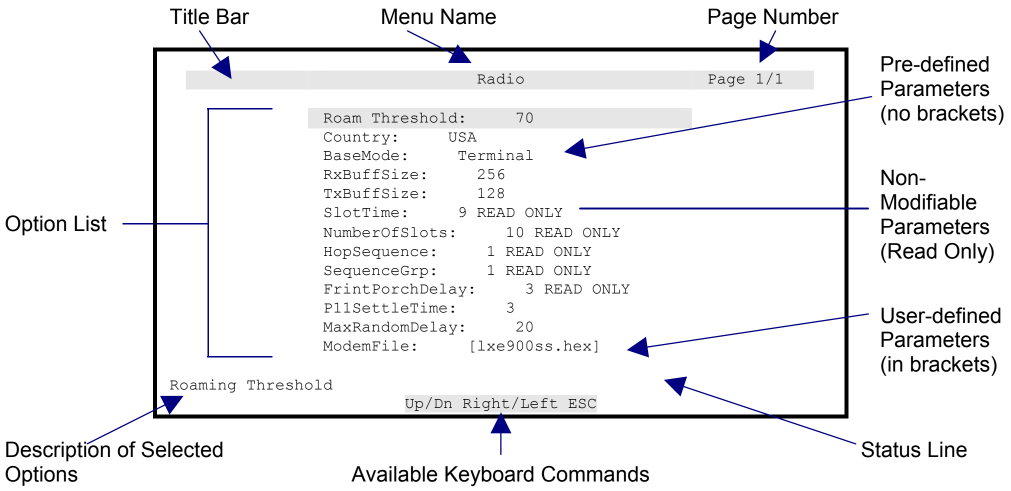
The set of keyboard commands used to select menu items and parameters, as well as modify parameter values are as follows:

Action	Keystrokes
Exit a menu	[Esc]
Acknowledge message or prompt	[Esc] or [Enter]
Select menu option (Menu or Parameter)	[UpArrow] or [DownArrow] or [LeftArrow] or [RightArrow]
Display menu	[Enter]
Modify parameter	[Space] or [LeftArrow] or [RightArrow] or type in value
Next/Previous page selection	[PageUp] or [PageDown]

IMPORTANT Configuration menu options discussed ***in this chapter*** are directed towards all LXE DOS equipment. Some parameters may be applicable to only specific LXE computers. In those cases, the description of the parameter indicates the applicable computers.

Menu Components

Each configuration menu is made up of several components.



```
                              Radio                    Page 1/1

               Roam Threshold:     70
               Country:     USA
               BaseMode:     Terminal
               RxBuffSize:     256
               TxBuffSize:     128
               SlotTime:     9 READ ONLY
               NumberOfSlots:     10 READ ONLY
               HopSequence:     1 READ ONLY
               SequenceGrp:     1 READ ONLY
               FrintPorchDelay:     3 READ ONLY
               P11SettleTime:     3
               MaxRandomDelay:     20
               ModemFile:     [lxe900ss.hex]

Roaming Threshold
                         Up/Dn Right/Left ESC
```

Figure 3-2 Configuration Utility - Menu Components

Note: Configuration menu options and components displayed are unique to the terminal being configured, the current user's access level, the terminal screen display size and installed radio type.

Title Bar

The title bar is the highlighted area at the top of the screen. The title bar contains:

Menu Name:	The name of the currently displayed menu. (Located in the center of the bar.)
Page Number:	The current page number as well as the total number of pages used to display the current menu. (Located in the upper right hand corner of the title bar.)

Option List

The list of parameters or additional menus available from the current menu.

Status Bar

The status bar is the highlighted area at the bottom of the screen. The status bar contains:

Option Description:	A brief description of the currently highlighted (selected) option. (Located at the top of the status line.)
Keyboard Commands:	The keyboard commands available from the current menu. Typically commands used to select or modify a parameter. (Located at the bottom of the status line.)

Modifiable Parameters

There are two types of modifiable parameters within the configuration utility:

- Pre-defined
- User Defined

Pre-Defined Parameters

Pre-defined parameters are parameters with a specific set of pre-defined options. These parameters may only be set to one of the pre-defined options.

User Defined Parameters

User defined parameters are parameter strings that have no pre-defined options. The user defined parameter strings are denoted with [brackets]. The space within the brackets indicates the maximum characters allowed:

[ ]	Short parameter field
[ ]	Longer parameter field

These parameters require the user to type in a parameter definition when modifying them. Some parameters have a default setting. This default can be recalled by using the Reset to Defaults option from the main menu.

ATTENTION
When using the Reset to Defaults option (and a 2.4GHz radio) all parameters available for that user level are reset to their factory default setting, with the exception of the passwords. Before using the "Reset to Defaults" option, record the values for "serial-number" and "authentication-key" parameters in the "Kernel" sub-menu.

Non-Modifiable Parameters

The non-modifiable parameters are those which cannot be modified. These parameters are marked as *read only*.

Main Menu

Configuration menu options and/or parameters displayed at the menu and activity screens are unique to the terminal being configured, the access level of the current user, terminal screen display size and installed radio type.

The main menu of the configuration utility allows access to the System, Emulation, Protocol, Radio, and Platform parameters, as well as saving the modified parameters and exiting the configuration utility program.

The Main menu is composed of the following menu options:

Main Menu Option	**Radio Type Generated Function**	**Access Level**
System	900MHz and 2.4GHz Opens submenu containing modifiable System parameters:	
	- Operating Mode	2 and 3
	- Passwords	1, 2 and 3
	- Reset Defaults	1, 2 and 3
Emulation	900MHz and 2.4GHz Modify screen display parameters such as LeftJustify, AutoWrap, etc.	2 and 3
Protocol	2.4GHz only Opens submenu containing modifiable protocol parameters: - TCPIP Autologin - Interface, DNS, General and Kernel parameters are changed by editing the SOCKET.CFG file.	2 and 3
Radio	900MHz - Modify radio parameters such as Network ID, Base Mode, etc.	2 and 3
	2.4GHz – Instructs the user to manually edit the NET.CFG to change radio parameters.	2 and 3
Platform	900MHz and 2.4GHz Modify platform parameters such as Beeper Volume, COM Port Settings, etc.	1, 2 and 3
Exit with Save	Saves current settings and exits the configuration program.	1, 2 and 3
Exit Without Save	Exits the configuration program without saving the changed parameter settings.	1, 2 and 3

System Menu Option

900MHz and 2.4GHz Radio Terminals

Upon choosing System at the Main Menu, the next screen displayed is the System submenu:

To return to the Main Menu press the [Esc] key. A detailed description of System parameters follows.

The last option on the System Submenu screen is the **Reset Defaults** parameter. Selecting this option allows the user to choose to reset the ANSI Plus TE parameters to their factory settings or let the parameters remain unchanged. Passwords are NOT affected by the Reset Defaults function.

Operating Mode Parameters

To return to the System Menu press the [Esc] key. A detailed description of Operating Mode parameters follows.

Parameter Access Levels

Refer to the “System Passwords” section of this chapter for more information on *access levels*.

Legacy ANSI TE Equivalent

In the following parameter descriptions, *Legacy ANSI TE* refers to the terminal emulation running on the 1280, 1290, 2280, 2285, and 2315 terminals.

OperatingMode – Operating Mode

IMPORTANT *RF is the only valid OperatingMode parameter for LXE DOS computers running ANSI Plus.*

Function

The OperatingMode parameter specifies what form of input/output the terminal should use for host communications and local I/O. When RF or File input is selected, set the RFFileModeCfg parameter accordingly. When Comm Port is selected, set the CommModeCfg parameter accordingly.

Options

Valid options:	RF, Comm Port, File
Default setting:	RF

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CommModeCfg – Comm Port Mode Config

Function

The CommModeCfg parameter specifies which comm port to use as host input and which one to use as local I/O (ie. for either bar code input or printer output). This parameter is only valid when the OperatingMode parameter is set to Comm Port. Not available on 2325/MX2.

Note *For use by LXE Engineering only.*

Options

Valid options:	Comm1 Host, Comm2 Aux Comm1 Aux, Comm2 Host
Default setting:	Comm1 Aux, Comm2 Host

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

NormalIO – Undocked I/O Configuration

Function

The NormalIO parameter specifies which comm port to use as local input (for a bar code scanner) and which one to use as local output (for a printer). This parameter only affects operation of the MX1 and MX3 when they are not docked. This parameter is also valid for 1380, 1390, VX1, VX2 and VX4.

The MX1 supports endcaps with an integrated scanner, an RS-232 port or both. A selection of a NormalIO configuration that is not fully supported by the endcap will result in partial functionality. The TE will not attempt to correct the selection.

Options

MX1

Valid options:	IS-IN:RS-OUT	COM1 RS-232	Output
		COM1 Scanner	Input
		COM2 IR	N/A
	RS-IN/OUT	COM1 RS-232	Input/Output
		COM1 Scanner	N/A
		COM2 IR	N/A
	IS-IN	COM1 RS-232	N/A
		COM1 Scanner	Input
		COM2 IR	N/A
Default setting:	IS-IN:RS-OUT		

MX3

Valid options:	CM1-IN:CM2 RS-IN/OUT	COM1	Input
		COM2 IR	N/A
		COM2 RS-232	Input/Output
	CM1-IN/OUT:CM2 RS-IN	COM1	Input/Output
		COM1 IR	N/A
		COM2 RS-232	Input
Default setting:	CM1-IN:CM2 RS-IN/OUT		

VX1, VX2, VX4, 1380, 1390

Valid options:	CM1-IN:CM2-IN/OUT	COM1	Input
		COM2	Input/Output
	CM1-IN/OUT:CM2-IN	COM1	Input/Output
		COM2	N/A
Default setting:	CM1-IN:CM2-IN/OUT		

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Related Parameters

DockedIIO, OutInDelay

DockedIO – Docked I/O Configuration

Function

The DockedIO parameter specifies which comm port to use as local input (for a bar code scanner) and which one to use as local output (for a printer). This parameter only affects operation of the MX1 and MX3 when they are docked. Not available on 1380/1390, VX1/VX2/VX4 and 2325/MX2.

The MX1 supports endcaps with an integrated scanner, an RS-232 port or both. A selection of a DockedIO configuration that is not fully supported by the endcap will result in partial functionality. The TE will not attempt to correct the selection.

The MX3 COM2 port supports both an IR port and an RS-232 port. The power is applied to the port configured as input until the terminal receives a print command. At that time, the power is switched to the port configured as output. When the print command is completed, power is returned to the input port. This configuration option allows dual scanning through a cradle.

Options

MX1

Valid options:	Not supported[1]	COM1 RS-232	Not supported
		COM1 Scanner	Not supported
		COM2 IR	Not supported
	RS-IN/OUT:IR:IN	COM1 RS-232	Input/Output
		COM1 Scanner	N/A
		COM2 IR	Input
	IS-IN:IR-IN/OUT	COM1 RS-232	N/A
		COM1 Scanner	Input
		COM2 IR	Input/Output
	RS-IN:IR-IN/OUT	COM1 RS-232	Input
		COM1 Scanner	N/A
		COM2 IR	Input/Output
	IS&RS-IN:RS-OUT	COM1 RS-232	Output
		COM1 Scanner	Input
		COM2 IR	Input
Default setting:	Not supported		

---

[1] Defaults to undocked configuration. See Normal IO.

MX3

Valid options:	Not supported[2]	COM1	Not supported
		COM2 IR	Not supported
		COM2 RS-232	Not supported
	CM1-IN:CM2RS-IN/OUT	COM1	Input
		COM2 IR	N/A
		COM2 RS-232	Input/Output
	CM1-IN/OUT:CM2RS-IN	COM1	Input/Output
		COM2 IR	N/A
		COM2 RS-232	Input
	CM1-IN:CM2IR-IN/OUT	COM1	Input
		COM2 IR	Input/Output
		COM2 RS-232	N/A
	CM1-IN/OUT:CM2IR-IN	COM1	Input/Output
		COM2 IR	Input
		COM2 RS-232	N/A
	CM1&CM2RS-IN:IR-OUT	COM1	Input
		COM2 IR	Output
		COM2 RS-232	Input
	CM1&IR-IN:CM2RS-OUT	COM1	Input
		COM2 IR	Input
		COM2 RS-232	Output
Default setting:	Not supported		

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Related Parameters

NormalIO, OutInDelay

---

[2] Defaults to undocked configuration. See Normal IO.

InputFName – Input File Name

Function

The InputFileName parameter specifies which file to use as host input. This parameter is only valid when the OperatingMode parameter is set to File.

Note *For use by LXE Engineering only.*

Options

Valid options:	any DOS file name
Default setting:	attrtest.txt

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BigBufferCount – Big Buffer Count

Function

The BigBufferCount parameter specifies how many times to execute the file given by the InputFName parameter. Note that a special version of the TE software is necessary to use this parameter. Not available on 2325/MX2.

Note *For use by LXE Engineering only.*

Options

Valid options:	1 - 30000
Default setting:	100

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Passwords

Note: Passwords are NOT affected by the Reset Defaults function.

There are three (3) access levels associated with the configuration utility. The access levels are:

Operator	(Access Level 1)
System Administrator	(Access Level 2)
LXE Technical staff	(Access Level 3)

Access to the configuration utility is gained by entering the appropriate password at the DOS prompt. The password determines the access level of the user. Passwords are masked upon entry. When a password is entered, each character is displayed as an "*" on the terminal display.

Passwords are modified by selecting the System option from the main menu. The Password menu then appears.

Only passwords for the current access level (or any lower access levels) can be modified. For example, the System Administrator may change his/her password and that of the Operator, but the Operator can only change his/her password.

The passwords menu is composed of the following menu items:

Menu Option	Description	Valid Parameters	Access Levels
OperatorPasswd	Operator's Password Access Level 1	user defined (3 character alphanumeric string)	1, 2 and 3
SupvPasswd	System Administrator's Password Access Level 2	user defined (4 character alphanumeric string)	2 and 3
LxeEngPasswd	LXE Technical Staff Password Access Level 3	user defined (5 character alphanumeric sring)	3

Default Passwords

Passwords entered by users are validated against user-defined entries found in the configuration file (LXE.INI). If the configuration file (LXE.INI) does not exist, the configuration utility will use the internally generated password.

The default password is generated daily, and therefore does not remain the same. To obtain the generated password, contact LXE customer support.

Operator Access – Level 1 Password

Function

This level allows modification of only a few basic parameters, such as operator password and platform parameters. The operator password is three characters long and consists of alphanumeric characters only.

Options

Valid options: user defined (3 character alphanumeric string)

Default setting: Tab 1 2 (pre-configured by LXE)

Access Level

Parameter can be modified by a user with an access level of 1, 2 or 3.

System Administrator Access – Level 2 Password

Function

This level allows modification of all parameters. The System Administrator password must be four characters long and consists of alphanumeric characters only.

Options

Valid options: user defined (4 character alphanumeric string)

Default setting: Tab 3 4 5 (pre-configured by LXE)

Access Level

Parameter can only be modified by a user with an access level of 2 or 3.

LXE Technical Staff Access – Level 3 Password

Function

This level allows modification of all parameters (except for read-only parameters). The LXE Technical Staff password must be five characters long and consists of alphanumeric characters only.

Options

Valid options: user defined (5 character alphanumeric string)

Default setting: Tab 6 7 8 9 (pre-configured by LXE)

Access Level

Parameter can be modified by a user with an access level of 3.

Reset Defaults

Note: Passwords are NOT affected by the Reset Defaults function.

The Reset Defaults option from the System menu allows the restoration of factory defaults. This feature only allows the restoration of defaults for the parameters currently accessible. In other words, only the parameters available to the current user's access level will be restored.

ATTENTION
When using the Reset to Defaults option (and a 2.4GHz radio) all parameters available for that user level are reset to their factory default setting, with the exception of the passwords.

To return to the System Menu press the [Enter] key.

Reset	Press the [Y] key and then the [Enter] key to reset the default values. System values are returned to their default values and configuration files are changed. See the sub-sections titled "Reset 900MHz Parameters" and "Reset 2.4GHz Parameters" in this Section.
	- or -
Ignore	Press the [N] key and then the [Enter] key to ignore the Reset command and remove the Reset text box from the screen display. System values are not changed.
	- or -
Ignore	At the "[N]" prompt, press the [Enter] key to remove the Reset box from the screen display. System values are not changed.

Reset 900MHz Parameters

When parameters in a 900MHz terminal are changed or reset to default values, the LXE.INI file is changed.

Reset 2.4GHz Parameters

When parameters in a 2.4GHz device are changed or reset to default values, the LXE.INI, the radio driver and the FTP stack files are changed.

Emulation Menu Option

900MHz and 2.4GHz Radio Terminals

The Emulation menu contains parameters pertaining to terminal emulation. Emulation parameters are displayed on multiple screens.

To access the second screen of emulation parameters, press the [PageDown] key.

Note: Configuration menu options and components displayed are unique to the terminal being configured, the current user's access level, and the terminal screen display size.

To return to the first screen, press the [PageUp] key. A detailed description of each emulation parameter follows. To access the third screen of emulation parameters, press the [PageDown] key.

To return to the Main Menu, press [Esc].

Access Levels

Refer to the “System Passwords” section of this chapter for more information on *access levels*.

Legacy ANSI TE

In the following parameter descriptions, *Legacy ANSI TE* refers to the terminal emulation running on the 1280, 1290, 2280, 2285, and 2315 terminals.

LockWindow – Lock Window Mode

Function

The LockWindow parameter determines whether the display window is locked in a permanent window position or if it follows the cursor in the virtual window.

Options

Valid options:	Does not track	(locked in permanent position)
	Track cursor	
Default setting:	Track cursor	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Lock Window

LockRow – Lock Row

Function

The LockRow parameter specifies the row coordinate of the upper left-hand corner (Home) position of the physical display window.

Note: No warning is given when an incorrect value is entered.

Options

Valid options:	1, 5, 9, 13, 17
Default setting:	1

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Lock Row

LockCol – Lock Column

Function

The LockCol parameter specifies the column co-ordinate of the upper left-hand corner (Home) position of the physical display window.

Note: *No warning is given when an incorrect value is entered.*

Options

Valid options:	1, 11, 21, 31, 41, 51, 61
Default setting:	1

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Lock Column

FullWinIncs – Full Window Mode

Function

The FullWinIncs parameter determines whether the terminal display moves in half or full window increments.

Options

Valid options:	Half, Full
Default setting:	Half

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Quad/Full

HostQuad – Host Quadrant

Function

The HostQuad parameter determines which quadrant the cursor appears in when the host sends a virtual screen to the terminal.

Options

Valid options:	1	Upper right
	2	Upper left
	3	Lower left
	4	Lower right
Default setting:	4	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Host Quadrant

StatusLine – Status Line Mode

Function

The StatusLine parameter specifies whether the status line will be displayed (usually on line 25) or not.

Options

Valid options:	Enabled, Disabled
Default setting:	Enabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Screen Setup|Status Line

LeftJustify – Left Justify Mode

Function

The LeftJustify parameter determines where the cursor appears in a field from one bar code entry to the next. The terminal must be in block mode for this option to work.

Options

When enabled, the cursor in a field justifies to the left each time a new bar code entry occurs. When disabled, the bar code entry appears at the end of the previous entry in that field.

Valid options:	Yes, No
Default setting:	Yes

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Bar/RS-232 Input|Left Justify

ClearData – Clear Data Mode

Function

The ClearData parameter determines if a field is automatically cleared when the next bar code entry occurs or if the data in the field is overwritten by the new entry. The terminal must be in block mode for this option to work.

Options

Valid options:	Clear, Overwrite
Default setting:	Clear

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Bar/RS-232 Input|Clear

TruncateData – Truncate Data Mode

Function

The TruncateData parameter determines whether the character position of a bar code entry that exceeds the field length are simply dropped off or if the leftover characters are entered in the next available field. The terminal must be in block mode for this option to work.

Options

Valid options:	Don’t Truncate, Truncate
Default setting:	Truncate

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Bar/RS-232 Input|Truncate

AutoAdvance – Automatic Advance Mode

Function

The AutoAdvance parameter determines whether the cursor advances to the next field at the end of a bar code entry. If the cursor does not advance to the next field, the next bar code entry is entered at the end of the previous bar code entry. The terminal must be in block mode for this option to work.

Options

Valid options:	Yes, No
Default setting:	Yes

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|Bar/RS-232 Input|Advance

EightBitMode – Eight Bit Mode

Function

The EightBitMode parameter determines whether to use the eight bit or seven bit code table when sending responses back to the host. The terminal can always receive either seven bit or eight bit codes from the host regardless of the setting of this mode.

This parameter is the equivalent to the DECSCL escape sequence.

Note: The value set in this parameter can be overriden by the host sending one of the following escape sequences: S7CIT, S8CIT, or DECSCL. See chapter 7, "DEC Private Escape Sequences", for details.

If the host does override the EightBitMode parameter, the terminal will revert to the value set in this parameter on the next power cycle.

Options

Valid options:	Eight bit, Seven bit
Default setting:	Seven bit

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Transmit|8 Bit Codes

AutoTransmit – Automatic Transmit

Function

The AutoTransmit parameter enables or disables the automatic transmit feature. When enabled, automatic transmit transmits the screen to the host when the last field is filled (either with bar code entry, keyboard entry or RS-232 port entry). When disabled, screen transmission to the host only occurs when the [Enter] key is pressed.

Options

Valid options:	Disabled, Enabled
Default setting:	Enabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

SoftwareOpModes|ANSI Mode|Transmit|AutoTransmit

TransmitCrLf – Transmit CR or CR/LF

Function

The TransmitCrLf parameter determines whether a single carriage return (CR) is transmitted when the [Enter] key is pressed or a carriage return and line feed (CR/LF) are transmitted when the [Enter] key is pressed.

Options

Valid options:	CR only CR and LF
Default setting:	CR only

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

LNM

EnterKeyChars – Enter Key Characters

Function

The EnterKeyChars parameter allows you to program the characters that ANSI Plus will use as "terminating" characters in messages it sends to the host computer when the [Enter] key is pressed. When this parameter is not programmed, each packet sent to the host is terminated by a single "carriage return" or a "carriage return" followed by a "line feed" (see the parameter TransmitCrLf in this section). Certain host computers require different and/or additional terminating characters. The EnterKeyChars parameter is provided to allow programming of the required terminating characters.

Up to five specially encoded terminating characters may be programmed in this parameter. The type of encoding is referred to as "hat encoding." The special encoding requires the use of the caret (^) and/or the tilde (~) characters in combination with a regular "printable" character.

A table of terminating characters is provided to assist you in the selection of the correct hat encoded character(s) to fulfill your requirements. Refer to the table titled "Hat Encoded Characters" during the following examples.

Note: The table titled "Hat Encoded Characters" is located at the end of this chapter.

Example 1

If the ASCII character ETX (hex value = 03) is required as a terminating character, you should enter **^C** (caret and upper case C) into the EnterKeyChars parameter. See the figure titled "Hat Encoded Characters."

Example 2

If C7 hex is required as a terminating character you should enter ~G (tilde and upper case G) into the EnterKeyChars parameter. See the figure titled "Hat Encoded Characters."

Example 3

Suppose you require the combination of 0E hex and 86 hex as terminating characters. In parameter EnterKeyChars enter the following: ^N~^F (that's caret, uppercase N, tilde, caret, uppercase F). See the figure titled "Hat Encoded Characters."

Options

Valid options: Any of the hat encoded characters in Figure titled "Hat Encoded Characters"

Default setting: None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

TransmitKeybdCSI – Transmit Keyboard CSI

Function

The TransmitKeybdCSI parameter provides the capability of programming ANSI Plus to either send CSIs when needed or never send CSIs to the host.

When ANSI Plus is **not** in "local echo" mode, certain key presses result in the sending of CSI (Control Sequence Introducer) functions to the host computer. Certain host computers do not want to receive CSIs.

Options

Valid options:	Transmit CSIs No CSI
Default setting:	Transmit CSIs

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CharColTimer – Character Collection Timer

Function

The CharColTimer parameter determines how long the terminal waits for the next key stroke before transmitting the characters entered. After the terminal waits the indicated amount of time, the terminal transmits the entered data.

Options

Each value must be multiplied by 50ms to get the actual time delay. For example, when the value 10 is selected, the actual time delay is 500ms or .5 seconds.

Valid options:	0 - 50 (x 50ms)
Default setting:	10

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Transmit|char coll tmr

ScrollMode – Scroll or Block Mode

Function

The ScrollMode parameter specifies whether the terminal operates in scroll or block mode.

Options

Valid options: Block Mode
Scroll Mode

Default setting: Scroll Mode

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

LXELM

EchoCrLf – Echo CR/LF to Terminal

Function

If the terminal is in scroll/block mode, the EchoCrLf parameter specifies whether to echo a carriage return and line feed to the terminal screen when the [Enter] key is pressed.

Options

Valid options: Send CR/LF
Don't send CR/LF

Default setting: Send CR/LF

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

LXENE

AutoWrap – Automatic Wrap

Function

The AutoWrap parameter enables or disables the automatic wrap feature. When enabled (on), AutoWrap automatically wraps characters (received from the host or typed at the keyboard) to the next line when the cursor reaches column 80 of the screen. When disabled (off), characters received by the terminal do not wrap to the next line when the right-hand border of the screen is reached.

Options

Valid options:	Off, On
Default setting:	On

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

DECAWM

OnlineMode – Online/Offline Mode

Function

The OnlineMode parameter specifies whether the [Enter] key will transmit messages to the host (via the RF link) or not.

Options

Valid options:	Offline	([Enter] disabled)
	Online	([Enter] enabled)
Default setting:	Online	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

SoftwareOpModes|ANSI Mode|Screen|OnLine

SendReceiveMode – Send/Receive Mode

Function

The SendReceiveMode parameter determines whether the terminal will operate in character mode or scroll/block mode.

Options

Valid options:	Char	(terminal operates in character mode)
	Line/Block	(terminal operates in scroll/block mode)
Default setting:	Line/Block	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

SRM

KpamEnable – Keypad Application Mode

Function

The KpamEnable parameter determines whether or not the keypad may be placed in application mode. When enabled, the keypad may be placed in either application mode or numeric mode. When disabled, the keypad may be placed in numeric mode only.

Options

Valid options:	Disable KPAM
	Enable KPAM
Default setting:	Enable KPAM

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

SoftwareOpModes|ANSI Mode|Keys|KpamEnable

KeypadAppMode – Keypad Mode

Function

The KeypadAppMode parameter specifies the mode of operation for the numeric keypad.

Options

Refer to the table labeled “Numeric Keypad Definitions” found in the “Default Key Definitions” section of Chapter 4, “Terminal Overview”.

Valid options:	Application	(sends application sequences from the numeric keypad)
	Numeric	(sends numeric characters from the numeric keypad)
Default setting:	Numeric	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Keys|Keypad Mode

CursorAppMode – Cursor Keys Mode

Function

The CursorAppMode parameter specifies the mode of operation for the arrow keys. Each mode sends a different application sequence to the host when a cursor key is pressed.

Options

Refer to the table labeled “Basic Default Arrow Key Definitions” found in the “Default Key Definitions” section of Chapter 4, “Terminal Overview”.

Valid options:	Application Mode
	Cursor Mode
Default setting:	Cursor Mode

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Keys|Cursor Keys

ComposeKey – Compose Key Enable

Function

The ComposeKey parameter determines if the compose key is active or not. Refer to the “Compose Key” section found in Chapter 4, “Terminal Overview” for more details.

Options

Valid options:	Disabled, Enabled
Default setting:	Enabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Keys|Compose

DisplayCtrlCodes – Display Control Codes

Function

The DisplayCtrlCodes parameter determines whether the control codes are executed or displayed on the screen.

Options

Valid options:	Displayed, Executed
Default setting:	Executed

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Debug|Ctrl codes

DisplayBadSeqs – Display Bad Sequences

Function

The DisplayBadSeqs parameter determines if the terminal displays a bad sequence of data or not. The terminal ignores the sequence in either case.

Options

Valid options: Displayed, Ignored

Default setting: Displayed

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software Op Modes|ANSI Mode|Debug|Bad Seqs

AlphaOnly – Alpha Only Wildcard

Function

The AlphaOnly parameter specifies the wildcard character that represents an alphabetic character. The characters represented by this wildcard are: A - Z, a - z, comma (,), period (.), dash (-) and space ( ).

Options

Enter the decimal equivalent of the ASCII character desired.

Valid options:	32 - 126	(Decimal)
Default setting:	63	(? in ASCII)

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software features|Local Field Edit|Alpha only

DigitsOnly – Digits Only Wildcard

Function

The DigitsOnly parameter specifies the wildcard character that represents a digit. The characters represented by this wildcard are: 0 - 9.

Options

Enter the decimal equivalent of the ASCII character desired.

Valid options: 32 - 126 (Decimal)

Default setting: 36 ($ in ASCII)

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software features|Local Field Edit|Digits only

NumericOnly – Numeric Only Wildcard

Function

The NumericOnly parameter specifies the wildcard character that represents a numeric character. The characters represented by this wildcard are: 0 - 9, plus (+), comma (,), period (.), dash (-) and space ( ).

Options

Enter the decimal equivalent of the ASCII character desired.

Valid options: 32 - 126 (Decimal)

Default setting: 35 (# in ASCII)

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software features|Local Field Edit|Numeric only

AlphaNumericOnly – Alphanumeric Only Wildcard

Function

The AlphaNumericOnly parameter specifies the wildcard character that represents an alphanumeric character. The characters represented by this wildcard are: A - Z, a - z, 0 - 9, plus (+), comma (,), period (.), dash (-) and space ( ).

Options

Enter the decimal equivalent of the ASCII character desired.

Valid options:	32 - 126	(Decimal)
Default setting:	43	(+ in ASCII)

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software features|Local Field Edit|Alphanumeric

MatchAll – Match All Wildcard

Function

The MatchAll parameter specifies the wildcard character that represents any non-control character (all characters included). The characters represented by this wildcard are: any GL or GR character.

Options

Enter the decimal equivalent of the ASCII character desired.

Valid options:	32 - 126	(Decimal)
Default setting:	42	(* in ASCII)

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

Software features|Local Field Edit|Match all

CursorVisible – Cursor Visible or Invisible

Function

The CursorVisible parameter determines whether the cursor will be visible or invisible.

Options

Valid options:	Invisible, Visible
Default setting:	Visible

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CursorType – Cursor Type

Function

The CursorType parameter determines whether the cursor will be displayed as a block or an underscore character.

Options

Valid options:	Block, Underscore
Default setting:	Block

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CursorBlink – Blinking Cursor

Function

The CursorBlink parameter determines whether the cursor will be blinking or steady.

Options

Valid options:	Blink, Steady
Default setting:	Blink

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

NationalMode – National or Multinational Mode

Function

The NationalMode parameter determines whether the terminal uses national replacement character sets or the DEC multinational character set.

Note: The KeyboardType parameter must be set to a value other than North American for NationalMode to function.

Options

Valid options: Multinational, National

Default setting: Multinational

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

KeyboardType – Installed Keyboard Type

Function

The KeyboardType parameter specifies the type of keyboard installed. (Refer to the hardware manual for more information.)

This parameter determines which national replacement character set is active when used in conjunction with the NationalMode parameter.

Options

Valid Options:

NorthAmerican
British
Flemish
Canadian (French)
Danish
Finnish
German
Dutch
Italian
Swiss (French)
Swiss (German)
Swedish
Norwegian
French/Belgian
Spanish
Custom (Allows use of CusomChar1 through CustomChar12)

Default setting: NorthAmerican

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CustomChar1 – Normal and Custom Character 1 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar1 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen. Not available on 2325/MX2 (Applies to all 12 CustomChar parameters).

For Example:

The first value of CustomChar1 is the ASCII screen character that is to be replaced.

```
CustomChar1: >23h<   20h
```

This switchover means: Display a space (20h) whenever a pound sign (23h) is to be placed on the screen.

The second value is the hexadecimal value of the new ASCII character displayed on the screen when the first key is pressed.

```
CustomChar1:  b2h   >23h<
```

This switchover means: Display a pound sign (23h) whenever a superscript ^{2} two (b2h) is to be placed on the screen.

Note: This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Options

Valid options:	First value:	hexadecimal values between 0x20 and 0x7e
	Second value:	hexadecimal values between 0x20 - 0x7e, 0xa1 -0xa3, 0xa5, 0xa7 - 0xab, 0xb0 - 0xb3, 0xb5 - 0xb7, 0xb9 - 0xbd, 0xbf, 0xc0 - 0xcf, 0xd1 - 0xdd, 0xdf - 0xef, 0xf1 - 0xfd.
	So, in explanation, you can choose any twelve characters from the ASCII table in locations 0x20 - 0x7e, and replace them with any other characters in that same range, or any of the printable characters from the DEC Supplemental Graphic table.	
Default setting:	First value:	20h
	Second value:	20h
	(0x20 or 20h is the hexadecimal value of Space)	

The cursor is in the “character to be replaced” field between the > and the <. Change the value by *either* typing a hexadecimal value *or* pressing the left and right arrow keys until the desired value is displayed.

Press the [Tab] key to move the cursor to the “replace the character with” field. The > and < will move to the second field.

Change the value by *either* typing a hexadecimal value *or* pressing the left and right arrow keys until the desired value is displayed.

Note: Pressing [Shift]+[Tab] causes the two fields to toggle between their hex and decimal values.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

TSIO Lettered Menus | Menu L :Character Sets

CustomChar2 – Normal and Custom Character 2 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar2 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar2 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar3 – Normal and Custom Character 3 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar3 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar3 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar4 – Normal and Custom Character 4 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar4 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar4 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar5 – Normal and Custom Character 5 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar5 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar5 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar6 – Normal and Custom Character 6 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar6 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar6 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar7 – Normal and Custom Character 7 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar7 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar7 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar8 – Normal and Custom Character 8 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar8 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar8 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar9 – Normal and Custom Character 9 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar9 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar9 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar10 – Normal and Custom Character 10 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar10 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar10 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar11 – Normal and Custom Character 11 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar11 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar11 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

CustomChar12 – Normal and Custom Character 12 of 12

Note: There are twelve (12) parameters available for custom characters (CustomChar1 through CustomChar12). The only difference in the parameter names is the number at the end of the parameter. The functions of each are the same.

Function

The CustomChar12 (of 12) parameter contains the hexadecimal value of one normal and one custom character switchover.

This feature does not change the keyboard or the characters sent to the host. It only affects the characters displayed on the screen.

Note: See the previous parameter description of CustomChar1 for equivalent CustomChar12 examples, options, valid values and defaults, edit instructions and Legacy ANSI equivalents.

ErrorCountPath – Error Count File Path

Function

The ErrorCountPath parameter is used to specify the path of the error count file. ErrorCountPath is a user defined parameter which can be from 1 to 12 alphanumeric characters in length. Only the path (or directory location) can be entered into this field, no filename is necessary.

Options

Valid options:	[user defined]	(up to 12 alphanumeric characters)
Default setting:	(none)	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ErrorCountFName – Error Count Filename

Function

The ErrorCountFName parameter is used to specify the name of the error count file. ErrorCountFName is a user defined parameter which can be from 1 to 12 alphanumeric characters in length. Only the filename can be entered into this field, no path statements are necessary.

Options

Valid options:	[user defined]	(up to 12 alphanumeric characters)
Default setting:	err.cnt	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

StoredFormsDir – Stored Forms Directory

Function

The StoredFormsDir parameter specifies the directory containing the stored forms. StoredFormsDir is a user defined parameter which can be from 1 to 12 alphanumeric characters in length. Only the path (or directory location) can be entered into this field, no filename is necessary.

Options

Valid options:	[user defined]	(up to 12 alphanumeric characters)
Default setting:	\stforms (indicates a directory called stforms found in the directory containing the ANSI Plus program, for example C:\APLUS\STFORMS)	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

SecondaryID – Secondary ID

2.4GHz Radio Terminals only

Function

This parameter is only used for terminals equipped with a 2.4GHz radio. However, use of this parameter is required **only** if the 2.4GHz RF terminal is being used in conjunction with the LXE 6224 Session Manager. If you are using **2.4GHz radios but not using the LXE 6224,** you may choose to implement or not implement this parameter.

Note: This parameter is functional only when the parameter "Enable 2nd ID" is set to "enabled."

When the terminal receives an "inquiry" (ENQ) command across the RF link it responds with an "answer back" message. One piece of information in the answer back message is data to identify this particular terminal to the sender of the ENQ command. When this parameter is not used in the 2.4GHz RF terminal, the terminal's IP address is included in the answer back message. When this parameter is used in the 2.4GHz terminal, the terminal's "secondary ID" is included in the answer back instead of the IP address.

Options

Valid options:	0001 to FFFF
Default:	0001

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Enable2ndID – Enable Secondary ID

Function

If the terminal is equipped with a **900MHz radio**, this parameter must be set to “disabled.”

If the terminal **is** equipped with a 2.4GHz radio **and** if your system uses the LXE 6224 Session Manager, this parameter must be set to “enabled.”

If the terminal **is** equipped with a 2.4GHz radio and you are **not** using the LXE 6224 Session Manager you may choose to use or not use the secondary ID parameter.

See the parameter above regarding the terminal’s secondary ID parameter.

When this parameter is set to enabled, a valid value must be entered into the “Secondary ID” parameter.

Options

Valid options:	Enabled, Disabled
Default:	Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

FCNMovement – Cursor Function Key Movement

Function

If disabled, there would is no cursor movement with default defined function keys.

Options

Valid options:	Enabled, Disabled
Default setting:	Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Protocol Menu Option

2.4GHz Radio Terminals only

Upon choosing Protocol at the Main Menu, the next screen displayed is the Protocol submenu. To return to the Main Menu press the [Esc] key. A detailed description of Protocol parameters follows.

TCPIP Autologin Parameters

2.4GHz Radio Terminals only

The Autologin feature enables terminals to automatically log on to a TELNET host computer. Autologin relieves the user from repeating the same log on information every time they power up a terminal.

The TCPIP Autologin screen contains parameters pertaining to terminal login. Up to four autologin hosts may be specified, and will be referred to as Host A, Host B, Host C and Host D. There are ten autologin parameters for each host. The TE Selection Menu allows the user to select the desired host for the Autologin process. Please see Chapter 2, "Getting Started" for more information. Multiple hosts can be specified for all 2.4GHz computers.

The Autologin process is slightly different for LXE terminals with an installed 900MHz radio as compared to the Autologin process for a terminal with a 2.4GHz radio.

Autologin data for a 900MHz radio terminal is stored on the Network Controller and edited using the Network Management Workstation. Autologin data for a 2.4GHz radio terminal is stored in the terminal and edited using the CONFIG program.

See Chapter 2, "Getting Started", sections "Starting ANSI Plus" and "Autologin" for more information.

Note: There is no automatic connection between setting up the Autologin parameters for Hosts 2 through 4 and the multiple host TE Selection Menu. Please refer to "Modifying the TE Selection Menu" in Chapter 2, Getting Started", for information on customizing the TE Selection Menu for multiple hosts.

Parameter Access Levels

Refer to the "System Passwords" section of this chapter for more information on *access levels*.

Legacy ANSI TE Equivalent

In the following parameter descriptions, *Legacy ANSI TE* refers to the terminal emulation running on the 1280, 1290, 2280, 2285, and 2315 terminals.

TermType_A – Type of Terminal (Host A)

Function

Terminal type used in TELNET negotiations when establishing a terminal emulator to the Host A connection.

Options

Valid options:	[user defined]
Default setting:	VT220

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

AutoLoginOn_A – Terminal Automatic Login (Host A)

Function

Turn autologin function on for Host A.

When this parameter is set to enabled, the TELNET client used by the terminal will use the autologin parameters and attempt to autologin.

Note: When this parameter is enabled and after the first failed attempt to connect during autologin, the autologin process stops and the TELNET prompt is displayed.

If your system is configured to suppress the TELNET prompt, the terminal *must* be setup to use the autologin function.

When it is disabled, the terminal will stop at the TELNET> prompt and the user must supply the response to the login: and Password: prompts to get logged in.

Open stack only.

Options

Valid options:	Disabled, Enabled
Default setting:	Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Host_A – Host Name (Host A)

Function

Specifies Host A that the autologin mechanism will attempt to connect to.

Note: If your system uses the LXE 6224 Session Manager, its IP address should be entered in this parameter.

Options

Valid options: [user defined]

Default setting: none

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Prompt1_A – First Host String (Host A)

Function

The host string that the terminal's autologin must see in order to send Reply1_A back to Host A. Normally this is the Login: prompt.

Options

Valid options: [user defined]

Default setting: Login:

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Reply1_A – First Response Sent to Host (Host A)

Function

The string that the terminal's autologin mechanism sends to Host A after Prompt1_A has been received. Normally this is the user name that is being logged in.

Options

Valid options: [user defined]

Default setting: none

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Prompt2_A – Second Host String (Host A)

Function

The host string that the terminal's autologin must see in order to send Reply2_A back to Host A. Normally this is the Password: prompt.

Options

Valid options:	[user defined]
Default setting:	Password:

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Reply2_A – Second Response Sent to Host (Host A)

Function

The string that the terminal's autologin mechanism sends to Host A after Prompt2_A has been received. Normally this is the user’s password.

Options

Valid options:	[user defined]
Default setting:	none

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Prompt3_A – Third Host String (Host A)

Function

The host string that the terminal's autologin must see in order to send Reply3_A back to Host A. Normally this is something distinctive in the host's prompt.

Use Prompt3_A and Reply3_A to issue a command (i.e. - to start a program) once the user has been auto connected to the host.

Options

Valid options:	[user defined]
Default setting:	none

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Reply3_A – Third Response Sent to Host (Host A)

Function

The string that the terminal's autologin mechanism sends to Host A after Prompt3_A has been received.

Options

Valid options:	[user defined]
Default setting:	none

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

PortNum_A – Port Number (Host A)

Function

The PortNum_A parameter is used to specify a TELNET port number for connection to Host A during the Autologin process. The PortNum_A is in decimal notation.

When AutoLoginOn_A is Enabled, the TELNET client used by the terminal will add the PortNum_A value to the autologin parameters during login. When AutoLoginOn_A is enabled and after the first failed attempt to connect during autologin, the autologin process stops and the TELNET prompt is displayed.

When AutoLoginOn_A is Disabled, when the user is entering responses at the TELNET prompt, the value entered into the PortNum_A parameter is ignored.

Options

Valid options:	0 - 9000
Default setting:	0

Note: Using the default setting will generate an invalid port number message and ANSI Plus will then connect using port number 23. Port 23 is a well-known port number for TELNET.

If you do not use the AutoLogin_A option, you can specify a port number at the TELNET prompt. Use any valid option in the range of 1 - 9000.

For example:

```
TELNET> open [ip-address] [Port Number]
```

- or-

```
TELNET> open [host-name] [PortNumber]
```

You can also use the “o” TELNET command abbreviation for “open”.

Access Level

Parameter may be modified by a user with an access level of 1, 2 or 3.

Legacy ANSI TE Equivalent

None

TermType_B – Type of Terminal (Host B)

Function

Terminal type used in TELNET negotiations when establishing a terminal emulator to Host B connection.

Note: See the previous parameter description of TermType_A for equivalent TermType_B valid values and defaults.

AutoLoginOn_B – Terminal Automatic Login (Host B)

Function

Turn autologin function on for Host B.

When this parameter is set to enabled, the TELNET client used by the terminal will use the autologin parameters and attempt to autologin.

Note: See the previous parameter description of AutoLoginOn_A for equivalent AutoLoginOn_B valid values, defaults and other information.

Host_B – Host Name (Host B)

Function

Specifies the name of Host B that the autologin mechanism will attempt to connect to.

Note: See the previous parameter description of Host_A for equivalent Host_B valid values, defaults and other information.

Prompt1_B – First Host String (Host B)

Function

The host string that the terminal's autologin must see in order to send Reply1_B back to Host B. Normally this is the Login: prompt.

Note: See the previous parameter description of Prompt1_A for equivalent Prompt1_B valid values, defaults and other information.

Reply1_B – First Response Sent to Host (Host B)

Function

The string that the terminal's autologin mechanism sends to Host B after Prompt1_B has been received. Normally this is the user name that is being logged in.

Note: See the previous parameter description of Reply1_A for equivalent Reply1_B valid values, defaults and other information.

Prompt2_B – Second Host String (Host B)

Function

The host string that the terminal's autologin must see in order to send Reply2_B back to Host B. Normally this is the Password: prompt.

Note: See the previous parameter description of Prompt2_A for equivalent Prompt2_B valid values, defaults and other information.

Reply2_B – Second Response Sent to Host (Host B)

Function

The string that the terminal's autologin mechanism sends to Host B after Prompt2_B has been received. Normally this is the user's password.

Note: See the previous parameter description of Reply2_A for equivalent Reply2_B valid values, defaults and other information.

Prompt3_B – Third Host String (Host B)

Function

The host string that the terminal's autologin must see in order to send Reply3_B back to Host B. Normally this is something distinctive in the host's prompt.

Use Prompt3_B and Reply3_B to issue a command (i.e. - to start a program) once the user has been auto connected to the host.

Note: See the previous parameter description of Prompt3_A for equivalent Prompt3_B valid values, defaults and other information.

Reply3_B – Third Response Sent to Host (Host B)

Function

The string that the terminal's autologin mechanism sends to Host B after Prompt3_B has been received.

Note: See the previous parameter description of Reply3_A for equivalent Reply3_B valid values, defaults and other information.

PortNum_B – Port Number (Host B)

Function

The PortNum_B parameter is used to specify a TELNET port number for connection to Host B during the Autologin process. The PortNum_B is in decimal notation.

Note: See the previous parameter description of Reply3_A for equivalent Reply3_B valid values, defaults and other information.

TermType_C – Type of Terminal (Host C)

Function

Terminal type used in TELNET negotiations when establishing a terminal emulator to Host C connection.

Note: See the previous parameter description of TermType_A for equivalent TermType_C valid values and defaults.

AutoLoginOn_C – Terminal Automatic Login (Host C)

Function

Turn autologin function on for Host C.

When this parameter is set to enabled, the TELNET client used by the terminal will use the autologin parameters and attempt to autologin.

Note: See the previous parameter description of AutoLoginOn_A for equivalent AutoLoginOn_C valid values, defaults and other information.

Host_C – Host Name (Host C)

Function

Specifies the name of Host C that the autologin mechanism will attempt to connect to.

Note: See the previous parameter description of Host_A for equivalent Host_C valid values, defaults and other information.

Prompt1_C – First Host String (Host C)

Function

The host string that the terminal's autologin must see in order to send Reply1_C back to Host C. Normally this is the Login: prompt.

Note: See the previous parameter description of Prompt1_A for equivalent Prompt1_C valid values, defaults and other information.

Reply1C – First Response Sent to Host (Host C)

Function

The string that the terminal's autologin mechanism sends to Host C after Prompt1_C has been received. Normally this is the user name that is being logged in.

Note: See the previous parameter description of Reply1_A for equivalent Reply1_C valid values, defaults and other information.

Prompt2_C – Second Host String (Host C)

Function

The host string that the terminal's autologin must see in order to send Reply2_C back to Host C. Normally this is the Password: prompt.

Note: See the previous parameter description of Prompt2_A for equivalent Prompt2_C valid values, defaults and other information.

Reply2_C – Second Response Sent to Host (Host C)

Function

The string that the terminal's autologin mechanism sends to Host C after Prompt2_C has been received. Normally this is the user’s password.

Note: See the previous parameter description of Reply2_A for equivalent Reply2_C valid values, defaults and other information.

Prompt3_C – Third Host String (Host C)

Function

The host string that the terminal's autologin must see in order to send Reply3_C back to Host C. Normally this is something distinctive in the host's prompt.

Use Prompt3_C and Reply3_C to issue a command (i.e. - to start a program) once the user has been auto connected to the host.

Note: See the previous parameter description of Prompt3_A for equivalent Prompt3_C valid values, defaults and other information.

Reply3_C – Third Response Sent to Host (Host C)

Function

The string that the terminal's autologin mechanism sends to Host C after Prompt3_C has been received.

Note: See the previous parameter description of Reply3_A for equivalent Reply3_C valid values, defaults and other information.

PortNum_C – Port Number (Host C)

Function

The PortNum_C parameter is used to specify a TELNET port number for connection to Host C during the Autologin process. The PortNum_C is in decimal notation.

Note: See the previous parameter description of Reply3_A for equivalent Reply3_C valid values, defaults and other information.

TermType_D – Type of Terminal (Host D)

Function

Terminal type used in TELNET negotiations when establishing a terminal emulator to Host D connection.

Note: See the previous parameter description of TermType_A for equivalent TermType_D valid values and defaults.

AutoLoginOn_D – Terminal Automatic Login (Host D)

Function

Turn autologin function on for Host D.

When this parameter is set to enabled, the TELNET client used by the terminal will use the autologin parameters and attempt to autologin.

Note: See the previous parameter description of AutoLoginOn_A for equivalent AutoLoginOn_D valid values, defaults and other information.

Host_D – Host Name (Host D)

Function

Specifies the name of Host D that the autologin mechanism will attempt to connect to.

Note: See the previous parameter description of Host_A for equivalent Host_D valid values, defaults and other information.

Prompt1_D – First Host String (Host D)

Function

The host string that the terminal's autologin must see in order to send Reply1_D back to Host D. Normally this is the Login: prompt.

Note: See the previous parameter description of Prompt1_A for equivalent Prompt1_D valid values, defaults and other information.

Reply1_D – First Response Sent to Host (Host D)

Function

The string that the terminal's autologin mechanism sends to Host D after Prompt1_D has been received. Normally this is the user name that is being logged in.

Note: See the previous parameter description of Reply1_A for equivalent Reply1_D valid values, defaults and other information.

Prompt2_D – Second Host String (Host D)

Function

The host string that the terminal's autologin must see in order to send Reply2_D back to Host D. Normally this is the Password: prompt.

Note: See the previous parameter description of Prompt2_A for equivalent Prompt2_D valid values, defaults and other information.

Reply2_D – Second Response Sent to Host (Host D)

Function

The string that the terminal's autologin mechanism sends to Host D after Prompt2_D has been received. Normally this is the user's password.

Note: See the previous parameter description of Reply2_A for equivalent Reply2_D valid values, defaults and other information.

Prompt3_D – Third Host String (Host D)

Function

The host string that the terminal's autologin must see in order to send Reply3_D back to Host D. Normally this is something distinctive in the host's prompt.

Use Prompt3_D and Reply3_D to issue a command (i.e. - to start a program) once the user has been auto connected to the host.

Note: See the previous parameter description of Prompt3-A for equivalent Prompt3_D valid values, defaults and other information.

Reply3_D – Third Response Sent to Host (Host D)

Function

The string that the terminal's autologin mechanism sends to Host D after Prompt3_D has been received.

Note: See the previous parameter description of Reply3_A for equivalent Reply3_D valid values, defaults and other information.

PortNum_D – Port Number (Host D)

Function

The PortNum_D parameter is used to specify a TELNET port number for connection to Host D during the Autologin process. The PortNum_D is in decimal notation.

Note: See the previous parameter description of PortNum_A for equivalent PortNum_D valid values, defaults and other information.

Interface Parameters

2.4GHz Radio Terminals only

The Interface parameters are changed by editing the SOCKET.CFG file. When the Interface option is selected, the user is instructed to manually edit the parameters in the SOCKET.CFG file in the PCTCP directory:

```
Manually edit SOCKET.CFG in the pctcp dir
```

For information on editing the SOCKET.CFG file, please refer to the reference guide for the appropriate LXE computer.

DNS Parameters

2.4GHz Radio Terminals only

The DNS parameters are changed by editing the SOCKET.CFG file. When the DNS option is selected, the user is instructed to manually edit the parameters in the SOCKET.CFG file in the PCTCP directory:

```
Manually edit SOCKET.CFG in the pctcp dir
```

For information on editing the SOCKET.CFG file, please refer to the reference guide for the appropriate LXE computer.

General Parameters

2.4GHz Radio Terminals only

The General parameters are changed by editing the SOCKET.CFG file. When the General option is selected the user is instructed to manually edit the parameters in the SOCKET.CFG file in the PCTCP directory:

```
Manually edit SOCKET.CFG in the pctcp dir
```

For information on editing the SOCKET.CFG file, please refer to the reference guide for the appropriate LXE computer.

Kernel Parameters

2.4GHz Radio Terminals only

The Kernel parameters are changed by editing the SOCKET.CFG file. When the Kernel option is selected the user is instructed to manually edit the parameters in the SOCKET.CFG file in the PCTCP directory:

```
Manually edit SOCKET.CFG in the pctcp dir
```

For information on editing the SOCKET.CFG file, please refer to the reference guide for the appropriate LXE computer.

Radio Menu Option

900MHz and 2.4GHz Radio Terminals

900MHz Radio Terminals

The Radio menu contains parameters pertaining to the Radio Frequency Network. Radio parameters are displayed on multiple screens.

To access the second screen of radio parameters, press the [PageDown] key.

To return to the first screen, press the [PageUp] key. A detailed description of each radio parameter follows.

To return to the Main Menu, press [Esc].

900MHz radios are not available on 2325/MX2.

Note: Configuration menu options and components displayed are unique to the terminal being configured, the current user's access level, the terminal screen display size and installed radio type.

Access Levels

Refer to the "System Passwords" section of this chapter for more information on *access levels*.

Legacy ANSI TE

In the following parameter descriptions, *Legacy ANSI TE* refers to the terminal emulation running on the 1280, 1290, 2280, 2285, and 2315 terminals.

AddrExtend – Extended Address Enabling

Function

The AddrExtend parameter extends the addresses of the Network Controllers and RF terminals by two (2) digits. This parameter works in conjunction with the terminal Station Identifier and the Network Controller Identifier.

When disabled, a four (4) digit address may still be configured, however only a two (2) digit address will be transmitted.

Options

Valid options:	Enabled	(2 or 4 digit address)
	Disabled	2 digit address only)
Default setting:	Enabled	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

ERF

VerboseExcMsgs – Exception Message Mode

Function

The VerboseExcMsgs parameter specifies the operating mode of exception messages. Exception messages may be displayed in two ways: terse mode and verbose mode. Terse mode displays a brief description of the problem. Verbose mode displays a more detailed description of the problem.

Options

Valid options:	Terse, Verbose
Default setting:	Terse

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

NetworkId – Network Identifier

Function

The NetworkId is the RF Network Identifier. This parameter is used to separate co-existing LXE RF networks. The terminal modem may only synchronize with a base station with the same network ID.

Options

Valid options:	1 - FE (Hex)
Default setting:	1

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

RFN

StationID – Station (or Terminal) Identifier

Function

The StationID parameter is used to specify the station (or terminal) identifier. The terminal identifier must be unique for each terminal on the same RF network.

Options

The length of the terminal identifier used by the RF network is determined by the AddrExtend parameter. Refer to the “AddrExtend” section of this chapter for more information on the extended address feature.

When **“AddrExtend” is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	01 - F0	(Hex)	The two right-most digits
Default setting:	01		

Note: Leading zeros are dropped when displaying the address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **“AddrExtend” is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options: 01 - F0 (Hex)

Default setting: 01

Note: When “AddrExtend” is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

TID

RadioType – Radio Type

Function

The RadioType parameter indicates the type of radio being used.

Options

Valid options:	Crystal	
	Etsi 1	(22 pin 12.5KHz)
	Etsi 2	(22 pin 20.0KHz)
	Spread Spectrum	
	Synth1	(30 pin 800MHz synthesized)
	Synth2	(30 pin 1.4GHz synthesized)
	Synth3	(22 pin 450MHz synthesized)
	Synth4	(22 pin 800MHz synthesized)
Default setting:	Spread Spectrum	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

RAD

PriNc1ID – Primary Network Controller #1 Identifier

Function

The PriNc1ID is the RF network identifier (or RF address) for the primary LDS Network Controller (or primary Network Controller #1). This parameter is used by the terminal to communicate with the primary Network Controller configured for an LDS host interface.

Note: This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the “AddrExtend” section of this chapter for more information on the extended address feature.

When **“AddrExtend” is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **“AddrExtend” is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: When “AddrExtend” is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC1 (PRI)

SecNc1ID – Secondary Network Controller #1 Identifier

Function

The SecNc1ID is the RF network identifier (or RF address) for the secondary LDS Network Controller (or secondary Network Controller #1). This parameter is used by the terminal to communicate with the secondary Network Controller configured for an LDS host interface.

Note: This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **"AddrExtend" is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NCI (SEC)

PriNc2ID – Primary Network Controller #2 Identifier

Function

The PriNc2ID is the RF network identifier (or RF address) for the primary 3270 Network Controller (or primary Network Controller #2). This parameter is used by the terminal to communicate with the primary Network Controller configured for a 3270 host interface.

Note: This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits

Default setting: F1

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **"AddrExtend" is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options: F1 - F6 (Hex)

Default setting: F1

Note: When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC2 (PRI)

SecNc2ID – Secondary Network Controller #2 Identifier

Function

The SecNc2ID is the RF network identifier (or RF address) for the secondary 3270 Network Controller (or secondary Network Controller #2). This parameter is used by the terminal to communicate with the secondary Network Controller configured for a 3270 host interface.

Note: This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **"AddrExtend" is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC2 (SEC)

PriNc3ID – Primary Network Controller #3 Identifier

Function

The PriNc3ID is the RF network identifier (or RF address) for the primary ANSI (TELNET) Network Controller (or primary Network Controller #3). This parameter is used by the terminal to communicate with the primary Network Controller configured for an ANSI (TELNET) host interface.

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **"AddrExtend" is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC3 (PRI)

SecNc3ID – Secondary Network Controller #3 Identifier

Function

The SecNc3ID is the RF network identifier (or RF address) for the secondary ANSI (TELNET) Network Controller (or secondary Network Controller #3). This parameter is used by the terminal to communicate with the secondary Network Controller configured for an ANSI (TELNET) host interface.

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: *Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)*

When **"AddrExtend" is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: *When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.*

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC3 (SEC)

PriNc4ID – Primary Network Controller #4 Identifier

Function

The PriNc4ID is the RF network identifier (or RF address) for the primary 5250 Network Controller (or primary Network Controller #4). This parameter is used by the terminal to communicate with the primary Network Controller configured for a 5250 host interface.

Note: This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the "AddrExtend" section of this chapter for more information on the extended address feature.

When **"AddrExtend" is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)

When **"AddrExtend" is disabled,** the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: When "AddrExtend" is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC4 (PRI)

SecNc4ID – Secondary Network Controller #4 Identifier

Function

The SecNc4ID is the RF network identifier (or RF address) for the secondary 5250 Network Controller (or secondary Network Controller #4). This parameter is used by the terminal to communicate with the secondary Network Controller configured for a 5250 host interface.

Note: *This parameter is not used in ANSI Plus. ANSI Plus only supports ANSI hosts (or NC #3).*

Options

The length of the RF network identifier is determined by the AddrExtend parameter. Refer to the “AddrExtend” section of this chapter for more information on the extended address feature.

When **“AddrExtend” is enabled**, the address is a 4 digit (2 byte) hex address.

Valid options:	00 - 10	(Hex)	The two left-most digits
	F1 - F6	(Hex)	The two right-most digits
Default setting:	F1		

Note: *Leading zeros are dropped when displaying the extended address in the configuration utility. (For example, address 0AF6 will be displayed as AF6.)*

When **“AddrExtend” is disabled**, the address is a 2 digit (1 byte) hex address.

Valid options:	F1 - F6	(Hex)
Default setting:	F1	

Note: *When “AddrExtend” is disabled, a 4 digit address may be configured, however only the 2 right-most digits will be used.*

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

NC4 (SEC)

AntennaNum – PC Card Antenna Port Number

Function

The AntennaNum parameter indicates the antenna port the antenna is physically connected to.

Options

Valid options:	1	(indicates the left antenna port)
	2	(indicates the right antenna port)
Default setting:	1	

Note: *There are two versions of the 900MHz Goshawk radio currently in use. One version has two antenna ports and the other has only one port. When using the two port version, the antenna should be connected only to the No. 1 port. Thus, with either version this parameter should be set to a value of 1.*

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

IrqNum – Interrupt Number

Function

The IrqNum parameter specifies the interrupt setting for the PC card radio.

Options

Valid options: 2 - 15

Default setting: 5

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

IoAddress – I/O Port Address

Function

The IoAddress parameter specifies the input/output port address for the PC card radio.

Options

Valid options:	0 - 3FF (Hex)
Default setting:	2F0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

MemBaseSeq – Memory Base Address

Function

The MemBaseSeq parameter specifies the memory base address to request from card services for the PC card radio.

Options

Valid options:	A000, B000, C000, D000, D100, E000, F000
Default setting:	D100

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

MemSize – Memory Size

Function

The MemSize parameter specifies the memory size (in hexadecimal format) to request from card services for the PC card radio.

Options

Valid options:	1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000
Default setting:	2000

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

MaxBadHops – Maximum # of Bad Hops

Function

The MaxBadHops parameter specifies the maximum number of missed time marks before terminal tries to re-sync.

Options

Valid options: 2 - 5

Default setting: 5

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

RHP

MaxAttempts – Maximum Delivery Attempts

Function

The MaxAttempts parameter specifies the maximum number of attempts the TE (Terminal Emulator) makes to deliver a single message.

Options

Valid options: 0 - F (Hex)

Default setting: 4

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

MNA

RoamThreshold – Roaming Threshold

Function

The RoamThreshold parameter specifies the strength (in dBm) that must be reached before ANSI Plus looks for another Radio Frequency Unit (RFU).

Options

The RoamThreshold setting represents a negative number. For example, when set to 30, ANSI Plus starts looking for an alternate RFU when the signal strength falls below -30dBm. Therefore, a higher RoamThreshold setting indicates that ANSI Plus will try to remain connected to the same RFU at a lower signal level than it would when a lower RoamThreshold setting is configured.

Valid options:	30 - 100 dBm
Default setting:	70

Access Level

The RoamThreshold parameter is a read only parameter for level 2 users.

Parameter may only be modified by level 3 users only.

Legacy ANSI TE Equivalent

None

Country – Regulatory Country Code

Function

The Country parameter specifies the regulatory country code the RF modem complies with.

Options

Valid options:	Australia, No Code, USA
Default setting:	USA

Note: *The Australia setting is also used in the Philippines*

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

RCC

BaseMode – Base Mode

Function

The BaseMode parameter determines whether you are configuring a terminal or a base station.

Options

Valid options: Base, Terminal

Default setting: Terminal

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

RxBuffSize – Receive Buffer Size

Function

The RxBuffSize parameter specifies the receive buffer size (in Kilobytes) for the PC card radio.

Options

Valid options: 128, 256, 512, 1024 kilobytes

Default setting: 256

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

TxBuffSize – Transmit Buffer Size

Function

The TxBuffSize parameter specifies the transmit buffer size (in Kilobytes) for the PC card radio.

Options

Valid options: 128, 265, 215, 1024 kilobytes

Default setting: 128

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

SlotTime – Slot Time

Function

The SlotTime parameter indicates the slot time (the length of a slot) in milliseconds.

Options

Valid options:	9 - 33 msec
Default setting:	9

Access Level

The SlotTime parameter is a **read only** parameter that may be viewed but not modified.

Parameter may only be viewed by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

NumberOfSlots – Number of Slots

Function

The NumberOfSlots parameter indicates the number of slots per hop.

Options

Valid options:	5 - 15 slots per hop
Default setting:	10

Access Level

The NumberOfSlots parameter is a **read only** parameter that may be viewed but not modified.

Parameter may only be viewed by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

HopSequence – Hop Sequence

Function

The HopSequence parameter indicates the hop sequence setting.

Options

Valid options:	1 - 52
Default setting:	1

Access Level

The HopSequence parameter is a **read only** parameter that may be viewed but not modified.

Parameter may only be viewed by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

SequenceGrp – Sequence Group

Function

The SequenceGrp parameter indicates the sequence group setting.

Options

Valid options: 1 - 5

Default setting: 1

Access Level

The SequenceGrp parameter is a **read only** parameter that may be viewed but not modified.

Parameter may only be viewed by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

FrontPorchDelay – Front Porch Delay

Function

The FrontPorchDelay parameter specifies the time (in milliseconds) allowed for the radio to stabilize before transmitting data within each slot time.

Options

Valid options: 0 - 15 msec

Default setting: 3

Access Level

The FrontPorchDelay parameter is a **read only** parameter that may be viewed but not modified.

Parameter may only be viewed by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

SFP

PllSettleTime – Phase Locked Loop Settle

Function

The PllSettleTime parameter specifies the phase locked loop settle time, which is the time allowed for the radio to change frequencies before the next hop.

Options

Valid options:	1 - 6 msec
Default setting:	3

Access Level

The PllSettleTime parameter is a **read only** parameter for **level 2** users.

Parameter may only be modified by level 3 users only.

Legacy ANSI TE Equivalent

PST

MaxRandomDelay – Maximum # of Slots

Function

The MaxRandomDelay parameter specifies the maximum number of slots to wait for transmit.

Options

Valid options:	10 - 50
Default setting:	20

Access Level

The MaxRandomDelay parameter is a **read only** parameter for **level 2** users.

Parameter may only be modified by level 3 users only.

Legacy ANSI TE Equivalent

None

ModemFile – Modem File Name

Function

The ModemFile parameter is used to specify the modem file that is to be loaded by the 900MHz spread spectrum radio driver (LXE900SS.EXE).

Options

ModemFile is a user defined parameter which can be from 1 to 12 alphanumeric characters in length. Only the filename can be entered into this field, no path statements are necessary. The file that is specified must be located in the same directory as the radio driver (i.e. LXE900SS.EXE).

Valid options:	[user defined]	(up to 12 alphanumeric characters)
Default setting:	lxe900ss.hex	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Driver Setting (6400, 6500, 6700 System 2.4GHz Radio)

2.4GHz radio configuration parameters are set by editing the NET.CFG file. When the Radio Config option is selected, the user is instructed to manually edit the NET.CFG file in the PCTCP directory:

```
Manually edit NET.CFG in the PCTCP dir
```

For information on editing the NET.CFG file, please refer to the reference guide for the appropriate LXE computer.

Platform Menu Option

900MHz and 2.4GHz Radio Terminals

The Platform parameter menu contains system parameters as well as miscellaneous parameter settings. The Platform parameter menu is a multiple paged menu.

To access the second screen of Platform parameters, press the [PageDown] key.

To return to the first screen, press the [PageUp] key. A detailed description of each platform parameter follows.

To return to the Main Menu, press [Esc].

Note: *Configuration menu options and components displayed are unique to the terminal being configured, the current user's access level, and the terminal screen display size.*

Access Levels

Refer to the "System Passwords" section of this chapter for more information on *access levels*.

Legacy ANSI TE

In the following parameter descriptions, *Legacy ANSI TE* refers to the terminal emulation running on the 1280, 1290, 2280, 2285, and 2315 terminals.

BeeperVolume – Beeper Volume

Function

The BeeperVolume parameter specifies the beeper volume for the terminal. This parameter allows you to increase or decrease the beeper volume. Each terminal may react differently to the beeper settings; some settings may appear to give the same audible tone. The volume ranges from inaudible (lower numbers) to loud (higher numbers).

Options

Valid options:	0	(Beeper off)
	1 - 15	
Default setting:	7	

Divide this number by 2 to get the equivalent volume setting to be used in the Beep Pattern private sequence command. That is, a setting of 12 in the configuration utility is equivalent to 6 using the beep pattern private sequence.

Access Level

Parameter may only be modified by users with access levels of 1, 2 or 3.

Legacy ANSI TE Equivalent

AAL

BeeperFrequency – Beeper Frequency

Function

Change the frequency of the beeper.

Options

Valid options:	500 - 3000 Hz
Default setting:	1500

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

InitialScrlDly – Initial Scroll Delay

Function

The InitialScrlDly parameter determines how long (in milliseconds) a key must be held down (depressed) before it repeats. A setting of 0 disables the repeat feature on MX1, MX3 and VX1 computers.

Options

Valid options:	0	(no repeat)
	250, 500, 750, 1000	(delay in msec)
Default setting:	500	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

ISD

RepeatScrlDly – Repeat Scroll Delay

Function

The RepeatScrlDly parameter specifies the speed (in cps - characters per second) at which the key is repeated or the cursor moves across the display.

Options

Valid options:	2, 8, 16, 24, 30 cps
Default setting:	16

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

RSD

Com1BaudRate – COM1 Baud Rate

Function

The Com1BaudRate parameter specifies the baud rate setting for communication port 1 (COM1).

Options

Valid options:	600, 1200, 2400, 4800, 9600, 19200, 38400 bps
Default setting:	9600

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com1DataBits – COM1 Data Bits

Function

The Com1DataBits parameter specifies the data bits setting for communications port 1 (COM1).

Options

Valid options:	Eight, Seven
Default setting:	Eight

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com1Parity – COM1 Parity

Function

The Com1Parity parameter specifies the parity setting for communications port 1 (COM1).

Options

Valid options:	Even, Odd, None
Default setting:	None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com1StopBits – COM1 Stop Bits

Function

The Com1StopBits parameter specifies the stop bits setting for communications port 1 (COM1).

Options

Valid options:	1 or 2
Default setting:	1

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com1HalfDuplex – COM1 Half Duplex

Function

The Com1HalfDuplex parameter determines whether or not characters received on communications port 1 (COM1) are echoed back to COM1. When Echo is selected, the characters received on COM1 are echoed back out to COM1. When No Echo is selected, the characters are not echoed.

Options

Valid options: Echo, No Echo

Default setting: No Echo

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

HDX

Com1Xon – COM1 Software Flow Control

Function

The Com1Xon parameter enables or disables the use of software flow control (XON/XOFF) for communications port 1 (COM1).

Options

Valid options: Disable, Enable

Default setting: Disable

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

AHS

Com1HwFlowCtrl – COM1 Hardware Flow Control

Function

The Com1HwFlowCtrl parameter specifies the hardware flow control setting for communications port 1 (COM1).

Options

Valid options: Off
RTS/CTS
DTR/CTS

Default setting: Off

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

DTR

Com1TimeOut – COM1 Time Out

Function

The Com1TimeOut parameter enables or disables the time out feature for communications port 1 (COM1). When enabled (On), a gap of more than 50ms between characters received on COM1 terminates the message. When disabled (Off), the message is not terminated by a time gap between characters.

Options

Valid options: Off, On

Default setting: Off

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

ATO

Com1Power – COM1 RS-232 Power

Function

Determines if power supplied to the COM1 RS-232 Port is On or Off. Not available on 2325/MX2.

Options

Valid options: On
Off

Default setting: On

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Com1PassTermChar – COM1 Pass Termination Character

Function

The Com1PassTermChar parameter determines whether or not communications port 1 (COM1) will propagate (transmit to host) the termination character. Termination characters are: 00, 0A, 0D.

Options

Valid options: Don't Propagate, Propagate

Default setting: Propagate

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Com2BaudRate – COM2 Baud Rate

Function

The Com2BaudRate parameter specifies the baud rate setting for communication port 2 (COM2). Not available on 2325/MX2.

Options

Valid options: 600, 1200, 2400, 4800, 9600, 19200, 38400 bps

Default setting: 9600

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Com2DataBits – COM2 Data Bits

Function

The Com2DataBits parameter specifies the data bits setting for communications port 2 (COM2). Not available on 2325/MX2.

Options

Valid options:	Eight, Seven
Default setting:	Eight

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com2Parity – COM2 Parity

Function

The Com2Parity parameter specifies the parity setting for communications port 2 (COM2). Not available on 2325/MX2.

Options

Valid options:	Even, Odd, None
Default setting:	None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com2StopBits – COM2 Stop Bits

Function

The Com2StopBits parameter specifies the stop bits setting for communications port 2 (COM2). Not available on 2325/MX2.

Options

Valid options:	1 or 2
Default setting:	1

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com2HalfDuplex – COM2 Half Duplex

Function

The Com2HalfDuplex parameter determines whether or not characters received on communications port 2 (COM2) are echoed back to COM2. When Echo is selected, the characters received on COM2 are echoed back out to COM2. When No Echo is selected, the characters are not echoed. Not available on 2325/MX2.

Options

Valid options:	Echo, No Echo
Default setting:	No Echo

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

HDX

Com2Xon – COM2 Software Flow Control

Function

The Com2Xon parameter enables or disables the use of software flow control (XON/XOFF) for communications port 2 (COM2). Not available on 2325/MX2.

Options

Valid options:	Disable, Enable
Default setting:	Disable

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

AHS

Com2HwFlowCtrl – COM2 Hardware Flow Control

Function

The Com2HwFlowCtrl parameter specifies the hardware flow control setting for communications port number 2 (COM2). Not available on 2325/MX2.

Options

Valid options:	Off RTS/CTS DTR/CTS
Default setting:	Off

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

DTR

Com2TimeOut – COM2 Time Out

Function

The Com2TimeOut parameter enables or disables the time out feature for communications port 2 (COM2). When enabled (On), a gap of more than 50ms between characters received on COM2 terminates the message. When disabled (Off), the message is not terminated by a time gap between characters. Not available on 2325/MX2.

Options

Valid options:	Off, On
Default setting:	Off

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

ATO

Com2Power – COM2 RS-232 Power

Function

Determines if power supplied to the COM2 RS-232 Port is On or Off. Not available on 2325/MX2.

Options

Valid options:	On Off
Default setting:	On

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Com2PassTermChar – COM2 Pass Termination Character

Function

The Com2PassTermChar parameter determines whether or not communications port number 2 (COM2) will propagate (transmit to host) the termination character. Termination characters are: 00, 0A, 0D. Not available on 2325/MX2.

Options

Valid options:	Don't Propagate, Propagate
Default setting:	Propagate

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

ConvBarCode1From – Convert Barcode

Function

The ConvBarCode1From parameter must be used in conjunction with the parameter "ConvBarCode1To." The combination of these two parameters provides the capability of converting any identified byte of a barcode scan to any identified one or two byte sequence. It is provided to accommodate certain host computers that require non-standard frame termination characters. For those certain host computers, this parameter needs to be used only if barcodes are being scanned while ANSI Plus is **not** operating in "local echo" mode.

Programming this parameter requires the use of "hat encoded" characters. A table is provided (see the figures titled "Hat Encoded Characters" at the end of this chapter) to assist you in the selection of the correct hat encoded character(s) to fulfill your requirements. Refer to the "Hat Encoded Characters" table during the following example.

Example

This is an example of indicating that you desire to change any "null" characters (ASCII hex 00), resulting from the scan, to some other character: Referring to the "Hat Encoded Characters" table, the hat encoded character that represents the "null" (00 hex) is ^@. Thus, the value that should be installed in this parameter is: ^@. When ANSI Plus encounters the null character in a barcode scan it will be converted into the character(s) that you program for the parameter "ConvBarCode1To."

Options

Valid options:	Any of the hat encoded characters in Figures titled "Hat Encoded Characters" at the end of this chapter.
Default setting:	None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ConvBarCode1To – Convert Barcode

Function

The ConvBarCode1To parameter must be used in conjunction with the parameter "ConvBarCode1From." The combination of these two parameters provides the capability of converting any identified byte of a barcode scan to any identified one or two byte sequence. It is provided to accommodate certain host computers that require non-standard frame termination characters. For those certain host computers, this parameter needs to be used only if barcodes are being scanned while ANSI Plus is **not** operating in "local echo" mode.

Using hat encoded characters, indicate the character, or characters, that will replace the character programmed in the previous parameter (ConvBarCode1From). The following example is presented to demonstrate how to program a "carriage return" plus a "line feed" as the replacement characters:

Example

The ASCII carriage return is represented by the hex value 0D. The ASCII line feed is represented by the hex value 0A. Referring to the Figure titled "Hat Encoded Characters", the hat encoded equivalent for 0D (carriage return) is ^M. The hat encoded equivalent for 0A (line feed) is ^J. Thus the value that should be programmed into this parameter is ^M^J.

Options

Valid options:	Any of the hat encoded characters in Figure titled "Hat Encoded Characters" at the end of this chapter.
Default setting:	None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ConvBarCode2From – Convert Barcode

Function

The ConvBarCode2From parameter must be used in conjunction with the parameter “ConvBarCode2To.” The combination of these two parameters provides the capability of converting any identified byte of a barcode scan to any identified one or two byte sequence. It is provided to accommodate certain host computers that require non-standard frame termination characters. For those certain host computers, this parameter needs to be used only if barcodes are being scanned while ANSI Plus is **not** operating in “local echo” mode.

*Note: Refer to the discussion on the parameter “ConvBarCode1From” for details on programming this parameter. The ConvBarCode2From parameter provides the capability of converting an **additional** terminating character resulting from a barcode scan.*

Options

Valid options: Any of the hat encoded characters in Figure titled “Hat Encoded Characters” at the end of this chapter.

Default setting: None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ConvBarCode2To – Convert Barcode

Function

The ConvBarCode2To parameter must be used in conjunction with the parameter “ConvBarCode2From.” The combination of these two parameters provides the capability of converting any identified byte of a barcode scan to any identified one or two byte sequence. It is provided to accommodate certain host computers that require non-standard frame termination characters. For those certain host computers, this parameter needs to be used only if barcodes are being scanned while ANSI Plus is **not** operating in “local echo” mode.

*Note: Refer to the discussion on the parameter “ConvBarCode1To” for details on programming this parameter. The ConvBarCode2To parameter provides the capability of converting an **additional** terminating character resulting from a barcode scan.*

Options

Valid options: Any of the hat encoded characters in Figure titled “Hat Encoded Characters” at the end of this chapter.

Default setting: None

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCPreStrip – Barcode Head Character Strip All

Function

Number of characters to remove and discard from the beginning of all barcode data.

Note: The specified number of characters are stripped from all types of scanned barcodes.

Options

Valid Options: 0 (disabled)
1 - 20

Default setting: 0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCPostStrip – Barcode Tail Character Strip All

Function

Number of characters to remove and discard from the end of all barcode data.

Note: The specified number of characters are stripped from all types of scanned barcodes.

Options

Valid Options: 0 (disabled)
1 - 20

Default setting: 0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

CodeID – Barcode Symbology Identification

Function

CodeID allows the computer to receive the barcode ID character as part of the scanned data.

By entering the corresponding Code ID Character in BCSymbologyID_1, for example, a user can then enter the number of characters to strip from the head (Sym_1PreStrip) or tail (Sym_1PostStrip) of the barcode data. Up to six different symbologies can be configured for character stripping. Valid for VX4 and 2325/MX2 only.

Note: *The appropriate barcode must be scanned to enable "Transmit Symbol Code ID Character" (Symbol) or "Enable Transmission of Symbology Identifiers" (PSC). For LXE computers with an integrated barcode scanner, please refer to the computer's reference guide for the appropriate barcode. For LXE computers with a tethered barcode scanner, please refer to the manual delivered with the scanner for the appropriate barcode.*

Note: *If using a Symbol scanner, be sure to enable the Symbol Code ID Character, not the AIM Code ID Character.*

Barcode Type	**Symbol Scanner Code ID Character**	**PSC Scanner Code Identifier**
Code 39	A	a
Code 128	C	f
UPC/EAN	E	
Codabar	F	h
Code 93	G	
Code 11	H	
Interleaved 2 of 5	I	b
MSI Plessy	M	
D2 of 5, IATA 2 of 5	S	
Code 39 Trioptic, Bookland EAN	X	
Code S2 of 5		c
UPC/EAN/JAN		d

Options

Valid Options: Enabled, Disabled

Default setting: Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCSymbologyID_1 – Barcode Symbology 1

Function

Character that identifies the first barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options:	A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h
Default setting:	z

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Sym_1PreStrip – Barcode Head Character Strip 1

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_1. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Sym_1PostStrip – Barcode Tail Character Strip 1

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_1. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options: 0 (disabled)
1 - 20

Default setting: 0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

BCSymbologyID_2 – Barcode Symbology 2

Function

Character that identifies the second barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options: A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h

Default setting: y

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Sym_2PreStrip – Barcode Head Character Strip 2

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_2. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options: 0 (disabled)
1 - 20

Default setting: 0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_2PostStrip – Barcode Tail Character Strip 2

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_2. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options: 0 (disabled)
1 - 20

Default setting: 0

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCSymbologyID_3 – Barcode Symbology 3

Function

Character that identifies the third barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options:	A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h
Default setting:	x

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_3PreStrip – Barcode Head Character Strip 3

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_3. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_3PostStrip – Barcode Tail Character Strip 3

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_3. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCSymbologyID_4 – Barcode Symbology 4

Function

Character that identifies the fourth barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options:	A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h
Default setting:	w

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_4PreStrip – Barcode Head Character Strip 4

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_4. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_4PostStrip – Barcode Tail Character Strip 4

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_4. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCSymbologyID_5 – Barcode Symbology 5

Function

Character that identifies the fifth barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options:	A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h
Default setting:	v

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_5PreStrip – Barcode Head Character Strip 5

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_5. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_5PostStrip – Barcode Tail Character Strip 5

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_5. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

BCSymbologyID_6 – Barcode Symbology 6

Function

Character that identifies the sixth barcode symbology for character stripping. Valid for VX4 and 2325/MX2 only.

Note: The scanner must be configured to identify the barcode symbology. Please see "CodeID" for more information.

Options

Valid Options: A, C, E, F, G, H, I, M, S, X, a, b, c, d, f, h

Default setting: u

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_6PreStrip – Barcode Head Character Strip 6

Function

Number of characters to remove and discard from the beginning of barcode data only when the barcode is of the type identified by BCSymbologyID_6. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Sym_6PostStrip – Barcode Tail Character Strip 6

Function

Number of characters to remove and discard from the end of barcode data only when the barcode is of the type identified by BCSymbologyID_6. The additional character identifying the symbology is also removed. Valid for VX4 and 2325/MX2 only.

Options

Valid Options:	0	(disabled)
	1 - 20	
Default setting:	0	

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

RFLogFS – RF Log File Size

Function

Size of RF log file in kilobytes. The RFLogFS parameter specifies how big (in K) the RF log file can become when the RF data stream is being logged. This parameter should be used by LXE engineering only.

Note *For use by LXE Engineering only.*

Options

Valid options:	1 - 9999
Default setting:	20

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com1LogFS – COM1 Log File Size

Function

Size of Com1 log file in kilobytes. The Com1LogFS parameter specifies how big (in K) the log file can become when the comm port 1 data stream is being logged. This parameter should be used by LXE engineering only.

Note *For use by LXE Engineering only.*

Options

Valid options:	1 - 9999
Default setting:	20

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

Com2LogFS – COM2 Log File Size

Function

Size of Com2 log file in kilobytes. The Com2LogFS parameter specifies how big (in K) the log file can become when the comm port 2 data stream is being logged. This parameter should be used by LXE engineering only. Not available on 2325/MX2.

Note For use by LXE Engineering only.

Options

Valid options:	1 - 9999
Default setting:	20

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

ScreenSize – Font Size Selections

Function

This option allows the user to configure the display fonts on DOS computers running the terminal emulation program. These settings are in effect only when the TE program is running. They do not supercede the font settings at the DOS prompt or on a batch unit.

Options

MX1

Valid options: 8 rows x 24 cols
10 rows x 20 cols
16 rows x 20 cols
16 rows x 24 cols
20 rows x 20 cols

Default setting: 20 rows x 20 cols

2325 / MX2

Valid options: 9 rows x 20 cols
16 rows x 20 cols

Default setting: 9 rows x 20 cols

MX3

Valid options: 8 rows x 40 cols
16 rows x 40 cols
12 rows x 80 cols
16 rows x 80 cols
25 rows x 80 cols

Default setting: 16 rows x 80 cols

1380 / VX1

Valid options: 8 rows x 40 cols
16 rows x 40 cols
12 rows x 80 cols
16 rows x 80 cols
25 rows x 80 cols

Default setting: 12 rows x 80 cols

1390 / VX2 / VX4

Valid options: 16 rows x 40 cols
25 rows x 40 cols
25 rows x 80 cols

Default setting: 25 rows x 80 cols

Access Level

This is a level 2 **read only** parameter that may be viewed but not modified by a user with an access level of 2. Parameter may be modified by a user with an access level of 3.

Legacy ANSI TE Equivalent

None

VT320NavKeys – VT320 Navigation Keys

Function

The VT320NavKeys allows you to enable VT320 escape sequences for navigation keys. When enabled, the table below shows the escape sequence sent to the host for each of the navigation keys. Available on VX4 and 2325/MX2 only.

Key	Sequence Sent to Host
Page-Up	`Esc[5~`
Page-Down	`Esc[6~`
Home	`Esc[4~`
End	`Esc[1~`

Options

Valid options:	Enabled, Disabled
Default:	Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

FnKeySize – Long Function Key Definitions

Function

The FNKeySize parameter defines the maximum length (in bytes) of the Function Key definitions. Valid for 2325/MX2 only.

Options

Valid options:	32, 100
Default setting:	32

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

NormIntensity – Normal Video Intensity Setting

Function

The NormIntensity parameter setting affects the foreground (normal video) intensity level on an LXE terminal's electroluminescent screen. Increasing the intensity makes bolded text characters appear darker than any surrounding normal text characters. This parameter has no effect on characters on a transmissive screen.

Options

Valid options:	High, MedH, MedL, Low
Default setting:	High

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ForceEcho – Force Echo

Function

Force TE to "Local Echo" when in char mode.

Options

Valid Options:	Enabled, Disabled
Default setting:	Disabled

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

OutInDelay – Shared COM Port Delay

Function

This option allows the user to set a delay before switching occurs between shared In/Out ports on MX1 and MX3 computers. There may be no delay, a delay for a period of time or a pause for user input. This delay only applies when the computer is docked.

If the user selects 0, there is no delay before the switch.

If the user selects a between 1 and 30 (inclusive), the delay is the specified number of seconds. For example, selecting 5 results in a delay of 5 seconds.

If the user selects 31, the terminal displays a pop-up message and waits for a keypress before switching back to the In port.

As an example, if an MX3 is configured with the Com2 RS-232 port for Output and the Com2 IR port for Input, the computer will keep the Com2 IR port active until it receives a print command. The Com2 RS-232 port is then enabled. When the print is finished, the MX3 will delay for a time period or user input, based on the OutInDelay value, before re-enabling the Com2 IR port for input.

Note: This delay is non-blocking. The TE continues to handle RF messages, screen updates, etc. while the switching delay is underway.

Options

Valid options:	0	(no delay)
	1 - 30	(sets the length of the delay, in seconds)
	31	(requires user input before switching)
Default setting:	0	

Access Level

This is a level 2 **read only** parameter that may be viewed but not modified by a user with an access level of 2. Parameter may be modified by a user with an access level of 3.

Legacy ANSI TE Equivalent

None

Related Parameters

DockedIO, NormalIO

SafeKeys – Disable Hotkeys

Function

When SafeKeys is set to Yes, the following hotkeys are disabled:

Alt-H	Help	
Alt-M	Stored Forms	
Alt-K	Function Key Edit	
Alt-E	Enable Local Echo	(2325/MX2 only)
Alt-G	Disable Local Echo	(2325/MX2 only)

Options

Valid options: No, Yes

Default setting: No

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

SuppressPrompt – Suppress Prompt

Function

This parameter provides the capability of restricting access to the DOS prompt to only privileged individuals with knowledge of a special "hot keys sequence." Also, the terminal will not display the TELNET prompt when this parameter is set to "suppressed."

If this parameter is set to **not** suppressed, when the terminal operator leaves the ANSI Plus program the DOS prompt is displayed allowing the operator to execute DOS commands. If this parameter is set to suppressed, attempted exit from ANSI Plus results in the display of a message advising the operator of an action to take. The default message states "Disconnected. Press ENTER to continue." Pressing the [Enter] key causes a loop program to execute which results in the terminal returning to the ANSI Plus program.

The default message can be modified/changed by editing the MCH.INI file which is located in the ANSIPLUS directory.

The hot key sequence to access the DOS prompt instead of looping back to ANSI Plus is: [control][d] [control][g] (that is the control key and the d key followed by the control key and the g key). This hot key sequence can be changed by editing the MCH.INI file. The default key sequence is indicated in the file as: ^d^g.

Options

Valid options: Suppressed, Not Suppressed

Default setting: Not Suppressed

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent

None

ForceDOSSeq1 – Exit ANSI Plus

Function

The ForceDOSSeq1 . allows you to exit ANSI Plus when the SuppressPrompt parameter is set to Suppressed.

This parameter may be used in conjunction with ForceDOSSeq2.

Options

Valid options:	Any character
Default setting:	D

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

ForceDOSSeq2 – Exit ANSI Plus

Function

The ForceDOSSeq2 allows you to exit ANSI Plus when the SuppressPrompt parameter is set to Suppressed.

This parameter may be used in conjunction with ForceDOSSeq1.

Options

Valid options:	Any character
Default setting:	G

Access Level

Parameter may only be modified by a user with an access level of 2 or 3.

Legacy ANSI TE Equivalent None

Exit With Save Option

The Exit with Save option on the main menu allows the user to save the current configuration settings and exit the utility.

To return to the System Menu press the [Esc] key.

No Press the [N] key and then the [Enter] key to ignore the Save command and remove the Save and Exit text box from the screen display. Parameters are not changed. You are returned to the Main Menu.

- or -

Ignore At the "[N]" prompt, press the [Enter] or [Esc] key to remove the text box from the screen display. You are returned to the Main Menu.

- or -

Yes Press the [Y] key and then the [Enter] key to save changes made to configuration parameters. Configuration files are updated upon exit. See the sub-sections titled "Change 900MHz Parameters" and "Change 2.4GHz Parameters" in this Section.

Press the [Enter] key to save changes made to configuration parameters and restart the terminal. The terminal will need to be turned off and then on again for new Protocol or Radio parameter values to take effect. Configuration files are updated upon exit. See the sub-sections titled "Change 900MHz Parameters" and "Change 2.4GHz Parameters" in this Section.

Change 900MHz Parameters

When parameters in a 900MHz terminal are changed or reset to default values, the LXE.INI file is changed.

Change 2.4GHz Parameters

When parameters in a 2.4GHz device are changed or reset to default values, the LXE.INI, PROTOCOL.INI, PCTCP.INI, the radio driver and the FTP stack files are changed.

Exit Without Save Option

The Exit without Save option on the main menu allows the user to exit the configuration utility without saving any changes made to the current settings.

To return to the System Menu press the [Esc] key.

Yes	Press the [Y] key and then the [Enter] key to exit the Configuration program. Configuration files are not changed.
	- or -
No	Press the [N] key and then the [Enter] key to ignore the Exit command and remove the Exit text box from the screen display. System values are not changed and you are returned to the Main Menu.
	- or -
Ignore	At the “[N]” prompt, press the [Esc] key twice to remove the Exit box from the screen display. System values are not changed and you are returned to the Main Menu.

Hat Encoded Characters

Used with the following parameters: EnterKeyChars, ConvBarCode1From, ConvBarCode1To, ConvBarCode2From, ConvBarCode2To.

Desired ASCII hex	Hat Encoded
00	^@
01	^A
02	^B
03	^C
04	^D
05	^E
06	^F
07	^G
08	^H
09	^I
0A	^J
0B	^K
0C	^L
0D	^M
0E	^N
0F	^O
10	^P
11	^Q
12	^R
13	^S
14	^T
15	^U
16	^V
17	^W
18	^X
19	^Y
1A	^Z
1B	^[
1C	^\\
1D	^]
1E	^^
1F	^_ Underscore
80	~^@
81	~^A
82	~^B
83	~^C
84	~^D
85	~^E
86	~^F
87	~^G
88	~^H
89	~^I
8A	~^J
8B	~^K
8C	~^L
8D	~^M
8E	~^N
8F	~^O
90	~^P
91	~^Q
92	~^R
93	~^S
94	~^T
95	~^U
96	~^V
97	~^W
98	~^X
99	~^Y
9A	~^Z
9B	~^[
9C	~^\\
9D	~^]
9E	~^^
9F	~^_ Underscore
A0	~ Tilde and Space
A1	~!
A2	~"
A3	~#
A4	~$
A5	~%
A6	~&
A7	~'
A8	~(
A9	~)
AA	~*
AB	~+
AC	~,

Figure 3-3 Hat Encoded Characters - Hex 00 through AC

Desired ASCII hex	Hat Encoded
AD	~- Tilde and Dash
AE	~. Tilde and Period
AF	~/
B0	~0 Tilde and Zero
B1	~1
B2	~2
B3	~3
B4	~4
B5	~5
B6	~6
B7	~7
B8	~8
B9	~9
BA	~:
BB	~;
BC	~<
BD	~=
BE	~>
BF	~?
C0	~@
C1	~A
C2	~B
C3	~C
C4	~D
C5	~E
C6	~F
C7	~G
C8	~H
C9	~I
CA	~J
CB	~K
CC	~L
CD	~M
CE	~N
CF	~O
D0	~P
D1	~Q
D2	~R
D3	~S
D4	~T
D5	~U

Desired ASCII hex	Hat Encoded
D6	~V
D7	~W
D8	~X
D9	~Y
DA	~Z
DB	~[
DC	~\\
DD	~]
DE	~\^
DF	~_ Underscore
E0	~`
E1	~a
E2	~b
E3	~c
E4	~d
E5	~e
E6	~f
E7	~g
E8	~h
E9	~i
EA	~j
EB	~k
EC	~l
ED	~m
EE	~n
EF	~o
F0	~p
F1	~q
F2	~r
F3	~s
F4	~t
F5	~u
F6	~v
F7	~w
F8	~x
F9	~y
FA	~z
FB	~{
FC	~|
FD	~}
FE	~~
FF	~^?

Figure 3-4 Hat Encoded Characters - Hex AD through FF

Table of Contents

CHAPTER 4 TERMINAL OVERVIEW 4-1

Illustrations

LXE®

An EMS Technologies Company

Chapter 4

Terminal Overview

LXE ANSI Plus Terminal

LXE (DOS) ANSI Plus terminals receive and transmit the standard ANSI data stream. They provide block mode communication connection to DEC, Hewlett-Packard, IBM (AIX), and other computer systems.

The following is a list of features for the LXE ANSI Plus terminal:

- Function keys that can be remotely defined and downloaded to the terminal (programmable function keys) [*900MHz radios only*]
- Single-high/double-wide, double-high/double-wide and single-high/single-wide character selection
- Hot key switching between primary and secondary Network Controllers [*900MHz radios only*]
- Keyboard help screen
- Emulates VT220 terminals
- Operates in 7-bit and 8-bit environments
- Contains a 25 line x 80 column virtual screen
- Supports window manager (a proprietary feature)
- Supports bar code entry
- The RS-232 port can be used for input or output
- Provides status line support
- Displays or executes control sequences, which help in debugging applications
- Supports three kinds of communication modes: character, scroll, and block
- Supports local field edit
- Supports remotely programmable parameters

Features unique to a radio type are tagged with the radio type name, e.g. 900MHz or 2.4GHz. Features that are not tagged with a specific radio type are supported by all terminals.

Block Mode Communication

Block mode communication is a communication mode that enables the terminal to send an entire block of data to the host at one time. (Refer to section titled “Sending the Terminal Screen to the Host” found later in this chapter for more details.) Block mode greatly reduces RF traffic by sending data in this way. Block mode minimizes RF overhead better than the other modes. Thus, block mode supports more users on an RF channel and allows faster response time.

Local echo is a process that allows each character to display locally at the terminal as the terminal operator presses the keys. The host does not see these characters until the operator presses the [Enter] key.

Note: In most cases, operations using programming tools that include screen painters cannot use block mode communication. Screen painters or designers usually come with programming tools such as Oracle, Ingress, Informix, and Powerhouse, which normally use character mode communications.

Block Mode Setup

Refer to the following rules for setting up block mode communication:

- The programmer must include the following commands in the command string for setting up block mode:

  ```
  ESC[12l      (turn local echo on in SRM)
  ESC[=1l      (turn block mode on in LXELM)
  ```

  Note: If local echo is off in send-receive mode (SRM), the terminal operates in character mode while the scroll or block mode setting in LXELM is ignored.

- The programmer must send formatted screens to the terminal from the host.

Scroll Mode Communication

Scroll mode communication is a communication mode where the terminal allows data entry only on one line. The terminal operator can view the rest of the form, but the cursor returns to the line the host placed the cursor in whenever the operator tries to enter data.

Scroll mode has the following advantages:

- Provides local protection of unformatted text.
- Protects all information on the screen except the command line, which is the line where the host positions the cursor. Character positions to the left of the cursor's current position are also protected.
- Enables the terminal operator to move the cursor around the screen with the arrow keys. If the operator moves the cursor, with the arrow keys, away from the command line and then tries to enter data, the cursor returns to the command line before accepting the data.

Note: Applications that edit individual characters do not perform well in scroll mode.

Scroll Mode Setup

Refer to the following rules for setting up scroll mode communications:

- The programmer must include the following commands in the command string for setting up scroll mode:

```
ESC[12l     (turn on local echo in SRM)
ESC[=1h     (turn on scroll mode in LXESM)
```

- The programmer must construct a form consisting of one unprotected field on one line.

Terminology

Term	Definition
Local input	Either keyboard, RS-232 or bar code input.
Command line	The only line on the screen where the user enters local input. The host determines which line this is by where it places the cursor. The command line is always 80 columns wide though the user may only be able to enter data on a portion of it.
Command start	The first column on the command line where the user enters local input. Screen cells to the left of command start are protected from alteration by the user. The column is defined by the column where the host places the cursor.
Command end	The last column on the command line where the user entered local input. This column starts as the same column as command start, then automatically expands as the user enters local input.
Active position	The current cursor position on the command line where the user positioned the cursor. The active position represents the location at which the next input char will go. This location is always somewhere between command start and command end (see below).
Active area	The area on the command line between command start and command end.

Initialization

The ANSI Plus TE powers up in scroll mode and stays that way until the host changes the TE's mode. The host may specify that scroll, block or char mode go into effect at the next power up by saving the appropriate modes to disk.

Host Setup

The command line, command start, command end and active position values are reset with every host command. For example, if the host sends a CUP 5 10 followed by the literal string “Enter name>“, the TE will define line 5 as the command line and (5,21) as command start (col 21 is the col after the “>“ that the host sent). This means that the TE user cannot enter data to the left of (5,21) or past (5,80). At this point the user will enter data and then press [Enter]. If the host then positions the cursor to (6,1) and prints “serial #”, the new command line is line 6 and command start is (6,9).

Data Entry

After the host sets up the command line, the TE user may enter data. Before the user enters any data, command start = command end = active position. If the cursor is not at the active position, it will jump there when the user presses the first key or scans the first char.

The simplest case is when the TE user starts typing data (without moving the cursor around with the arrow keys). In this case, when the user presses the first char, both command end and the active position increase by 1. When the second char is entered, they both increase by one again. This continues until the user reaches the right margin of the virtual screen or presses [Enter].

When [Enter] is pressed all screen cells between command start and command end are sent to the host. The user may use the left and right arrow keys (see below) to move around in the command line before pressing [Enter].

VT220 Key Equivalents

An LXE terminal running the ANSI Plus TE emulates VT220 terminals. The VTXXX keypresses and functions supported by LXE's ANSI Plus terminal emulation program are described in this reference guide.

Active Keys

The following keys affect scroll mode.

Key(s)	Function
[Del]	Delete Character - The cursor will jump back to the active position if necessary. If the active position is to the right of command start, the TE will backspace followed by a DCH 1. The active position will move to the left by one and so will command end (the active area will shrink by one).
[Alt]+[L]	Clear Screen - The cursor will jump back to the command line if necessary. The command end, active position and the cursor will be set to command start. The columns between command start and the right edge of the virtual screen will be erased on the command line. In effect, this sequence will reset and erase the entire command line.
[Alt]+[F]	Clear Field - The cursor will jump back to the active position if necessary. Command end will be set to the active position and the line will be erased from the active position to the end of the virtual line. In effect this key will reset and erase the command line from the active position to the end of the line.
[Enter]	[Legacy SEND] - The cursor will jump back to the active position if necessary. The cells between command start and command end will be sent to the host. The GATM/SATM/MATM modes are ignored in scroll mode.

Key(s)	Function
[RightArrow]	The cursor and the active position will move to the right. Command end does not change. If the user moves past command end and then enters data, command end will be set to the active position. If the cursor is not on the command line, the right arrow will move the cursor but not the active position or command end.
[LeftArrow]	The cursor and the active position will move to the left. Command start and end do not change. If the user moves before command start and then enters data, the cursor and the active position will jump to command end before displaying the data. If the cursor is not on the command line, the left arrow will move the cursor but not the active position.
[UpArrow]	The cursor will move up one line on the virtual display. The active position does not change.
[DownArrow]	The cursor will move down one line on the virtual display. The active position does not change.
[Bksp]	The cursor will jump back to the active position if necessary. Then the cursor will backspace by one and move the active position back by one. The user cannot backspace to the left of command start.
[LINE FEED]	A line feed may be sent by the host or sent by the TE user when [Enter] is pressed depending on how various modes are set up. When a line feed is encountered the cursor position, active position, command start and command end are updated. If the cursor was on the last line of the scrolling region when the line feed was received, the screen will scroll. Note that there are two ways to process a line feed. The first is to do a CR/LF pair which resets the cursor position at the left margin on the next line. The second is to do only a line feed which drops the cursor down directly below its previous position. In either case, the appropriate settings (active position, command start, etc.) are updated appropriately just as if the host had positioned the cursor.

Data Editing

The only local editing that is enabled in scroll mode is the following DAQ (Define Area Qualifications) options: alphabetic, numeric, alphanumeric, and all graphics. The other local editing commands (local edit match field CSI...+A) are disabled in scroll mode. Also, the TE ignores SPA and SSA areas while in scroll mode.

Scroll mode communication is a communication mode where the terminal allows data entry only on one line. The terminal operator can view the rest of the form, but the cursor returns to the line the host placed the cursor in whenever the operator tries to enter data.

Character Mode Communication

Character mode communication is a communication mode that requires the terminal to send each character to the host for host echo back to the terminal display. The terminal operator does not see the character entered at the terminal until the host echoes the character back for display.

Character mode is the best communication mode when

- operations use screen painters or designers with programming tools such as Oracle, Ingress, Informix, and Powerhouse, because these tools normally require character mode communication.
- the host requires the terminal operators to log in with passwords that should not display on the screen.

Character mode is much slower than the other communication modes because the host must echo each keystroke the operator makes before the terminal operator can see that character on the terminal screen. Therefore, another communication mode should be used when possible.

Local Echo

Local echo is a process that allows each character to display locally at the terminal, as opposed to waiting for the host to echo back the character to the terminal before displaying the character. Local echo can be used when a host does not perform the echo back to the terminal.

The "Local Echo" feature of character mode communication may be toggled via hot keys:

Hot Key	Action	
[Alt]+[E]	Enabled local echo	Characters are displayed locally on the terminal
[Alt]+[G]	Disabled local echo	Characters are displayed on the terminal after the character is sent to the host and the host echoes the character back to the terminal

Character Mode Setup

The programmer must include the following command in the command string for setting up character mode:

```
ESC[12h(turn local echo off in SRM)
```

If local echo is not turned off in send-receive mode (SRM), the terminal will operate in scroll or block mode.

Printing And The RS-232 Port

You can print data out the RS-232 port on the terminal by enabling the RS-232 port with the MC (Media Copy) command. Refer to the “MC” section of Chapter 6 “ANSI Escape Sequences” for more information on the Media Copy command. However, if you have trouble printing, you need to check several factors.

Troubleshooting

Refer to the following list of possible problems that can inhibit you from printing out the RS-232 port:

- RS-232 port is not enabled. You must enable this port with the Media Copy command to print using the RS-232 port. Refer to “MC” in Chapter 6, “ANSI Escape Sequences.”
- Com1Power or Com2Power parameters are set to OFF. The power mode parameter for the communications port being used must be set to ON to print using the RS-232 port. (Refer to the “Platform Menu Option” section of Chapter 3, “Configuration Utility”, for more information on these parameters.)
- The communication parameters (i.e. parity, stop bits, data bits, baud rate, etc.) for the device connected to the RS-232 port are set incorrectly. (Refer to the “Platform Menu Option” section of Chapter 3, “Configuration Utility”, for more information on these parameters.)

RS-232 Input/Output

The RS-232 port supports both input and output. One example of input is from a weight scale. The weight of the item is entered in an input field on an application screen. The data from the scales comes in the RS-232 port as input. For more information on input from the RS-232 port, refer to the reference guide for your specific terminal. One example of output is printing labels using command sequences sent from the host to a printer connected to the terminal's RS-232 port.

Line Attributes

Line attributes are display features that affect the entire line containing the cursor's current position. Line attributes include:

- Single-width / Single-height lines
- Double-width / Double-height lines
- Double-width / Single-height lines

When these attributes are applied to the current line, the cursor position does not change. When lines scroll, the line attributes move with them. If the entire line is erased using the Erase in Display (ED) command, the line attribute changes to single-height and single-width.

Note: *EBM (Editing Boundary Mode) must be set to the virtual display for the lines to change to single-height, single-width.*

Line attributes may be applied at any time. They will stay in effect until explicitly overridden by another attribute type, scrolled off the display or erased.

DEC Special Graphics Character Set

The terminal includes the DEC special graphics set (also known as the VT100 line drawing character set). The character set has about two-thirds of the ASCII graphic characters. It also has special symbols and short line segments. The line segments let you create a limited range of pictures while still using text mode. This character set may be designated by the host (see the section titled "Designating Hard Character Sets" found in this chapter).

DEC Special Graphics - Code Table

Refer to the 8-bit code table below that lists all the decimal, hexadecimal, and octal values for each character:

	COLUMN				0		1		2		3		4		5		6		7	
	BITS b8	b7	b6	b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4	b3	b2	b1																
0	0	0	0	0	**NUL**	0 0 0	**DLE**	20 16 10	**SP**	40 32 20	**0**	60 48 30	**@**	100 64 40	**P**	120 80 50	♦	140 96 60	— SCAN3	160 112 70
1	0	0	0	1	**SOH**	1 1 1	**DC1** (XON)	21 17 11	**!**	41 33 21	**1**	61 49 31	**A**	101 65 41	**Q**	121 81 51	▒	141 97 61	— SCAN5	161 113 71
2	0	0	1	0	**STX**	2 2 2	**DC2**	22 18 12	**"**	42 34 22	**2**	62 50 32	**B**	102 66 42	**R**	122 82 52	␉	142 98 62	— SCAN7	162 114 72
3	0	0	1	1	**ETX**	3 3 3	**DC3** (XOFF)	23 19 13	**#**	43 35 23	**3**	63 51 33	**C**	103 67 43	**S**	123 83 53	␌	143 99 63	— SCAN9	163 115 73
4	0	1	0	0	**EOT**	4 4 4	**DC4**	24 20 14	**$**	44 36 24	**4**	64 52 34	**D**	104 68 44	**T**	124 84 54	␍	144 100 64	├	164 116 74
5	0	1	0	1	**ENQ**	5 5 5	**NAK**	25 21 15	**%**	45 37 25	**5**	65 53 35	**E**	105 69 45	**U**	125 85 55	␊	145 101 65	┤	165 117 75
6	0	1	1	0	**ACK**	6 6 6	**SYN**	26 22 16	**&**	46 38 26	**6**	66 54 36	**F**	106 70 46	**V**	126 86 56	°	146 102 66	┴	166 118 76
7	0	1	1	1	**BEL**	7 7 7	**ETB**	27 23 17	**'**	47 39 27	**7**	67 55 37	**G**	107 71 47	**W**	127 87 57	±	147 103 67	┬	167 119 77
8	1	0	0	0	**BS**	10 8 8	**CAN**	30 24 18	**(**	50 40 28	**8**	70 56 38	**H**	110 72 48	**X**	130 88 58	␤	150 104 68	│	170 120 78
9	1	0	0	1	**HT**	11 9 9	**EM**	31 25 19	**)**	51 41 29	**9**	71 57 39	**I**	111 73 49	**Y**	131 89 59	␋	151 105 69	≤	171 121 79
10	1	0	1	0	**LF**	12 10 A	**SUB**	32 26 1A	*****	52 42 2A	**:**	72 58 3A	**J**	112 74 4A	**Z**	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1	0	1	1	**VT**	13 11 B	**ESC**	33 27 1B	**+**	53 43 2B	**;**	73 59 3B	**K**	113 75 4B	**[**	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1	1	0	0	**FF**	14 12 C	**FS**	34 28 1C	**,**	54 44 2C	**<**	74 60 3C	**L**	114 76 4C	****	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1	1	0	1	**CR**	15 13 D	**GS**	35 29 1D	**-**	55 45 2D	**=**	75 61 3D	**M**	115 77 4D	**]**	135 93 5D	└	155 109 6D	£	175 125 7D
14	1	1	1	0	**SO**	16 14 E	**RS**	36 30 1E	**.**	56 46 2E	**>**	76 62 3E	**N**	116 78 4E	**^**	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1	1	1	1	**SI**	17 15 F	**US**	37 31 1F	**/**	57 47 2F	**?**	77 63 3F	**O**	117 79 4F	(BLANK)	137 95 5F	— SCAN1	157 111 6F		177 127 7F

|← C0 Codes →|← GL Codes (DEC Special Graphic) →|

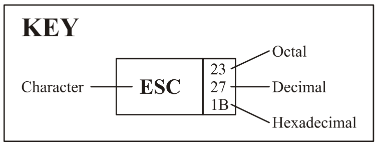


Figure 4-1 DEC Special Graphics - Code Table

National Replacement Character Sets (NRC Sets)

These character sets are basically the standard ASCII table with a handful of characters replaced by characters specific to a country or language. They are included with the terminal and may be designated by the host (see the G0/G1/G2/G3 commands in Chapter 7, "DEC Private Escape Sequences" for more details).

The NRC sets are only available if National Mode is selected, and only one NRC set may be designated at a time. The keyboard type must also be set to the correct language. The actual sets are shown in the following tables.

KeyboardType and NRC Sets

The KeyboardType is used by the TE to determine the font table to use.

KeyboardType Selected	**NRC Set Displayed**
British	British
Canadian(French)	French Canadian
Danish	Norwegian
Dutch	Dutch
Finnish	Finnish
Flemish	Dutch
French/Belgian	French
German	German
Italian	Italian
NorthAmerican	ASCII
Norwegian	Norwegian
Spanish	Spanish
Swedish	Swedish
Swiss(French)	Swiss
Swiss(German)	Swiss

ASCII Set with NRC Replacement

Note: If National Mode is in effect, all GR characters are treated as GL characters i.e. the GR characters have their high bit stripped.

The tables below show which characters from the standard ASCII set are replaced by each type of NRC set. Refer to the 8-Bit code tables that follow.

Standard ASCII Column/Rows 2/3 to 5/14

	Column/Row Position					
Character Set	**2/3**	**4/0**	**5/11**	**5/12**	**5/13**	**5/14**
ASCII	#	@	[	\	]	^
British	£	@	[	\	]	^
Danish	#	@	Æ	Ø	Å	^
Dutch	£	¾	ÿ	½	|	^
Finnish	#	@	Ä	Ö	Å	Ü
Flemish	£	¾	ÿ	½	|	^
French/Belgian	£	à	°	ç	§	^
French Canadian	#	à	â	ç	ê	î
German	#	§	Ä	Ö	Ü	^
Italian	£	§	°	ç	é	^
Norwegian	#	@	Æ	Ø	Å	^
Spanish	£	§	¡	Ñ	¿	^
Swedish	#	É	Ä	Ö	Å	Ü
Swiss (French)	ù	à	é	ç	ê	î
Swiss (German)	ù	à	é	ç	ê	î
Swiss	ù	à	é	ç	ê	î

Standard ASCII Column/Rows 5/15 to 7/14

	Column/Row Position					
Character Set	**5/15**	**6/0**	**7/11**	**7/12**	**7/13**	**7/14**
ASCII	-	`	{	|	}	~
United Kingdom	-	`	{	|	}	~
Danish	-	`	æ	ø	å	~
Dutch	-	`	¨	ƒ	¼	´
Finnish	-	é	ä	ö	å	ü
Flemish	-	`	¨	ƒ	¼	´
French/Belgian	-	`	é	ù	è	¨
French Canadian	-	ô	é	ù	è	û
German	-	`	ä	ö	ü	ß
Italian	-	ù	à	ò	è	ì
Norwegian	-	`	æ	ø	å	~
Spanish	-	`	`	°	ñ	ç
Swedish	-	é	ä	ö	å	ü
Swiss (French)	è	ô	ä	ö	ü	û
Swiss (German)	è	ô	ä	ö	ü	û
Swiss	è	ô	ä	ö	ü	û

Down-Line-Loadable Character Set

The host may create and download a character set containing up to 94 characters. This character set is called a Dynamically Re-definable Character Set (DRCS). After the characters are created, they may be downloaded into the DRCS buffer by using a DECDLD device control string.

Since the character set is not saved to disk, the character set is lost when the terminal is powered off.

Designing A Character Set

The figure below shows an example of the cell size of the DRCS cell. Each pixel in a character cell is represented by a bit with a binary value of 1 (pixel on) or 0 (pixel off).

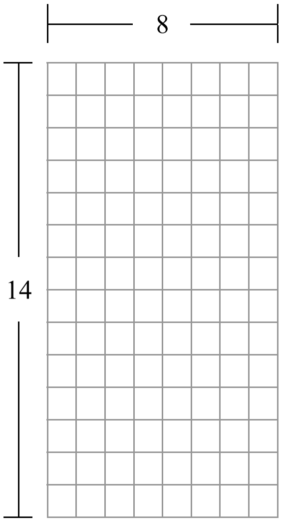


Figure 4-2 DRCS Cell Size

Note: Refer to the reference guide for your specific LXE terminal for the correct cell size.

Example

To design the character `A` follow this four step process.

1. Designate which pixels will be on and which pixels will be off. The character might look like this:

			X				
		X		X			
	X				X		
X						X	
X						X	
X						X	
X	X	X	X	X	X	X	
X						X	
X						X	
X						X	
X						X	

Figure 4-3 Example of An "A" Character

2. After the character pixels are specified, divide the pixels of the character cell into columns of 6 bits each, using the format shown in the next figure. The column numbers designate the order in which the columns are sent to the terminal.

b0									Upper Columns
b1				X					
b2			X		X				
b3		X				X			
b4	X						X		
b5	X						X		
b0	X						X		Middle Columns
b1	X	X	X	X	X	X	X		
b2	X						X		
b3	X						X		
b4	X						X		
b5	X						X		
b0									Lower Columns
b1									

Figure 4-4 Example of "A" Divided Into Columns

Each column is now represented as a vertical 1 X 6 pixel matrix, called a sixel. The least significant bit is at the top, and the most significant bit is at the bottom. Because the character height (14 pixels) is not a multiple of 6, the columns on the bottom of the character cell have only 2 bits each. The 6 highest order bits are ignored.

3. After the cell is divided into six-pixel columns (sixels), convert the binary values of each column to its equivalent character. Because column codes are restricted to characters in the range of ? (octal 077) to ~ (octal 176), an offset of octal 077 must be added to each

column octal value. For example, binary value 000000 is converted to octal 077 (octal 0 + octal 77). Binary value 110101 is converted to octal 164 (octal 65 + octal 077) and binary value 111111 is converted to octal 176 (octal 077 + 077).

4. After the binary column codes are converted to octal values (using the offset), convert the octal value for each column to its equivalent character by using the standard ASCII table. The next three figures provide this conversion procedure for the example of the character `A`.

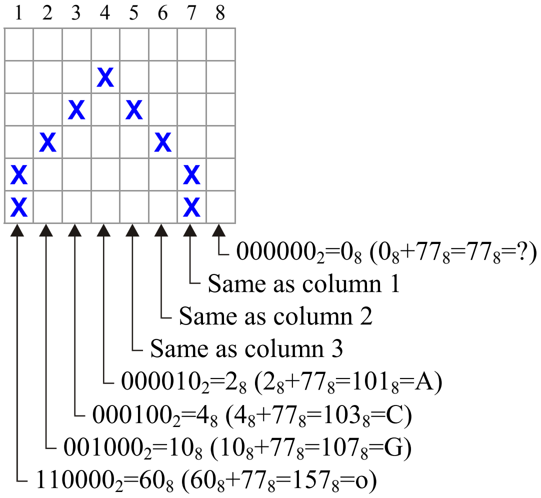


Figure 4-5 Upper Column Codes For "A" Character

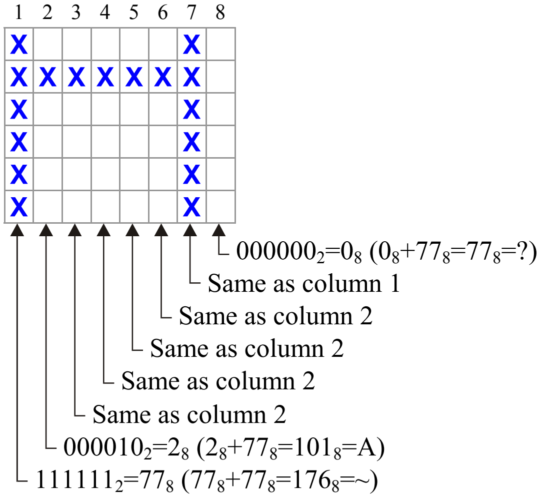


Figure 4-6 Middle Column Codes For "A" Character

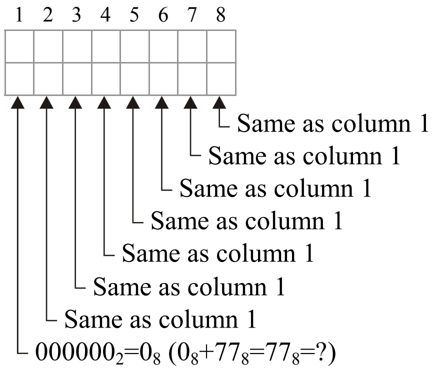


Figure 4-7 Lower Column Codes For "A" Character

Use this procedure to design each DRCS character you want. Then download your characters with the DECDLD device control string described below.

Device Control Strings (DCS)

A Device Control String is a delimited string of characters used in a data stream as a logical entity for control purposes. It consists of an opening delimiter, a command string and a closing delimiter. They are used for down-line loading character sets.

The terminal uses the following device control string format.

9/0 DCS	... Data	9/12 ST
Device Control String (opening delimiter)	Character set	String Terminator (closing delimiter)

DCS is an 8-bit control character. You can also express it as ESC P (1/11, 5/0) when coding for a 7-bit environment.

ST is an 8-bit control character. You can also express it as ESC \ (1/11, 5/12) when coding for a 7-bit environment.

Downloading DRCS Characters

Download the DRCS character set with the following DECDLD device control string format.

DCS Pfn;Pcn;Pe;Pcms;Pw;Pt { Dscs Sxbp1;Sxbp2;...;Sxbpn ST

where:

DCS (9/0) is the device control string introducer. It is an 8-bit control character that can also be expressed as ESC P (1/11, 5/0) when coding for a 7-bit environment.

Pfn;Pcn;Pe;Pcms;Pw;Pt are parameter characters, separated by semicolons. See Figure titled "DECDLD Parameter Characters" for a description of these parameters.

{ (7/11) is the final character that marks the end of the parameter characters and specifies a DECDLD function.

Dscs defines the character set name for the soft font, and is used in the SCS (select character set) escape sequence.

Sxbp1;Sxbp2;...;Sxbpn are sixel bit patterns (1 to 94 patterns) for character separated by semicolons. Each sixel bit pattern has the form:

S...S/S...S/S...S

where:

the first S...S represents the upper columns (sixels) of the DRCS character,

the first slash (2/5) advances the sixel pattern to the middle columns of the DRCS character,

the second S...S represents the middle columns (sixels) of the DRCS character,

the second slash (2/5) advances the sixel pattern to the bottom columns of the DRCS character,

and the third S...S represents the bottom columns (sixels) of the DRCS character.

ST (9/12) is the string terminator. It is an 8-bit control character that can also be expressed as ESC \ (1/11, 5/12) when coding for a 7-bit environment.

Note: Any characters that are not explicitly specified will appear as a gray box on the display.

DECDLD Parameter Characters

Parameter	Name	Description
Pfn	Font Number	Selects the DRCS font buffer to load. The terminal only has one DRCS font buffer. 0 = Device default (set 1) 1 = set 1
Pcn	Starting	Selects starting character to load in DRCS font buffer. The first replaceable character is ! (hex 21) and the last is ~ (hex 7E). For example, a parameter of 1 specifies a column 2/row 1 character; parameter 20 specifies a column 3/row 4 character; parameter 94 specifies a column 7/row 14 character. Row and column designations are 0 (zero) based as shown in the “8-Bit Code Table” in Chapter 5, “Control Codes”. Valid values: 1-94.
Pe	Erase	Selects which characters are erased before loading. 0 = erase all characters in this DRCS set. 1 = erase only the characters being reloaded. 2 = erase all characters in all DRCS sets (effectively the same as 0 since there is only one set).
Pcms	Character	Defines the expected limit of the character matrix size. 0 = Device default (8 X 14)6 = 8 X 14
Pw	Width	Specifies the width attribute. 0 = Device default (80 columns) 1 = 80 columns.
Pt	Text/Full	Allows software to treat the font as a text font or a full cell font. 0 = Device default (text) 1 = Text.

DECDLD Example

Download a character set starting with the character A, designed in the example shown in the "Designing a Character Set" section of this chapter. Designate this soft character set as G1. Invoke this G1 set into GL.

Use the following device control string to define your character set.

DCS 1;33;1 { sp @ oGCACGo?/~AAAAA~?/????????; (character 2); ... ; (character n) ST

where

DCS	introduces the sequence.
1;33;1	specifies loading the DRCS font buffer, selects the starting character as column 4/row 1

Note: Pcms, Pw, and Pt are not included; they default to 0 values.

{	indicates the end of the parameter characters and specifies that this is a DECDLD control string.
sp @	defines a character set as an unregistered soft set. This value is the only currently supported value for this parameter.
oGCACGo?	are the character codes for the upper columns of the example DRCS character A.
/	advances the sixel sequence to the middle columns of the example DRCS character A.
~AAAAA~?	are the character codes for the middle columns of the example DRCS character A.
/	advances the sixel sequence to the lower columns of the example DRCS character A.
????????	are the character codes for the lower columns of the example DRCS character A.
;	signals the end of the current DRCS character being loaded and the beginning of another DRCS character to be loaded.
ST	indicates the end of the device control string.

After defining the character set with the device control string, designate the character set as G1 (or any of the four character sets). To do this, use the following escape sequence.

ESC) sp @

Refer to the "Designating Soft Character Sets" section in this chapter for a detailed discussion of the subject.

After designating the character set as G1, invoke the G1 set into GL (or GR) by performing a shift out (S0). To do this, use the following sequence:

^N

The dynamically re-definable character set should now be loaded and ready for use.

Note: Any characters not explicitly defined by the host will appear as a gray box on the terminal display.

Designating Soft Character Sets

A soft, down-line loadable, character set (font) can be defined that may or may not replace one of the existing hard sets. The soft character set that replaces a hard character set remains in effect until the soft character set is cleared, redefined or the terminal is powered off. The soft set is redefined by a DECDLD device control string described in this chapter.

Designate a soft character set by using the following escape sequences.

Escape Sequence				Designates Soft Set As
1/11 ESC	2/8 (	2/0 sp	4/0 @	G0
1/11 ESC	2/9)	2/0 sp	4/0 @	G1
1/11 ESC	2/10 *	2/0 sp	4/0 @	G2
1/11 ESC	2/11 +	2/0 sp	4/0 @	G3

Figure 4-8 Soft Character Set Escape Sequences

After designating the soft character set into any of the G0-G3 slots, invoke it into either GL or GR with the single shifts or locking shifts (i.e. LS0 or SS2).

Sending The Terminal Screen To The Host

Character Mode

Since each keypress is sent to the host as it is pressed, there is no concept of sending the screen in character mode. Pressing the [Enter] key in this case will simply send a carriage return and/or line feed to the host. No other sections of the screen are sent as a result of pressing [Enter].

Scroll Mode

Scroll mode allows user input on only one line of the display at a time until the terminal user presses [Enter]. When this happens, the cursor will jump back to the active position if necessary. The cells between command start and command end will be sent to the host. The GATM/SATM/MATM modes are ignored in scroll mode. See the section titled “Scroll Mode Communication” found previously in this chapter for more details.

Block Mode

Block mode allows the user to input in any unprotected area on the display until the terminal user presses [Enter]. When this happens, the terminal will look at the current settings of GATM, SATM and MATM to determine exactly which screen cells are transmitted back to the host. The settings of these three modes yields 6 different combinations of cells (see Figure titled “Sending The Screen In Block Mode”).

Protected cells are cells tagged with SPA (Start Protected Area) or DAQ 1 (Define Area Qualification).

Unprotected cells are cells that are neither SSA (Start Selected Area) nor SPA nor DAQ 1.

Note: DAQ 8 cells are considered unprotected.

			(x = don't care)		
Case #	**Description of Cells Sent**	**Kind of Cells Sent**	**GATM**	**SATM**	**MATM**
1	Entire screen	All	set 1h	set 17h	x
2	Current/next selected only	1 SSA	set 1h	reset 17l	reset 15l
3	All selected and all unprotected	SSA + !(SPA/DAQ1)	reset 1l	reset 17l	set 15h
4	Current/next selected and all unprotected	1 SSA + !(SPA/DAQ1/SSA)	reset 1l	reset 17l	reset 15l
5	All selected	SSA	set 1h	reset 17l	set 15h
6	All unprotected	!SSA + !(SPA/DAQ1)	reset 1l	set 17h	x

Figure 4-9 Sending The Screen In Block Mode

To use the table, first decide which of the 6 cases is desired. For example, to use case #6 set SATM and reset GATM. Since MATM is don't care in this case, it does not need to be set either way. Set up the screen so that it is marked protected or selected except where the terminal user will enter data. Now when the user presses [Enter], all cells that are not SSA, not SPA and not DAQ1 will be sent back to the host.

Sending the screen always operates on the virtual screen so EBM (Editing Boundary Mode) is ignored. Tab fields do not affect how the screen is sent.

When sending the screen, the “current/next” selected area means that if the cursor is currently on a selected area, that area will be sent. If not, search forward in the screen to find one and use that one. If none are found before the end of the screen, reset to the screen origin and look from there. If the terminal returns to the cursor position without finding one, it sends nothing.

In all cases the screen is scanned starting from the origin and each cell is transmitted in the order in which it is programmed on the display. For example, #5 means start at the origin and scan forward for all selected cells (type SSA only) sending them in the order they are found. For case #4, the current/next selected area is found first based on the current cursor position and it is remembered. However, the cells are still transmitted starting from the origin. The selected area is simply transmitted when the terminal comes across it in the display.

A single area separator (HT 0/9) byte is inserted into the transmit stream for each protected area on the screen (for case #3, 4, 5 and 6 only). However, if the first or last area on the screen is protected, no separator byte is inserted.

For example, assume case #3 is in effect. If the first few cells of the screen are defined as follows where p is a protected cell, s is an SSA selected cell and u is an unprotected cell:

p1 p1 p1 s1 s1 s1 p2 p2 p2 s2 s2 s2 p3 p3 p3 s3 s3 s3 u1 u1 u1 ...

yields a transmit stream of:

s1 s1 s1 HT s2 s2 s2 HT s3 s3 s3 u1 u1 u1...

Note: No separator is inserted for the first protected area on the screen. No separator appears between s3 and u1 since there are no protected cells between them.

If the last few cells on the same display are defined as follows:

...p4 p4 p4 p4 u2 u2 u2 u2 u2 p5 s4 s4 s4 s4 p6 p6

yields a transmit stream of:

...HT u2 u2 u2 u2 u2 HT s4 s4 s4 s4

Note: No separator is inserted for the last protected area on the screen.

To reduce traffic between the terminal and the host when sending the screen, use cases #3, #5 and #6. Tag most of the screen as protected by using SPA or DAQ1. Leave only the cells that the operator will type/scan into as unprotected/selected. When the terminal operator presses [Enter] to send the screen, only the unprotected/selected cells will be sent separated by tabs.

Tabbing

Three types of tabs are provided in the terminal:

- default
- area
- regular

Only one type of tab can be active at any time. Previously set tabs may be suppressed by a tab type with a higher priority. However, the suppressed tabs are still defined in the terminal.

Block mode must be active for the [Tab] key to work. If the [Tab] key is pressed in character mode, it will be transmitted back to the host. It is up to the host to echo it back to actually perform the tab. In scroll mode, pressing the [Tab] key is equivalent to pressing the [Spacebar] key.

The host may perform tabbing by sending the appropriate commands (i.e. HT) to the terminal regardless of the mode it is in.

When a tab command is received or the terminal user presses the [Tab] key, the cursor will move forward until the next tab stop is found. If none is found before the end of the display, the cursor will move to the origin of the virtual display and continue searching. The terminal will not tab to a tab that is in a protected area. Tabs are stationary and do not scroll when the screen does. EBM is ignored when the terminal tabs.

Default Tabs

These tabs are present when the terminal is first powered up. They appear on every line starting in column one and are spaced 8 columns apart (i.e. 1, 9, 17, etc.). These tabs have the lowest priority and they can be neither set nor erased by the host.

Area Tabs

These tabs can be set by the DAQ, SSA, EPA and ESA commands. They can be set by the host on any screen cell within the virtual display. These tabs have a medium priority. Setting one or more area tabs suppresses all of the default tabs. They can be erased only by the ED (Erase in Display) and EL (Erase in Line) commands when ERM (ERasure Mode) is set. If all of the area tabs are erased (and there are no regular tabs) the default tabs are restored.

Regular Tabs

These tabs can be set by the HTS, CTC and DAQ 7 commands. They can be set by the host on any screen cell within the virtual display. These tabs have high priority. Setting one or more regular tabs suppresses all default and area tabs. Regular tabs can be erased by the CTC, HTS and TBC commands. If all of the regular tabs are erased, any existing area tabs are restored. If no area tabs exist, the default tabs are restored.

Compose Key

The compose key can be used to generate characters that do not exist as standard keys on the keyboard. To use a compose sequence, press the compose key ([Alt]+[C]) followed by two other standard keys. When the terminal sees this compose sequence, only the special “composed” character will be printed on the display.

For example, to generate the letter e with an acute accent: press [Alt]+[C], then press [E], then press the apostrophe [‘]. Or you may press [Alt]+[C], then press apostrophe [‘], then press the [E].

You may press the keys in the following table in any order unless the table states they are order sensitive.

The compose key must be enabled within the configuration utility or with the CSI...+E private host sequence.

Valid Compose Keys

VALID COMPOSE SEQUENCES

Composite Character		**Key Sequence [note: (sp)=space]**
“	(quotation mark)	“ (sp)
#	(number sign)	++
'	(apostrophe)	' (sp)
@	(commercial at)	aa or AA
[	(opening bracket)	((
\	(backslash)	// or /<
]	(closing bracket)	))
^	(circumflex accent)	^ (sp)
'	(single quote)	' (sp)
{	(opening brace)	(-
|	(vertical line)	/^
}	(closing brace)	)-
~	(tilde)	~ (sp)
¡	(inverted !)	!!
¢	(cent sign)	c/ or C/ or c| or C|
£	(pound sign)	l- or L- or l= or L=
¥	(yen sign)	y- or Y- or y= or Y=
§	(section sign)	so or SO or s! or S! or s0 or S0
¤	(currency sign)	xo or XO or x0 or X0
©	(copyright sign)	co or CO or c0 or C0
ª	(feminine ordinal indicator)	a- or A-

VALID COMPOSE SEQUENCES

Composite Character		Key Sequence [note: (sp)=space]
«	(angle quotation mark left)	<<
°	(degree sign)	0^ or (sp) #
±	(plus/minus sign)	+-
²	(superscript 2)	2^
³	(superscript 3)	3^
µ	(micro sign)	/u or /U (order sensitive)
¶	(paragraph sign)	p! or P!
·	(middle dot)	.^
¹	(superscript 1)	1^
º	(masculine ordinal indicator)	o- or O-
»	(angle quotation mark right)	>>
¼	(fraction one-quarter)	1 4 (order sensitive)
½	(fraction one-half)	1 2 (order sensitive)
¿	(inverted ?)	??
À	(A grave)	A`
Á	(A acute)	A'
Â	(A circumflex)	A^
Ã	(A tilde)	A~
Ä	(A umlaut)	A”
Å	(A ring)	A*
Æ	(A E ligature)	AE (order sensitive)
Ç	(C cedilla)	C,
È	(E grave)	E`
É	(E acute)	E'
Ê	(E circumflex)	E^
Ë	(E umlaut)	E”
Ì	(I grave)	I`
Í	(I acute)	I'
Î	(I circumflex)	I^

VALID COMPOSE SEQUENCES

Composite Character		Key Sequence [note: (sp)=space]
Ï	(I umlaut)	I”
Ñ	(N tilde)	N~
Ò	(O grave)	O`
Ó	(O acute)	O'
Ô	(O circumflex)	O^
Õ	(O tilde)	O~
Ö	(O umlaut)	O”
Œ	(O E ligature)	O E (order sensitive)
Ù	(U grave)	U`
Ú	(U acute)	U'
Û	(U circumflex)	U^
Ü	(U umlaut)	U”
Ÿ	(Y umlaut)	Y”
ß	(German small sharp s)	ss
à	(a grave)	a`
á	(a acute)	a'
â	(a circumflex)	a^
ã	(a tilde)	a~
ä	(a umlaut)	a”
å	(a ring)	a*
æ	(a e ligature)	ae (order sensitive)
ç	(c cedilla)	c,
è	(e grave)	e`
é	(e acute)	e'
ê	(e circumflex)	e^
ë	(e umlaut)	e”
ì	(i grave)	i`
í	(i acute)	i'
î	(i circumflex)	i^
ï	(i umlaut)	i”
ñ	(n tilde)	n~
ò	(o grave)	o`

VALID COMPOSE SEQUENCES

Composite Character		Key Sequence [note: (sp)=space]
ó	(o acute)	o'
ô	(o circumflex)	o^
õ	(o tilde)	o~
ö	(o umlaut)	o”
œ	(o e ligature)	o e (order sensitive)
ø	(o slash)	o/
ù	(u grave)	u`
ú	(u acute)	u'
û	(u circumflex)	u^
ü	(u umlaut)	u”
ÿ	(y umlaut)	y”

Default Key Definitions

The following tables show the key definitions for the arrow keys and the numeric keypad. The default key definitions for the function keys are included in Appendix D, “ANSI Plus Function Key Definitions”.

Note that the function key definitions can be changed with the Function Key Editor ([Alt]+[K]). In a 6200 system, download them with the NMWS (Network Management Workstation) console program.

Also, when the terminal is in 8-bit mode, ESC[will be converted to CSI and ESCO will be converted to SS3.

Basic Default Arrow Key Definitions

Keystroke	**Cursor Key Numeric Mode (DECCKM reset)**	**Cursor Key Application Mode (DECCKM set)**
[UpArrow]	ESC[A	ESCOA
[DownArrow]	ESC[B	ESCOB
[RightArrow]	ESC[C	ESCOC
[LeftArrow]	ESC[D	ESCOD

Extended Arrow Key Definitions

Keystroke	**Shift+**	**Alt+**	**Ctrl+** [1]
[RightArrow]	ESC[1~	nothing	Move physical display one unit right
[UpArrow]	ESC[2~	ESC[5~	Move physical display one unit up
[LeftArrow]	ESC[3~	nothing	Move physical display one unit left
[DownArrow]	ESC[4~	ESC[6~	Move physical display one unit down

Function Key Defaults

Please refer to Appendix D, “ANSI Plus Function Key Definitions” for the default values for ANSI Plus Function keys.

---

[1] These keys only function on screens smaller than 24x80. One unit may be either half window increments or full window increments depending on how Window Movement Mode is set.

Numeric Keypad Definitions

Keystroke	Keypad Application Mode (DECKPAM)	Keypad Numeric Mode (DECKPNM)
Numeric 0	ESCOp	0
Numeric 1	ESCOq	1
Numeric 2	ESCOr	2
Numeric 3	ESCOs	3
Numeric 4	ESCOt	4
Numeric 5	ESCOu	5
Numeric 6	ESCOv	6
Numeric 7	ESCOw	7
Numeric 8	ESCOx	8
Numeric 9	ESCOy	9
Numeric .	ESCOn	. (period)
Numeric -	ESCOm	- (dash)
Numeric Enter	ESCOM	Enter
Numeric /	ESCOl	/
Numeric +	+	+
Numeric *	*	*

Window Manager

When the host sends a 25 line x 80 column virtual screen to the terminal, window manager determines how the display window moves on the virtual screen. The window manager positions the physical display window on the virtual screen based on the cursor's position and on the terminal's current window manager settings.

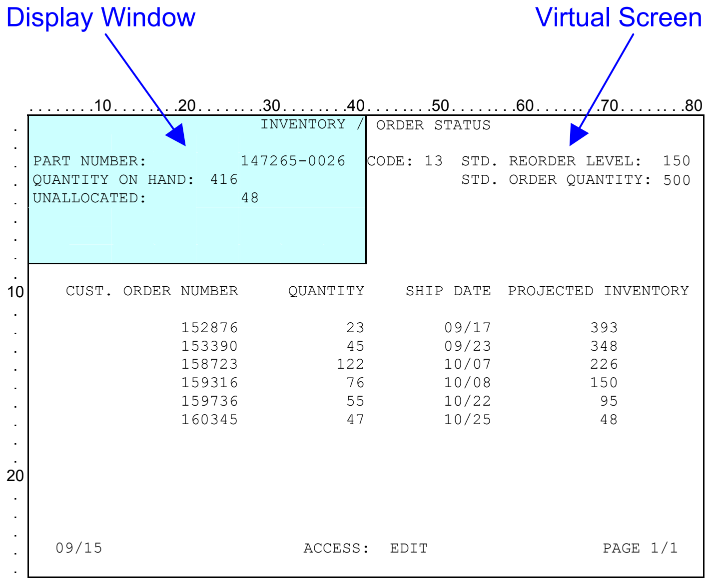
All of the window manager parameters may be set by either the host program or by the user in the configuration utility. In a 6200 system, window manager parameters may be set using the NMW console program.

Terminology

Virtual Screen	The full 25 line x 80 column screen generated by the host computer. An example of a virtual screen is the screen you view when you use a personal computer.
Display Window	The portion of the virtual screen visible in the physical display window. The size of this window depends on the type of terminal. Usually this is a window 8 x 40 or 8 x 20, but other sizes are possible.
Window Manager	The feature that enables the smaller display window to show portions of the virtual screen at the terminal.
Lock Window Mode	Locks the display window in a position relative to the virtual screen. The display window will remain stationary even if the cursor moves outside the lock window.
Cursor Tracking Mode	The display window follows the movement of the cursor. The display window can also be moved by moving the cursor or by the window movement keys.

Example

The host computer sends a virtual screen to the terminal where the display window shows only a section of the virtual screen. The illustration below shows the relationship between the virtual screen and the terminal display window.



```
                        INVENTORY / ORDER STATUS

PART NUMBER:          147265-0026 CODE: 13  STD. REORDER LEVEL:  150
QUANTITY ON HAND:  416                      STD. ORDER QUANTITY: 500
UNALLOCATED:          48




   CUST. ORDER NUMBER     QUANTITY    SHIP DATE  PROJECTED INVENTORY

                152876          23        09/17          393
                153390          45        09/23          348
                158723         122        10/07          226
                159316          76        10/08          150
                159736          55        10/22           95
                160345          47        10/25           48






  09/15                      ACCESS:  EDIT                  PAGE 1/1
```

Figure 4-10 8 Line x 40 Column Display Window

Note: This display window is an 8 line x 40 column display.

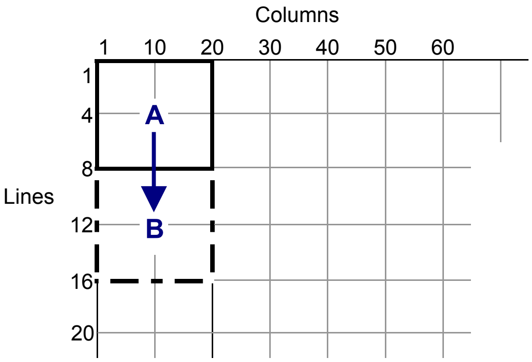
Window Movement

Two types of window movement can be specified: full window movement or quadrant movement.

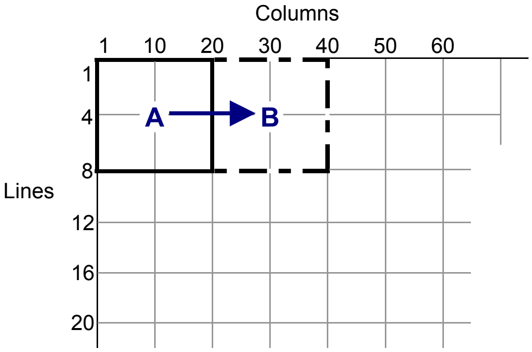
Full Window Movement

For an 8x20 display, when the display window moves vertically in full window movement, the window moves 8 lines at a time and does not overlap its previous position.

When the display window moves horizontally in full window movement, the display window moves 20 columns at a time and does not overlap its previous position.



Vertical Movement



Horizontal Movement

Figure 4-11 Full Window Movement - 8 x 20 Display

Quadrant Movement

When the display window moves vertically in quadrant movement, the display window moves 4 lines at a time, causing the screen to overlap a portion of its previous position.

When the display window moves horizontally in quadrant movement, the display window moves 10 columns at a time to overlap a portion of its previous position.

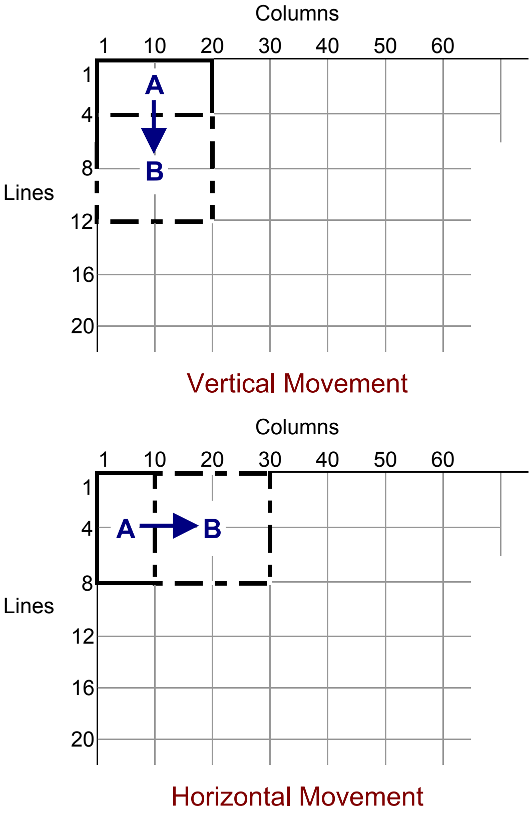


Figure 4-12 Quadrant Movement - 8 x 20 Display

Moving the Display Window

There are two ways the user may move the display window.

Arrow Keys

The first is by using the arrow keys. When the cursor crosses the display window boundary, it will adjust itself accordingly. When the terminal is in character mode, it will send escape sequences to the host and it is up to the host to send the appropriate CUF, CUB, CUU, CUD commands back to the terminal to actually move the cursor. When the terminal is in scroll/block mode, DECCKM should be reset for the arrow keys to work properly. In either case, the terminal will prevent the cursor from moving outside the boundaries of the virtual screen.

Window Movement Keys

The second way to move the window is to use the following key sequences to manually move the display window around the virtual screen. The display window moves according to the type of window movement (full window movement or quadrant movement) enabled. These keys work in both cursor tracking and lock window mode (except for [Shift]+ [UpArrow]).

Key Sequence	Window Direction
[Ctrl]+[UpArrow]	Up
[Ctrl]+[RightArrow]	Right
[Ctrl]+[LeftArrow]	Left
[Ctrl]+[DownArrow]	Down
[Shift]+[UpArrow]	Moves window to lock window position (lock window mode only)

Figure 4-13 Window Movement Keys

Cursor Position

Window movement keys do not affect the position of the cursor on the screen. The cursor remains where it is on the virtual screen while the display window moves.

Display Wrapping

If the display window is along any of the four edges of the virtual screen and you use a window movement key sequence to direct the window beyond the edge of the virtual screen, window manager wraps the display window to the opposite edge of the screen. A window that is wrapped from top to bottom or bottom to top will change line position number and not column position. A window that is wrapped from left to right or right to left will change column number position, but not line position.

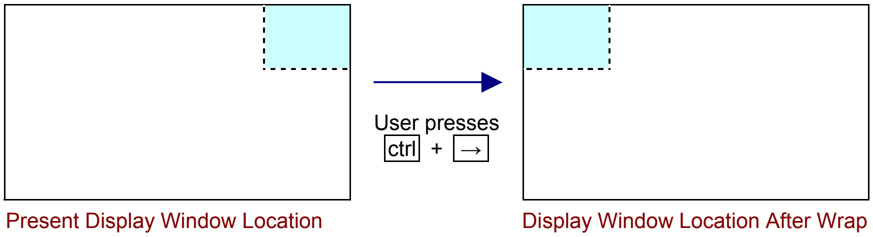


Figure 4-14 Horizontal Display Wrapping

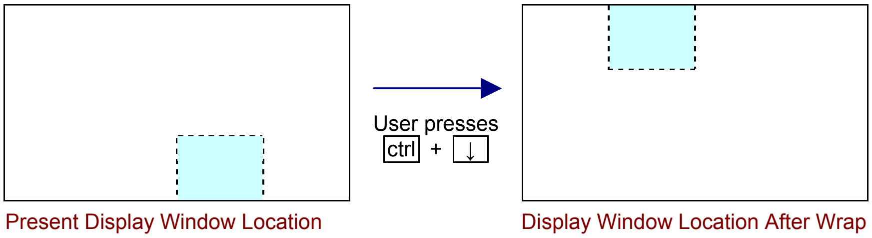


Figure 4-15 Vertical Display Wrapping

Lock Window Mode

Lock window mode locks the display window in a permanent window position relative to the virtual screen. Although you can move the display window around the virtual screen with window movement keys, the display will automatically return to the coordinates specified in window manager when the host sends a new virtual screen or when you press any data entry key. You may also move the cursor outside the display window. However, since the display window is locked, you must manually pan the display window to view the cursor again. Manually panning the display window does not change the lock window setting.

If the operator moves a locked display window using the window movement keys, the window will remain in this new position until one of the following events occurs to get the window to return to the locked position:

- The host sends any character or escape sequence.
- Pressing the “Window lock home” key ([Shift]+[UpArrow]).
- Entering data in the locked window.

Editing Boundary Mode

When EBM=phys, all editing commands take place in the locked window regardless of cursor position.

Window Movement in Lock Window Mode

Cursor location and the position of the display window on the virtual screen combine in the following four ways to affect what is displayed when you enter data in lock window mode.

Note: The terminal behaves in the same way when host input is received except for case 3 as noted below.

When the cursor is located...	**and the display window...**	**then, when you enter a data key...**
in the lock window	is in the same position as the lock window	that data appears and the display does not move.
in the lock window	is outside the lock window	that data appears and the display window returns to the lock window position.
outside the lock window	is outside the lock window (cursor is in the display window)	that data appears in the display window and the window does not move even if the cursor crosses a window boundary. When host input is received, that data appears and the display window returns to the lock window position.
outside the lock window and outside the display window	anywhere on the virtual screen	that data is put in the virtual screen but does not appear in the display window. The display window does not move.

Figure 4-16 Window Movement in Lock Window Mode

The following figure shows valid lock window line and lock window column values that the terminal will accept. If an invalid line or column is entered, the terminal will automatically adjust the values to the nearest valid value that is less than the one entered.

Valid Window Positions

	8 row x 20 column display							**8 row x 40 column display**		
Quadrant Mode	1,1	1,11	1,21	1,31	1,41	1,51	1,61	1,1	1,21	1,41
	5,1	5,11	5,21	5,31	5,41	5,51	5,61	5,1	5,21	5,41
	9,1	9,11	9,21	9,31	9,41	9,51	9,61	9,1	9,21	9,41
	13,1	13,11	13,21	13,31	13,41	13,51	13,61	13,1	13,21	13,41
	17,1	17,11	17,21	17,31	17,41	17,51	17,61	17,1	17,21	17,41

	8 row x 20 column display				**8 row x 40 column display**	
Full Window Mode	1,1	1,21	1,41	1,61	1,1	1,41
	9,1	9,21	9,41	9,61	9,1	9,41
	17,1	17,21	17,41	17,61	17,1	17,41

Figure 4-17 Valid Window Positions

Cursor Tracking Mode

In cursor tracking mode, the display window moves with the cursor as the cursor moves around the screen. The type of window movement depends on your choice of either full window movement or quadrant movement.

Window Moving Methods

Window manager uses two methods to move the display window up, down, left and right when you are in cursor tracking mode:

1. Arrow (cursor movement) keys cause the cursor to cross the window boundaries at the edges of the display window you are currently in.
2. Window movement keys cause the display window to move leaving the cursor stationary (refer to the "Window Movement Keys" section found in this chapter).

Window Moving Rules

When the terminal receives host input, it will adjust the display window as described in the following Quadrant Choice section. When the terminal receives local input (keyboard/bar code/RS-232), it will adjust the display window based on the following rules.

Note: Assume an 8 line x 20 column display for all examples in this section.

Basic Movement Up/Down/Left/Right - The new cursor position is exactly one quadrant Up/Down/Left/Right from the old cursor position. The cursor will be placed in the new quadrant.

Straight Movement Up/Down/Left/Right - The new cursor position is more than one quadrant Up/Down/Left/Right from the old cursor position. The new and old position of the window is in the same horizontal or vertical plane. For Up/Down movement, the cursor will be placed in the lower half of the screen when possible. For Left/Right movement, the cursor will be placed in the right half of the screen when possible.

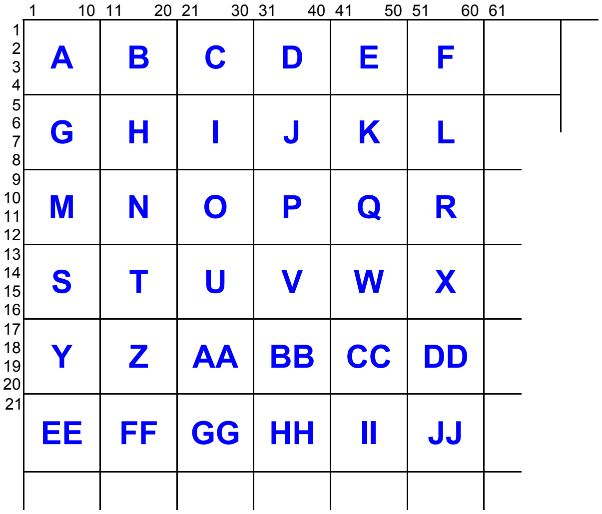
Diagonal Movement - The new cursor position is one or more diagonal quadrants from the old cursor position. The cursor will be placed in the lower right quadrant when possible.

Note: The only way diagonal movement can be achieved is by tabbing to the next unprotected area, or normal data entry when the current area is completely filled and the terminal automatically jumps to the next unprotected area. In both of these cases, it is assumed that the next unprotected area is in a quadrant that is diagonally positioned from the current one.

Note: Sometimes it is not possible to place the cursor in the quadrant specified by one of these 3 rules because doing so would move portions of the display window outside of the virtual display. In these cases, the terminal will adjust the display window so that it never moves outside of the virtual screen.

Display Example

This illustration and the following tables show how window manager moves the display window around the virtual screen in cursor tracking mode. The display example divides windows into four equal pieces, or *quadrants*. For example, the window positioned at 1, 1 consists of upper left quadrant A, upper right quadrant B, lower left quadrant G and lower right quadrant H.



A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	AA	BB	CC	DD
EE	FF	GG	HH	II	JJ

Figure 4-18 Virtual Screen With Arbitrary Labels

The following table describes how the display window moves when tracking the cursor. Use this table with the figure labeled “Virtual Screen with Arbitrary Labels”.

A window is made of quadrants...	The cursor is in quadrant	The operator moves the cursor to quadrant	In quadrant movement, the new window becomes...	In full window movement, the new window becomes...	The rule invoked for this type of move is...
O,P,U,V	O	I	I,J,O,P	C,D,I,J	basic up
P,Q,V,W	P	O	O,P,U,V	N,O,T,U	basic left
C,D,I,J	D	V	O,P,U,V	O,P,U,V	straight down (cursor placed in bottom half)
H,I,N,O	I	Q	J,K,P,Q	J,K,P,Q	straight right (cursor placed in right half)
V,W,BB,CC	V	Y	S,T,Y,Z	S,T,Y,Z	straight left (cursor cannot be placed in right half)
W,X,CC,DD	W	N	G,H,M,N	G,H,M,N	diagonal up and left (cursor placed in lower right quadrant)
Y,Z,EE,FF	Y	C	B,C,H,I	B,C,H,I	diagonal up and right (cursor cannot be placed in lower right quadrant)
M,N,S,T	S	DD	W,X,CC,DD	W,X,CC,DD	diagonal down and right (cursor placed in lower right quadrant)

Figure 4-19 Cursor Tracking Display Examples

Quadrant Choice

The terminal display window is divided into four quadrants that identify sections of the virtual screen. The terminal will always adjust its display window so that the cursor ends up in the specified quadrant. The terminal will effectively update the display window location after every character or escape sequence is received from the host. This parameter only applies to host input. See the section Window Moving Rules above to see how the terminal responds to local input.

Note: Each quadrant has some locations on the virtual screen that would make it impossible for the terminal to follow this rule.

The default for this parameter is quadrant 4.

The display window is divided into four numbered quadrants:

1. Upper right
2. Upper left
3. Lower left
4. Lower right

Example

The quadrant is set to 4 and the host positions the cursor to row 1, column 1 on the virtual screen. In this case, the terminal cannot adjust the display window so that the cursor ends up in the lower right quadrant since that would mean that quadrants 1, 2 and 3 would end up off the virtual display (see next figure).

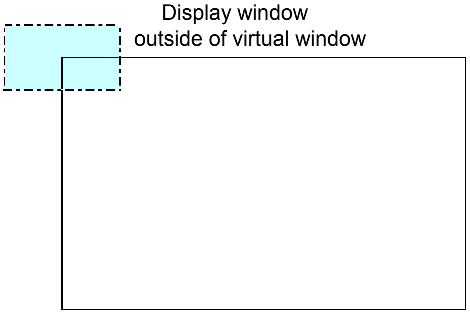


Figure 4-20 Display Window Outside of Virtual Screen

Therefore the terminal places the display window as close as it can to the specified quadrant while still keeping the entire display window within the virtual screen (see next figure).

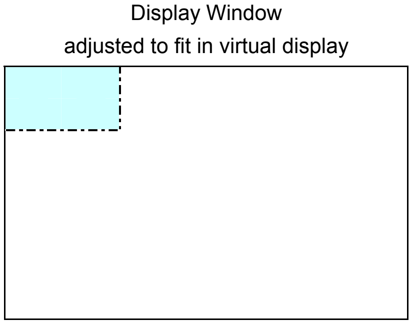


Figure 4-21 Display Window Adjusted to Virtual Screen

Editing Boundary Mode and Cursor Tracking

LXE does not recommend resetting the Editing Boundary Mode while cursor tracking is enabled. While the TE will correctly execute any command with respect to EBM, the window manager may update the physical display after the fact. This may cause the cursor to be located outside the original physical display window.

Physical Display Size and Cursor Tracking

On terminals where the physical display size is not an even divisor of the virtual screen (i.e. a 20x20 physical screen on a 25x80 virtual screen), there will be some overlapping areas on the virtual screen which may show the physical display. In this case the TE will always adjust the physical display towards the top (or left) of the virtual screen.

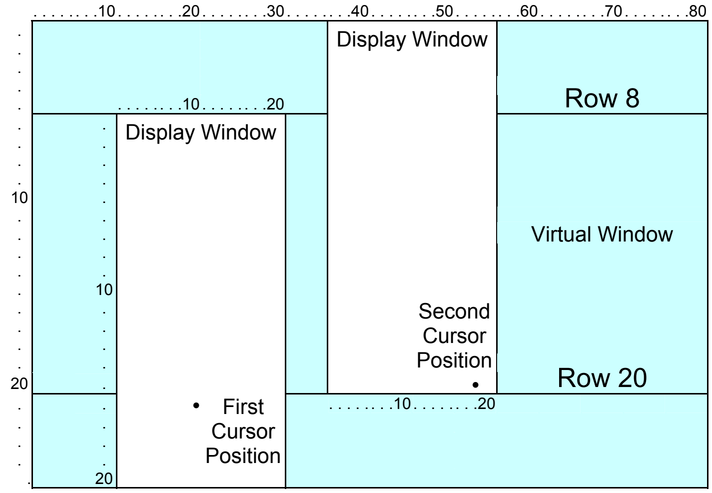


Figure 4-22 When Physical Display Is Not An Even Divisor Of Virtual Screen - Example 1

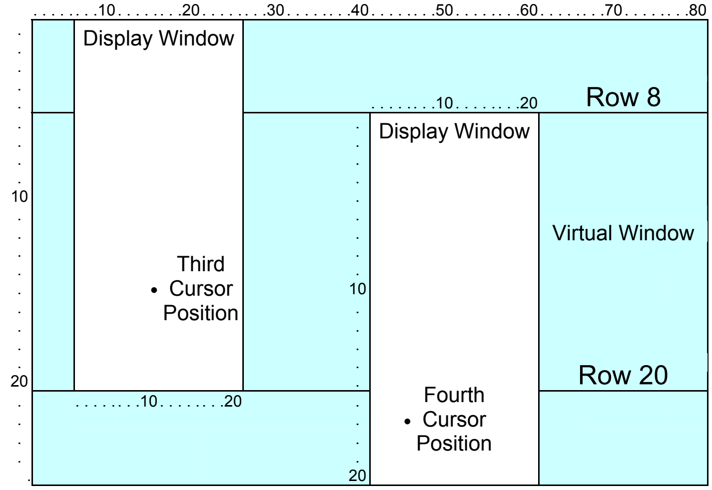


Figure 4-23 When Physical Display Is Not An Even Divisor Of Virtual Screen - Example 2

Status Line Functionality

The status line is row 25 on the virtual screen. The programmer decides whether the status line is displayed some of the time or none of the time by using the LXE private sequence for transmit and screen setup (CSI...+F). It uses up to 80 character positions, and stores data for display.

When the status line is displayed, it overwrites the last line(s) of the display window. The number of lines overwritten depends on the width of the physical display and the length of the status line. The number of lines used for the status line will be:

$$\texttt{int}\left(\frac{\texttt{status line length -1}}{\texttt{physical display width}}\right)\texttt{+1}$$

If the cursor is on the bottom line(s) of the display, the status line can overwrite the cursor position. However, the previous virtual screen appears when the operator enters valid data.

If the status line overwrites the cursor and the terminal operator enters valid data, the status line is removed from the display window and the operator views the entered data.

When the window movement keys are used to move the window or when cursor tracking occurs, the status line is not removed from the display window. The status line always appears at the bottom of the display window.

How To Activate The Status Line

The status line must first be enabled in the config program or by sending the private sequence for transmit and screen setup. Then the programmer must position the cursor to row 25 by using the CUP command. Any text sent to the TE from this point on will be written to the status line.

Text sent past column 80 will be lost. Text will continue to be redirected to the status line until the host positions the cursor to some other line or until the TE user presses the up arrow key. The length of the status line is determined by the location of the last non-blank character sent to the status line. Note that the normal cursor positioning commands work while the cursor is on the status line such as CUF, CUB, CHA, etc. The TE will automatically adjust how many rows are used to display the status line depending on its length. The length of the status line can change, for example, by simply appending text to the existing line or by using the EL command on line 25.

The TE will beep and the user will not be permitted to enter any data if the host has positioned the cursor on the status line. The TE user will however be able to use the cursor keys to move within the status line or off the status line.

When the status line is on screen and the TE user moves into the status line by autowrapping or by using the arrow keys, the status line will automatically be removed. It will not reappear until the host issues another CUP 25 command.

Most commands ignore the status line when the cursor is not positioned on line 25. For example, ED will only erase to the end of line 24 and not include the status line. If the cursor is on line 25, however, the ED will include the status line in its execution.

Table of Contents

CHAPTER 5 CONTROL CODES 5-1

Illustrations

Chapter 5
Control Codes

Overview

Control codes enable a programmer to control or modify an LXE terminal and attached device. ANSI control codes consist of 7-bit and 8-bit control codes. This chapter provides information for the programmer on formatting control codes for LXE equipment. All the control codes supported by LXE are listed in this chapter. If a control code is not listed, LXE does not support it, and it is treated as a no operation (NOP).

Using the 8-Bit Code Table

The 8-bit code table has the same number of rows as the 7-bit code table, but twice as many columns. The codes on the left half of the table (columns 0 - 7) are used in a 7-bit and 8-bit environment. The eighth bit of these codes is 0. The codes on the right half of the table (columns 8 - 15) have an eighth bit of 1. These codes are used in an 8-bit environment and indirectly in a 7-bit environment.

The mnemonics in columns 0 and 1 are C0 (control 0) codes, which can be used in 7-bit environments. The mnemonics in columns 8 and 9 are C1 (control 1) codes, which can be used in 8-bit environments and indirectly in 7-bit environments as escape sequences.

A *code table* is a tool used to group all the characters in a character set with their codes.

A *mnemonic* is an abbreviated name that can be a single control character, escape sequence, or a control sequence.

8-Bit Code Table

Refer to the 8-bit code table below that lists all the decimal, hexadecimal, and octal values for each character:

	COLUMN				0		1		2		3		4		5		6		7	
	BITS b8 b7 b6 b5				0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4	b3	b2	b1																
0	0	0	0	0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0	0	0	1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0	0	1	0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0	0	1	1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0	1	0	0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0	1	0	1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0	1	1	0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0	1	1	1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1	0	0	0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1	0	0	1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1	0	1	0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1	0	1	1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1	1	0	0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1	1	0	1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1	1	1	0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1	1	1	1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

C0 Codes (columns 0–1) — GL Codes (ASCII Graphic) (columns 2–7)

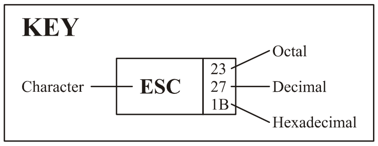


Figure 5-1 8-Bit Code Table, Columns 0 Through 7

8		9		10		11		12		13		14		15		COLUMN				
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		BITS b8 b7 b6 b5				
																b4	b3	b2	b1	ROW
	200 128 80	**DCS**	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0	0	0	0	**0**
	201 129 81	**PU1**	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0	0	0	1	**1**
	202 130 82	**PU2**	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0	0	1	0	**2**
	203 131 83	**STS**	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0	0	1	1	**3**
IND	204 132 84	**CCH**	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0	1	0	0	**4**
NEL	205 133 85	**MW**	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0	1	0	1	**5**
SSA	206 134 86	**SPA**	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0	1	1	0	**6**
ESA	207 135 87	**EPA**	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0	1	1	1	**7**
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1	0	0	0	**8**
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1	0	0	1	**9**
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1	0	1	0	**10**
PLD	213 139 8B	**CSI**	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1	0	1	1	**11**
PLU	214 140 8C	**ST**	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1	1	0	0	**12**
R1	215 141 8D	**OSC**	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1	1	0	1	**13**
SS2	216 142 8E	**PM**	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1	1	1	0	**14**
SS3	217 143 8F	**APC**	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1	1	1	1	**15**

← C1 Codes → ← GR Codes (DEC Supplemental Codes) →

KEY

Character — **ESC** 23 — Octal

27 — Decimal

1B — Hexadecimal

Figure 5-2 8-Bit Code Table, Columns 8 Through 15

C0 Codes: 7-Bit Control Codes

C0 codes are 7-bit control codes that enable the programmer to perform basic functions with LXE terminals and attached devices. This section lists all the C0 codes supported by LXE equipment. The examples in this chapter show applications for each control code. Not all C0 codes are supported by LXE equipment. If a code is not listed, LXE does not support it, and it is treated as a no operation (NOP).

BEL – Bell

Function

BEL causes the terminal to beep.

Note: *Each BEL causes the terminal to beep one time until six or more successive BELs are received by the terminal, which initiates one long beep.*

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
07	^G	(none)

Example

Refer to the following example on how a programmer can format a BEL command in the C programming language:

```
printf("Alert the operator to this message!");
printf("\007\007\007\007");     /*Beep 4 times*/
printf("\007\007\007\007\007\007");/*Beep 1 long beep*/
```

BS – Backspace

Function

BS moves the cursor one space to the left. The cursor stops at the left margin of the virtual display.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
08	^H	(none)

Example

Refer to the following example on how a programmer can format a BS command in the C programming language:

```
printf("The cursor will backspace one space.");
printf("\010");   /*(Octal for hex 08)*/
```

CR – Carriage Return

Function

CR returns the cursor to the left margin (of the virtual screen) of the row that the cursor is currently in.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0D	^M	(none)

Example

Refer to the following example on how a programmer can format a CR command in the C programming language:

```
printf("Return cursor to beginning of \n");
printf("this second line.");
printf("\015");   /*Octal for hex 0D*/
```

DC1/XON – Device Control 1

Function

If XON/XOFF handshaking is enabled and this code is received from the RS-232 port, the terminal will start transmitting bytes, until an XOFF code is received or until the terminal has completed the transmission. This code is ignored when received from the host.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
11	^Q	(none)

DC2 – Device Control 2

Function

DC2 is ignored by the terminal while media copy is enabled. The DC2 code will not be sent to the printer.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
12	^R	(none)

Example

Refer to the following example on how a programmer can format an DC2 command in the C programming language:

```
printf("\033[5i");   /*Enables following text to copy to the
                     RS-232 port*/
printf("This message will copy to the RS-232 port.");
printf("\022");      /*Terminal ignores this code and does
                     not send it to the RS-232 port.*/
printf("\033[4i");   /*Disables RS-232 port*/
```

DC3/XOFF – Device Control 3

Function

If XON/XOFF handshaking is enabled and this code is received from the RS-232 port, the terminal will stop transmitting bytes, until an XON code is received. This code is ignored when received from the host.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
13	^S	(none)

DC4 – Device Control 4

Function

DC4 is an alternative way to turn off media copy.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
14	^T	(none)

Example

Refer to the following example on how a programmer can format an DC4 command in the C programming language:

```
printf("\033[5i");   /*Enables following text to copy to the
                     RS-232 port*/
printf("This message will copy to the RS-232 port.");
printf("\020");      /*Disables RS-232 port*/
```

DEL – Delete

Function

DEL erases a character in the data stream. DEL is for terminal operator use only. The host does not send this command.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
7F	[Delete] *or* [Ctrl] = [8]	(none)

Example

If an operator enters a keystroke by mistake, the operator can press [Del] or [Ctrl]+[8] causing the terminal to backspace, which deletes the last character.

ENQ – Inquiry

Function

ENQ requests a response from the terminal. The terminal responds with an answerback message.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
05	^E	(none)

Default Answerback Message

Refer to the following formats of a default answerback message:

```
LXE/x/number of lines/software-radio-platform-rev./terminal
ID
```

where:

- **x** indicates the keyboard type.
- q = QWERTY keyboard
- a = ABCD keyboard
- **number of lines** is the number of lines on the display.
- **software-radio-platform-rev.** is the software, radio type, platform and software revision.
- terminal ID is the

  900MHz terminal's RF channel identification number.

  2.4GHz terminal’s IP Address. If the terminal “secondary ID” parameter is enabled and a valid secondary ID value programmed, the secondary ID will replace the IP address.

Example:

```
LXE/q/25/APLUSxyy02D/255.255.255.255
```

if extended addressing is ON and

```
LXE/q/25/APLUSxyy02D/0110
```

if extended addressing is OFF

Note: If the terminal “Secondary ID” parameter is enabled and a valid Secondary ID value programmed, the Secondary ID will replace the IP address.

Note: The terminal will always append a carriage return and line feed to the end of the ENQ response.

Example

The following example illustrates how two typical answerback messages appear:

[*900MHz radio*] LXE/q/25/APLUS00101A/0010

[*2.4Gz radio*] LXE/q/20/APLUS00101A/100.118.100.118

Example

Refer to the following example on how a programmer can format an ENQ command in the C programming language:

```
printf("Requesting terminal response now.");
printf("\005");    /*Octal for hex 05*/
gets(input_str);  /*input_str = LXE/q/08/TERMS67001C/0010*/
```

ESC – Escape

Function

ESC extends the set of ANSI Plus controls by acting as a code extension.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
1B	^[	(none)

Example

Refer to the following example on how a programmer can format an ESC command in the C programming language:

```
printf("\033E");  /*Octal for hex 1B. ESC E is a NEL
                command*/
```

FF – Form Feed

Function

FF is processed as a line feed (LF) in LXE terminals and advances the cursor down one line. Refer to the Line Feed control code for a more detailed description.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0C	^L	Scroll Region, LNM

Example

Refer to the following example on how a programmer can format an FF command in the C programming language:

```
printf("This appears on line 1.\n");
printf("\014");   /*(Octal for hex 0C)*/
printf("\014");
printf("This appears on line 4.");
```

See Also

LF

FS – File Separator

Function

FS separates and qualifies data or information logically. The specific meaning of FS must be defined by the programmer for each application.

Control Keys (from keyboard only)	Modes that affect this code...(refer to Appendix A for mode definitions)
(none)	(none)

GS – Group Separator

Function

When the GS control keys are pressed, the terminal will send a control code to the network controller and the network controller will send back a TELNET escape prompt.

Control Keys (from keyboard only)	Modes that affect this code...(refer to Appendix A for mode definitions)
[Ctrl]+[4] or [Ctrl]+[5]	(none)

HT – Horizontal Tab

Function

HT moves the cursor to the next predetermined position (tab stop) to the right, within the virtual screen.

If there are no more tab stops after the cursor, it will jump to the origin of the screen and continue looking for the next tab stop from there.

If no tab stops are defined, the cursor does not move from its original location.

Note: The cursor will not tab to any protected screen cells.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
09	^I	(none)

Example

Refer to the following example on how a programmer can format the HT command in the C programming language:

```
printf("\033[1;25H");   /*CUP-place cursor at 1,25*/
printf("\033H");     /*HTS-Set Tab*/
printf("\033[1;1H");/*CUP-place cursor at 1,1*/
printf("\011");      /*HT-tab to 1,25. Octal for hex 09*/
```

LF – Line Feed

Function

LF moves the cursor down one line.

If the cursor is outside the scrolling region when the Line Feed is received, it will not move past the bottom of the virtual screen. Scrolling does not occur outside the scrolling region.

If the cursor is on the bottom row of the scrolling region when the Line Feed is received and the terminal is in scroll mode, a blank line is inserted at the cursor and the screen scrolls up one (1) row.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0A	^J	Scroll Region, LNM

Example

Refer to the following example on how a programmer can format an LF command in the C programming language:

```
print("\033[20h");      /*Set LNM*/
printf("This appears on line 1.\n");
printf("\012");   /*Octal for hex 0A*/
printf("\012");
printf("This appears on line 4.");
```

LS0 – Lock Shift G0

Function

This command is equivalent to SI (Shift In).

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0F	^O	(none)

See Also

SI for details

LS1 – Lock Shift G1

Function

This command is equivalent to SO (Shift Out).

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0E	^N	(none)

See Also

SO for details

NUL – Null

Function

NUL acts as a space-fill or time-fill character.

Note: You can insert a NUL anywhere in the data stream without affecting the information content.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
00	^@	(none)

Example

Refer to the following example on how a programmer can format a NUL command in the C programming language:

```
printf("There will \000not be any\n");
printf("extra \000characters here.");
```

RS – Record Separator

Function

LXE uses RS as a private remote command to switch the terminal to line mode. When the host sends a message to the terminal to turn line mode on, the terminal returns an RS command to the Network Controller so that the entire line of communication is in line mode. The Network Controller does not send the RS command to the host. This command is reserved for LXE use only.

Control Keys (from keyboard only)	Modes that affect this code...(refer to Appendix A for mode definitions)
[Ctrl]+[6]	(none)

SI – Shift In (Lock Shift G0)

Function

Invokes G0 character set into GL.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0F	^O	(none)

Example

Refer to the following example on how a programmer can format an SI command in the C programming language:

```
printf("\033)0"); /* Designate DEC special graphics into G0.
                  */
printf("\017");   /* Shift G0 into GL */
printf("lqqk\n"); /* These two lines print a small box on
                  the display. */
printf("mqqj");
```

SO – Shift Out (Lock Shift G1)

Function

Invokes G1 character set into GL.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
0E	^N	(none)

Example

Refer to the following example on how a programmer can format an SO command in the C programming language:

```
printf("\033)0"); /* Designate DEC special graphics into G1.
                  */
printf("\016");   /* Shift G1 into GL */
printf("lqqk\n"); /* These two lines print a small box on
                  the display. */
printf("mqqj");
```

US – Unit Separator

Function

LXE uses US as a private remote command to switch the terminal to character mode. When the host sends a message to the terminal to turn character mode on, the terminal returns a US command to the Network Controller so that the entire line of communication is in character mode. The Network Controller does not send the US command to the host. This command is reserved for LXE use only.

Control Keys (from keyboard only)	Modes that affect this code...(refer to Appendix A for mode definitions)
[Ctrl]+[7]	(none)

VT – Vertical Tab

Function

VT is processed as a line feed (LF) in LXE terminals and advances the cursor down one line. Refer to the Line Feed control code for a more detailed description.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
07	^G	Scroll Region, LNM

Example

Refer to the following example on how a programmer can format a VT command in the C programming language:

```
printf("This appears on line 1.\n");
printf("\013");    /*(Octal for hex 0B)*/
printf("\013");
printf("This appears on line 4.");
```

See Also

LF

XON – Transmit On

Function

This command is equivalent to DC1.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
11	^Q	(none)

See Also

DC1 for details

XOFF – Transmit Off

Function

This command is equivalent to DC3.

Hexadecimal Value	Control Code	Modes that affect this code...(refer to Appendix A for mode definitions)
13	^S	(none)

See Also

DC3 for details

C1 Codes: 8-Bit Control Codes

C1 codes are 8-bit control codes that enable the programmer to perform various functions with LXE terminals and attached devices. Some environments operate in 7-bit code; therefore, 8-bit control codes also have 7-bit equivalents. This section lists all the C1 codes supported by LXE equipment. The examples show applications for each control code. LXE supports most of the C1 codes but not all of them. If a code is not listed, LXE does not support it, and it is treated as a no operation (NOP).

CSI – Control Sequence Introducer

Function

CSI indicates the start of a control sequence.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
9B	ESC[	(none)

Example

Refer to the following example on how a programmer can format a CSI command in the C programming language:

```
printf("\033[10C");      /*CUF moves cursor 10 spaces
forward. 033[ represents hex 9B*/
```

DCS – Device Control String

Function

DCS determines the start of a data stream. It is followed by device dependent data such as a down-loadable character set and finally by a string terminator to indicate the end of the data stream.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
90	ESC P	(none)

Example

Refer to the following example on how a programmer can format a DCS command in the C programming language:

```
printf("\033P");  /*Data stream and string terminator
follow*/
```

EPA – End of Protected Area

Function

EPA indicates that the cursor's position is the end of a string of character positions protected from alteration by the terminal and guarded against transmission back to the host. The area from the cursor position forward is marked to accept all input as if a DAQ 0 command was issued from that position.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
97	ESC W	(none)

Example

Refer to the following example on how a programmer can format an EPA command in the C programming language:

```
printf("\033V");  /*SPA*/
printf("This is the start of a protected area\n");
printf("This is the end of a protected area");
printf("\033W");  /*EPA*/
```

See Also

SPA (which precedes an EPA command)

ESA – End of Selected Area

Function

ESA indicates that the cursor position is the end of a string of character positions selected for transmission in a data stream to the host. The area from the cursor position forward is marked to accept all input as if a DAQ 0 command was issued from that position.

Note: The ESA does not initiate the RF transmission.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
87	ESC G	(none)

Example

Refer to the following example on how a programmer can format an ESA command in the C programming language:

```
printf("\033F");  /*SSA-start of selected area*/
printf("This area is selected for transmission");
printf("\033G");  /*ESA-end of selected area*/
```

See Also

SSA (which precedes an ESA command).

HTJ – Horizontal Tab with Justify

Function

HTJ shifts the characters between the preceding tab position and the left of the cursor position, up to the next tab position. The screen cells to the left of the justified text will be space filled.

The cursor will also move to the left of the next tab stop. If the cursor was on a tab stop when the HTJ command was received, the cursor will move to the next tab stop with no justification. This command will act on regular tabs if they exist, otherwise it will act on default tabs. HTJ will not act on area tabs.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
89	ESC I	(none)

Example

Refer to the following example on how a programmer can format an HTJ command in the C programming language:

```
printf("\033[5;5H");  /*CUP-move cursor to 5;5*/
printf("\033H");      /*HTS-tab set*/
printf("\033[5;12H"); /*CUP-move cursor to 5;12*/
printf("\033H");      /*HTS-tab set*/
printf("\033[5;5H");  /*CUP-move cursor to 5;5*/
sleep(10);            /*Wait for operator to enter data*/
printf("\033I");      /*HTJ-justify data to the  tab set*/
gets(s);              /*Wait for operator to press <Enter>
                      and read in data*/
```

HTS – Horizontal Tabulation Set

Function

HTS sets a tab position at the cursor position.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
88	ESC H	TSM

Example

Refer to the following example on how a programmer can format an HTS command in the C programming language:

```
printf("\033[4,20H");    /*CUP-move cursor to 4,20*/
printf("\033H");  /*HTS-Set Tab Stop*/
```

IND – Index

Function

IND moves the cursor down one row without changing column position.

If the cursor is on the bottom row of the scrolling region, the display within the region will scroll up by one (1) row.

No scrolling occurs if the cursor is outside of the scrolling region.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
84	ESC D	Scroll Region

Example

Refer to the following example on how a programmer can format an IND command in the C programming language:

```
printf("\033D");  /*Index-move down 1 row*/
```

NEL – Next Line

Function

NEL moves the cursor to the left margin on the next line.

If the cursor is on the bottom row of the scrolling region, the display within the region will scroll up by one (1) row.

No scrolling occurs if the cursor is outside of the scrolling region.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
85	ESC E	Scroll Region

Example

Refer to the following example on how a programmer can format an NEL command in the C programming language:

```
printf("\033E");  /*NEL-down one line to the left margin*/
```

PM – Privacy Message

Function

PM indicates the start of a string of character positions containing information that will not be displayed on the terminal screen.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
9E	ESC ^	(none)

Example

Refer to the following example on how a programmer can format a PM command in the C programming language:

```
printf("Only the *'s appear on the display\n");
printf("***\033^This will not appear\033\x5c**");
      /*\033\x5c = ST (String Terminator)*/
```

See Also

ST (which follows a PM command)

RI – Reverse Index

Function

RI moves the cursor up one line without changing the column position.

If the cursor is on the top row of the scrolling region, the display within the region will scroll down by one (1) row.

No scrolling occurs if the cursor is outside the scrolling region.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
8D	ESC M	Scroll Region

Example

Refer to the following example on how a programmer can format an RI command in the C programming language:

```
printf("This prints on line 3.");
printf("\015");   /*Carriage Return. Move to  column 1*/
printf("\033M");  /*RI-move up 1 line*/
printf("This prints on line 2.");
```

SPA – Start of Protected Area

Function

SPA indicates the start of a string of character positions protected from alteration by the terminal and guarded against transmission back to the host. This command is equivalent to DAQ 1.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
96	ESC V	(none)

Example

Refer to the following example on how a programmer can format an SPA command in the C programming language:

```
printf("\033V");  /*SPA*/
printf("This is the start of a protected area\n");
printf("This is the end of a protected area");
printf("\033W");  /*EPA*/
```

See Also

EPA (which follows the SPA command)

SS2 – Single Shift G2

Function

SS2 temporarily invokes G2 into GL for the next graphic character sent by the host or entered locally at the terminal.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
8E	ESC N	(none)

Example

Refer to the following example on how a programmer can format an SS2 command in the C programming language:

```
printf("\033*0");    /*Designate DEC Special graphics into
                     G2*/
printf("\033Nl\033Nq\033Nq\033Nk\n");     /*These two lines
                     print a small box on the display*/
printf("\033Nm\033Nq\033Nq\033Nj");
```

SS3 – Single Shift G3

Function

SS3 temporarily invokes G3 into GL for the next graphic character sent by the host or entered locally at the terminal.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
8F	ESC O	(none)

Example

Refer to the following example on how a programmer can format an SS3 command in the C programming language:

```
printf("\033+0"); /*Designate DEC Special graphics into G3*/
printf("\033Ol\033Oq\033Oq\033Ok\n");
printf("\033Om\033Oq\033Oq\033Oj"); /* These two lines print
                  a small box on the display*/
```

SSA – Start of Selected Area

Function

SSA indicates the start of a string of character positions selected for transmission back to the host. This command is similar to DAQ 8, except that SSA allows user input whereas DAQ 8 does not.

Note: The SSA does not initiate the RF transmission.

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
86	ESC F	(none)

Example

Refer to the following example on how a programmer can format an SSA command in the C programming language:

```
printf("\033F");   /*SSA-start of selected area*/
printf("This area is selected for transmission");
printf("\033G");   /*ESA-end of selected area*/
```

See Also

ESA (which follows the SSA command)

ST – String Terminator

Function

ST indicates the end of a privacy message (PM).

Hexadecimal Value	7-Bit Equivalent	Modes that affect this code...(refer to Appendix A for mode definitions)
9C	ESC \	(none)

Example

Refer to the following example on how a programmer can format an ST command in the C programming language:

```
printf("Only the *'s  appear on the display\n");
printf("***\033^This will not appear\033\x5c**");
                  /*\033\x5c = ST (String Terminator)*/
```

See Also

PM

An EMS Technologies Company

Table of Contents

Chapter 6
ANSI Escape Sequences

Overview

This chapter provides information on coding ANSI escape sequences, which extend the functions available with the control codes. If an ANSI escape sequence is not listed, LXE does not support it, and it is treated as a no operation (NOP).

ANSI Escape Sequences

CBT – Cursor Backward Tabulation

Function

CBT moves the cursor left, to the preceding tab position within the virtual screen.

If there are no tab stops before the cursor, it will jump to the lower right corner of the screen and continue looking backward for the next tab stop.

If no tab stops are defined, the cursor does not move from its original position.

Note: The cursor will not tab to any protected screen cells.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnZ	(none)

where:

Pn is the number of tab positions the cursor moves to the left.

Z is the final character.

Parameter Values

Refer to the following table for the movement of the cursor according to the value that you enter:

When you enter...	Then the cursor moves...
0	back one tab stop because the value defaults to 1 when you enter 0.
1 (default)	back one tab stop.
a number greater than 1	back the number of tab stops the value indicates within the bounds of the virtual screen.

Example

Refer to the following example on how a programmer can format a CBT command in the C programming language:

```
printf("\033[1;25H");    /*CUP-place cursor at row 1, column
                         25*/
printf("\033H");         /*HTS-set tab set*/
printf("\033[1;35H");    /*CUP-place cursor at row 1, column
                         35*/
printf("\033[1Z");       /*CBT-move cursor back to tab set*/
```

CHA – Cursor Horizontal Absolute

Function

CHA moves the cursor to the column (on the current row) that the parameter indicates. If a number greater that 80 is entered, the cursor will move to column 80 by default.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnG	(none)

where:

Pn is the column number the cursor moves to.

G is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	to column one because the value defaults to 1 when you enter 0.
1 (default)	to column one.
a number greater than 1	to the column the value indicates in the current line.

Example

Refer to the following example on how a programmer can format a CHA command in the C programming language:

```
printf("\033[7G"); /*move cursor to column 7 of the current
line*/
```

CHT – Cursor Horizontal Tab

Function

CHT moves the cursor right, to the next tab position within the virtual screen. If there are no tab stops found after the cursor, it will jump to the origin of the screen and continue looking for the next tab stop. If no tab stops are defined, the cursor does not move from its original position.

Note: The cursor will not tab to any protected screen cells.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnI	(none)

where:

Pn is the number of tab positions the cursor moves to the right.

I is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	forward one tab position because the value defaults to 1 when you enter 0.
1 (default)	forward one tab position.
a number greater than 1	forward the number of tab positions that the value indicates within the bounds of the virtual screen

Example

Refer to the following example on how a programmer can format a CHT command in the C programming language:

```
printf("\033[1;25H");     /*CUP-place the cursor at row 1,
column 25*/
printf("\033H");    /*HTS-tab stop*/
printf("\033[1;1H");      /*CUP-move cursor up to  row 1,
column 1*/
printf("\033[1I"); /*CHT-place cursor horizontal tab*/
```

CNL – Cursor Next Line

Function

CNL moves the cursor down one or more lines to the left margin from the cursor's current position. The cursor will stick at the bottom margin if a large number is entered.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnE	EBM

where:

Pn is the number of lines the cursor moves down.

E is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	to the left margin on the next line from its current position because the value defaults to 1 when you enter 0.
1 (default)	to the left margin on the next line from its current position.
a number greater than 1	down, as many lines as the value indicates, to the left margin.

Example

Refer to the following example on how a programmer can format a CNL command in the C programming language:

```
printf("\033[1;20H");    /*CUP-place cursor at row 1, column
20*/
printf("\033[1E"); /*CNL-move cursor to next line, column 1*/
```

CPL – Cursor Preceding Line

Function

CPL moves the cursor up one or more lines to the left margin from the cursor's current position. The cursor will stick at the top margin if a large number is entered.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnF	EBM

where:

Pn is the number of lines the cursor moves up.

F is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	up one line to the left margin because the value defaults to 1 when you enter 0.
1 (default)	up one line to the left margin.
a number greater than 1	up, as many lines as the value indicates, to the left margin.

Example

Refer to the following example on how a programmer can format a CPL command in the C programming language:

```
printf("\033[3;20H");    /*CUP-move the cursor to row 3,
column 20*/
printf("\033[2F"); /*CPL-move the cursor up 2 lines*/
```

CPR – Cursor Position Report

Function

CPR reports the cursor's position to the host. DECOM is ignored when reporting the cursor position. The position reported is always relative to the origin of the virtual screen.

Note: This report is sent from the terminal to the host in response to a device status report 6 (DSR 6).

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pr;PcR	(none)

where:

Pr is the row position.

Pc is the column position.

R is the final character.

Example

Refer to the following example of a CPR sent from the terminal to the host:

```
printf("\033[2;5H");      /*CUP-Place cursor*/
printf("\033[6n"); /*DSR-Request cursor position*/
gets(s);      /*String returned=
 "\033[2;5R"*/
```

Note: This example requires that the [Enter] key is pressed to exit from the `gets` *command.*

See Also

DSR

CTC – Cursor Tabulation Control

Function

CTC sets or clears the horizontal tab positions.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnW	(none)

where:

Pn is the tab position you want to set.

W is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's tab position:

When you enter...	Then this parameter value...
0 (default)	with TSM set, sets the tab stop at the cursor's position. with TSM reset, sets the tab stop at the cursor's column, on all rows in the virtual screen.
1	LXE does not support this parameter value.
2	with TSM set, clears the tab stop at the cursor's position. with TSM reset, clears the tab stop at the cursor's column, on all rows in the virtual screen.
3	LXE does not support this parameter value.
4	with TSM set, clears all of the tab stops on the cursor's row in the virtual screen. with TSM reset, clears all of the tab stops on the virtual screen.
5	with TSM set or reset, clears all of the tab stops on the virtual screen.
6	LXE does not support this parameter value.

Example

Refer to the following example on how a programmer can format a CTC command in the C programming language:

```
printf("\033[5W");       /*clear all horizontal tab
                  positions*/
```

CUB – Cursor Back

Function

CUB moves the cursor one or more positions to the left, up to the left margin. The cursor will stick at the left margin if a large number is entered.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnD	EBM

where:

Pn is the number of character spaces the cursor moves to the left.

D is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	to the left one space because the value defaults to 1 when you enter 0.
1 (default)	to the left one space.
a number greater than 1	to the left the number of spaces the value indicates up to the left margin.
	Note: Each line contains 80 character positions.

Example

The following example illustrates how a programmer can format a CUB command in the C programming language:

```
printf("\033[4D");  /*CUB-move cursor back 4 spaces*/
```

CUD – Cursor Down

Function

CUD moves the cursor down one or more lines, without changing the cursor's column position.

If the cursor is outside the scrolling region when this command is received, it will not move past the bottom of the virtual screen.

If the cursor is inside the scrolling region, it will not move past the bottom row of the scrolling region.

If Editing Boundary Mode (EBM) is set to the physical display, the scrolling region is ignored and the cursor will not move past the bottom of the physical display.

Note: Scrolling does not occur with this command.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnB	Scroll Region, EBM

where:

Pn is the number of lines the cursor moves down.

B is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	down one line in the same column because the value defaults to 1 when you enter 0.
1 (default)	down one line in the same column.
a number greater than 1	down the number of lines the value indicates without changing column position.

Example

Refer to the following example on how a programmer can format a CUD command in the C programming language:

```
printf("\033[4B"); /*CUD-move cursor down 4 lines*/
```

CUF – Cursor Forward

Function

CUF moves the cursor forward (to the right) one or more spaces up to the right margin. The cursor will stick at the right margin if a large number is entered.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnC	EBM

where:

Pn is the number of character positions the cursor moves to the right.

C is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	one position to the right because the value defaults to 1 when you enter 0.
1 (default)	one position to the right.
a number greater than 1	to the right the number of times the value indicates up to the right margin.
	Note: Each line contains 80 character positions.

Example

Refer to the following example on how a programmer can format a CUF command in the C programming language:

```
printf("\033[4C"); /*CUF-moves cursor forward 4 spaces*/
```

CUP – Cursor Position Absolute

Function

CUP moves the cursor to a specified row/column position. If DECOM (Origin Mode) is set, the cursor is moved relative to the scrolling region. In this case, the origin of the scrolling region is considered (1,1). If DECOM is reset, the cursor is moved to the absolute position within the virtual screen. In this case the origin of the virtual screen is considered (1,1).

If the row or column positions are out of range, the terminal will adjust them and be within range of the appropriate display.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pr,PcH	Scroll Region, DECOM

where:

Pr is the row position.

Pc is the column position.

H is the final character.

Parameter Values

Refer to the following table for the values that you can enter for CUP:

Parameter	Value	
Pr	1-25	Line 25 is the status line.
Pc	1-80	

Note: If you enter 0 for Pr or Pc, the parameter defaults to 1.

Example

Refer to the following example on how a programmer can format a CUP command in the C programming language:

```
printf("\033[3;17H");     /*CUP-move cursor to row 3, column
17*/
```

CUU – Cursor Up

Function

CUU moves the cursor up one or more lines, without changing the cursor's column position.

If the cursor is outside the scrolling region when this command is received, it will not move past the top of the virtual screen.

If the cursor is inside the scrolling region, it will not move past the top row of the scrolling region.

If Editing Boundary Mode (EBM) is set to the physical display, the scrolling region is ignored and the cursor will not move past the bottom of the physical display.

Note: Scrolling does not occur with this command.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnA	Scroll Region, EBM

where:

Pn is the number of lines the cursor moves up.

A is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the cursor moves...
0	up one line because the value defaults to 1 when you enter 0.
1 (default)	up one line.
a number greater than 1	up the number of lines the value indicates up to the top of the virtual screen.

Example

Refer to the following example on how a programmer can format a CUU command in the C programming language:

```
printf("\033[4A");  /*CUU-move cursor up 4 lines*/
```

CVT – Cursor Vertical Tab

Function

CVT is processed the same as the Cursor Next Line command (CNL).

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnY	EBM

where:

Pn is the number of lines the cursor moves down.

Y is the final character.

See

CNL for more details.

DA Primary – Primary Device Attribute

Function

DA Primary is sent from the host to the terminal requesting the terminal's service class code, conformance level, and supported extensions.

Escape Sequence	Modes that affect this code...
ESC[c or ESC[0c	(none)

where:

0 is a constant.

c is the final character.

Response

The terminal's response is in the following format:

```
CSI?Psc;Psnc
```

String Elements of a Response

Refer to the table below for a description of the string elements in the terminal's response to the DA Primary command:

String Elements	Description
?	Indicates that private parameters follow.
Psc	Indicates the terminal's service class code. The value for Psc reflects the terminal's operating level: PSC Operating Level 62 Level 2 (VT200 family)
;	Required delimiter that must precede all parameters.
Psn	Parameter indicating that the terminal supports the following extension: 6 selective erase Example: Refer to the following message that an LXE terminal can send to the host in response to a DA command: `CSI[?62;6c` where: 62 Level 2 (VT200 family) 6 Selective erase
c	Final character.

Example

Refer to the following example on how a programmer can format a DA command in the C programming language:

```
printf("\033[c");   /*host requesting a terminal's service
code and supporting extensions*/
```

DA Secondary – Secondary Device Attribute

Function

DA Secondary is sent from the host to the terminal requesting the terminal's type, version number and installed options.

Escape Sequence	Modes that affect this code...
ESC>c or ESC>0c	(none)

where:

0 is a constant.

c is the final character.

Response

The terminal's response is in the following format:

`CSI>1;Pv;Poc`

String Elements of a Response

Refer to the table below for a description of the string elements in the terminal's response to the DA Secondary command:

String Elements	Description
>	Indicates that private parameters follow.
1	1 indicates VT220 type terminal.
Pv	Parameter indicating the software version number. "10" means 1.0.
Po	0 indicates no options installed.
c	Final character.

DAQ – Define Area Qualifications

Function

DAQ indicates that the cursor's position is the start of a string of consecutive character positions that constitute a qualified area. The end of a qualified area is indicated by the beginning of the next qualified area. A qualified area restricts the type of input the terminal operator enters.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pno	(none)

where:

Pn indicates the type of data that the terminal accepts.

o is the final character.

Multiple parameters may be sent in a single DAQ command if they are separated by semi-colons.

Parameter Values

Refer to the following table for the effect the value that you enter has on the qualified area:

When you enter...	Then the qualified area...
0 (default)	accepts all data.
1	does not accept any input and is not selected to transmit. *Note: Entering 1 for this value is equivalent to the 8-bit control code start of protected area (SPA) discussed in C1 Codes: 8-Bit Control Codes.*
2	accepts only displayable characters, 20-7E hex.
3	accepts only numeric characters.
4	accepts only alphabetic characters.
5	right justifies the data.
6	is filled with zeros.
7	has a horizontal tab stop set at the start of area.
8	does not accept any input, but it is selected to transmit.
9	is filled with spaces.
10	accepts input, but the characters are invisible.
15	accepts only keyboard input.

Every new DAQ command that the terminal receives will always overwrite the existing DAQ attribute type from the cursor to the end of the area.

Note: Parameter types 6 and 9 are executed immediately and are not stored in the TE.

Rule

All DAQ parameters are supported in block mode. Parameters 0, 2, 3, 4 and 10 are supported in scroll mode. No parameters are supported in character mode.

However, if multiple DAQ parameters are specified in a single command, they will combine as described in the table below.

Parameter in DAQ command	**+**	**Previous Parameter in DAQ command**	**=**	**Parameter Stored in TE**
All data (0)	+	any	=	All data
Guarded (1)	+	any	=	Guarded
Displayable (2)	+	any	=	Displayable
Numeric (3)	+	Alphabetic	=	Alphanumeric
Numeric (3)	+	any other	=	Numeric
Alphabetic (4)	+	Numeric	=	Alphanumeric
Alphabetic (4)	+	any other	=	Alphabetic
Right Justify (5)	+	any	=	Previous+Right Justified
				(If 5 is the only parameter received, the area will be an “All data, right justified” one.)
Regular Tab (7)	+	any	=	All data+Regular Tab Stop
				(Ignores TSM)
Selected (8)	+	any	=	Selected

For example, if the host specified `ESC[1;5;0o`, the terminal would mark the area as simply “All data” since `0` was the last parameter. However, if the host specified `ESC[3;4;5o`, the terminal would mark the area as alpha-numeric and right justified.

Example

Refer to the following example on how a programmer can format a DAQ command in the C programming language:

```
printf("\033[1o"); /start of a protected area*/
printf("Accept numeric character input only.\n");
printf("\033[3o"); /*Define numeric entry area start*/
printf("____");    /*4 underscore characters*/
printf("\033[1o"); /*define numeric entry area stop*/
printf("\033[4D"); /*CUB-moves the cursor back 4 spaces*/
gets(s);      /*string returned will be numeric value only*/
```

DCH – Delete Character

Function

DCH deletes the character(s) at the cursor position and shifts all the characters right of the cursor to the left (toward the cursor).

The terminal will not delete past the first protected screen cell to the right of the cursor. If the cursor is on a protected cell when the DCH command is received, it will move one position to the right (i.e. CUF 1). No deletion takes place in this case.

Note: The vacated character position at the other end is erased.

Rule

Block mode must be enabled to use DCH.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnP	SEM, EBM

where:

Pn is the number of characters to delete.

P is the final character.

Parameter Values

Refer to the following table for the effect the value that you enter has on the cursor's position:

When you enter...	Then the terminal...
0	deletes one character because the value defaults to 1 when you enter 0.
1 (default)	deletes one character.
a number greater than 1	deletes the number of characters the value indicates.

Example

Refer to the following example on how a programmer can format a DCH command in the C programming language:

```
printf("All X's on this XX line will be deleted");
printf("\033[23D");/*CUB-23 spaces back*/
printf("\033[2P"); /*DCH-deletes 2 x's*/
printf("\033[12D");/*CUB-12 spaces back*/
printf("\033[1P"); /*DCH-deletes x's remaining*/
```

DL – Delete Line

Function

DL deletes the line the cursor is in and shifts all following lines up. If the cursor is outside the scrolling region, this command is ignored.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnM	EMB, Scroll Region

where:

Pn is the number of lines the device deletes.
M is the final character.

Parameter Values

Refer to the following table for the number of lines the terminal deletes according to the value that you enter:

When you enter...	Then the terminal deletes...
0	one character because the value defaults to 1 when you enter 0.
1 (default)	one line.
a number greater than 1	the number of lines the value indicates without going beyond the bottom of the virtual screen or the scrolling region.

Example

Refer to the following example on how a programmer can format a DL command in the C programming language:

```
printf("This line will be deleted.\n");
printf("This line will be deleted.\n");
printf("This line moves up and is not deleted.\n");
printf("\033[3A"); /*CUU-cursor moves up 3 lines*/
printf("\033[2M"); /*DL-2 lines are deleted and remaining
lines move up*/
```

DMI – Disable Manual Input

Function

DMI disables keyboard and RS-232 input. This command is equivalent to SM KAM.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC`	(none)

Example

Refer to the following example on how a programmer can format a DMI command in the C programming language:

```
printf(“\033`”);    /* Keyboard and RS-232 now locked */
```

DSR – Device Status Report

Function

DSR requests information or reports information. DSR from the host requests information from the terminal receiving the sequence (Pn = 5, 6). DSR from the terminal reports the status of the terminal (Pn = 0).

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pnn	(none)

where:

Pn is either a value that you enter (5 or 6) or it is a value returned from the terminal (0).

n is the final character.

Parameters Indicating a Request

Refer to the following table for the requests the host sends to a terminal according to the parameter value:

Note: *Parameter 5, 6 and 26 do not append a CR or LF to the response. Parameters 90, 92 and 93 append CR and LF as if the [SEND] key was pressed.*

Parameter	Report that the Host is Requesting
5	Status Report. *Note:* *The terminal returns a CSI0n indicating ready status.*
6	Cursor Position Report (CPR).

Parameter	Report that the Host is Requesting
26	Keyboard report. The TE sends the sequence `CSI?27;Pnn` (where Pn is the keyboard type). The keyboard type may be set in the configuration utility.
90	Sends the TE display, just as if the terminal operator had pressed the [Enter] key. However, if the cursor is in a justified area, the area will not be justified before the display is sent. The TE will lock its keyboard until it receives more host input or [Alt]+[U] is pressed.
92	Sends the TE display just as if the terminal operator had pressed [Enter]. If the cursor is in a justified area, the area will be justified before the display is sent. The TE will lock its keyboard until it receives more host input or [Alt]+[U]is pressed.
93	Same as 90, except the TE does not lock its keyboard.
94	Sends the upper left and lower right coordinates of the physical display in the format `CSI?UL Row;UL Col;LR Row,LR Col R` The coordinates are 1-based.

Note: DSR 90 - 94 are useful for debugging host applications.

Response to a CPR Request

The terminal sends a CPR response when the host requests a CPR. Refer to *CPR* in this section for more information on the cursor position report.

Note: A carriage return (CR) and a line feed (LF) are not returned in the response to a CPR request.

Example

Refer to the following example on how a programmer can format a DSR command that requests a terminal to send a status report in C programming language:

```
printf("\033[5n");
gets (s);    /*string returned is \033[0n for a ready
status*/
```

Note: The example above requires that the [Enter] key be pressed at the terminal to exit from the `gets` *command.*

EA – Erase in Area

Function

EA erases characters in the qualified area where the active position (cursor's position) resides according to the value that you enter. This command will not erase the type of qualified area that the cursor is in.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnO	ERM, EBM

where:

Pn indicates what portion of the area is erased.
O is the final character.

Parameter Value

Refer to the following table on how characters are erased in the qualified area:

When you enter...	Then the terminal ...
0 (default)	erases characters from the cursor to the end of the qualified area.
1	erases characters from the start of the qualified area to the cursor.
2	erases all characters in the qualified area.

Example

Refer to the following example on how a programmer can format an EA command in the C programming language:

```
printf("\033[6l"); /*Reset ERM*/
printf("\033[1o")  /*DAQ-Start of protected area*/
printf("This will not be erased.\n");
printf("\033[0o"); /*DAQ-Accepts all input*/
printf("_ _ _ _ \033[1o");      /*DAQ-Accept no input*/
printf("\033[4D"); /*CUB-4 spaces*/
printf("\033[2O"); /*EA-Erase in Area-all*/
```

ECH – Erase Character

Function

ECH erases the character at the cursor's position and possibly other characters to the right of the cursor, depending on the value that you enter. If a large number of characters to erase is specified, the erase will continue on succeeding lines.

Note: *The active (cursor) position is unchanged.*

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnX	ERM, EBM

where:

Pn is the number of characters to erase.
X is the final character.

Parameter Values

Refer to the following table on how the terminal erases characters according to the value that you enter:

When you enter...	Then the terminal ...
0	erases one character because the value defaults to 1 when you enter 0.
1 (default)	erases one character.
a number greater than 1	erases the number of characters the value indicates.

Example

Refer to the following example on how a programmer can format an ECH command in the C programming language:

```
printf("This Z will be erased.");
printf("\015");     /*Carriage return*/
printf("\033[5c");  /*Cursor forward 5 spaces*/
printf("\033[1X");  /*Erase character Z*/
```

ED – Erase in Display

Function

ED erases some or all of the characters in the display according to the value that you enter. This command will erase the type of qualified area that the cursor is in. It will also erase qualified area boundaries.

If EBM is set to the virtual screen, every row that is completely erased will have its line attribute reset to single high and single wide.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnJ	ERM, EBM

where:

Pn indicates what portion of the display is erased.

J is the final character.

Parameter Values

Refer to the following table on how the terminal erases characters according to the value that you enter:

When you enter...	Then the terminal ...
0 (default)	erases characters starting with the cursor position to the end of the display.
1	erases the characters from the start of the display to the cursor.
2	erases all the characters on the display. The cursor is moved to the origin of the physical display or virtual screen, depending on the EBM setting.

Example

Refer to the following example on how a programmer can format an ED command in the C programming language:

```
printf("\033[2J"); /*Erases entire display*/
```

EF – Erase in Field

Function

EF erases characters in the current field depending on the value that you enter. This command will not erase the type of qualified area that the cursor is in.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnN	EBM, ERM

where:

Pn indicates what portion of the current field is erased.

N is the final character.

Parameter Values

Refer to the following table on how the terminal erases characters in a field according to the value that you enter:

When you enter...	Then the terminal ...
0 (default)	erases characters from the cursor position to the end of the field.
1	erases characters from the beginning of the field to the cursor position.
2	erases the entire field that the cursor is in.

Example

Refer to the following example on how a programmer can format an EF command in the C programming language:

```
printf("\033[5W"); /*CTC-Clear all tabs*/
printf("\033[0W"); /*CTC-Set a tab*/
printf("_ _ _ _ \033[0W");     /*CTC-Set a tab*/
printf("\033[3D"); /*CUB-3 spaces*/
printf("\033[2N"); /*EF-Erases all of field*/
```

EL – Erase in Line

Function

EL erases characters in the line the cursor is in according to the value that you enter. This command will erase the type of qualified area that the cursor is in. It will also erase qualified area boundaries.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnK	EBM, ERM

where:

Pn indicates what portion of the current line is erased.

K is the final character.

Parameter Values

Refer to the following table on how the terminal erases characters according to the value that you enter:

When you enter...	Then the terminal ...
0 (default)	erases characters from the cursor position to the end of line.
1	erases characters from the beginning of the line to the cursor.
2	erases the entire line that the cursor is in.

Example

Refer to the following example on how a programmer can format an EL command in the C programming language:

```
printf("\033[6l"); /*Reset ERM*/
printf("\033[1o"); /*DAQ-Start of protected area*/
printf("This will not be erased.\n");
printf("\033[0o"); /*DAQ-Accepts all input*/
printf("_ _ _ _ \033[1o");     /*DAQ-Accepts no input*/
printf("\033[4D"); /*CUB-4 spaces*/
printf("\033[2K"); /*Erase in Line*/
```

EMI – Enable Manual Input

Function

EMI enables keyboard and RS-232 input. This command is equivalent to RM KAM.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESCb	(none)

Example

Refer to the following example on how a programmer can format an EMI command in the C programming language:

```
printf(“\033b”);   /* Keyboard and RS-232 now unlocked */
```

HVP – Horizontal Vertical Position

Function

HVP moves the cursor to the specified row/column position. HVP is the same as the CUP command in LXE Legacy terminals. Refer to the CUP command for more details.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pr;Pcf	Scroll Region, DECOM

where:

Pr is the row position.

Pc is the column position.

f is the final character.

Parameter Values

Refer to the following table for the values that you can enter for HVP:

Parameter	Value	
Pr	1 - 25	(Line 25 is the status line.)
Pc	1 - 80	

Note: If you enter 0 for Pr or Pc, the parameter defaults to 1.

Example

Refer to the following example on how a programmer can format an HVP command in the C programming language:

```
printf("\033[3;17f");    /*Move the cursor to row 3, column
17*/
```

ICH – Insert Character

Function

ICH shifts all characters from the cursor to the right and inserts a space character at the cursor position.

The terminal will not insert past the first protected screen cell to the right of the cursor. If the cursor is on a protected cell when this command is received, it will move one position to the right (i.e. CUF 1). No insertion takes place in this case.

Escape Sequence	**Modes that affect this code... (refer to Appendix A for mode definitions)**
ESC[Pn@	EBM, SEM

where:

Pn is the number of character spaces the cursor shifts to the right.

@ is the final character.

Rule

Block mode must be enabled for ICH to work.

Parameter Values

Refer to the table below on how the characters move according to the value that you enter:

When you enter...	**Then the terminal shifts the characters...**
0	one space to the right because the value defaults to 1 when you enter 0.
1 (default)	one space to the right.
a number greater than 1	the number of spaces the value indicates to the right.

Example

Refer to the following example on how a programmer can format an ICH command in the C programming language:

```
printf("Insert space here");
printf("\015");     /*CR-carriage return*/
printf("\033[6C"); /*CUF-cursor forward*/
printf("\033[1@"); /*ICH-insert space character*/
printf("\033[1C"); /*CUF-cursor forward*/
printf("\033[1@"); /*ICH-insert space character*/
```

IL – Insert Line

Function

IL inserts one or more blank lines at the cursor's position and shifts all following lines down. If the cursor is outside the scrolling region, this command is ignored.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnL	EBM, Scroll Region

where:

Pn is the number of blank lines inserted at the cursor's position.

L is the final character.

Parameter Values

Refer to the following table on how to insert blank lines at the cursor's position:

When you enter...	Then the terminal inserts...
0	one blank line because the value defaults to 1 when you enter 0.
1 (default)	one blank line.
a number greater than 1	the number of blank lines the value indicates at the cursor's position. The inserted lines do not go beyond the bottom of the virtual screen or the scrolling region.

Example

Refer to the following example on how a programmer can format an IL command in the C programming language:

```
printf("This line will move down 2 lines.");
printf("\015");     /*CR-carriage return*/
printf("\033[2L");  /*Insert 2 lines*/
```

MC – Media Copy

Function

MC enables or disables the RS-232 port. When enabled, the terminal sends received characters to the RS-232 port without displaying them on the screen. All characters (except: Nul, XON, XOFF, DC4, DC2, CSI5i and CSI4i) are sent to the port. When disabled, the terminal sends received chars to the display. MC has a higher priority than autoprint mode.

Note: An alternative way to disable MC is to send a DC4.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pni	(none)

where:

Pn indicates whether the RS-232 port is off or on.

i is the final character.

Parameter Values

Refer to the table below on how to enable or disable the RS-232 port:

When you enter...	Then the RS-232 port is ...
4	off.
5	on.

Example

Refer to the following example on how a programmer can format an MC command in the C programming language:

```
printf("\033[5i"); /*Enables following text to copy to the
RS-232 port*/
printf("This message will copy to the RS-232 port.");
printf("\033[4i"); /*Disables RS-232 port*/
```

NP – Next Page

Function

NP moves the display window and the cursor forward the number of pages indicated by the value that you enter. If a large number of pages is entered, the display window will move to the lower right corner of the virtual screen.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnU	(none)

where:

Pn is the number of pages to advance forward.

U is the final character.

Parameter Values

Refer to the following table on how the display window and cursor move forward the number of pages indicated by the value that you enter.

Note: The terminal will automatically calculate the page size based on the size of the physical display. The number of pages across the virtual display is 80/(physical display width). This command will always move the physical display window by full window units.

When you enter...	Then the display window and cursor move forward ...
0	one page because the value defaults to 1 when you enter 0.
1 (default)	one page.
a number greater than 1	the number of pages the value indicates within the bounds of the virtual display.

Example

Refer to the following example on how a programmer can format an NP command in the C programming language for a 40 column terminal:

```
printf("\033[1;1H");      /*CUP-move cursor to 1;1*/
printf("Page 1");
printf("\033[1;41H");     /*CUP-move to 1;41*/
printf("Page 2");
printf("\015");     /*CR-carriage return*/
printf("\033[1U"); /*Next page-to page 2*/
```

PP – Previous Page

Function

PP moves the display window and the cursor to the previous page of the virtual screen indicated by the value that you enter. If a large number of pages is entered, the display window will move to the origin of the virtual screen.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnV	(none)

where:

Pn is the number of pages to move back.

V is the final character.

Parameter Values

Refer to the table below on how the display window and cursor move to the previous page according to the value that you enter.

Note: *The terminal will automatically calculate the page size based on the size of the physical display. The number of pages across the virtual display is 80/(physical display width). This command will always move the physical display window by full window units.*

When you enter...	Then the display window and cursor move back ...
0	one page because the value defaults to 1 when you enter 0.
1 (default)	one page.
a number greater than 1	the number of pages the value indicates.

Example

Refer to the example below on how a programmer can format a PP command in the C programming language for a 40 column terminal:

```
printf("\033[1;1H");      /*CUP-move to 1;1*/
printf("Page 1");
printf("\033[1;41H");     /*CUP-move to 1;41*/
printf("Page 2");
printf("\033[1V");  /*Previous page*/
```

REP – Repeat Character

Function

REP repeats the last character transmitted by the host the number of times indicated by the value. Only characters are repeated. Escape sequences are not repeated.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pnb	(none)

where:

Pn is the number of times a character is repeated.

b is the final character.

Parameter Values

Refer to the table below on how many times the character is repeated according to the value that you enter:

When you enter...	Then the terminal repeats the character ...
0	once because the value defaults to 1 when you enter 0.
1 (default)	once.
a number greater than 1	the number of time that the value indicates.

Example

Refer to the following example on how a programmer can format an REP command in the C programming language:

```
printf("The number 3 is repeated:\n);
printf("4 times-3");
printf("\033[4b"); /*repeat 4 times*/
```

RM – Reset Mode

Function

RM resets one or more modes for the device indicated by the value that you enter. Several modes can be reset with the same command if the parameter values are separated by semicolons. They can be sent in any order except #3.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pnl	(none)

where:

Pn indicates which mode to reset.

l (lower-case L) is the final character.

Parameter Values

Refer to the following table for the modes RM resets:

Parameter	Mode to Reset	Description
1	Guarded Area Transfer Mode (GATM)	Enables the terminal to transmit only the unguarded areas to the host and auxiliary device. See the “Sending the Terminal Screen to the Host” section for more details.
2	Keyboard Action Mode (KAM)	Unlocks the keyboard and RS-232 data entry.
3	Control Representation Mode (CRM)	Enables the terminal to execute the control codes. This parameter value must immediately precede the final character.
4	Insert-Replace Mode (IRM)	Enables the terminal to replace the information during editing by overwriting the content of the line.
6	Erasure Mode (ERM)	Determines that the editing commands such as EA cannot erase protected characters.
12	Send-Receive Mode (SRM)	Enables local echo in the terminal so that the terminal operates in scroll or block mode. Character mode communication is disabled.
15	Multiple Area Transmit Mode (MATM)	Enables the terminal to transmit only the selected area containing the cursor. See the “Sending the Terminal Screen to the Host” section for more details.
17	Selected Area Transmit Mode (SATM)	Enables the terminal to transmit only the selected areas. See the “Sending the Terminal Screen to the Host” section for more details.

Parameter	Mode to Reset	Description
18	Tab Stop Mode (TSM)	Determines that the tab stop settings affect the entire virtual display.
19	Editing Boundary Mode (EBM)	Determines that the editing commands affect the display window only. A scrolling region can not be defined in this case.
20	Line Feed/New Line Mode (LNM)	Enables the terminal to perform only a line feed while the cursor remains in its current column.

Example

Refer to the following example on how a programmer can format an RM command in the C programming language:

```
printf("\033[4l")  /*reset IRM*/
```

See Also

Refer to *SM* in this section for more information on how to set the modes discussed in the previous table.

SD – Scroll Down

Function

SD moves the entire contents of the scrolling region down the number of lines indicated by the value that you enter. If the cursor is outside the scrolling region, this command is ignored. The cursor will scroll with the display.

Note: If the number of lines exceed the boundaries of the scrolling region, a blank line moves into the top line of the display window. The number of blank lines that move into the top of the display window depends on how many lines exceed the bounds of the scrolling region.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnT	Scroll Region

where:

Pn is the number of lines to move down.

T is the final character.

Parameter Values

Refer to the table below on how many lines the contents of the display window scroll down according to the value that you enter:

When you enter...	Then the contents of the display window scroll down ...
0	one line because the value defaults to 1 when you enter 0.
1 (default)	one line.
a number greater than 1	the number of lines the value indicates.

Example

Refer to the following example on how a programmer can format an SD command in the C programming language:

```
printf("\033[2T"); /*Scroll down 2 lines*/
```

SEM – Select Editing Extent Mode

Function

SEM indicates the extent to which the delete character (DCH) command and the insert character (ICH) command affect the display window.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnQ	EBM

where:

Pn indicates how much of the display window is affected by the editing commands.

Q is the final character.

Parameter Values

Refer to the following table for the affect that SEM has on the DCH and ICH editing commands:

Note: EBM controls whether SEM works within the virtual or physical display.

When you enter...	Then DCH and ICH commands affect...
0 (default)	the entire display.
1	only the row the cursor is in.
2	only the field containing the cursor (between horizontal tab stops).
3	only the selected area containing the cursor (defined by DAQ, SPA and EPA).

Example

Refer to the following example on how a programmer can format an SEM command in the C programming language:

```
printf("\033[3Q"); /*SEM-3*/
```

See Also

Refer to the “DAQ” section in this chapter for more information on define area qualification. Refer to the “EPA” and “SPA” sections of Chapter 5 “Control Codes” for more information on end of protected area and select protected area.

SGR – Select Graphic Rendition

Function

SGR indicates that the characters following the SGR command are displayed according to the selected graphic renditions (attributes). These attributes apply until the next SGR command occurs.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnM	(none)

where:

Pn is the graphic rendition selection.

m is the final character.

Parameter Values

Refer to the values that you can enter for the SGR command:

When you enter...	Then the character attribute is ...
0	normal.
1	enable bold.
4	enable underline.
5	enable blink.
7	enable reverse video.
22	disable bold.
24	disable underline.
25	disable blink.
27	disable reverse video.

Example

Refer to the following example on how a programmer can format an SGR command in the C programming language:

```
printf("\033[0m"); /*SGR-normal*/
printf("Normal\n");
printf("\033[4;5;7m");   /*SGR-underlined, blinking, reverse
video*/
printf(" NOTICE! ");
```

SL – Scroll Left

Function

SL moves the display window to the left the number of display window widths indicated by the value that you enter. If a large number is entered, the display window does not move past the left margin of the virtual screen in the current row.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pn @	(none)

where:

Pn indicates the number of display window widths that the display window scrolls to the left.

An intermediate space (20 hex) is between the Pn and @.

@ is the final character.

Parameter Values

Refer to the table below for the movement of the display window according to the type of terminal that you have:

Note: This command will always move the physical display window by full window units.

Terminal	Effect of Value Entered for Pn
80 column display	The value does not cause the display window to move because the display is an 80 column display.
40 column display	Any value entered causes the display window to move one display window width to columns 1-40 of the virtual display.
20 column display	The value entered causes the display window to move to the left the number of display window widths indicated by the value entered, up to the left margin. *EXAMPLE If you enter 1, the display window moves to the left 20 columns, up to the left margin. If you enter 2, the display window moves to the left 40 columns, up to the left margin.*

Example

Refer to the following example on how a programmer can format an SL command in the C programming language:

```
printf("\033[1 @");/*Scroll 40 columns for  a 2280 terminal*/
```

Note: There must be a space (20 hex) between the 1 and @.

SM – Set Mode

Function

SM sets one or more modes for the terminal as specified by the parameter. Several modes can be set with the same command if the parameter values are separated by semicolons. They can be sent in any order except #3.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pnh	(none)

where:

Pn indicates the mode to set.

h is the final character.

Parameter Values

Refer to the following table for the modes SM sets:

Parameter	Mode to Set	Description
1	Guarded Area Transfer Mode (GATM)	Enables the terminal to transmit the entire screen to the host or auxiliary device. See the “Sending the Terminal Screen to the Host” section for more details.
2	Keyboard Action Mode (KAM)	Locks the keyboard and RS-232 data entry. The keyboard can not be unlocked from the terminal.
3	Control Representation Mode (CRM)	Enables the terminal to display the control codes. This parameter value must immediately precede the final character.
4	Insert-Replace Mode (IRM)	Enables the terminal to insert the information while the contents of the line shift to the right.
6	Erasure Mode (ERM)	Determines that the editing commands such as EA can erase protected characters.
12	Send-Receive Mode (SRM)	Disables local echo in the terminal so that the terminal operates in character mode. Scroll or block mode communication is disabled.
15	Multiple Area Transmit Mode (MATM)	Enables the terminal to transmit all selected areas. See the “Sending the Terminal Screen to the Host” section for more details.
17	Selected Area Transmit Mode (SATM)	Enables the terminal to transmit the entire virtual screen. See the “Sending the Terminal Screen to the Host” section for more details.

Parameter	Mode to Set	Description
18	Tab Stop Mode (TSM)	Determines that the tab stop settings only affect the cursor's row.
19	Editing Boundary Mode (EBM)	Determines that the editing commands affect the entire virtual screen.
20	Line Feed/New Line Mode (LNM)	Enables the terminal to perform a line feed and carriage return, placing the cursor at the beginning of the next line.

Example

Refer to the following example on how a programmer can format an SM command in the C programming language:

```
printf("\033[4h"); /*Set IRM*/
```

See Also

RM (for more information on how to reset the modes discussed in the previous table).

SR – Scroll Right

Function

SR moves the display window to the right the number of display window widths indicated by the value that you enter. If a large number is entered, the display window does not move past the right margin of the virtual screen in the current row.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pn A	(none)

where:

Pn indicates the number of display window widths that the display window scrolls to the right.

An intermediate space (20 hex) is between Pn and A.

A is the final character.

Parameter Values

Refer to the table below for the movement of the display window according to the type of terminal that you have and the value that you enter:

Note: This command will always move the physical display window by full window units.

Terminal	Effect of Value Entered for Pn
80 column display	The value does not cause the display window to move because the display is an 80 column display.
40 column display	Any value entered causes the display window to move one display window width to columns 41-80 of the virtual screen.
20 column display	The value entered causes the display window to move to the right the number of display window widths indicated by the value entered (up to the right margin). *EXAMPLE If you enter 1, the display window moves to the right 20 columns, up to the left margin.If you enter 2, the display window moves to the right 40 columns, up to the left margin.*

Example

Refer to the following example on how a programmer can format an SR command in the C programming language for a 2285 terminal:

```
printf("\033[4  A");      /*Scroll right 4 display window
widths*/
```

Note: There must be a space (20 hex) between the 4 and the A.

SU – Scroll Up

Function

SU moves the entire contents of the scrolling region up the number of lines indicated by the value that you enter. If the cursor is outside the scrolling region, this command is ignored. The cursor will scroll with the display.

Note: If the number of lines exceed the boundaries of the scrolling region, a blank line moves into the bottom line of the display window. The number of blank lines that move into the bottom of the display window depends on how many lines exceed the bounds of the scrolling region.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[PnS	Scroll Region

where:

Pn is the number of lines the display window scrolls up.

S is the final character.

Parameter Values

Refer to the table below on how SU affects the contents of the display window according to the value that you enter:

When you enter...	Then the contents of the display window scroll up ...
0	one line because the value defaults to 1 when you enter 0.
1 (default)	one line.
a number greater than 1	the number of lines the value indicates.

Example

Refer to the following example on how a programmer can format an SU command in the C programming language:

```
printf("\033[3S"); /*Scroll up 3 lines*/
```

TBC – Tab Clear

Function

TBC clears one or more tab stops according to the value that you enter.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Png	(none)

where:

Pn is the type of tab stops to clear.

g is the final character.

Parameter Values

Refer to the following table on how TBC affects the tab stop positions:

When you enter...	Then the parameter value ...
0 (default)	with TSM set, clears the tab stop at the cursor's position. with TSM reset, clears the tab stop at the cursor's column on all rows in the virtual screen.
1	LXE does not support this parameter.
2	with TSM set, clears all of the tab stops on the cursor's row in the virtual screen.
3	with TSM set or reset, clears all of the tab stops on the virtual screen.
4	LXE does not support this parameter.

Example

Refer to the following example on how a programmer can format a TBC command in the C programming language:

```
printf("\033[0g"); /*Clear tab stop*/
```

LXE®

An EMS Technologies Company

Table of Contents

LXE®

An EMS Technologies Company

Chapter 7
DEC Private Escape Sequences

Overview

This chapter provides information for the programmer on how to code DEC private escape sequences for LXE equipment. All the private escape sequences supported by LXE are listed in this chapter. If a DEC private escape sequence is not listed, LXE does not support it, and it is treated as a no operation (NOP).

DEC Private Escape Sequences

APM – Auto Print Mode

Function

APM turns auto print mode on or off. When auto print is on, the virtual screen row that the cursor is presently on prints when the cursor is moved off the line with a line feed, form feed, vertical tab or auto wrap. The printed line ends with a carriage return and a line feed. Trailing spaces on a screen row will not print.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?Pni	(none)

where:

Pn is the parameter value that indicates on or off.
i is the final character.

Parameter Values

Refer to the following table on the parameter values indicating the mode to set:

When you enter...	Auto Print mode is turned ...
4 (default)	off.
5	on.

DECALN – Screen Alignment Pattern

Function

DECALN fills the virtual screen with uppercase E's. All lines on the display have their line attributes reset to single high single wide. The cursor is placed at the origin of the virtual screen.

Note: DECALN is used by service personnel for diagnostics on the screen display.

Escape Sequence	**Modes that affect this code...** **(refer to Appendix A for mode definitions)**
ESC#8	(none)

where:

and **8** are final characters.

Example

Refer to the following example on how a programmer can format a DECALN command in the C programming language:

```
printf("\033#8");   /*Screen alignment pattern*/
```

DECDHL – Double-Height Line

Function

Makes the line with the cursor on it the top or bottom half of a double-height, double-width line. Pairs of this command must be used on adjacent lines.

The same screen character must be used on both the top and bottom lines in that column to form a full character.

Note: When this command is received, the terminal will reset the right half of the virtual screen to spaces and remove any tabs or areas contained there. If the cursor is in the right half, it will be moved to the last column in the left half.

Escape Sequence	**Modes that affect this code...** **(refer to Appendix A for mode definitions)**
ESC#3 (Top half) ESC#4 (Bottom half)	(none)

Example

Refer to the following example on how a programmer can format a DECDHL command in the C programming language:

```
printf("\033#3 Double High Line\n");
printf("\033#4 Double High Line\n");
```

DECDWL – Double-Width Line

Function

Makes the line with the cursor on it double-width, single-height.

Note: When this command is received, the terminal will reset the right half of the virtual screen to spaces and remove any tabs or areas contained there. If the cursor is in the right half, it will be moved to the last column in the left half.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC#6	(none)

Example

Refer to the following example on how a programmer can format a DECDWL command in the C programming language:

```
printf("\033#6 Double-Width Line\n");
```

DECKPAM – Keypad Application Mode

Function

DECKPAM enables the numeric keypad to send application sequences.

DECKPNM

When you turn on or reset the terminal, the terminal automatically defaults to DECKPNM.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC =	(none)

Refer to the following table for the application sequences sent to the host:

Key	Application Sequence Sent to Host
0	SS3 p
1	SS3 q
2	SS3 r
3	SS3 s
4	SS3 t
5	SS3 u
6	SS3 v
7	SS3 w
8	SS3 x
9	SS3 y
/ (slash)	SS3 l (lowercase L)
- (minus)	SS3 m
. (period)	SS3 n
Numeric Enter	SS3 M

Example

Refer to the following example on how a programmer can format a DECKPAM command in the C programming language:

```
printf("\033=");
```

DECPNM – Keypad Numeric Mode

Function

DECKPNM causes the numeric keypad to send the characters shown on the key.

IMPORTANT When you turn on or reset the terminal, the terminal automatically defaults to DECKPNM.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC >	(none)

Example

Refer to the following example on how a programmer can format a DECKPNM command in the C programming language:

```
printf("\033>");
```

DECRC – Restore Cursor

Function

DECRC restores the terminal to the states saved with the DECSC command. The saved states are listed below:

- Cursor position
- Graphic Rendition (SGR)
- Autowrap Attribute (DECAWM)
- Origin Mode Attribute (DECOM)
- GL and GR Character Set Shift States

Exception

If nothing was saved by DECSC, DECRC performs the following actions:

- Moves the cursor to the home position (upper-left corner of the screen)
- Resets the origin mode (DECOM)
- Turns all character attributes off
- Maps the ASCII character set into GL and the DEC supplemental graphic set into GR

Note: The state of DECAWM is not changed in this case.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC 8	DECAWM, DECOM, SGR

Example

Refer to the following example on how a programmer can format a DECRC command in the C programming language:

```
printf("\033\070");/*Octal for character 8*/
```

DECRM – Reset Mode

Function

DECRM resets the mode you choose.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?Pnl	(none)

where:

? is a constant.

Pn is the value of the mode to reset.

l (lowercase L) is the final character.

Parameter Values

Refer to the following table for the modes DECRM can reset:

Param.	Mode to Reset	Description
1	Cursor Keys Mode (DECCKM)	Enables cursor keys mode (disables application mode). The arrow keys operate in cursor keys mode, and they send special commands to the host. To see the codes sent to the host, refer to the "Arrow Key Mode/Application Sequences Sent to Host" subject/topic in Chapter 8, "LXE Private Escape Sequences". *Note: The terminal must be in character mode for this mode to operate. In scroll/block mode, DECCKM is ignored and the arrow keys always move the cursor.*
5	Screen Mode (DECSCNM)	Selects normal screen (light characters on a dark background). The entire virtual screen is instantly affected by this command. Any text that was normal will become reversed.
6	Origin Mode (DECOM)	Enables the terminal to set the home cursor position at the upper-left corner of the screen regardless of the margins. Also moves the cursor to the origin of the virtual screen.
7	Autowrap Mode (DECAWM)	Disables autowrap mode. In other words, the characters received by the terminal do not automatically wrap to the next line when the cursor reaches the right border of the screen.
18	Print Form Feed Mode (DECPFF)	Disables the terminating character (form feed) after print screen operations. (Print full screen or scroll region with the PS command.)

Param.	**Mode to Reset**	**Description**
19	Print Extent Mode (DECPEX)	Selects the scrolling region to print for each Print Screen (PS) command issued.
25	Text Cursor Enable Mode (DECTCEM)	Makes the cursor invisible.
42	Character Set Mode (DECNRCM)	Enables the multinational character set and disables National Replacement Character Sets.

Example

Refer to the following example on how a programmer can format a DECRM command in the C programming language:

```
printf("\033[?7l");/*DECAWM-disable autowrap mode*/
```

DECSC – Save Cursor

Function

DECSC saves the following states in the terminal's memory:

- Cursor position
- Graphic Rendition (SGR)
- Autowrap Attribute (DECAWM)
- Origin Mode Attribute (DECOM)
- GL and GR Character Set Shift States

This setting can be restored with the DECRC command.

Escape Sequence	**Modes that affect this code...** **(refer to Appendix A for mode definitions)**
ESC 7	DECAWM, DECOM, SGR

Example

Refer to the following example on how a programmer can format a DECSC command in the C programming language:

```
printf("\033\067");/*Save cursor state. 067 is octal for
character 7*/
```

DECSCA – Select Character Attribute

Function

DECSCA indicates the start of a protected or unprotected string of character positions.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pn”q	(none)

where:

Pn is the parameter value that indicates the character attribute.

“ and **q** are the final characters.

Parameter Values

Refer to the following table on the parameter values indicating the character attribute as protected or unprotected:

When you enter...	Then the character(s) is ...
0 (default)	unprotected. *Note:* *This is the same as the EPA command.*
1	protected. *Note:* *This is the same as the SPA command.*
2	unprotected. *Note:* *This parameter has the same effect as parameter 0.*

Example

Refer to the example below on how a programmer can format a DECSCA command in the C programming language:

```
printf("\033[1\"q");      /*DECSCA-1 cannot erase*/
printf("This text is protected from erasure.\n");
printf("\033[0\"q");      /*DECSCA-0 can erase*/
printf("This text is not protected.");
```

DECSCL – Select an Operating Level

Function

DECSCL enables the programmer to select whether the terminal transmits C1 codes as 7-bit or 8-bit sequences. The terminal can always receive either 7 or 8 bit C1 codes from the host regardless of the setting of this mode.

The terminal only supports VT200 mode. This escape sequence is equivalent to the EightBitMode entry in the Emulation Menu of the Configuration program.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[62;Pn"p	(none)

where:

Pn is the parameter value that indicates the character attribute.

" and p are the final characters.

Parameter Values

Refer to the following table on the parameter values:

When you enter...	Then the terminal transmits C1 codes as ...
0 (default)	8-bit sequences.
1	7-bit sequences.

Example

Refer to the example below on how a programmer can format a DECSCL command in the C programming language:

```
printf("\033[62;1\"p");   /* Set to 7-bit */
```

DECSED – Selective Erase in Display

Function

DECSED enables the programmer to erase some or all of the erasable characters in the display. This command will not erase qualified area boundaries or the type of qualified area the cursor is in.

If EBM is set to the virtual screen, every row that is completely erased will have its line attribute reset to single high and single wide.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?PnJ	EBM

where:

? is a constant.

Pn is the portion of the display to erase.

J is the final character.

Parameter Values

Refer to the following table on how the characters are erased according to the value that you enter:

When you enter...	Then the characters are erased from ...
0 (default)	the cursor through the end of the display.
1	the beginning of the display through the cursor.
2	the entire display. The cursor is moved to the origin of the physical or virtual display depending on the EBM setting.

Example

Refer to the example below on how a programmer can format a DECSED command in the C programming language:

```
printf("\033[0\"q");     /*DECSCA = 0*/
printf("Text from this letter x to the end of \n");
printf("the display should be erased.\n");
printf("\033[2A"); /*CUU-Cursor up 2 lines*/
printf("\033[22C");/*CUF-Cursor Forward 22 to the x*/
printf("\033[?0J");/*Erase from x to the end of the display*/
```

DECSEL – Selective Erase in Line

Function

DECSEL enables the programmer to erase some or all of the erasable characters in a single line of text. This command will not erase qualified area boundaries or the type of qualified area the cursor is in.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?PnK	EBM

where:

? is a constant.

Pn is the portion of the line to erase.

K is the final character.

Parameter Values

Refer to the following table on how the characters are erased in relation to the cursor:

When you enter...	Then the characters are erased from ...
0 (default)	the cursor through the end of the line.
1	the beginning of the line through the cursor.
2	the entire line.

Example

Refer to the example below on how a programmer can format a DECSEL command in the C programming language:

```
printf("\033[0\"q");      /*DECSCA = 0*/
printf("Text from this letter x to the end of \n");
printf("the above line should be erased.\n");
printf("\033[2A"); /*CUU-Cursor up 2 lines*/
printf("\033[22C");/*CUF-Cursor forward 22*/
printf("\033[0K"); /*Erase from x to the end of the line*/
```

DECSM – Set Mode

Function

DECSM sets the mode you choose.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?Pnh	(none)

where:

? is a constant.

Pn is the value of the mode to set.

h is the final character.

Parameter Values

Refer to the following table for the modes that DECSM can set:

Parameter	Mode to Reset	Description
1	Cursor Keys Mode (DECCKM)	Enables application mode (disables cursor keys mode). The arrow keys operate in application mode, and they send special commands to the host. To see the codes sent to the host, refer to the “Arrow Key Mode / Application Sequences Sent to Host” topic in Chapter 8, “LXE Private Escape Sequences”. *Note: The terminal must be in character mode for this mode to operate. In scroll/block mode, the terminal will actually perform the single shift operation which displays G3 characters.*
5	Screen Mode (DECSCNM)	Selects reverse video (dark characters on a light background). The entire virtual screen is instantly affected by this command. Any text that was reversed will become normal.
6	Origin Mode (DECOM)	Enables the terminal to set the home cursor position at the upper-left corner of the screen within the margin. Also moves the cursor to the origin of the scrolling region.
7	Autowrap Mode (DECAWM)	Enables autowrap mode. In other words, the characters received by the terminal automatically wrap to the next line when the cursor reaches the right border of the screen.

Parameter	Mode to Reset	Description
18	Print Form Feed Mode (DECPFF)	Enables the terminating character (form feed) after print screen operations. (Print full screen or scroll region with the PS command.)
19	Print Extent Mode (DECPEX)	Selects the full virtual screen to print for each Print Screen (PS) command issued.
25	Text Cursor Enable Mode (DECTCEM)	Makes the cursor visible.
42	Character Set Mode (DECNRCM)	Enables National Replacement Character Sets and disables the multinational character set.

Example

Refer to the following example on how a programmer can format a DECSM command in the C programming language:

```
printf("\033[?7h");/*DECAWM-enable autowrap mode*/
```

DECSTBM – Set Top and Bottom Margins

Function

DECSTBM enables the programmer to set the top and bottom margins of the virtual screen. The display only scrolls between the margins and only when the cursor is between them. In other words, vertical scrolling is restricted to these margins.

The minimum height of a scrolling window is two rows. The top and bottom margin must be within the boundaries of the virtual screen. If not, the TE will correct them automatically.

If DECOM is reset (to absolute positioning), the cursor will move to the origin of the virtual screen. If DECOM is set to relative positioning, the cursor will move to the origin of the new scrolling region.

Rule

You must have a full screen terminal to use DECSTBM.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pt;Pbr	DECOM

where:

Pt is the top margin.

; is a required delimiter.

Pb is the bottom margin.

r is the final character.

Default Values

Refer to the table below for the default values for Pt and Pb:

Parameter	Description	Default Value
Pt	Top Margin	1
Pb	Bottom Margin	24

Example

Refer to the example below on how a programmer can format a DECSTBM command in the C programming language:

```
printf("1)This line cannot be scrolled.\n");
printf("2)This line cannot be scrolled.\n);
printf("3)This line cannot be scrolled.\n");
printf("4)This line can be scrolled.\n");
printf("5)This line can be scrolled.\n");
printf("6)This line can be scrolled.\n");
printf("7)This line cannot be scrolled.\n");
printf("8)This line cannot be scrolled.\n");
printf("\033[4;6r");    /*Scroll region to lines 4 through
                        6.*/
```

DECSTR – Soft Terminal Reset

Function

DECSTR changes the terminal's current parameter settings to the power-up default settings for each mode. The scrolling region is reset to the entire virtual screen, the cursor position remains unchanged.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[!p	(none)

where:

! is a constant.

p is the final character.

Default Settings

Refer to the following table for a listing of the default settings for the modes affected by DECSTR:

Mode	Default Setting
Autowrap Mode (DECAWM)	Reset (no autowrap)
Erasure Mode (ERM)	Set (erase all characters)
Cursor Key Modes (DECCKM)	Reset (cursor mode)
Guarded Area Transmit Mode (GATM)	Reset (only unguarded data)
Insert-Replace Mode (IRM)	Reset (replace mode)
Keyboard Action Mode (KAM)	Reset (unlocked)
Keypad Mode (DECKPNM)	Numeric
Multiple Area Transmit Mode (MATM)	Set (all selected fields)
Origin Transmit Mode (DECOM)	Reset (absolute)

Mode	Default Setting
Selected Area Transmit Mode (SATM)	Reset (send all selected fields)

Example

Refer to the example below on how a programmer can format a DECSTR command in the C programming language:

```
printf("This changes the terminal setting to\n");
printf("the power-up default states.");
printf("\033[!p");
```

DECSWL – Single-Width Line

Function

Makes the line with the cursor on it single-width, single-height. This is the line attribute for all new lines on the display.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC#5	(none)

Example

Refer to the following example on how a programmer can format a DECSWL command in the C programming language:

```
printf("\033#5 Single-height, Single-width Line\n");
```

G0/G1/G2/G3 – Designate G0/G1/G2/G3

Function

G0 designates one of the hard character sets (ASCII, DEC Supplemental Graphics, DEC Special Graphics, and National Replacement character sets) as G0 through G3.

Escape Sequence	To Designate	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC(F	G0	(none)
ESC)F	G1	(none)
ESC*F	G2	(none)
ESC+F	G3	(none)

where:

F is the final character from the following table.

Character set to designate...	Final character...
ASCII	B
DEC Supplemental	<
DEC Special Graphics	0
National Replacement Character Sets	
British	A
Dutch	4
Finnish	C or 5
French	R
French Canadian	Q
German	K
Italian	Y
Norwegian/Danish	E or 6
Spanish	Z
Swedish	H or 7
Swiss	=

LS1R – Lock Shift G1, Right

Function

Invoke G1 into GR.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC~	(none)

Example

Refer to the example below on how a programmer can format a LS1R command in the C programming language:

```
printf("\033)0");  /*Designate DEC Special graphics into G1*/
printf("\033~");   /*Invoke G1 into GR*/
printf("\354\361\361\353\n");  /*These two lines print a
small box on the display*/
printf("\355\361\361\352");
```

LS2 – Lock Shift G2

Function

Invoke G2 into GL.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESCn	(none)

Example

Refer to the example below on how a programmer can format a LS2 command in the C programming language:

```
printf("\033*0");  /*Designate DEC Special graphics into G2*/
printf("\033n");   /*Invoke G2 into GL*/
printf("lqqk\n");  /*These two lines print a small box on the
display*/
printf("mqqj");
```

LS2R – Lock Shift G2, Right

Function

Invoke G2 into GR.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC}	(none)

Example

Refer to the example below on how a programmer can format a LS2R command in the C programming language:

```
printf("\033*0");   /*Designate DEC Special graphics into G2*/
printf("\033}");    /*Invoke G2 into GR*/
printf("\354\361\361\353\n");   /*These two lines print a
small box on the display*/
printf("\355\361\361\352");
```

LS3 – Lock Shift G3

Function

Invoke G3 into GL.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESCo	(none)

Example

Refer to the example below on how a programmer can format a LS3 command in the C programming language:

```
printf("\033+0");   /*Designate DEC Special graphics into G3*/
printf("\033o");    /*Invoke G3 into GL*/
printf("lqqk\n");   /*These two lines print a small box on the
display*/
printf("mqqj");
```

LS3R – Lock Shift G3, Right

Function

Invoke G3 into GR.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC|	(none)

Example

Refer to the example below on how a programmer can format a LS3R command in the C programming language:

```
printf("\033+O");  /*Designate DEC Special graphics into G3*/
printf("\033|");   /*Invoke G3 into GR*/
printf("\354\361\361\353\n");   /*These two lines print a
small box*/
printf("\355\361\361\352");
```

PS – Print Screen

Function

PS prints the display to the RS-232 port. A carriage return and line feed is automatically printed after each line.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[Pni	DECPEX, DECPFF, Scroll Region

where

Pn is either 0 or omitted.

Example

Refer to the following example on how a programmer can format a PS command in the C programming language:

```
printf("\033[0i"); /*Causes the display to be printed*/
```

PL – Print Line

Function

PL prints the display line containing the cursor. A carriage return and line feed are automatically printed after the line.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[?1i	(none)

Example

Refer to the following example on how a programmer can format a PL command in the C programming language:

```
printf("\033[?1i");/*Prints the line the cursor is on*/
```

S7C1T – Select 7-Bit C1 Control Transmission

Function

S7C1T informs the terminal to convert all C1 codes to their 7-bit equivalent before sending them back to the host.

Note: The terminal will always accept either 7-bit or 8-bit control codes regardless of the state of this command.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC sp F	(none)

Example

Refer to the following example on how a programmer can format a S7C1T command in the C programming language:

```
printf("\033 F");  /*There is a single space between the 3
and the F*/
```

S8C1T – Select 8-Bit C1 Control Transmission

Function

S8C1T informs the terminal to send all C1 codes back to the host without converting them to their 7-bit equivalent.

Note: The terminal will always accept either 7-bit or 8-bit control codes regardless of the state of this command.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC sp G	(none)

Example

Refer to the following example on how a programmer can format a S8C1T command in the C programming language:

```
printf("\033 G");  /*There is a single space between the 3
and the G*/
```

LXE®

An EMS Technologies Company

Table of Contents

Chapter 8
LXE Private Escape Sequences

Overview

LXE offers special escape sequences as well as command strings that modify terminal emulation functions, window parameters, and wild card local edit characters. LXE also provides command strings for stored forms, using a backspace function with the [CLEAR] key, enabling an echo suppress, and a wide range of beep patterns.

This chapter describes the applications for each command string and information on how the programmer can code the escape sequences and command strings.

LXE Private Escape Sequences

LXERM – Reset Mode

Function

LXERM resets the mode that you choose.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[=Pnl	(none)

where:

= is a constant.

Pn is the value of the mode to reset.

l (lowercase L) is the final character.

Parameter Values

Refer to the following table for the modes LXERM can reset:

Param.	Mode to Reset	Description
1	Scroll/Block Mode (LXELM)	Enables block mode. Rule: Send-receive mode (SRM) must be reset for LXELM to work.
2	New Line Echo Mode (LXENE)	Disables new line local echo. If the terminal is in scroll/block mode, does not execute carriage return and line feed to the terminal display when the [Enter] key is pressed.
3	Autosend (LXEAS)	Disables autosend. In other words, the terminal transmits only when the [Enter] key is pressed. Rule: Send-receive mode (SRM) must be reset for LXEAS to work.

Example

Refer to the following example on how a programmer can format an LXERM command in the C programming language:

```
printf("\033[=3l");/*Reset LXEAS- Reset autosend mode*/
```

LXESM – Set Mode

Function

LXESM sets the mode you choose.

Escape Sequence	Modes that affect this code... (refer to Appendix A for mode definitions)
ESC[=Pnh	(none)

where:

= is a constant.

Pn is the value of the mode to set.

h is the final letter.

Parameter Values

Refer to the following table for the modes LXESM can set:

Param.	Mode to Set	Description
1	Scroll/Block Mode (LXELM)	Enables scroll mode. Rule: Send-receive mode (SRM) must be reset for LXELM to work.
2	New Line Echo Mode (LXENE)	Enables new line local echo. If the terminal is in scroll/block mode, execute carriage return and line feed to the terminal display when the [Enter] key is pressed.
3	Autosend (LXEAS)	Enables autosend. In other words, the terminal transmits the screen when the last field is filled with a bar code entry, keyboard entry or RS-232 port entry. Rule: Send-receive mode (SRM) must be reset for LXEAS to work.

Example

Refer to the following example on how a programmer can format an LXESM command in the C programming language:

```
printf("\033[=3h");/*Set LXEAS-Set autosend mode*/
```

Specifying Bar Code, Keypad and Control Modes

This section discusses the elements of the command string that enable the programmer to set the bar code, keypad and control modes for the terminal. The different settings for each parameter of the command string are discussed in detail.

Command String for Bar Code, Keypad and Control Modes

The programmer can specify bar code, keypad and control modes with the appropriate command string. This section discusses elements of the command string and its structure.

Command String

Use the following command string for specifying bar code, keypad, and control modes:

```
CSIc1;p1;p2;p3;p4;p5;p6;p7;p8;p9;p10+E
```

Rule

Each parameter must be preceded by a semicolon (;).

You can use the semicolon as a placeholder if you want to change only one parameter. For example, you can send the following string to change only p5 of the command string:

```
CSI1;;;;;1+E
```

String Elements

The following table describes the different elements of the command string for specifying bar code, keypad, and control modes:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
c1	A compatibility byte that ensures the terminal emulation understands the command string sent to it. The c1 is a revision level character for compatibility purposes, which increments with each revision. The original value for this parameter is 1.
;	A required delimiter that must precede all parameter entries.
p1	Save to disk parameter. 0 = Use the command string in the terminal but do not save. The configuration is used only as long as the terminal is powered on. 1 = Use the command string in the terminal and save. The configuration is saved even after the terminal is powered off.

String Element	Description
p2	Bar Code/RS-232 Input Left Justify Mode 0 = Left Justify Mode Off 1 = Left Justify Mode On The section “Left Justify Mode: p2” found in this chapter discusses this parameter in more detail.
p3	Bar Code/RS-232 Input Clear Mode 0 = Clear Off 1 = Clear On The section “Clear Mode: p3” found in this chapter discusses this parameter in more detail.
p4	Bar Code/RS-232 Input Truncate Mode 0 = Truncate OFF 1 = Truncate ON The section “Truncate Mode: p4” found in this chapter discusses this parameter in more detail.
p5	Bar Code/RS-232 Input Advance Mode 0 = Advance OFF 1 = Advance ON The section “Advance Mode: p5” found in this chapter discusses this parameter in more detail.
p6	Keypad Mode 0 = Keypad in Numeric Mode (DECKPNM) 1 = Keypad in Application Mode (DECKPAM) The section “Keypad Mode: p6” found in this chapter discusses this parameter in more detail.
p7	Arrow Key Mode (DECCKM) 0 = Arrow key in Cursor Mode 1 = Arrow key in Application Mode The section “Arrow Key Mode: p7” found in this chapter discusses this parameter in more detail.

String Element	**Description**
p8	Compose Mode 0 = Compose Mode OFF 1 = Compose Mode ON The section “Compose Mode: p8” found in this chapter discusses this parameter in more detail.
p9	Control Code Mode (CRM) *Note: This private sequence cannot be used to change the terminal to execute control codes. Use RM3 instead.* 1 = Display Control Code The section “Control Code Mode: p9” found in this chapter discusses this parameter in more detail.
p10	Bad Sequence Mode 0 = Ignore Bad Control Sequence Code 1 = Display Bad Control Sequence Code The section “Bad Sequence Mode: p10” found in this chapter discusses this parameter in more detail.
+E	The final characters of the ANSI command string. This combination indicates the end of the command string and identifies the command string as a bar code, keypad, and control mode command.

Left Justify Mode: p2

Left justify mode determines where the cursor appears in a field from one bar code entry to the next. The programmer must decide if the cursor in a field justifies to the left each time a new bar code entry occurs or if the bar code entry appears at the end of the previous entry.

Rule

You must enable block mode to use left justify mode.

Parameter Values

Refer to the following table on how to set p2 of the command string to disable or enable left justify mode:

When you enter ...	Then left justify is ...
0	off.
1 (default)	on.

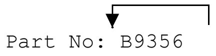
Left Justify Off

Left Justify Off enables a bar code read to appear in the next available cursor position of a field directly following a previous bar code entry. For example, the first bar code entry fills the first five positions of the first field available as shown below:

The next bar code read is entered starting at the cursor's position and proceeding to the next unprotected field.

Left Justify On

Left Justify On enables the cursor to left justify in a field each time a bar code read occurs. For example, the following bar code read appears in the field and then the cursor left justifies when the next bar code entry occurs. The next bar code read appears in the field starting at the B position:



Clear Mode: p3

Clear mode determines if a field is automatically cleared when the next bar code entry occurs or if the data in the field is overwritten by the new entry.

IMPORTANT If a field is not cleared after the data is transmitted to the host, there is a chance that some of the previous data will not be completely overwritten by the new entry. If the data is not completely overwritten, it appears at the end of the new entry and is transmitted with the new entry to the host.

Rule

You must enable block mode to use clear mode.

Parameter Values

Refer to the following table on how to set p3 of the command string to disable or enable clear mode:

When you enter ...	Then clear mode is ...
0	off.
1 (default)	on.

Clear Mode Off

When clear mode is off, the terminal does not clear the data in a field. The new data entered overwrites portions of the left-over data. An operation that has bar code entries with equal character lengths might choose clear mode off.

Clear Mode On

When clear mode is on, the terminal clears the data entered in a field each time a new bar code entry occurs. An operation that has bar code entries with varying lengths would need to use clear mode on to ensure accurate transmission of each entry.

Truncate Mode: p4

Truncate mode determines whether the character positions of a bar code entry that exceeds the field length are simply dropped off or if the leftover characters are entered in the next available field.

Rule

You must enable block mode to use truncate mode.

Parameter Values

Refer to the following table on how to set p4 of the command string to disable or enable truncate mode:

When you enter ...	Then truncate mode is ...
0	off.
1 (default)	on.

Truncate Mode Off

When truncate mode is off, the character positions of a bar code entry exceeding the field length are entered in the next available field. An operation that has two kinds of information in one bar code entry would use truncate mode off so that one bar code read can fill two fields on a form.

Truncate Mode On

When truncate mode is on, the character positions of a bar code entry exceeding the field length are dropped off. If the field has 10 available character positions and the bar code entry has 22, the first 10 character positions of the entry are entered in the field and the remaining 12 character positions are dropped. An operation that wants to ensure that a bar code entry from one field is not carried over into the next field would want to have truncate mode on.

Advance Mode: p5

Advance mode determines whether the cursor advances to the next field at the end of a bar code entry. If the cursor does not advance to the next field, the next bar code entry is entered at the end of the previous bar code entry.

Rule

You must enable block mode to use advance mode.

Parameter Values

Refer to the following table on how to set p5 of the command string to disable or enable advance mode:

When you enter ...	Then advance mode is ...
0	off.
1 (default)	on.

Advance Mode Off

When advance mode is off, the cursor does not advance to the next field at the end of a bar code entry. The next bar code entry is entered directly following the previous bar code entry.

Advance Mode On

When advance mode is on, the cursor advances to the next field at the end of each entry regardless of whether the field is completely filled or not. An operation that has bar code entries that do not always fill the field would use advance mode on.

If autosend (LXEAS) is also enabled, the terminal will automatically send the screen even if the last bar code entry did not completely fill the last field on the display.

Keypad Mode: p6

Keypad mode enables the programmer to choose the mode of operation for the keypad. There are two types of keypad modes: numeric mode and application mode.

Parameter Values

Refer to the following table on how to set p6 of the command string for the keypad mode:

When you enter ...	Then the keypad operates in ...
0 (default)	numeric mode (DECKPNM).
1	application mode (DECKPAM).

Numeric Mode

Numeric mode is a keypad mode that enables the terminal to send numeric characters from the keypad. For instance, if the operator presses a 0, the host receives a 0.

Application Mode

Application mode is a keypad mode that enables the terminal to send application sequences from the keypad. For instance, if the operator presses a 0, the terminal transmits an SS3 p application sequence to the host.

Application Sequences

Refer to the following table for the application sequences sent to the host according to the key pressed at the keyboard:

Key	Application Sequence Sent to Host
0	SS3 p
1	SS3 q
2	SS3 r
3	SS3 s
4	SS3 t
5	SS3 u
6	SS3 v
7	SS3 w
8	SS3 x
9	SS3 y

Key	Application Sequence Sent to Host
/ (slash)	SS3 l (lowercase L)
- (minus)	SS3 m
. (period)	SS3 n
Numeric Enter	SS3M

Arrow Key Mode: p7

Arrow key mode enables the programmer to choose the mode of operation for the arrow keys while the terminal is in character mode. There are two types of arrow key modes: arrow key in cursor mode or arrow key in application mode. If the terminal is in scroll/block mode, cursor mode will ignore DECCKM and the cursor will move depending on which arrow key was pressed. In application mode, the terminal will actually perform the single shift operation which displays G3 characters.

Parameter Values

Refer to the following table on how to set p7 of the command string for the arrow key mode:

When you enter ...	Then arrow key mode operates in ...
0 (default)	cursor mode (DECCKM).
1	application mode (DECCKM).

Application Sequences Sent to Host

Refer to the following table for the application sequences sent to the host according to the arrow key pressed at the keypad and the arrow key mode the terminal is operating in:

Arrow Key	Application Sequence in Cursor Mode	Application Sequence in Application Mode
[ArrowUp]	ESC[A	ESC OA
[ArrowDown]	ESC[B	ESC OB
[ArrowRight]	ESC[C	ESC OC
[ArrowLeft]	ESC[D	ESC OD

Compose Mode: p8

The compose mode determines if the compose key is active or not.

Parameter Values

Refer to the following table on how to set p8 of the command string to disable or enable the compose mode:

When you enter ...	Then the international character set is ...
0 (default)	off, the compose key is not recognized.
1	on, the compose key is recognized. See the section on the compose key in Chapter 4, “Terminal Overview” for more information.

Control Code Mode: p9

Control code mode determines whether the control codes are executed or displayed on the screen.

Parameter Values

Refer to the following table on how to set p9 of the command string for executing or displaying control codes:

When you enter ...	Then the terminal ...
1	displays the control codes (CRM parameter in SM command).

Note: *This private sequence can not be used to change the terminal to execute control codes, use RM3 instead.*

Bad Sequence Mode: p10

The bad sequence mode determines if the terminal displays a bad sequence of data or not. The terminal ignores the sequence in either case.

Definition

A *bad sequence* is an invalid control sequence that is neither identifiable nor executable by the terminal.

Parameter Values

Refer to the following table on how to set p10 of the command string to display bad sequences or not:

When you enter ...	Then the terminal ...
0	ignores the bad sequence and does not display it on the screen.
1 (default)	ignores the bad sequence but displays it on the screen so that the programmer can either remove or correct the bad sequence. This parameter helps in debugging applications.

Specifying Transmit and Screen Function Setup

Command String for Transmit and Screen Function Setup

You must specify the transmit and screen function setup with the appropriate command string. This section discusses elements of the command string and its structure.

Command String

Use the following command string for the transmit and screen function setup:

```
CSIc1;p1;p2;p3;p4;p5;p6;p7;p8;p9;p10+F
```

Rule

Each parameter must be preceded by a semicolon (;).

You can use the semicolon as a placeholder if you want to change only one parameter. For example, you can send the following string to change only p5 of the command string:

```
CSI1;;;;;1+F
```

String Elements

The following table describes the different elements of the command string for the transmit and screen function setup:

String Element	Description	Related Sections
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or hex 1B5B for 7-bit code).	-
c1	A compatibility byte that ensures the terminal emulation understands the command string sent to it. The c1 is a revision level character for compatibility purposes, which increments with each revision. The original value for this parameter is 1.	-
;	A required delimiter that must precede all parameter entries.	-
p1	Save to disk parameter. 0 = Use the command string in the terminal but do not save. The configuration in the terminal is used only as long as the terminal is powered on. 1 = Use the command string in the terminal and save. The configuration is saved even after the terminal is powered off.	-

String Element	Description	Related Sections
p2	8-Bit Code control from the host. *Note: The terminal can always accept either 7-bit or 8-bit codes from the host. The value of this parameter is therefore ignored.*	8-Bit Code Table
p3	Auto Transmit Mode 0 = Auto Transmit Mode is OFF. The [Enter] key must be pressed to transmit the data. 1 = Auto Transmit Mode is ON. The data is automatically transmitted when the last field is filled with a bar code entry, an RS-232 port entry or keyboard entry.	LXEAS
p4	LF Transmission on Send 0 = The [Enter] key places a carriage return in the data stream. 1 = The [Enter] key places a carriage return and line feed in the data stream.	LNM
p5	Screen Mode 0 = Screen Mode is Block Mode. 1 = Screen Mode is Scroll Mode.	LXELM
p6	New Line Echo Mode 0 = New Line Echo Mode OFF. 1 = New Line Echo Mode ON.	LXENE
p7	Auto Wrap Mode 0 = Auto Wrap Mode OFF. 1 = Auto Wrap Mode ON.	DECAWM
p8	On Line Mode 0 = Operate in On Line Mode. The [Enter] key is enabled to send messages to the host through the RF link. 1 = Operate in Local Mode. The [Enter] key is disabled.	-
p9	Status Line Mode 0 = Status Line Mode OFF. 1 = Status Line Mode ON.	Status Line
p10	Transmit Mode 0 = Transmit in Block or Scroll Mode. 1 = Transmit in Character Mode.	SRM

String Element	Description	Related Sections
+F	The final character of the ANSI command string. This combination indicates the end of the command string and identifies the command string as a transmit and screen function mode.	-

Local Edit Commands and Beep Patterns

Local edit commands enable a programmer to control data entry from the terminal. This section discusses the two types of local edit commands and the command string elements for each. Additionally, this section discusses beep patterns available in LXE terminals.

Command String for Local Edit Match Field

A local edit match field restricts operator data entry to exact match values or ranges of accepted values. A programmer can use this feature to prevent incorrect part number or bin location entries.

Definitions

An *exact match field* is a field with predetermined values that must be matched exactly by operator entry.

Local edit match field is a feature provided by LXE that enables the terminal to perform data comparisons between the match buffer in the terminal and the data entered by the terminal operator. In other words, the terminal locally edits information as the terminal operator enters data instead of the host performing the edit.

A *match buffer* is an area of memory used by the local edit match field command to restrict data entry by the terminal operator.

Rules

Refer to the following rules for using local edit match field:

- You must specify a local edit field by placing the cursor at the beginning of the appropriate field and sending the correct command string.
- You must place match data between two identical LXE local edit command strings.
- You must enable Local echo. Refer to RM (parameter value 12) in this manual for more information on local echo.
- You must enable block mode. If character mode or scroll mode is enabled, local edit match fields are ignored.
- Use characters entered into the match buffer field that do not match the local edit wildcard symbols (defaults: ?, $, #, +, *) as exact match characters.

Command String

Use the following command string for specifying a local edit match field:

```
CSId+A
```

String Elements

The following table describes the different elements of the command string for specifying a local edit match field:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
d	Keyboard Input 0 (default) = Display the keyboard input if operator entry matches the requirements of the match buffer. 1 = Do not display the keyboard input if operator entry matches the requirements of the match buffer. Host input on these cells is displayed.
+A	The final character of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a local edit match field.

Example

Refer to the following example of a local edit match field command:

```
printf(" _ _ _ _ _ _ "); /*6 character fields*/
printf("\033[6D"); /*CUB-move cursor back 6*/
printf("\033[0+A");/*turn on local edit setup mode*/
printf("P?$$$$");  /*part number format*/
printf("\033[0+A");/*turn off local edit setup mode*/
printf("\033[6D"); /*CUB-move cursor back 6*/
gets(s)/*wait for operator to enter data*/
```

Command String for Local Edit Wildcard

Local edit wildcard enables the programmer to change the default match symbols. Wildcard match values represent a range of acceptable operator input. The host may change the definition of a particular wildcard at any time without affecting any previously set local edit match fields. (The TE stores the *type* of wildcard rather than the actual wildcard value.)

Command String

Use the following command string for local edit wildcard:

`CSIc1;p1;p2;p3;p4;p5;p6+G`

Rule

The values you enter for the symbols must be in their decimal equivalent forms. Refer to the *8-Bit Code table* in this manual for these decimal values.

String Elements

The following table describes the different elements of the command string for specifying a local edit wildcard:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
c1	A compatibility byte that ensures the terminal emulation understands the command string sent to it. The c1 is a revision level character for compatibility purposes, which will increment with each revision. The original value for this parameter is 1.
;	A required delimiter that must precede all parameter entries.
p1	Save to disk parameter. 0 = Use the command in the terminal but do not save. The configuration in the terminal is used only as long as the terminal is powered on. 1 = Use the command string in the terminal and save. The configuration is saved even after the terminal is powered off.
p2	? (Alpha Only Match) Alpha only match includes the following characters: A-Z, a-z, comma, period, dash, space.
p3	$ (Digits Only Match) Digits only match includes the following characters: 0-9.

String Element	Description
p4	# (Numeric Only Match) Numeric only match includes the following characters: 0-9, plus, comma, period, dash, space.
p5	+ (Alphanumeric Only Match) Alphanumeric only match includes the following characters: A-Z, a-z, 0-9, plus, comma, period, dash, space.
p6	* (Match All) Match all includes any GL or GR characters.
+G	The final characters of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a local edit wildcard command.

Using Local Edit

Using the local edit command enables the programmer to control data entry at the terminal. The local edit field length can be as long as the virtual screen.

The terminal edits the field locally as data is entered and not when the field is exited. During the editing of a field, if the data fails the match test, the cursor does not move.

EXCEPTION *When a bar code entry fails the match test, the cursor goes to the first position where the character did not match the test. The remaining characters in the input field are checked against the match test.*

Programming Local Edit Fields

1. Position the cursor to the beginning of the local edit field.
2. Send the CSId+A command to put the TE in local edit setup mode.
3. Send the local edit match characters which could be either exact match or wildcards. These are sent as regular GL or GR characters. They will be stored internally in the TE and not displayed on the screen. The cursor position will automatically be updated after each char is received.

   Note: Setting a screen cell as a local edit cell overrides any previous DAQ/SPA/SSA settings for that cell and vice versa.
4. If desired, position the cursor to a new location and send more local edit match characters to define more fields. Repeat as necessary.
5. Send a matching CSId+A command to put the TE in normal mode.
6. If automatic tabbing is desired to the start of each field, the host must send the standard area or tab commands at the start of each local edit field. This can be accomplished even when the TE is in local edit mode.
7. The TE user may now enter data through keyboard or RS-232 inputs.

If the operator input passes the local edit checks, the data will be displayed (if the local edit field was set up as CSI0+A). The data will not be displayed if the local edit field was set up as CSI1+A. In either case, the cursor position will advance by 1.

If the data does not pass the local edit checks, the TE will beep to inform the user and the cursor will not advance.

Requirements

The following requirements must be met for local edit to work properly:

- The terminal must be in block mode (LXELM).
- Local echo must be enabled in send-receive mode (SRM).
- The terminal must be an LXE terminal to have local edit capabilities.

Beep Pattern Command

The beep pattern command is an LXE private remote setup command that enables the programmer to set a beep pattern that sounds at the terminal under certain conditions.

Command String

The programmer must send the following command string for specifying a beep pattern:

```
CSIp1;p2+B
```

String Elements

The following table describes the different elements of the command string for specifying a beep pattern:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
p1	Beep Pattern. Enter a number from 1 - 255.
;	A required delimiter.
p2	Volume Control. Enter a number from 1 (lowest) to 7 (highest). *Note: The lower values may not provide an audible beep on some terminals. If a volume is specified, the volume in the TE is changed.*
+B	The final character of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a beep pattern command.

Beep Pattern: p1

The programmer must derive the number required for p1 of the command string for the beep pattern. The following information should help the programmer derive this number.

Diagram

Refer to the diagram below of a template used to derive the beep pattern number for p1 of the command string:

0	1	2	4	8	16	32	64	128
B								

Description

The template for deriving the beep pattern is divided into boxes. Each box in the template corresponds to a unit of time when either a beep or silence occurs. The leftmost box always contains a B that represents a beep. This B always occurs, regardless of the rest of the beep pattern.

Range

The range for the possible beep patterns is 1 - 255.

IMPORTANT The beep pattern generated with the above template begins with the leftmost bit. Refer to the example on the following page.

Procedure

1. Place a “B” in each box of the template where you want a beep to occur. Leave the boxes blank where you want a silence to occur.

   Note: Box 0 will always have a beep.

2. Total the numbers listed above each box that has a B in it.

   Note: The total of the numbers above the boxes is the number value used in the command string for p1.

3. Enter the total from Step 2 for parameter p1 of the command string.

Example

Refer to the following example on producing a beep pattern:

1. Put “B” in the box where you want a beep to occur and leave the box empty where you want a silence to occur.

0	1	2	4	8	16	32	64	128
B	B		B	B		B	B	

2. Add the numbers above the boxes that have a B in them.

   0 + 1 + 4 + 8 + 32 + 64 = 109

3. Enter the total from Step 2 (109) in the command string for setting the beep pattern:

```
printf("Beep pattern will be\n");
printf("on on off on on off on on off");
printf("\033[109;6+B");     /*pattern = 109,
                     volume = 6*/
```

Result: *This command string produces three short beeps at the terminal because successive “Bs” produce a continuous beep until a silence occurs.*

Stored Forms

The stored forms feature enables the terminal to store forms at the terminal. This section discusses the command string used to access the stored forms. The components of the command string and the different commands available with the stored form command string are discussed in detail.

Definition

A *stored form* is a combination of ANSI, DEC, and LXE commands with or without text that appear as a fill-in form at the terminal. A stored form is typically stored in a directory (for example, C:\APLUS\STFORMS) on the terminal’s hard drive. The directory name and path are user configurable (refer to Chapter 3 “Configuration Utility” for more information).

About Stored Forms

Form Name

The form name is terminated when a space or non-printable character occurs. The form name follows DOS naming conventions and can be any printable character string with a maximum length of 12 characters (8 character name and a 3 character extension separated by a period). Use uppercase letters in the form name because the terminal changes all lowercase alpha characters to uppercase alpha characters.

Memory Storage

The maximum storage capacity needed for stored forms is 200K bytes.

Number of Stored Forms

The maximum number of stored forms allowed in the form list directory is 100 forms.

Form Data

The form data can be a maximum of 2048 bytes. If a form requires more than 2048 bytes, the form should be divided into two forms. The form data can consist of any combination of ANSI, DEC, and LXE commands except the PM command) discussed in this reference guide. The form data can also include text.

Command String for Stored Forms

A stored form command string begins with a privacy message (PM) command and ends with a string terminator (ST) command. The structure of the command string changes according to the command you want to perform.

Command String

Use the following command string for stored forms:

```
ESC^FCMDFORM(SPACE)DATAESC\
```

String Elements

The following table describes the different elements of the command string for specifying stored forms:

String Element	Description	Related Sections
ESC^	A privacy message (PM) command indicating that the following data is a private sequence.	PM
F	A constant indicating that the privacy message is a stored form message.	-
CMD	A single, capital letter indicating the command to perform. These commands are checksum, delete, execute, list directory, rename, and write.	Commands for ANSI Stored Forms
FORM	A form name consisting of a maximum of eight characters that identifies a stored form.	Rename Command and Write Command
(SPACE)	An intermediate space character (20 hex) that must follow the form name.	-
DATA	The contents of the form.	Write Command
ESC\	A string terminator (ST) command that follows all PM commands to indicate the end of the privacy message.	ST

Stored Form Commands

All stored form commands begin with a 2-character sequence. The first character of the command string is:

```
F
```

followed by another capital letter indicating the specific command to perform.

Rule

Each letter representing a command must be a capital letter.

Table

The following table describes the specific commands you can perform with the stored form command string:

CMD Element	Description	Related Sections
C	Checksum Requests the size and checksum of a form.	Checksum Command
D	Delete Deletes one form at a time or all the forms.	Delete Command
L	List Directory Lists all of the stored forms.	List Directory Command
R	Rename Renames an existing form.	Rename Command
W	Write Creates a new form using any combination of the ANSI, DEC and LXE commands discussed in this manual (except for the PM command).	Write Command
X	Execute Recalls a form.	Execute Command

Checksum Command

The checksum command causes the terminal to return the size and checksum of a specified form. You can use the checksum command to determine whether a form already exists before you create a form or rename a form.

Syntax

ESC^FCFORMESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

C identifies the stored form command as a checksum command.

FORM is the form name.

ESC is a string terminator (ST) command, which ends a privacy message.

Terminal Response

When the terminal receives a checksum command from the host concerning an existing form, the terminal returns the following response:

ESC^FCFORM XXXX YYYY$^{C}{}_{R}{}^{L}{}_{F}$ESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

C identifies the stored form command as a checksum command.

FORM is the form name consisting of a maximum of eight characters.

An intermediate space (20 Hex) must occur between the FORM and XXXX.

XXXX is a four-digit form length with leading zeros.

An intermediate space (20 Hex) occurs between the XXXX and YYYY.

YYYY is a four-digit hexadecimal checksum with leading zeros.

$\mathbf{C_{R}L_{F}}$ is a carriage return and line feed.

ESC is a string terminator (ST) command, which ends a privacy message.

When the terminal receives a checksum command concerning a form that does not exist, the terminal returns the following response:

ESC^FCNAESC\

where:

NA indicates that the form is not available.

Delete Command

The delete command either deletes the specified form or all the forms.

Delete One Form

When you want to delete one form, send the following command string:

ESC^FDFORM ESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

D identifies the stored form command as a delete command.

FORM is the form name consisting of a maximum of eight characters.

An intermediate space character (20 Hex) must occur between FORM and ESC\.

ESC is a string terminator (ST) command, which ends a PM command.

Delete All Forms

To delete all the stored forms, send the following command string:

ESC^FD*_*ESC\

Execute Command

The execute command recalls an existing form that is stored in the terminal. When you recall a form, the form is displayed at the terminal for the operator to use. If the form that you execute (attempt to recall) does not exist, the execute command is treated as a no operation (NOP). No error response occurs to indicate that the form does not exist.

Syntax

ESC^FXFORMESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

X identifies the stored form command as an execute command.

FORM is the form name consisting of a maximum of eight characters.

ESC is a string terminator (ST) command, which ends a PM command.

List Directory Command

The list directory command requests the terminal to send a list of forms stored at the terminal. When you send a list directory command from the host to the terminal, the terminal responds with a privacy message. The forms in the list directory are not in alphabetical order.

Syntax

ESC^FLESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

L identifies the stored form command as a list directory command.

ESC is a string terminator (ST) command, which ends a PM command.

Terminal Response

When the terminal receives a list directory command concerning forms stored in the terminal, the terminal returns the following response:

ESC^FLFORM XXXX YYYY${}^{C}{}_{R}{}^{L}{}_{F}$FORM2...ESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form command.

L identifies the stored form command as a list directory command.

FORM is the form name consisting of a maximum of eight characters.

An intermediate space character (20 Hex) must occur between FORM and XXXX.

XXXX is a four-digit form length with leading zeros.

An intermediate space character (20 Hex) must occur between XXXX and YYYY.

YYYY is a four-digit hexadecimal checksum with leading zeros.

${}^{\mathbf{C}}{}_{\mathbf{R}}{}^{\mathbf{L}}{}_{\mathbf{F}}$ is a carriage return and line feed.

FORM2 is the next form in the list directory.

... indicates the repetition of information for any other form names in the list directory.

ESC is a string terminator (ST) command.

When the terminal receives a list directory command before any forms are created, the terminal returns the following response:

ESC^FLNAESC\

where:

NA indicates that the list directory is not available.

When the terminal receives a list directory command after all the forms in the list directory have been deleted, the terminal returns the following response:

ESC^FLESC\

Rename Command

The rename command renames a form. When a form is renamed, the old form name is deleted. However, if a form name already exists and you try to use this name to rename another form, the rename command is treated as a no operation (NOP). In other words, the form you are trying to rename is not renamed. No error response occurs to indicate that the form was not renamed.

Syntax

ESC^FRFORM1 FORM2ESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form message.

R identifies the stored form command as a rename command.

FORM1 is the name of the form you want to rename.

An intermediate space (20 Hex) must occur between FORM1 and FORM2.

FORM2 is the new form name.

ESC is a string terminator (ST) command, which ends a PM command.

Write Command

The write command creates a form that is saved as a form name in the terminal. Each form consists of data that is included in the stored form command string.

Syntax

ESC^FWFORM DATAESC\

where:

ESC^ is a privacy message (PM) command.

F identifies the message as a stored form message.

W identifies the stored form command as a write command.

FORM is the form name consisting of a maximum of eight characters.

An intermediate space (20 Hex) must occur between FORM and DATA.

DATA consists of text, ANSI, DEC, and LXE commands (except the PM command) discussed in this manual, which make up the form.

ESC is the string terminator (ST) command, which ends a privacy message.

Overwriting an Existing Form

When you use an existing form name in the write command, the form created with the write command overwrites the existing form. No error response occurs to indicate that the form was overwritten.

To prevent overwriting an existing form, perform a checksum command on the form name that you want to create with the write command. If the form name exists, the checksum response will include the form name, size, and checksum. If the form does not exist, the checksum response will indicate NA (not available).

Command String for Window Setup

Window setup determines how the display window shows portions of the virtual screen. You can set the parameters for window setup by using the correct command string. For detailed information on how these parameters work, refer to the "Window Manager" section found in Chapter 4 "Terminal Overview".

Command String

Use the following command string for window setup:

```
CSIc1;p1;p2;p3;p4;p5;p6+D
```

String Elements

The following table describes the different elements of the command string for window setup:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
c1	A compatibility byte that ensures the terminal emulation will understand the command string sent to it. The c1 is a revision level character for compatibility purposes, which will increment with each revision. The value for this parameter is 1.
;	A required delimiter that must precede all parameter entries.
p1	Save to disk parameter. 0 = Use the command string in the terminal but do not save. The configuration in the terminal is used only as long as the terminal is powered on. 1 = Use the command string in the terminal and save. The configuration is saved even after the terminal is powered off.
p2	Lock Window Mode. 0 = Lock window mode off/cursor tracking on. 1 = Lock window mode on/cursor tracking off. *Note: "Display Modes: p2" in this chapter discusses lock window mode and cursor tracking mode in more detail.*

String Element	**Description**
p3	Lock Window Line Number. This number identifies the line of the top left corner position of the display window. “Valid Window Positions: p3 and p4” in this chapter discusses the valid window positions. *Note: This parameter is only used when lock window mode is enabled.*
p4	Lock Window Column Number. This number identifies the column of the top left corner position of the display window. “Valid Window Positions: p3 and p4” in this chapter discusses the valid window positions. *Note: This parameter is only used when lock window mode is enabled.*
p5	Window Movement Mode. 0 = Move by Quadrant. 1 = Move by Full Window. *Note: Quadrant and full window are window movement modes discussed in “Window Movements: p5” in this chapter.* *This parameter is used for both lock window mode and cursor tracking mode.*
p6	Quadrant for Cursor. The quadrant the cursor is placed in when the host sends a screen to the terminal. 1 = Top right quadrant. 2 = Top left quadrant. 3 = Bottom left quadrant. 4 = Bottom right quadrant. *Note: This parameter indicates the placement of the cursor in the display window when the host sends a screen to the terminal. “Quadrant for Cursor: p6” in this chapter discusses this parameter in more detail.* *This parameter is only used when cursor tracking mode is enabled.*
+D	The intermediate final character of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a window setup host command.

Display Modes: p2

Parameter p2 of the window setup command string enables the programmer to choose the display mode. Window manager supports two types of display modes for displaying the information from the virtual screen at the terminal: cursor tracking mode and lock window mode.

In cursor tracking mode, the terminal will automatically adjust the physical display to follow the cursor around the virtual screen. The exact position that the terminal leaves the display window in depends on p6.

In lock window mode the terminal will not adjust it's display window to follow the cursor around the virtual screen. It is possible for the cursor to move outside the lock window from either host or user input. The programmer must specify valid line or column positions in p3 and p4 of the command string which specify the location of the upper left corner of the lock window.

Parameter Values

Refer to the table below on how to select cursor tracking mode or lock window mode:

When you enter ...	Then ...
0	lock window mode is off and cursor tracking is on.
1	lock window mode is on and cursor tracking is off.

Valid Window Positions: p3 and p4

When the terminal operates in lock window mode (p2 set to 1), the programmer must specify valid line and column positions in p3 and p4 of the command string.

Parameters p3 And p4

Parameter p3 sets the line position for the upper left corner of the display window.

Parameter p4 sets the column position of the upper left corner of the display window.

Note: When p2 is set to 0 (lock window mode off), the display window defaults to cursor tracking mode, which does not use pre-defined window positions. Parameters p3 and p4 are ignored.

If an invalid line or column position is specified, the terminal will automatically correct either one by choosing the closest correct value that is less than the one specified.

Rules

Refer to the following rules on valid window positions:

- The line positions or column positions must be valid numbers that represent the location of the top left corner of the display window.
- When lock mode is on, the programmer must specify the display window's line/column positions in p3 and p4.

Note: Refer to the "Window Manager" section found in Chapter 4 "Terminal Overview" for valid line and column values for different display sizes.

Window Movement: p5

The terminal uses two types of window movements: full window movement and quadrant movement. When cursor tracking mode is enabled, this parameter determines how far the display window moves when the user moves the cursor past the display window edge or when the user pans the window with the window movement keys. When lock window mode is enabled this parameter determines how far the display window moves when the user pans the window with the window movement keys.

Vertical Movement

When the display window moves vertically in full window movement, the display window moves 8 lines at a time (for an 8 line display) so that the window does not overlap. When the display window moves vertically in quadrant movement, the display window moves 4 lines causing the screen to overlap a portion of its current screen.

Horizontal Movement

When the display window moves horizontally in full window movement, the display window moves 40 columns if it is a 40 column display. When the display window moves horizontally in quadrant movement, the display window moves half the number of columns the display has. In other words, the display moves 20 columns to the right.

Parameter Values

Refer to the table below on how to choose full window movement and quadrant movement:

When you enter ...	Then the window moves in ...
0	quadrant movement.
1	full window movement.

Quadrant for Cursor: p6

The display window is divided into four quadrants that identify sections of the virtual screen. The programmer must choose which quadrant the cursor appears in when the host sends a virtual screen to the terminal. The terminal will effectively update the display window location after every character or escape sequence received from the host.

The display window is divided into four quadrants:

1. Upper right
2. Upper left
3. Lower left
4. Lower right

Purpose

Parameter p6 enables you to determine where the cursor appears when the host sends a screen down to the terminal. The cursor appears in the specified quadrant entered in p6.

The default for this parameter is quadrant 4.

Additional Commands

This section discusses additional commands that LXE supports that go beyond the capability of the standard ANSI commands. These additional commands include character collection timer, and suppress. The collection timer command string and the suppress command string are discussed in detail.

Character Collection Timer Command String

The character collection timer string sets the time that the terminal waits for the next key stroke before transmitting the characters entered.

Rule

You must enable character mode to use this command string; otherwise, this command string is ignored.

Command String

Use the following command string to specify a character collection timer value:

```
CSIc1;p1;p2+H
```

String Elements

The following table describes the different elements of the command string for specifying the character collection timer:

String Element	Description
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
c1	A compatibility byte that ensures the terminal emulation will understand the command string sent to it. The c1 is a revision level character for compatibility purposes, which will increment with each revision. The value for this parameter is 1.
;	A required delimiter that must precede all parameter entries.
p1	Save to disk parameter. 0 = Use the command in the terminal but do not save. The configuration in the terminal is used only as long as the terminal is powered on. 1 = Use the command string in the terminal and save. The configuration is saved even after the terminal is powered off.

String Element	Description
p2	Character Collection Timer. 0 = Disabled 1-50 = Enabled Any value in this range enables the character collection timer. The values are measured in units of 55ms. For example, entering 50 indicates that the terminal will wait 2.75 seconds before transmitting the entered data. The character collection timer determines how long the terminal waits between one character being entered in a field and the next character. After the terminal waits the indicated amount of time, the terminal transmits the entered data.
+H	The final character of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a character collection timer command.

Suppress Command String

The suppress command string performs two functions. One of these functions is the echo suppress function, which prevents the terminal from displaying data from the bar code and RS-232 ports on the screen. The characters from the terminal keypad are not suppressed. You cannot program echo suppress from the terminal setup menu. You must send this command from the host.

Command String

Use the following command string for the suppress command:

```
CSIc1;;p2;;p4+J
```

String Elements

The following table describes the different elements of the command string for specifying echo suppress and display suppress:

String Element	**Description**
CSI	ANSI control sequence introducer (Hex 9B for 8-bit code or Hex 1B5B for 7-bit code).
c1	A compatibility byte that ensures the terminal emulation will understand the command string sent to it. The c1 is a revision level character for compatibility purposes, which will increment with each revision. The value for this parameter is 1.
;;	Two required delimiters.
p2	Echo Suppress. 0 = Disable (default, RS-232 input appears on the display) 1 = Enable (suppresses the echo, RS-232 input does not appear on the display) *Note: If you are using a barcode scanner connected to the RS-232 port, the setting of this parameter will affect barcode reads also.*
;;	Two required delimiters.

String Element	**Description**
p4	Bar Code Input. 0 = Enable bar code input. 1 = Disable bar code input. This parameter information is not saved to disk. You must re-enter 1 whenever the terminal is powered on if you want to disable bar code input. Also, this parameter does not apply to keyboard entry and this parameter can not be changed in the setup menus. *Note: If you are using a barcode scanner connected to the RS-232 port, this parameter has no function.*
+J	The final character of the ANSI command sequence. This combination indicates the end of the command string and identifies the command string as a suppress command.

LXE®
An EMS Technologies Company

Table of Contents

APPENDIX A MODE DEFINITIONS A-1

An EMS Technologies Company

Appendix A
Mode Definitions

Overview

Many of the control codes and escape sequences (discussed in chapters 5, 6, 7 and 8) are affected by the state (set or reset) of certain modes. These modes are controlled by the SM/DECSM (Set Mode) and RM/DECRM (Reset Mode) commands.

SM and RM Modes

Refer to Chapter 6, "ANSI Escape Sequences", for more information on the modes controlled by SM and RM.

Line Feed New Line Mode

If line feed new line mode (LNM) is set, then the escape sequence causes the cursor to move to column 1 of the next line.

If LNM is reset, then the control code or escape sequence causes the cursor to move down one line while staying in the same column.

Tab Stop Mode

Whether the tab position you indicate with the control code or escape sequence affects the line that the cursor is in or all the lines on the screen depends on the set and reset state of tab stop mode (TSM).

Editing Boundary Mode

Whether the entire virtual screen or only the display window is affected by the control code or escape sequence depends on the set and reset state of editing boundary mode (EBM).

Erasure Mode

Whether unprotected areas or all areas are erased with the control code or escape sequence depends on the reset or set state of erasure mode (ERM).

Select Editing Extent Mode

SEM affects the way the terminal deletes characters. Refer to the following table for the effect the value that you enter for SEM has on the characters:

When SEM is set to ...	Then the character(s) following the cursor ...
0	shift on the entire virtual screen.
1	shift only on the row the cursor is in.
2	shift only in the field the cursor is in.
3	shift only in the selected area the cursor is in.

DECSM and DECRM Modes

Refer to Chapter 7, “DEC Private Escape Sequences”, for more information on the modes controlled by DECSM and DECRM.

DEC Autowrap Mode

If DECAWM is set, autowrap mode is enabled. In other words, the characters received by the terminal automatically wrap to the next line when the cursor reaches the right border of the virtual screen.

If DECAWM is reset, autowrap mode is disabled. In other words, the characters received by the terminal do not automatically wrap to the next line when the cursor reaches the right border of the virtual screen.

DEC Origin Mode

If DECOM is set, it enables the terminal to set the home cursor position at the upper-left corner of the screen within the margin. Also moves the cursor to the origin of the scrolling region.

If DECOM is reset, it enables the terminal to set the home cursor position at the upper-left corner of the screen regardless of the margins. Also moves the cursor to the origin of the virtual screen.

DEC Print Extent

If DECPEX is set, the entire virtual screen will be printed.

If DECPEX is reset, only the scrolling region will be printed.

DEC Print Form Feed

IF DECPFF is set, a form feed will be sent after the display or scrolling region is printed.

If DECPFF is reset, no form feed will be sent.

Additional Modes

Selected Graphic Rendition

SGR indicates that the characters following the SGR command are displayed according to the selected graphic renditions (attributes). These attributes apply until the next SGR command occurs. Refer to Chapter 6, “ANSI Escape Sequences”, for the parameter values of SGR.

LXE®

An EMS Technologies Company

Table of Contents

APPENDIX B ANSI PLUS MESSAGES B-1

Appendix B
ANSI Plus Messages

Overview

At LXE we strive to be the best in everything we do, but no matter how hard we try, no matter how careful we are, and no matter how many tests we perform there are still times when everything does not go as smoothly for our customers as we would like.

This appendix contains an alphabetical listing of ANSI Plus messages that will or may appear on your terminal's screen. They are split into General Messages and Fatal Error messages. General Messages do not require ANSI Plus be shut down while Fatal Error Messages usually require the ANSI Plus application be closed.

1. Write down what happened and what activity you were performing just before the problem occurred.
2. Write down ANSI Plus message text and what actions you took immediately after the problem occurred. This information will help when you call LXE for assistance, if necessary.

If you cannot resolve the problem using the information in this reference guide, contact LXE as described in "Contacting LXE".

Legend:

TN	Message appears during 2.4GHz radio operation.
RF	Message appears during 900MHz radio operation.
SCR	Screen message directed toward current activity.
LOG	Log file message.

Note: Check log file (file AP.LOG) periodically to review messages.

General Messages

Already connected.

Message ID: OpenAmConnMsg
It Means ... TELNET client is already connected to a host.
User Action: Press <Enter> to return to host connection.
Message Type: TN, SCR

Ambiguous command. (‘?’ for help).

Message ID: AmbCmd
It Means ... TELNET system did not understand command entered at TELNET prompt.
User Action: Press ? key for help.
Message Type: TN, SCR

Ambiguous display cmd arg. ('display ?’ for help).

Message ID: AmbDisArg
It Means ... Argument to the Display command was ambiguous.
User Action: Type Display command with no argument.
Message Type: TN, SCR

Ambiguous help cmd arg.

Message ID: AmbHlpArg
It Means ... Argument to the Help command was ambiguous.
User Action: Type HELP command with no argument.
Message Type: TN, SCR

Ambiguous mode cmd arg. ('mode ?' for help).

Message ID: AmbModArg
It Means ... Argument to the Mode command was ambiguous.
User Action: Type mode ? for help.
Message Type: TN, SCR

Ambiguous send cmd arg. ('send ?' for help).

Message ID: AmbSndArg
It Means ... Argument to the Send command was ambiguous.
User Action: Type send ? for help.
Message Type: TN, SCR

Ambiguous set cmd arg. ('set ?' for help).

Message ID: AmbSetArg
It Means ... Argument to the Set command was ambiguous.
User Action: Type set ? for help.
Message Type: TN, SCR

Ambiguous toggle cmd argument ('toggle ?' for help).

Message ID: AmbTogArg
It Means ... Argument to the Toggle command was ambiguous.
User Action: Type toggle ? for help.
Message Type: TN, SCR

Can’t set attrid:

Message ID: LxeprivaAttrIdErr
It Means ... LXE private sequence parameter indicated was out of its possible range.
User Action: Check the host application.
Message Type: LOG

Cannot Determine Current NC. Swap Anyway? Press Y or N

Message ID: SwitchToOtherNC
It Means ... Confirm the switch to another NC.
User Action: Press Y to switch or N to cancel action.
Message Type: RF, SCR

Comm Failure. Press Enter Key to Retry.

Message ID: CommFailure
It Means ... General communication failure.
User Action: Press Enter key to retry communication.
Message Type: RF, SCR

Communication Link Okay. Press Any Key to Continue

Message ID: InqPassed
It Means ... Communications test passed.
User Action: Press any key to continue.
Message Type: RF, SCR

Configured for Primary NC. Switch to Secondary NC? Press Y or N

Message ID: SwitchToSecondary
It means ... Confirm the switch to Secondary NC.
User Action: Press Y to switch or N to cancel action.
Message Type: RF, SCR

Configured for Secondary NC. Switch to Primary NC? Press Y or N

Message ID: SwitchToPrimary
It means ... Confirm the switch to Primary NC.
User Action: Press Y to switch or N to cancel action.
Message Type: RF, SCR

Connect failed (AutoLogin).

Message ID: ConnErrAutoLog
It means ... Connection to host failed while using Autologin.
User Action: Try to connect manually using the same host, user Id and password. Change the Autologin parameters in configuration utility, if necessary.
Message Type: TN, SCR

Connect Failed: FTP Error

Message ID: FTPConErr
It means ... Connection to host failed, giving FTP error number.
User Action: Look up the error in the FTP documentation. (Error number 13 is a timeout.)
Message Type: TN, SCR

Connect Pending. Awaiting Connect Confirm from RFLIB

Message ID: ConnectingVerbose
It means ... Protocol stack is connecting now.
User Action: None.
Message Type: RF, SCR.

Connecting

Message ID: ConnectingTerse
It means ... Connection is in progress.
User Action: None.
Message Type: RF, SCR

Connecting to Primary NC

Message ID: ConnectingToPrimary
It means ... Message is displayed during an NC swap.
User Action: None.
Message Type: RF, SCR

Connecting to Secondary NC

Message ID: ConnectingToSecondary
It means ... Message is displayed during an NC swap.
User Action: None.
Message Type: RF, SCR

Connecting...

Message ID: ConnectingMsg
It means ... Connection to host is in progress.
User Action: None.
Message Type: TN, SCR

Connection abnormally closed.

Message ID: ConnAbnormClose
It means ... The TELNET connection was unexpectedly closed.
User Action: Re-connect to host.
Message Type: TN, SCR

Connection closed.

Message ID: ConnClosed
It means ... Displayed when the user or host drops the TELNET connection.
User Action: Re-connect to host.
Message Type: TN, SCR

Couldn't resolve host

Message ID:	ResvHostErr
It means ...	The host name specified in the Open or Connect command could not be resolved to an IP address.
User Action:	Verify the Domain Name Server specified in configuration utility.
Message Type:	TN, SCR

Delete error on:

Message ID:	FrmutilsDeleteErr
It means ...	Had an error when deleting a stored form.
User Action:	Verify the file is not write protected.
Message Type:	LOG

Delete Operation Failed. File Could Be Read Only. Press Any Key to Continue

Message ID:	DeleteFailed
It means ...	Had an error when deleting a stored form.
User Action:	Verify the file is not write protected.
Message Type:	SCR

Delete the Selected Form? Press Y or N

Message ID:	ConfirmDelete
It means ...	Confirm that you want to delete the selected stored form.
User Action:	Press Y to delete or N to cancel action.
Message Type:	SCR

Disconnected. Channel busy (NB CSAM). Please Power Down.

Message ID:	DiscBusy
It means ...	RF channel was so busy that the terminal lost connection.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Disconnected. Comm lost, cause unknown. Please Power Down.

Message ID:	DiscUnknown
It means ...	Lost radio connection to network.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Disconnected. Configuration error. Please Power Down.

Message ID:	DiscConfig
It means ...	There was a radio configuration problem.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Disconnected. Modem/radio H/W failure. Please Power Down.

Message ID:	DiscHW
It means ...	Radio is not installed in terminal, or it is not working.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Disconnected. Network connection lost (explicit). Please Power Down

Message ID:	DiscNCFailure
It means ...	Terminal was disconnected by some agent.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. Normal remote user initiated disconnect. Please Power Down

Message ID:	DiscNormal
It means ...	You have been disconnected by a remote user.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. Protocol Error. Please Power Down

Message ID:	DiscProtErr
It means ...	Protocol error caused disconnect.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. Remote provider congested. Please Power Down

Message ID:	DiscRemProvCong
It means ...	Disconnected due to remote provider congestion.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. Remote user rejected connection. Please Power Down.

Message ID:	DiscRemUserRej
It means ...	Remote user rejected connection.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. Remote user unknown to remote provider. Please Power Down.

Message ID:	DiscRemUserNotFound
It means ...	User unknown to remote provider.
User Action:	Power the terminal down.
Message Type:	RF, SCR

Disconnected. SS modem out of sync. Please Power Down.

Message ID:	DiscSync
It means ...	Spread spectrum radio lost connection with network radios.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Disconnected. Undefined reason. Please Power Down.

Message ID:	DiscUnknownReason
It means ...	User was disconnected due to an undetermined error.
User Action:	Turn terminal off.
Message Type:	RF, SCR

Execute the Selected Form? Press Y or N

Message ID: ConfirmExecute
It means ... Confirm execution of the selected stored form.
User Action: Press Y to execute the form or N to cancel action.
Message Type: SCR

Failed to execute ansi file

Message ID: AfplayerExecuteFailure
It means ... Failed to open a file during stored form execution.
User Action: Check that the file being executed is correct.
Message Type: LOG

Failed to read VtConfig group of lxe.ini

Message ID: NmfilehaVtConfigReadErr
It means ... LXE.INI file read error.
User Action: None.
Message Type: LOG

Flow Control Delay

Message ID: FlowControlTerse
It means ... Radio is busy sending data to the host.
User Action: Wait for message to be removed from the display.
Message Type: RF, SCR

Flow Control. RFLIB Buffers Filled by TE. Please Stand By...

Message ID: FlowControlVerbose
It means ... Radio is busy sending data to the host.
User Action: Wait for message to be removed from the display.
Message Type: RF, SCR

Format is 'set Name Value'. 'set ?' for help.

Message ID: SetFormatMsg
It means ... TELNET Set command was used incorrectly.
User Action: Type set ? for help.
Message Type: TN, SCR

Incomplete command line, or line too long.

Message ID: LineTooLongMsg
It means ... The line typed at the TELNET prompt was not terminated with a CR or LF.
User Action: Make sure you are not in character mode.
Message Type: TN, SCR

Initializing

Message ID: Initializing
It means ... 900MHz terminal is initializing the RF link.
User Action: None. On most platforms this message is displayed and removed so quickly the user never sees it.
Message Type: RF, SCR

Invalid command. ('?' for help).

Message ID: InvCmd
It means ... Unknown TELNET command was entered at the TELNET prompt.
User Action: Type ? for help.
Message Type: TN, SCR

Invalid display cmd arg. ('display ?' for help).

Message ID: InvDisArg
It means ... An invalid argument was used after the TELNET Display command.
User Action: Type display ? for help.
Message Type: TN, SCR

Invalid help cmd arg.

Message ID: InvHlpArg
It means ... An invalid argument was used after a TELNET Help command.
User Action: Type help ? for help.
Message Type: TN, SCR

Invalid mode cmd arg. ('mode ?' for help).

Message ID: InvModArg
It means ... An invalid argument was used after a TELNET Mode command.
User Action: Type mode ? for help.
Message Type: TN, SCR

Invalid send cmd arg. ('send ?' for help).

Message ID: InvSndArg
It means ... An invalid argument was used after a TELNET Send command.
User Action: Type send ? for help.
Message Type: TN, SCR

Invalid set cmd arg. ('set ?' for help).

Message ID: InvSetArg
It means ... An invalid argument was used after a TELNET Set command.
User Action: Type set ? for help.
Message Type: TN, SCR

Invalid toggle cmd arg. ('toggle ?' for help).

Message ID: InvTogArg
It means ... An invalid argument was used after a TELNET Toggle command.
User Action: Type toggle ? for help.
Message Type: TN, SCR

Missing 'send' cmd arg. ('send ?' for help).

Message ID: NeedSndArg
It means ... TELNET Send command did not have the required argument.
User Action: Type send ? for help.
Message Type: TN, SCR

Need an arg to 'toggle' cmd. ('toggle ?' for help).

Message ID: NeedTogArg
It means ... TELNET Toggle command did not have the required argument.
User Action: Type toggle ? for help.
Message Type: TN, SCR

Need to be connected first.

Message ID: NeedConnectMsg
It means ... TELNET command applies only if there is already a host connection established.
User Action: Connect to host before using this command.
Message Type: TN, SCR

No connection.

Message ID: NoConn
It means ... Connection status displayed for a TELNET Status command.
User Action: None.
Message Type: TN, SCR

No Stored Forms Available. Press Any Key to Continue

Message ID: NoStoredFormsAvailable
It means ... There are no stored forms available.
User Action: If stored forms are to be used, have the host application download them to the terminal.
Message Type: SCR

Not Connected. Please Power Down

Message ID: NotConnectedTerse
It means ... Disconnected.
User Action: Power the terminal down and power on again.
Message Type: RF, SCR

Open cmd, bad # args.

Message ID: OpenBadNumArgs
It means ... The TELNET Open command was used with an incorrect number of arguments.
User Action: Type open ? for help.
Message Type: TN, SCR

Open cmd, name too long.

Message ID:	OpenNameLongMsg
It means ...	The TELNET Open command was used with a host name that was too long.
User Action:	Use a shorter host name *or* IP address.
Message Type:	TN, SCR

Out Of Range. Channel busy (NB CSAM). Press Enter Key to Test Link

Message ID:	OORBusy
It means ...	Terminal is out of range because the RF channel is busy.
User Action:	Press Enter key.
Message Type:	RF, SCR

Out Of Range. Comm lost, cause unknown. Press Enter Key to Test Link.

Message ID:	OORUnknown
It means ...	Terminal is out of range for an undetermined reason.
User Action:	Press Enter key.
Message Type:	RF, SCR

Out Of Range. Configuration error. Press Enter Key to Test Link.

Message ID:	OORConfig
It means ...	Terminal is out of range due to an invalid configuration.
User Action:	Press Enter key.
Message Type:	RF, SCR

Out Of Range. Modem/radio H/W failure. Press Enter Key to Test Link.

Message ID:	OORHW
It means ...	Terminal is out of range due to a terminal radio problem.
User Action:	Press Enter key.
Message Type:	RF, SCR

Out Of Range. SS modem out of sync. Press Enter Key to Test Link.

Message ID:	OORSync
It means ...	Spread spectrum radio has lost connection with network radio.
User Action:	Press Enter key.
Message Type:	RF, SCR

Out Of Range. Undefined Reason. Press Enter Key to Test Link.

Message ID:	OORUnknownReason
It means ...	Terminal is out of range for an undetermined reason.
User Action:	Press Enter key.
Message Type:	RF, SCR

Rename error on:

Message ID:	FrmutilsRenameErr
It means ...	There was an error renaming a stored form.
User Action:	None.
Message Type:	LOG

Rename File from [formname] to [formname]

Message ID:	RenameText
It means ...	Confirm stored form name change.
User Action:	Press ESC to ignore change, press ENTER to accept change.
Message Type:	SCR

Rename Operation Failed. File already exists or name invalid. Press Any Key to Continue

Message ID:	RenameFailed
It means ...	Stored Form cannot be renamed.
User Action:	Press any key to continue.
Message Type:	SCR

Resolving hostname...

Message ID:	ResolveHostMsg
It means ...	Host name specified in Open command is being resolved to an IP Address.
User Action:	None.
Message Type:	TN, SCR

Stored form read error on:

Message ID:	FrmutilsReadErr
It means ...	Form read error on file indicated.
User Action:	None.
Message Type:	LOG

Swapping NCs

Message ID:	SwappingNCs
It means ...	Connecting to the other NC.
User Action:	None.
Message Type:	RF, SCR

Timed Out Before Receiving Connect Confirm from RFLIB. Please Power Down

Message ID:	ConnectTimedOut
It means ...	Allowable connect time expired before connect was successful.
User Action:	Check radio configuration: NetworkId, StationId, and NC3ID.
Message Type:	RF, SCR

Tried to Connect Before Last Session Was Dropped. Retrying...

Message ID:	ConnectErrorVerbose
It means ...	Error on connect.
User Action:	None.
Message Type:	RF, SCR

Unable to write form:

Message ID:	PmstoredWriteErr
It means ...	Error writing to indicated form.
User Action:	None.
Message Type:	LOG

WARNING. Communications Failure. Press Enter Key to Retry

Message ID:	InqFailed
It means ...	There was an RF communication failure.
User Action:	Press the Enter key to run communications inquiry test again.
Message Type:	RF, SCR

WARNING. Communications Test in Progress

Message ID:	InqInProgress
It means ...	There is a communications inquiry test in progress.
User Action:	None.
Message Type:	RF, SCR

Fatal Error Messages

Fatal error: Error extracting attribute. Press Any Key to Exit Program

Message ID: InifileExtractErr

It means ... Error when extracting data from .INI file.

User Action: Press any key. Verify that the .INI file has not been corrupted.

Fatal error: Error loading system font. Press Any Key to Exit Program

Message ID: WinmgrSysFontErr

It means ... Windows manager system font error.

User Action: Press any key. Check font files.

Fatal error: Error reading NRC font file. Press Any Key to Exit Program

Message ID: ChsetNRCError

It means ... Could not read font file.

User Action: Press any key. Make sure the font file exists.

Fatal error: Error reading ReadFile. Press Any Key to Exit Program

Message ID: InifileReadErr

It means ... Error when reading INI file.

User Action: Press any key. Check that the .INI file(s) are not corrupted.

Fatal error: Error reading. Press Any Key to Exit Program

Message ID: VirttermUpErrCntReadErr

It means ... Failed to read the Error count to File.

User Action: Press any key. Check the Errcount path and filename in configuration utility.

Fatal error: Error writing. Press Any Key to Exit Program

Message ID: VirttermUpErrCntWriteErr

It means ... Failed to read the Error count to File.

User Action: Press any key. Check the Errcount path and filename in configuration utility.

Fatal error: Failed to open keyboard driver interrupt file. Press Any Key to Exit Program

Message ID: LxeiocFileOpenError

It means ... The keyboard driver was loaded in a different directory, and/or ANSI Plus can’t open the keyboard data file.

User Action: Press any key.

Fatal error: Failed to read value from keyboard driver interrupt file. Press Any Key to Exit Program

Message ID: LxeiocFileReadError

It means ... Failed to read value from keyboard driver interrupt file.

User Action: Press any key.

Fatal error: LXE I/O driver not installed. Press Any Key to Exit Program

Message ID:	LxeiocDriverNotInstalled
It means ...	Required LXE driver not installed.
User Action:	Press any key. Add the driver to the AUTOEXEC.BAT file and restart the terminal.

Fatal error: Read error. Press Any Key to Exit Program

Message ID:	FdatasrcReadErr
It means ...	Error when reading data from the input file.
User Action:	Press any key. If you are running with file input instead of the normal RF, and the file specified does not exist, this would happen. Make sure the specified file exists.

Fatal error: RFLIB is not loaded. Press Any Key to Exit Program

Message ID:	RFbiosRFlibNotLoaded
It means ...	LXE’s RFLIB kernel is not loaded.
User Action:	Press any key. Add radio drivers to AUTOEXEC.BAT file and restart the terminal.

Fatal error: TCPIP Kernel is not loaded. Press Any Key to Exit Program

Message ID:	TnclientNoKernel
It means ...	TCPIP Kernel was not found.
User Action:	Press any key. Check radio and protocol sections in configuration utility and reboot the terminal.

Table of Contents

An EMS Technologies Company

Appendix C
Quick Reference Guide

Functionality List

Notes

Reset mode terminating character is l, set mode is h.

C	The configuration utility equivalent of this command.
M	The modes that affect this command.
R	The effect generated when this mode is reset.
S	The effect generated when this mode is set.

BKC	Bar code/Keypad Setup Private Sequence
CCT	Character Collection Timer Private Sequence
DECOM	DEC Origin Mode
DECPEX	DEC Print Extent Mode
DECPFF	DEC Print Form Feed Mode
DRM/SM	DECRM/DECSM
EBM	Editing Boundary Mode
ERM	Erasure Mode
LNM	Line Feed/New Line Mode
LRM/SM	LXERM/LXESM
NRC	National Replacement Character Sets
SEM	Select Editing Extent Mode
SGR	Select Graphic Rendition
SRM	Send/Receive Mode
TSM	Tab Stop Mode
TSS	Transmit/Screen Setup Private Sequence

Cursor Positioning

Cursor Pos.	Description	Hex Value / Esc Seq	Notes
BS	Backspace	0x08 (^H)	
CUB	Cursor Back	ESC[PnD	M=EBM
CUF	Cursor Forward	ESC[PnC	M=EBM
CUD	Cursor Down	ESC[PnB	M=Scroll Reg, EBM
CUU	Cursor Up	ESC[PnA	M=Scroll Reg, EBM
CUP	Cursor Position	ESC[Pr;PcH	M=Scroll Region, DECOM
HVP	Horizontal Vertical Pos	ESC[Pr;Pcf	M=Scroll Region, DECOM; Same as CUP
CHA	Cursor Horizontal Absolute	ESC[PnG	
DECOM	DEC Origin Mode	ESC[?6l (or h)	R=absolute, S=relative (DRM/SM)
CR	Carriage Return	0x0d (^M)	
LF	Line Feed	0x0a (^J)	M=Scroll Region, LNM
FF	Form Feed	0x0c (^L)	M=Scroll Region, LNM; Same as LF
VT	Vertical Tab	0x0b (^K)	M=Scroll Region, LNM; Same as LF
IND	Index	0x84 (ESC D)	M=Scroll Region
RI	Reverse Index	0x8d (ESC M)	M=Scroll Region
LNM	Line Feed/New Line Mode	ESC[20l (or h)	R=LF, S=LF+CR (RM/SM); See Transmission to Host
LXENE	New Line Echo Mode	ESC[=2l (or h)	R=neither, S=CR+LF (LRM/SM); Also TSS 6; C=EchoCrLf
CNL	Cursor Next Line	ESC[PnE	M=EBM
NEL	Next Line	0x85 (ESC E)	M=Scroll Region
CVT	Cursor Vertical Tab	ESC[PnY	M=EBM
CPL	Cursor Previous Line	ESC[PnF	M=EBM

Forms

Forms	Description	Hex Value/ ESC Seq	Notes
HT	Horizontal Tab	0x09 (^I)	No tab to protected cells
CHT	Cursor Horizontal Tab	ESC[PnI	No tab to protected cells
CBT	Cursor Back Tab	ESC[PnZ	No tab to protected cells
HTJ	Horizontal Tab w/Justify	0x89 (ESC I)	No tab to protected cells
HTS	Horizontal Tab Set	0x88 (ESC H)	M=TSM
CTC	Cursor Tab Clear	ESC[PnW	(TSM reset/TSM set); 0,2=column/position, 4=all/row, 5=all/all
TBC	Tab Clear	ESC[Png	(TSM reset/TSM set); 0=column/position, 2=all/row, 3=all/all
TSM	Tab Set Mode	ESC[18l (or h)	R=virtual, S=curr line (RM/SM)
DAQ	Define Area Qualification	ESC[Pno	0=all data, 1=prot, 2=display, 3=num, 4=alpha, 5=right just, 6=zero filled, 7=tab, 8=selected, 9=space filled, 10=hidden, 15=keyboard input only
DECSCA	DEC Select Char Attrib	ESC[Pn"q	0,2=unprot, 1=prot; Same as EPA/SPA
SPA	Start Protected Area	0x96 (ESC V)	Same as DAQ 1
EPA	End Protected Area	0x97 (ESC W)	Current area is all input
SSA	Start Selected Area	0x86 (ESC F)	
ESA	End Selected Area	0x87 (ESC G)	Current area is all input
EA	Erase Area	ESC[PnO	0=to end, 1=from start, 2=all; M=ERM, EBM; Does not erase area type
EF	Erase Field	ESC[PnN	0=to end, 1= from start, 2=all; M=ERM, EBM; Does not erase area type
LEM	Local Edit Match	ESC[Pn+A	R=display, S=hide

Forms	Description	Hex Value/ ESC Seq	Notes
LEM Alpha WC	Alpha only wildcard	ESC[c1;;p2+G	Matches A-Z, a-z, comma, period, dash, space; C=AlphaOnly
LEM Digits WC	Digits only wildcard	ESC[c1;;;p3+G	Matches 0-9; C=DigitsOnly
LEM Numeric WC	Numeric only wildcard	ESC[c1;;;;p4+G	Matches 0-9, plus, comma, period, dash, space; C=NumericOnly
LEM Alphanum WC	Alphanum only wildcard	ESC[c1;;;;;p5+G	Matches A-Z, a-z, 0-9, plus, comma, period, dash, space; C = AlphaNumericOnly
LEM Match All WC	Match all wildcard	ESC[c1;;;;;;p6+G	All GL and GR; C=MatchAll
Stored forms checksum	Stored forms checksum	ESC^FCformESC\	
Stored forms delete	Stored forms delete	ESC^FDform ESC\	A space (0x20) follows the form name
Stored forms execute	Stored forms execute	ESC^FXformESC\	
Stored forms list	Stored forms list	ESC^FLESC\	
Stored forms rename	Stored forms rename	ESC^FRform1 form2ESC\	A space (0x20) follows name of form1
Stored forms write	Stored forms write	ESC^FWform dataESC\	A space (0x20) follows the form name
Stored Forms Dir	Stored Forms Directory	-	C=StoredFormsDir

Transmission To Host

Transmission To Host	Description	Hex Value/ Esc Seq	Notes
GATM	Guarded Area Transmit Mode	ESC[1l (or h)	R=unguarded, S=entire screen (RM/SM)
MATM	Multiple Area Transmit Mode	ESC[15l (or h)	R=cursor selected, S=all selected (RM/SM)
SATM	Selected Area Transmit Mode	ESC[17l (or h)	R=selected, S=virtual (RM/SM)
LXEAS	Auto Send Mode	ESC[=3l (or h)	R=disable, s=enable (LRM/SM); SRM must be reset; Also TSS 3; C=AutoTransmit
SRM	Send Receive Mode	ESC[12l (or h)	R=local echo, S=char (RM/SM); Also TSS 10; C=SendReceiveMode
LXELM	Line Mode	ESC[=1l (or h)	R=block, S=scroll (LRM/SM); Also TSS 5; C=ScrollMode
DECCKM	Cursor Keys Mode	ESC[?1l (or h)	R=cursor, S=application (DRM/SM); Also BKC 7; C=CursorAppMode
DECKPAM	Keypad Application Mode	ESC=	Also BKC 6; C=KeypadAppMode
DECKPNM	Keypad Numeric Mode	ESC>	Also BKC 6; C=KeypadAppMode
Keypad Enable Mode	-	-	C=KpamEnable
DECSCL	Select Operating Level	ESC[62;Pn"p	R=8 bit, S=7 bit
S7C1T	Select 7 Bit Transmission	ESC F	A space (0x20) precedes F; C=EightBitMode
S8C1T	Select 8 Bit Transmission	ESC G	A space (0x20) precedes G; C=EightBitMode
LNM	Line Feed/New Line Mode	ESC[c1;;;;p4+F	R=CR, S= CR+LF; Also TSS 4; See cursor positioning; C=TransmitCR/LF
OLM	Online Mode	ESC[c1;;;;;;;;p8+F	R=online, S=offline; Also TSS 8; C=OnlineMode
Char Coll Timer	Character Collection Timer	ESC[c1;;p2+H	R=disabled, S=enabled; Also CCT 2; C=CharColTimer

Editing

Editing	Description	Hex Value/ Esc Seq	Notes
ECH	Erase Character	ESC[PnX	M=ERM, EBM
DCH	Delete Character	ESC[PnP	M=SEM, EBM; Aborts at protected cells
ICH	Insert Character	ESC[Pn@	M=SEM, EBM; Aborts at protected cells
SEM	Select Editing Extent Mode	ESC[PnQ	0=display, 1=row, 2=field, 3=area; M=EBM
DL	Delete Line	ESC[PnM	M=EBM, Scroll Region
IL	Insert Line	ESC[PnL	M=EBM, Scroll Region
ED	Erase Display	ESC[PnJ	0=to end, 1=from start, 2=all; M=ERM, EBM; Erases area type/boundaries, homes cursor.
EL	Erase Line	ESC[PnK	0=to end, 1=from start, 2=all; M=EBM, ERM; Erases area type/boundaries
DECSED	DEC Selective Erase In Display	ESC[?PnJ	0=to end, 1=from start, 2=all; M=EBM; Never erases protected cells or area types/boundaries, homes cursor
DECSEL	DEC Selective Erase In Line	ESC[?PnK	0=to end, 1=from start, 2=all; M=EBM; Never erases protected cells or area types/boundaries
ERM	Erasure Mode	ESC[6l (or h)	R=cannot erase prot, S=erase prot (RM/SM)
SD	Scroll Down	ESC[PnT	M=Scroll Region
SU	Scroll Up	ESC[PnS	M=Scroll Region
DECSTBM	DEC Set Top/ Bottom Margins	ESC[Pt;Pbr	M=DECOM; Must have full screen terminal to use; Min height of region is 2 rows, Homes cursor according to DECOM
IRM	Insert/Replace Mode	ESC[4l (or h)	R=overwrite, S=insert (RM/SM)
EBM	Editing Boundary Mode	ESC[19l (or h)	R=phys disp, S=virt disp (RM/SM)

Character Rendition

Character Rendition	Description	Hex Value/ Esc Seq	Notes
SGR	Select Graphics Rendition	ESC[Pnm	0=normal, 1,22=bold, 4,24=underline, 5,25=blink, 7,27=reverse; First num enables, second disables
DECDHL	DEC Double High Line	ESC#3 (top) ESC#4 (bot)	Must be used in pairs, same text must appear on each pair of lines
DECDWL	DEC Double Wide Line	ESC#6	
DECSWL	DEC Single High/ Single Wide Line	ESC#5	
DECSCNM	DEC Screen Mode	ESC[?5l (or h)	R=normal, S=reverse (DRM/SM)
DECTCEM	DEC Text Cursor Enable Mode	ESC[?25l (or h)	R=invisible, S=visible (DRM/SM); C=CursorVisible
Cursor Type	Cursor Type	-	Block/underline; C=CursorType
Cursor Blink	Cursor Blink	-	Blink/steady; C=CursorBlink

Character Sets

Character Sets	**Description**	**Hex Value/ Esc Seq**	**Notes**
SI	Shift In	0x0f (^O)	
SO	Shift Out	0x0e (^N)	
LS0	Lock Shift G0	0x0f (^O)	Same as SI
LS1	Lock Shift G1	0x0e (^N)	Same as SO
SS2	Single Shift G2	0x8e (ESC N)	
SS3	Single Shift G3	0x8f (ESC O)	
LS1R	Lock Shift G1 Right	ESC~	
LS2	Lock Shift G2	ESCn	
LS2R	Lock Shift G2 Right	ESC}	
LS3	Lock Shift G3	ESCo	
LS3R	Lock Shift G3 Right	ESC|	
G0	Designate G0	ESC(F	
G1	Designate G1	ESC)F	
G2	Designate G2	ESC*F	
G3	Designate G3	ESC+F	
DECNRCM	DEC Char Set Mode	ESC[?42l (or h)	R=multinat, S=NRC (DRM/SM); C=NationalMode
Keyboard Type	Keyboard Type	-	C=KeyboardType

Bar Code

Bar code	**Description**	**Hex Value/ Esc Seq**	**Notes**
Left Justify	Left Justify	ESC[c1;;p2+E	R=off, S=on (BKC); Requires block mode; C=LeftJustify
Clear	Clear	ESC[c1;;;p3+E	R=off, S=on (BKC); Requires block mode; C=ClearData
Truncate	Truncate	ESC[c1;;;;p4+E	R=off, S=on (BKC); Requires block mode; C=TruncateData
Auto Advance	Auto Advance	ESC[c1;;;;;p5+E	R=off, S=on (BKC); Requires block mode; C=AutoAdvance
Suppress Barcode Input	Suppress Barcode Input	ESC[c1;;;;p4+J	R=enable, S=suppress

Printing

Printing	Description	Hex Value/ Esc Seq	Notes
MC	Media Copy	ESC[Pni	4=off, 5=on
DC2	Device Control 2	0x12 (^R)	Ignored from host
DC4	Device Control 4	0x14 (^T)	
APM	Auto Print Mode	ESC[?Pni	4=off, 5=on
PL	Print Line	ESC[?1i	
PS	Print Screen	ESC[0i	M=DECPEX, DECPFF, Scroll Region
DECPFF	DEC Print Form Feed Mode	ESC[?18l (or h)	R=no term, S=term (DRM/SM)
DECPEX	DEC Print Extent Mode	ESC[?19l (or h)	R=scroll region only, S=virtual screen (DRM/SM)

Reports / Status / Test

Reports /Status/ Test	Description	Hex Value/ Esc Seq	Notes
ENQ	Enquire	0x05 (^E)	
CPR	Cursor Position Report	ESC[Pr;PcR	Only sent from terminal; Position relative to virtual display
DSR	Device Status Report	ESC[Pnn	5=status, 6=CPR, 26=keyboard
DA Primary	Device Attribute, Primary	ESC[c	
DA Secondary	Device Attribute, Secondary	ESC>c	
DECALN	DEC Screen Alignment Pattern	ESC#8	

Window Manager

Window Manager	Description	Hex Value/ Esc Seq	Notes
NP	Next Page	ESC[PnU	Stops at lower right of virt disp
PP	Previous Page	ESC[PnV	Stops at upper left of virt disp
SL	Scroll Left	ESC[Pn @	A space (0x20) precedes @
SR	Scroll Right	ESC[Pn A	A space (0x20) precedes A
Lock win Mode	Lock Window Mode	ESC[c1;;p2+D	R=tracking, S=lock; C=LockWindow
Lock win row	Lock Window Row	ESC[c1;;;p3+D	1-24; C=LockRow
Lock win col	Lock Window Column	ESC[c1;;;;p4+D	1-80; C=LockCol
Win move mode	Window Movement Mode	ESC[c1;;;;;p5+D	R=quad, S=full; C=HalfWinIncs
Cursor quad	Cursor Quadrant	ESC[c1;;;;;;p6+D	1=top right, 2=top left, 3=bottom left, 4=bottom right; C=HostQuad

Keyboard Control

Keyboard Control	Description	Hex Value/ Esc Seq	Notes
KAM	Keyboard Action Mode	ESC[2l (or h)	R=unlock, S=lock (RM/SM)
DMI	Disable Manual Input	ESC`	Same as SM KAM
EMI	Enable Manual Input	ESCb	Same as RM KAM
Compose Mode	Compose Mode	ESC[c1;;;;;;;;p8+E	R=off, S=on (BKC); C=ComposeKey
Initial Scroll Delay	Initial Scroll Delay	-	C=InitialScrlDly
Repeat Scroll Delay	Repeat Scroll Delay	-	C=RepeatScrlDly

Debugging

Debugging	Description	Hex Value/ Esc Seq	Notes
DSR (90+)	Device Status Report	ESC[Pnn	90=screen w/o just+lock, 92=screen w/just+lock, 93=screen w/o just or lock 94=phys window coords
CRM	Control Representation Mode	ESC[3l (or h)	R=execute, S=display (RM/SM); Also BKC p9; C=DisplayCtrlCodes
Display Bad Seqs	Display Bad Sequences	ESC[c1;;;;;;;;;p10+E	R=ignore, S=display (BKC); C=DisplayBadSeqs
Error Count Path	Error Count Path	-	C=ErrorCountPath
Error Count FName	Error Count File Name	-	C=ErrorCountFName
Verbose RF Exp Messages	Verbose RF Exception Messages	-	900 MHz only; C=VerboseExcMsgs

Local Input/Output

Local I/O	Description	Hex Value/ Esc Seq	Notes
COM1/2 Baud Rate	COM1/2 Baud Rate	-	C=Com1BaudRate (or 2)
COM1/2 Data Bits	COM1/2 Data Bits	-	C=Com1DataBits (or 2)
COM1/2 Parity	COM1/2 Parity	-	C=Com1Parity (or 2)
COM1/2 Stop Bits	COM1/2 Stop Bits	-	C=Com1StopBits (or 2)
COM1/2 Half Duplex	COM1/2 Half Duplex	-	C=Com1HalfDuplex (or 2)
COM1/2 XON	COM1/2 XON	-	C=Com1XON (or 2)
COM1/2 HW Flow Ctrl	COM1/2 HW Flow Ctrl	-	C=Com1HwFlowCtrl (or 2)
COM1/2 Time Out	COM1/2 Time Out	-	C=Com1TimeOut (or 2)
COM1/2 Power	COM1/2 Power	-	C=Com1Power (or 2)
COM1/2 Pass Term Char	COM1/2 Pass Term Char	-	C=Com1PassTermChar (or 2)
DC1/XON	Device Control 1 (XON)	0x11 (^Q)	Ignored from host
DC3/XOFF	Device Control 3 (XOFF)	0x13 (^S)	Ignored from host

Miscellaneous

Miscellaneous	Description	Hex Value/ Esc Seq	Notes
BEL	Bell	0x07 (^G)	
Beep Pattern	Beep Pattern	ESC[p1;p2+B	p1=pattern (1-255), p2=volume (1-7); C=BeeperVolume
DEL	Delete	0x7f	Ignored from host
ESC	Escape	0x1b (^[)	
CSI	Control Sequence Introducer	0x9b (ESC[)	
DCS	Device Control String	0x90 (ESC P)	
PM	Privacy Message	0x9e (ESC^)	
ST	String Terminator	0x9c (ESC\)	
FS	File Separator	0x1c (^\)	
GS	Group Separator	0x1d (^])	
RS	Record Separator	0x1e (^^)	
US	Unit Separator	0x1f (^_)	
REP	Repeat	ESC[Pnb	Only repeats GL/GR
DECRC	DEC Restore Cursor	ESC8	M=SGR, DECOM, DECAWM
DECSC	DEC Save Cursor	ESC7	M=SGR, DECOM, DECAWM
DECSTR	DEC Soft Terminal Reset	ESC[!p	
Suppress RS-232 Echo	Suppress RS-232 echo	ESC[c1;;p2+J	R=echoes, S=suppress
DECAWM	DEC Autowrap Mode	ESC[?7l (or h)	R=disable, S=enable (DRM/SM), Also TSS7; C=AutoWrap
Status Line Enable	Status Line Enable	ESC[c1;;;;;;;;;p9+F	R=off, S=on (TSS); C=StatusLine
NUL	Null	0x00 (^@)	

Terminating Character List

0x00 (^@)	NUL	Null
0x05 (^E)	ENQ	Enquire
0x07 (^G)	BEL	Bell
0x08 (^H)	BS	Backspace
0x09 (^I)	HT	Horizontal Tab
0x0a (^J)	LF	Line Feed
0x0b (^K)	VT	Vertical Tab
0x0c (^L)	FF	Form Feed
0x0d (^M)	CR	Carriage Return
0x0e (^N)	SO	Shift Out
0x0f (^O)	SI	Shift In
0x11 (^Q)	DC1/XON	Device Control 1 (XON)
0x12 (^R)	DC2	Device Control 2
0x13 (^S)	DC3/XOFF	Device Control 3 (XOFF)
0x14 (^T)	DC4	Device Control 4
0x1b (^[)	ESC	Escape
0x1c (^\)	FS	File Separator
0x1d (&])	GS	Group Separator
0x1e (^^)	RS	Record Separator
0x1f (^_)	US	Unit Separator
0x7f	DEL	Delete
ESC[Pn @	SL	Scroll Left
ESC[Pn A	SR	Scroll Right
ESC F	S7C1T	Select 7 Bit Transmission
ESC G	S8C1T	Select 8 Bit Transmission
ESC[!p	DECSTR	DEC Soft Terminal Reset
ESC[62;Pn"p	DECSCL	Select Operating Level
ESC[Pn"q	DECSCA	DEC Select Char Attrib
ESC#3 (top)		
ESC#4 (bot)	DECDHL	DEC Double High Line
ESC#5	DECSWL	DEC Single High/ Single Wide Line
ESC#6	DECDWL	DEC Double Wide Line
ESC#8	DECALN	DEC Screen Alignment Pattern
ESC(F	G0	Designate G0
ESC)F	G1	Designate G1
ESC*F	G2	Designate G2
ESC[Pn+A	LEM	Local Edit Match
ESC[+B	Beep Pattern	Beep Pattern
ESC[+D	Win Setup	Window Setup
ESC[+E	BKC	Bar code/Keypad/ Control
ESC+F	G3	Designate G3
ESC[+F	TSS	Transmit/Screen Setup
ESC[+G	LEW	Local Edit Wildcard
ESC[+H	Char Coll	Character Collector
ESC[+J	Suppress Command	Suppress Command
ESC7	DECSC	DEC Save Cursor
ESC8	DECRC	DEC Restore Cursor
ESC=	DECKPAM	Keypad Application Mode
ESC>	DECKPNM	Keypad Numeric Mode
ESC>c	DA Secondary	Device Attribute, Secondary
ESC[Pn@	ICH	Insert Character
ESC[PnA	CUU	Cursor Up

ESC[PnB	CUD	Cursor Down
ESC[PnC	CUF	Cursor Forward
ESC[PnD	CUB	Cursor Back
0x84 (ESC D)	IND	Index
ESC[PnE	CNL	Cursor Next Line
0x85 (ESC E)	NEL	Next Line
ESC[PnF	CPL	Cursor Previous Line
0x86 (ESC F)	SSA	Start Selected Area
ESC[PnG	CHA	Cursor Horizontal Absolute
0x87 (ESC G)	ESA	End Selected Area
ESC[Pr;PcH	CUP	Cursor Position
0x88 (ESC H)	HTS	Horizontal Tab Set
ESC[PnI	CHT	Cursor Horizontal Tab
0x89 (ESC I)	HTJ	Horizontal Tab w/Justify
ESC[PnJ	ED	Erase Display
ESC[?PnJ	DECSED	DEC Selective Erase In Display
ESC[PnK	EL	Erase Line
ESC[?PnK	DECSEL	DEC Selective Erase In Line
ESC[PnL	IL	Insert Line
0x8d (ESC M)	RI	Reverse Index
ESC[PnM	DL	Delete Line
ESC[PnN	EF	Erase Field
0x8e (ESC N)	SS2	Single Shift G2
ESC[PnO	EA	Erase Area
0x8f (ESC O)	SS3	Single Shift G3
ESC[PnP	DCH	Delete Character
0x90 (ESC P)	DCS	Device Control String
ESC[PnQ	SEM	Select Editing Extent Mode
ESC[Pr;PcR	CPR	Cursor Position Report
ESC[PnS	SU	Scroll Up
ESC[PnT	SD	Scroll Down
ESC[PnU	NP	Next Page
0x96 (ESC V)	SPA	Start Protected Area
ESC[PnV	PP	Previous Page
ESC[PnW	CTC	Cursor Tab Clear
0x97 (ESC W)	EPA	End Protected Area
ESC[PnX	ECH	Erase Character
ESC[PnY	CVT	Cursor Vertical Tab
ESC[PnZ	CBT	Cursor Back Tab
0x9c (ESC\)	ST	String Terminator
ESC^FCMDform dataESC\	Stored Forms	Stored Forms
0x9e (ESC^)	PM	Privacy Message
ESC`	DMI	Disable Manual Input
ESCb	EMI	Enable Manual Input
ESC[Pnb	REP	Repeat
ESC[c	DA Primary	Device Attribute, Primary
ESC[Pr;Pcf	HVP	Horizontal Vertical Pos
ESC[Png	TBC	Tab Clear
ESC[Pnh	SM	Set Mode
ESC[?Pnh	DECSM	DEC Set Mode
ESC[=Pnh	LXESM	LXE Set Mode
ESC[Pni	MC	Media Copy
ESC[?Pni	APM	Auto Print Mode
ESC[?1i	PL	Print Line
ESC[0i	PS	Print Screen

ESC[Pnl	RM	Reset Mode
ESC[?Pnl	DECRM	DEC Reset Mode
ESC[=Pnl	LXERM	LXE Reset Mode
ESC[Pnm	SGR	Select Graphics Rendition
ESCn	LS2	Lock Shift G2
ESC[Pnn	DSR	Device Status Report
ESC[Pnn	DSR (90+)	Device Status Report
ESC[Pno	DAQ	Define Area Qualification
ESCo	LS3	Lock Shift G3
ESC[Pt;Pbr	DECSTBM	DEC Set Top/Bottom Margins
ESC|	LS3R	Lock Shift G3 Right
ESC}	LS2R	Lock Shift G2 Right
ESC~	LS1R	Lock Shift G1 Right

LXE®

An EMS Technologies Company

Table of Contents

APPENDIX D ANSI PLUS FUNCTION KEY DEFINITIONS D-1

An EMS Technologies Company

Appendix D
ANSI Plus Function Key Definitions

Introduction

There are differences between the ANSI Plus and VT100/VT220 standards regarding Function Key definitions. This appendix lists the codes generated by a VT100/VT220 Function Key and an ANSI Plus Function Key.

In general, the differences between ANSI Plus and VT100/VT220 Function Keys is summarized below:

- ANSI Plus defines default values for F1-F4. VT11/VT220 does not.
- ANSI Plus uses **ESC [** instead of **CSI** (**ESC [** is the seven bit equivalent for **CSI**).
- ANSI Plus maps to PC function keys, not to native VT220 keys:
    - ANSI plus: PC Function keys F1-F12, alone and modified by Alt, Ctl, Shift
    - VT100/VT220: F1-F20
- ANSI Plus provides a Function Key Editor (Alt K), allowing the definition to be changed. Please refer to Chapter 2, “Getting Started” for more information.

The following tables show the codes generated by VT100/VT220 and ANSI Plus Function Keys.

Note: Please refer to “Using the 8-Bit Code Table” in Chapter 5, “Control Codes” for more information about the 7-Bit codes used in the following tables.

VT100/VT220 Standard, Codes Generated by Top Row Function Keys

Name on Legend Strip	Generic Name	Code Generated				
		VT200 Mode				VT100, VT52 Modes
HOLD SCREEN	(F1)*	- -				- -
PRINT SCREEN	(F2)*	- -				- -
SET-UP	(F3)*	- -				- -
DATA/TALK	(F4)*	- -				- -
BREAK	(F5)*	- -				- -
F6	F6	9/11 **CSI**	3/1 **1**	3/7 **7**	7/14 **~**	- -
F7	F7	9/11 **CSI**	3/1 **1**	3/8 **8**	7/14 **~**	- -
F8	F8	9/11 **CSI**	3/1 **1**	3/9 **9**	7/14 **~**	- -
F9	F9	9/11 **CSI**	3/2 **2**	3/0 **0**	7/14 **~**	- -
F10	F10	9/11 **CSI**	3/2 **2**	3/1 **1**	7/14 **~**	- -
F11 (ESC)	F11	9/11 **CSI**	3/2 **2**	3/3 **3**	7/14 **~**	1/11 **ESC**
F12 (BS)	F12	9/11 **CSI**	3/2 **2**	3/4 **4**	7/14 **~**	0/8 **BS**
F13 (LF)	F13	9/11 **CSI**	3/2 **2**	3/5 **5**	7/14 **~**	0/10 **LF**
F14	F14	9/11 **CSI**	3/2 **2**	3/6 **6**	7/14 **~**	- -
HELP	(F15)	9/11 **CSI**	3/2 **2**	3/8 **8**	7/14 **~**	- -
DO	(F16)	9/11 **CSI**	3/2 **2**	3/9 **9**	7/14 **~**	- -
F17	F17	9/11 **CSI**	3/3 **3**	3/1 **1**	7/14 **~**	- -
F18	F18	9/11 **CSI**	3/3 **3**	3/2 **2**	7/14 **~**	- -
F19	F19	9/11 **CSI**	3/3 **3**	3/3 **3**	7/14 **~**	- -
F20	F20	9/11 **CSI**	3/3 **3**	3/4 **4**	7/14 **~**	- -

Note: *Numbers above the* **dark red** *characters identify the character's position in the 7-bit code table. For example,* 9/11 **CSI** *refers the* **CSI** *character; located at column 9, row 11.*

ANSI Plus, Default Codes Generated by Function Keys

Function Key	Code Generated				
F1	1/11 **ESC**	4/15 **O**	5/0 **P**		
F2	1/11 **ESC**	4/15 **O**	5/1 **Q**		
F3	1/11 **ESC**	4/15 **O**	5/2 **R**		
F4	1/11 **ESC**	4/15 **O**	5/3 **S**		
F5			- -		
F6	1/11 **ESC**	5/11 **[**	3/1 **1**	3/7 **7**	7/14 **~**
F7	1/11 **ESC**	5/11 **[**	3/1 **1**	3/8 **8**	7/14 **~**
F8	1/11 **ESC**	5/11 **[**	3/1 **1**	3/9 **9**	7/14 **~**
F9	1/11 **ESC**	5/11 **[**	3/2 **2**	3/0 **0**	7/14 **~**
F10	1/11 **ESC**	5/11 **[**	3/2 **2**	3/1 **1**	7/14 **~**
F11	1/11 **ESC**	5/11 **[**	3/2 **2**	3/3 **3**	7/14 **~**
F12	1/11 **ESC**	5/11 **[**	3/2 **2**	3/4 **4**	7/14 **~**
Shift F1	1/11 **ESC**	5/11 **[**	3/2 **2**	3/5 **5**	7/14 **~**
Shift F2	1/11 **ESC**	5/11 **[**	3/2 **2**	3/6 **6**	7/14 **~**
Shift F3	1/11 **ESC**	5/11 **[**	3/2 **2**	3/8 **8**	7/14 **~**
Shift F4	1/11 **ESC**	5/11 **[**	3/2 **2**	3/9 **9**	7/14 **~**
Shift F5	1/11 **ESC**	5/11 **[**	3/3 **3**	3/0 **0**	7/14 **~**
Shift F6	1/11 **ESC**	5/11 **[**	3/3 **3**	3/2 **2**	7/14 **~**
Shift F7	1/11 **ESC**	5/11 **[**	3/3 **3**	3/3 **3**	7/14 **~**
Shift F8	1/11 **ESC**	5/11 **[**	3/3 **3**	3/4 **4**	7/14 **~**
Shift F9			- -		
Shift F10			- -		
Shift F11			- -		
Shift F12			- -		

Function Key	**Code Generated**				
Ctl F1	- -				
Ctl F2	- -				
Ctl F3	- -				
Ctl F4	- -				
Ctl F5	- -				
Ctl F6	- -				
Ctl F7	- -				
Ctl F8	- -				
Ctl F9	- -				
Ctl F10	- -				
Ctl F11	- -				
Ctl F12	- -				
Alt F1	1/11 **ESC**	5/11 **[**	3/2 **2**	3/3 **3**	7/14 **~**
Alt F2	1/11 **ESC**	5/11 **[**	3/2 **2**	3/4 **4**	7/14 **~**
Alt F3	1/11 **ESC**	5/11 **[**	3/2 **2**	3/5 **5**	7/14 **~**
Alt F4	1/11 **ESC**	5/11 **[**	3/2 **2**	3/6 **6**	7/14 **~**
Alt F5	1/11 **ESC**	5/11 **[**	3/2 **2**	3/8 **8**	7/14 **~**
Alt F6	1/11 **ESC**	5/11 **[**	3/2 **2**	3/9 **9**	7/14 **~**
Alt F7	1/11 **ESC**	5/11 **[**	3/3 **3**	3/1 **1**	7/14 **~**
Alt F8	1/11 **ESC**	5/11 **[**	3/3 **3**	3/2 **2**	7/14 **~**
Alt F9	1/11 **ESC**	5/11 **[**	3/3 **3**	3/3 **3**	7/14 **~**
Alt F10	1/11 **ESC**	5/11 **[**	3/3 **3**	3/4 **4**	7/14 **~**
Alt F11	- -				
Alt F12	- -				

Note: Numbers above the **dark red** *characters identify the character's position in the 7-bit code table. For example,* 1/11 **ESC** *refers the* **ESC** *character; located at column 1, row 11.*

Table of Contents

APPENDIX E KEY MAPS E-1

LXE®

An EMS Technologies Company

Appendix E
Key Maps

The MX1 Keypad

Alpha-Numeric Keypad (60 Key)

Please refer to the "MX1 Reference Guide", Appendix A "Key Maps", for detailed key-mapping information for the 60-key Alpha-Numeric keypad.

This section reflects the ANSI Plus specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an MX1 with an Alpha-Numeric keypad.

When using a sequence of keys that includes the 2nd key, press the 2nd key first then the rest of the key sequence. Set the On/Off condition of NumLock before pressing a key sequence. There is no visual indication of the condition of NumLock.

Note: When the MX1 boots, the default condition of NumLock is On. NumLock can be set using the MX1 BIOS Setup or toggled with a 2nd-Shift-Right Arrow key sequence.

Note: When the MX1 boots, the default condition of Caps (or CapsLock) is Off. The Caps (or CapsLock) condition can be set using the MX1 BIOS Setup or toggled with a 2nd-Shift-Left Arrow key sequence. CAPS is displayed on the screen when CapsLock is On.

To get this ANSI+ result	Press These Keys and Then				Press this key
	2nd	Shift	Ctrl	Alt	
Answerback			x		Enter
Backspace			x		H
Clear Field				x	F
Clear Screen				x	L
Compose				x	C
DEL	x				. (DOT) or Ctrl-8
Down Arrow					Down Arrow
Exit ANSI +				x	X
F1 through F10					F1 through F10
F11 through F20				x	F1 through F10
F21 through F30		x			F1 through F10
F31 through F40			x		F1 through F10
Function Key Editor				x	K
Help				x	H
INQ[1]				x	I
Left Arrow					Left Arrow

---

[1] 900MHz Radio Equipped Only

To get this ANSI+ result	Press These Keys and Then				Press this key
	2nd	Shift	Ctrl	Alt	
Page Down (in Help)	x				Down Arrow
Page Up (in Help)	x				Up Arrow
Right Arrow					Right Arrow
Send					Enter
Stored Forms				x	M
Swap Network Controller[2]			x		1
TN Escape	x		x		2
Unlock Keyboard				x	U
Up Arrow					Up Arrow
Window Down			x		Down Arrow
Window Left			x		Left Arrow
Window Lock/Home		x			Up Arrow
Window Right			x		Right Arrow
Window Up			x		Up Arrow

---

[2] 900MHz Radio Equipped Only

Numeric-Alpha Keypad (41 Key)

Please refer to the "MX1 Reference Guide", Appendix A "Key Maps", for detailed key-mapping information for the 41-key Numeric-Alpha keypad.

This section reflects the ANSI Plus specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an MX1 with a Numeric-Alpha keypad.

When using a sequence of keys that include the A/# key, first press A/#.

When using a sequence of keys that include the A/# key and the 2nd key, first press the A/# key then the 2nd key.

When using a sequence of keys that do not include the A/# key and includes the 2nd key, press the 2nd key first then the rest of the key sequence.

To get this ANSI+ result	Press These Keys and Then					Press this key
	A/#	2nd	Shift	Ctrl	SPC	
Answerback				x		Enter
Backspace						
Clear Field	x	x			x	4 (F)
Clear Screen	x	x			x	. DOT (L)
Compose	x	x			x	7 (C)
Delete						DEL or (num) Ctrl-8
Exit ANSI +	x	x			x	F10 (X)
F1 through F10						F1 through F10
F11 through F20		x			x	F1 through F10
F21 through F30			x			F1 through F10
F31 through F40				x		F1 through F10
Function Key Editor	x	x			x	3 (K)
Hot Key Help	x	x			x	6 (H)
INQ[3]	x	x			x	1 (I)
Page Down (in Help)		x				Down Arrow
Page Up (in Help)		x				Up Arrow
Send						Enter
Stored Forms	x	x			x	0 (M)
Swap Network Controller[3]				x		1
TN Escape		x		x		6
Unlock Keyboard	x	x			x	F7 (U)
Window Down				x		Down Arrow
Window Left				x		Left Arrow
Window Lock/Home			x			Up Arrow
Window Right				x		Right Arrow
Window Up				x		Up Arrow

---

[3] 900MHz Radio Equipped Only

The MX2 Keypad

38-key Keypad

Please refer to the "MX2 Reference Guide", Appendix A "Key Maps", for detailed key-mapping information for the 38-key keypad.

This section reflects the ANSI Plus specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an MX2 with a 38-key keypad.

Note: When running ANSI Plus, the Flag key (International or Fn+ALT) operates as a Shift key.

When using a sequence of keys that include the Fn key, first press Fn.

The MX2 does not use the following keys normally found on a computer keyboard:

F11	F12	Open/Closed Parentheses
Scroll Lock	Pause/Break	Print Screen/SysReq
Insert	Num Lock	Numeric keypad number keys
Right Alt		Open/Closed Square Brackets
Right Ctrl		Open/Closed Curly Braces

The Caps key provides the function of the following computer keyboard keys:

Left Shift	Right Shift	Caps Lock

Note: DblStk means to press the key twice, or doublestrike the key.

To get this ANSI+ result	Press These Keys and Then				Press this key
	Fn	DblStk	Ctrl	Alt	
Answerback			x		Enter
Backspace	x				Space
Clear Field		x		x	E**F**
Clear Screen		x		x	K**L**
Compose				x	**C**D
Delete	x				Esc
Exit ANSI +		x		x	W**X**
F1 through F5					F1 through F5
F6 through F10	x				F1 through F5
F11 through F15				x	F1 through F5
F16 through F20[4]	x			x	F1 through F5
F31 through F35			x		F1 through F5
F36 through F40	x		x		F1 through F5
Function Key Editor				x	**K**L

---

[4] ANSI Plus Function keys F21 through F30 are not supported on the MX2.

To get this ANSI+ result	Press These Keys and Then				Press this key
	Fn	DblStk	Ctrl	Alt	
Hot Key Help		x		x	G**H**
INQ[5]					N/A
Page Down (in Help)	x				3
Page Up (in Help)	x				9
Send					Enter
Stored Forms				x	**M**N
Swap Network Controller[5]					N/A
TN Escape			x		4
Unlock Keyboard				x	**U**V
Window Down	x		x		2
Window Left	x		x		4
Window Lock/Home	x			x	8
Window Right	x		x		6
Window Up	x		x		8

---

[5] 900MHz Radio Equipped Only

48-key Keypad

Please refer to the "MX2 Reference Guide", Appendix A "Key Maps", for detailed key-mapping information for the 48-key keypad.

This section reflects the ANSI Plus specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an MX2 with a 38-key keypad.

When using a sequence of keys that include the Fn key, first press Fn.

The MX2 does not use the following keys normally found on a computer keyboard:

F11	F12	Open and Closed Curly Braces
Caps Lock	Num Lock	Scroll Lock
Left Shift	Right Shift	Numeric Keys
Right Alt	Right Ctl	Home
Page Up	Page Down	End
Tab / Next		

The Caps key provides the function of the following computer keyboard keys:

Left Shift	Right Shift	Caps Lock

To get this ANSI+ result	**Press These Keys and Then**				**Press this key**
	Fn	**Shift**	**Ctrl**	**Alt**	
Answerback			x		Enter
Backspace					BkSp
Backtab				x	BkSp
Clear Field				x	F
Clear Screen				x	L
Compose				x	C
Delete	x				BkSp
Exit ANSI +				x	X
F1 through F10	x				1 – 0
F11 through F20				x	S then 1 through 0
F21 through F30				x	B then 1 through 0
F31 through F40				x	A then 1 through 0
Function Key Editor				x	K
Hot Key Help				x	H
INQ[6]				x	I
Page Down (in Help)				x	A then B
Page Up (in Help)				x	A then A
Send					Enter

---

[6] 900MHz Radio Equipped Only

To get this ANSI+ result	Press These Keys and Then				Press this key
	Fn	Shift	Ctrl	Alt	
Stored Forms				x	M
Swap Network Controller[7]					N/A
Tab				x	Enter
TN Escape			x		4
Unlock Keyboard				x	U
Cursor Down	x				B
Cursor Up	x				A
Cursor Left	x				E
Cursor Right	x				F
Window Down			x		B
Window Up			x		A
Window Lock/Home				x	A then H
Window Right			x		F
Window Left			x		A
Local Echo On				x	E
Local Echo Off				x	G

---

[7] 900MHz Radio Equipped Only

The MX3 Keypad

Please refer to the "MX3 Reference Guide", Appendix A "Key Maps", for detailed key-mapping information.

This section reflects the ANSI Plus specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an MX3.

When using a sequence of keys that includes the 2nd key, press the 2nd key first then the rest of the key sequence. Set the On/Off condition of NumLock before pressing a key sequence. There is no visual indication of the condition of NumLock.

Note: When the MX3 boots, the default condition of NumLock is On. NumLock can be set using the MX3 BIOS Setup.

Note: When the MX3 boots, the default condition of Caps (or CapsLock) is Off. The Caps (or CapsLock) condition can be set using the MX3 BIOS Setup or toggled with a 2nd-F1 key sequence. CAPS is illuminated when CapsLock is On.

To get this ANSI+ result	**Press These Keys and Then**				**Press this key**
	2nd	**Shift**	**Ctrl**	**Alt**	
Answerback			x		Enter
Clear Field				x	F
Clear Screen				x	L
Compose				x	C
DEL	x				. (DOT) or Ctrl-8
Down Arrow					Down Arrow
Exit ANSI +				x	X
F1 through F10					F1 through F10
Alt+F1 through F10				x	F1 through F10
Shift+F1 through F10		x			F1 through F10
Ctrl+F1 through F10			x		F1 through F10
Function Key Editor				x	K
Help				x	H
INQ				x	I
Left Arrow					Left Arrow
Numeric Enter	x				Enter
Page Down (in Help)	x				Down Arrow
Page Up (in Help)	x				Up Arrow
Right Arrow					Right Arrow
Send					Enter
Stored Forms				x	M
TN Escape	x		x		2
Unlock Keyboard				x	U
Up Arrow					Up Arrow

To get this ANSI+ result	Press These Keys and Then				Press this key
	2nd	Shift	Ctrl	Alt	
Window Down			x		Down Arrow (N/A on 25 rows)
Window Left					Left Arrow (N/A on 80 column)
Window Lock/Home		x			Up Arrow
Window Right					Right Arrow (N/A on 80 column)
Window Up			x		Up Arrow (N/A on 25 rows)

The VX1, VX2 and VX4 Keyboards

Please refer to the "VX1 Reference Guide", "VX2 Reference Guide" and "VX4 Reference Guide".

This section reflects the TE specific keys and keypress sequences that can be made when the ANSI Plus terminal emulator is running on an LXE VX1, or for the VX2, the LXE VMT keyboard.

When using a sequence of keys that includes the 2nd key, press the 2nd key first then the rest of the key sequence. For the VX computers, NumLock is always On.

When the VX computers bootup, the default condition of Caps (or Capslock) is Off. The Caps (or Capslock) condition can be set using the CMOS Setup or toggled with a 2nd + F1 key sequence. The CAPS LED is illuminated when Caps is On.

To get this ANSI+ result	Press These Keys and Then				Press this key
	2nd	Shift	Ctrl	Alt	
Answerback			x		Enter
BackTab (CBT)	x				Tab
Clear Field				x	F
Clear Screen				x	L
Compose				x	C
DEL	x				. (DOT) *or see below*
DEL			x		8
Down Arrow					Down Arrow
Exit ANSI +				x	X
F1 through F10					F1 through F10
Alt+F1 through F10				x	F1 through F10
Shift+F1 through F10		x			F1 through F10
Ctrl+F1 through F10			x		F1 through F10
Function Key Editor				x	K
Help				x	H
INQ				x	I
Left Arrow					Left Arrow
Numeric Enter	x				Enter
Page Down (in Help)	x				Down Arrow
Page Up (in Help)	x				Up Arrow
Right Arrow					Right Arrow
Send					Enter
Stored Forms				x	M
TN Escape	x		x		2
Unlock Keyboard				x	U
Up Arrow					Up Arrow

To get this ANSI+ result	Press These Keys and Then				Press this key
	2nd	Shift	Ctrl	Alt	
Window Down[8]			x		Down Arrow
Window Left			x		Left Arrow (N/A on 80 column)
Window Lock/Home		x			Up Arrow
Window Right			x		Right Arrow (N/A on 80 column)
Window Up			x		Up Arrow

---

[8] Window movement commands are not used with the VX2 as it has a 640x480 pixel wide screen.

LXE®

An EMS Technologies Company

Index

An EMS Technologies Company