# Transport API Java Edition 3.8.0.L1

## ANSI LIBRARY REFERENCE MANUAL

Document Version: 3.8.0
Date of issue: April 2024
Document ID: ETAJ380REANS.240

REFINITIV™

# Contents

# 1 Introduction

## 1.1 Overview

The AnsiPage API offers an easy to use programming interface that has the ability to interpret data received from ANSI page sources, as well as the ability to generate page data suitable for publishing.

What is page data? Page data is market data that is organized by row and column. In fact, you can think of page data as a two-dimensional array of data. Figure 1 below shows a sample page based on TDPF page 19000. The page illustrated is 25 rows by 80 columns, a typical page size. If appropriate for your Page service, this API will allow you to define pages using arbitrary row and column sizes.

Cursor-positioning controls are also supported so that updates may be applied to a page without the need for retransmitting the entire page. Pages and updates are assembled from characters plus escape sequences and control characters in a format that conforms to the ISO 2022 or ANSI X3.64 standard. The AnsiPage API interprets the ANSI standard described in the DEC VT200 Reference Manual with additions for special character sets and attributes. Refer to Appendix A of this document for more detail on page data formatting.

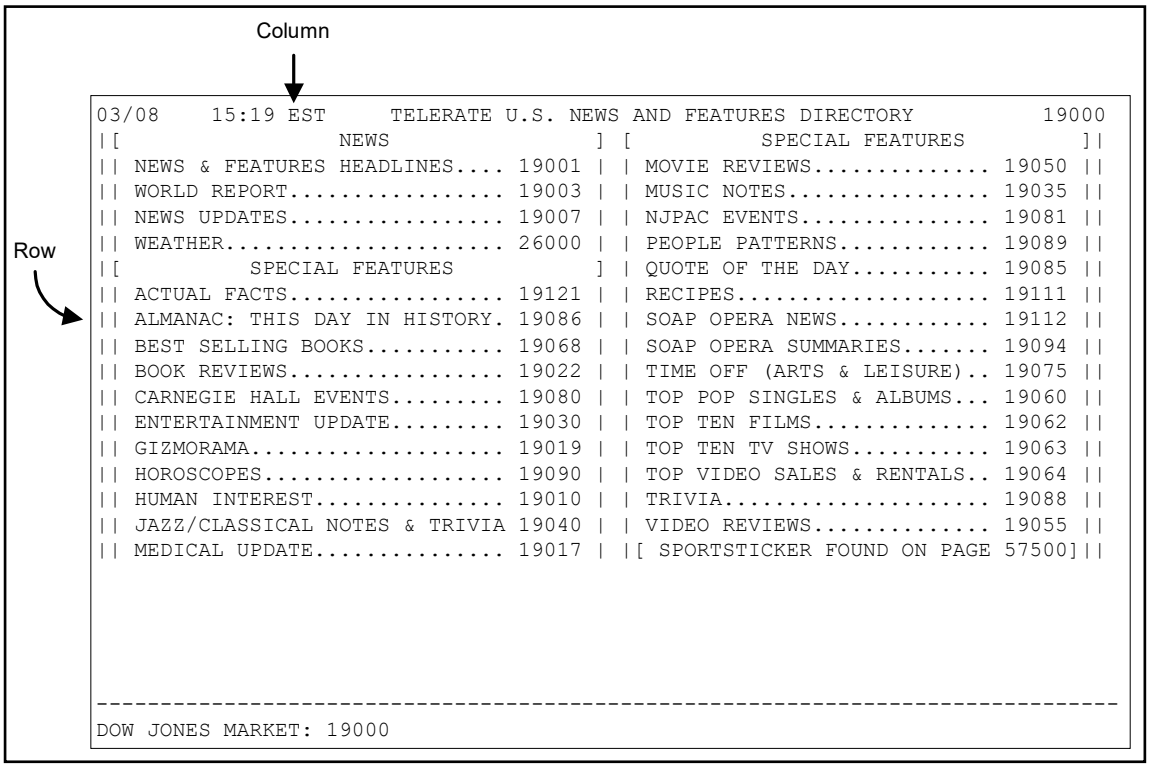


**Figure 1. Sample Page**

## 1.2 Audience

This guide is intended for software programmers who wish to incorporate the AnsiPage API into the development of their applications.

## 1.3 Organization of Manual

This manual is organized into the following sections:

| CHAPTER | CONTENTS |
|---|---|
| Chapter 1 | Product description, manual description and layout, and definitions of terms relevant to the AnsiPage API. |
| Chapter 2 | Describes basic concepts and product architecture. |
| Chapter 3 | Includes a full list of supported control functions and escape sequences. |
| Appendix A | Discusses things to keep in mind while programming with the AnsiPage API. |

**Table 1: Manual Overview**

## 1.4 Additional References

- *ANSI X3.64 Protocol Specification*, New York, N.Y. 1979
- *Transport API Java Edition Developers Guide*
- *DEC VT220 Programmers Reference Manual*, Digital Equipment Corporation, 1984

## 1.5 Conventions

### 1.5.1 Typographic Conventions

- Directories, filenames, parameters, and list entries appear in bold, e.g.: **filename**
- When in text, keywords, function names, and method names appear in **`black, Courier New`** font.
- Method names omit their arguments unless the arguments are needed for clarity.
- Variables and document titles appear in italics, e.g.: *Transport API Java Edition Developers Guide*
- Code samples appear in a Courier New font with a grey background, e.g.:

```
typedef struct
{
    unsigned char  flags;
    unsigned char   code;
} SSL_STATUS_CODE;
```

### 1.5.2 Programming

Transport API Java Edition standard conventions were followed.

### 1.5.3 Diagrams

All class diagrams follow the Unified Modeling Language (UML) standard.

## 1.6 Glossary

Below is a list of terms and definitions which are used frequently throughout this document.

| TERM | DEFINITION |
|---|---|
| ANSI<br>(American National Standards Institute) | American National Standards Institute – This agency develops standards widely used as guidelines by US firms. Data processing standards developed by ANSI range from the definition of ASCII to the determination of overall system performance. |
| ANSI Sequence | A string of characters that invokes a mode or status change in the display system. ANSI sequences start with an ESC character (033 octal) and then follow one of several patterns. The receiving device must recognize these patterns and act upon the supported sequences while ignoring unsupported sequences. |
| API | Application Programming Interface |
| ASCII | American Standard Code for Information Interchange |
| Fading | A feature that many users and sources require, where an area of updated text must be highlighted for a short time to draw attention to it. After the end of the fade period, the text returns to its original color. Applications implement fading as follows: when a character is changed, it is initially displayed with the “fading” set of styles. After a fixed period of time, it is re-displayed with the default styles. |
| Image | An image is the complete data representation of an item. |
| ISO | International Organization for Standardization |
| Item | An item is a named data entity. There are two types of items: pages and records. They are identified by the name of the source that supplies them, and the name of the individual item within that source. |
| Page | A page is a type of item formatted for distribution to display systems. The data includes text and control information in accordance with ANSI X3.64 and X3.41 (plus various extensions and private sequences). |
| Source | An application or server capable of supplying or transmitting information. |
| TDPF | Telerate Digital Page Feeder |
| UML | Unified Modeling Language |
| Update | An update is a modification to the contents of an item. |

**Table 2: Glossary of Terms**

# 2 Architecture

## 2.1 Feature Overview

This chapter discusses important AnsiPage API features that are implemented in this package, namely:

- AnsiPage API Interfaces
- Error Handling
- Threading Support
- Scrolling Support
- Unsupported ANSI Features

One purpose of this chapter is to introduce some concepts that are helpful in understanding what the AnsiPage API consists of as well as how to use it.

## 2.2 AnsiPage API Interfaces

The AnsiPage API is composed of three interfaces: **PageCell**, **PageUpdate**, and **Page**. Through the use of these interfaces the ability for your application to generate and interpret updates encoded as ANSI character sequences can be achieved. Refer to the online reference for the usage of each of these interfaces in detail.

## 2.3 Error Handling

### 2.3.1 Return Values

The AnsiPage API makes use of return values for an application to determine whether or not function calls have succeeded or not. Details on how each method reacts to errors will be included in the corresponding interface descriptions in the following chapters.

### 2.3.2 Exceptions

Because the AnsiPage API makes use of return values, it is not necessary for applications that use this API to incorporate the use of exceptions and the try/catch paradigm.

## 2.4 Threading Support

The AnsiPage API is thread-safe at the class level. This implies that there may be no more than a single thread of execution running on a given page at any time. Having multiple threads simultaneously executing on the same page is not supported and will likely yield unpredictable results.

## 2.5 Scrolling Support

The AnsiPage API only supports vertical scrolling of entire lines (up or down). There are provisions to avoid more than one set of entries in the update list if an update causes multiple scroll operations.

There is also support for the DEC escape sequences to restrict the scrolling region.

## 2.6 Unsupported ANSI Features

The AnsiPage API does not support Double Height and Double Width character formats.

# 3 AnsiPage Data Formats

## 3.1 Overview

Each position on a page, in addition to the character displayed, may have the following values associated with it:

- Character-set indicators
- Attributes (dim, bold, blink, underscore, reverse-video)
- Foreground and background color attributes

A complete page image consists of a full page of data and control codes. Updates can be broadcast as cursor controls followed by data or attributes. Pages must support cursor-positioning controls to make efficient use of screen-display capabilities, such as when updates to specific parts of a page are being transmitted.

The *Reuters Multilingual Text Encoding Standard* describes the encoding of graphical character sets and control functions. It gives standards for selecting character sets and switching between character sets in use. The Reuters supported character sets and control functions are defined in the text encoding standard as well. For example, the Reuters Basic Character Set is essentially ASCII. ASCII is a national variant of the ISO 646 standard for character sets. The UK national variant of ISO 646 (the main difference being the national version of the 'pound' sign) is also supported.

A subset of the Reuters standard as defined in this appendix is supported. There are cases where inheritance of private character sets clashes with the international standard for control sequences.

The private escape sequence **SFR** is used to control fading. Fading is a feature of display systems that allows the presentation software to enhance the display of recently modified data for a specified period of time. This is most typically used for real-time updates.

A full list of control functions and escape sequences supported is given in this appendix. Supported commands consist of ANSI sequences for information exchange, DEC private sequences, and Rich private sequences. The definitions that follow are used in the transmission of information in the Data portion of the **SSL_ET_ITEM_IMAGE** and **SSL_ET_ITEM_UPDATE** events. All references to these command sequences refer explicitly to the use of 7-bit character encoding.

## 3.2 Introducer Sequence

The following control sequence introducer is supported.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| CSI | ESC [ | 1B 5B | Control sequence introducer. The optional code sets the hex values for the control sequence introducer. |

**Table 3: Introducer Sequence**

## 3.3 Cursor Control Sequences

The following cursor control sequences are supported.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| BS | ^h | 0/8 | The cursor is moved one space to the left, but not past the beginning of the line. |
| HT | ^i | 0/9 | The cursor is moved to the next tab stop (default is set to 8-column tabs) or to the right margin if no more tabs exist on the line. |
| LF | ^j | 0/10 | The cursor is moved down one position, scrolling if necessary. |
| CR | ^m | 0/13 | The cursor is moved to the left margin of current line. |
| DECSC | ESC 7 | | This command is a DEC private request used to store the current cursor position and attribute for later use. |
| DECRC | ESC 8 | | This command is a DEC private request used to restore cursor position and attribute from the values saved by DECSC request. |
| IND | ESC D | | The cursor is moved to the next lower line (without changing column position), scrolling if necessary. |
| NEL | ESC E | | The cursor is moved to the first position of the next lower line, scrolling if necessary. |
| RI | ESC M | | The cursor is moved to the preceding line (without changing column position), scrolling down if necessary. |
| HVP | CSI Pl ; Pc f | | The cursor is moved to the screen position on line Pl and column Pc. Lines and columns are numbered starting at 1. A parameter value of **0** or null is interpreted as a value of **1**. An HVP command with no parameters has the same effect as a home command. Parameter values are tested to restrict the cursor location to the current screen. |
| CUU | CSI Pn A | | The cursor is moved upwards Pn positions, stopping at the top line. The default is set to one position and moves one position if **Pn = 0**. |
| CUD | CSI Pn B | | The cursor is moved down Pn positions, stopping at the last line. The default is set to one position and also moves one position if **Pn = 0**. |
| CUF | CSI Pn C | | The cursor is moved forward Pn positions, stopping at right margin. The default is set to one position and also moves one position if **Pn = 0**. |
| CUB | CSI Pn D | | The cursor is moved backward Pn positions, stopping at left margin. The default is set to one position and also moves one position if **Pn = 0**. |
| CUP | CSI Pl ; Pc H | | The cursor is moved to position Pl,Pc. This is the same as HVP control sequence. |

**Table 4: Cursor Control Sequences**

## 3.4 Scrolling Sequences

The following scrolling sequences are supported:

- Margins
- Movement

### 3.4.1 Margins

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| DECSTBM | CSI Pt ; Pb r | | This command selects the top and bottom margins defining the scrolling region. Pt is the top line and Pb is the bottom line. Lines are numbered starting at 1. The default state is (1;24). |

**Table 5: Scrolling Sequences: Margins**

### 3.4.2 Movement

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| SU | CSI Pn S | | Scroll up Pn lines. |
| SD | CSI Pn T | | Scroll down Pn lines. |

**Table 6: Scrolling Sequences: Movement**

## 3.5 Mode Control Sequences

The following control sequences control the type or mode of the ANSI Page Data Format.

- Mode Control Sequence Types
- Mode Control Sequences

### 3.5.1 Mode Control Sequence Types

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| SM | CSI Ps h<br>CSI ? Ps h | | Commands of this type set (enable) modes as requested by the parameter Ps. The ? character after the CSI defines the sequence to be a DEC private sequence |
| RM | CSI Ps l<br>CSI ? Ps l | | Commands of this type reset (disable) modes as requested by the parameter Ps. The ? character after the CSI defines the sequence to be a DEC private sequence.<br>**NOTE:** The last character in this sequence is a lower case L. |

**Table 7: Mode Control Sequence Types**

### 3.5.2 Mode Control Sequences

The following are supported mode control sequences.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| DECCOLM | CSI ? 3 h<br>CSI ? 3 l | | These DEC private sequences control the number of columns displayed on the terminal. When this mode is set, the terminal displays 132 columns. When reset, only 80 columns are displayed. |
| DECSCLM | CSI ? 4 h<br>CSI ? 4 l | | These DEC private sequences control the scrolling mechanism used by the terminal. When this mode is set, smooth scrolling is used. When reset, jump scrolling is used. |
| DECAWM | CSI ? 7 h<br>CSI ? 7 l | | These DEC private sequences control the auto wrap feature. When invoked, auto wrap performs a new line upon an attempt to write past the right margin of the display area. When disabled, an attempt to write past the right margin causes the right most character of the display to be replaced with the current character. |
| DECTCEM | CSI ? 25 h<br>CSI ? 25 l | | These DEC private sequences control the display status of the terminal's cursor. When this mode is set, the cursor is displayed. When reset, the cursor becomes invisible. |

**Table 8: Mode Control Sequences**

## 3.6 Display Editing Control Sequences

The following sequences are used to alter or edit the visual display.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| ED | CSI Ps J | | This command erases some or all of the image according to value of Ps:<br>• **Ps = 0**, from cursor to end inclusive<br>• **Ps = 1**, from start to cursor inclusive<br>• **Ps = 2**, all of display<br>Defaults to **Ps = 0**. |
| EL | CSI Ps K | | This command erases some or all of the current line according to value of Ps:<br>• **Ps = 0**, from cursor to end inclusive<br>• **Ps = 1**, from start to cursor inclusive<br>• **Ps = 2**, all of line<br>Defaults to **Ps = 0**. |
| IL | CSI Pn L | | This command inserts Pn erased lines starting at the active line and working downward. The remainder of the page is moved downward Pn lines, and the last Pn lines are discarded. The default is set to one line. One line is also inserted if **Pn = 0**. |
| DL | CSI Pn M | | This command deletes Pn lines starting at the active line and working downward. The remainder of the page is shifted to begin at the active line. The default is set to one line. One line is also deleted if **Pn = 0**. |
| DCH | CSI Pn P | | This command deletes Pn characters starting at the active position and working to the right. The remainder of the line is moved to the left of the active position. The default is set to one character. One character is also deleted if **Pn = 0**. The deletion of characters is restricted to the active line exclusively. |
| ICH | CSI Pn @ | | This command inserts Pn erased positions starting at the cursor and working to the right. The remainder of the line is moved Pn positions to the right and the last Pn positions are discarded. The default is set to one erased position. One erased position is also inserted if **Pn = 0**. The insertion of the erased characters is limited to the active line. |

**Table 9: Display Editing Control Sequences**

## 3.7 Graphic Selection Sequences

The following graphic control sequences are supported. Please refer to the *Reuters Multilingual Text Encoding Standard* for the Reuters Basic Character Sets 1 and 2.

A character set is mapped to a "working character set" in the range G0, G1, G2, or G3.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| SO | ^n | 0/14 | This command shifts out the current working graphics set and invokes the G1 graphics set. |
| SI | ^o | 0/15 | This command shifts in the current G0 graphics set as the working graphics set. |
| | ESC ( Pf | | This command designates the G0 graphics set as the primary graphics set. The physical character set associated with the selection character **Pf** is invoked whenever G0 is invoked.<br>The supported parameter values of **Pf** corresponding to character set indicators are as follows:<br>• **B**: ASCII (U.S. national variant of ISO 646)<br>• **A**: U.K. national variant of ISO 646<br>• **0**: VT100 special graphics (e.g., diamond)<br>• **:** (colon): Chapdelaine special set (e.g., fractions, superscripts, arrows)<br>• **;** (semi-colon): RMJ special set (e.g., checkmark)<br>• **<**: Garban character set<br>• **m**: Mabon special character set<br>• **>**: Mosaic graphics set (e.g., solid block)<br>• **?**: FBI special character set (e.g., down arrow, subscript 6)<br>• **f**: FBI special graphics set (e.g., yen, subscript 2, 4 and 8)<br>• **g**: General graphics set<br>• **s**: SOP Telerate ASCII set<br>• **v**: Viewdata mosaic set |
| | ESC ) Pf | | This command designates the G1 graphics set as the primary graphics set. The parameter values of **Pf** are identical to the G0 case. |

**Table 10: Graphic Selection Sequences**

## 3.8 Character Attribute Sequences

The following character attribute sequences are supported.

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| SGR | CSI Ps m | | Set Graphic Rendition (SGR) sets display attributes. Defaults to **Ps = 0**.<br>**Ps** supports the following parameter values:<br>• 0: All attributes off<br>• 1: High-intensity (bold)<br>• 2: Half-intensity (dim)<br>• 4: Underscore on<br>• 5: Blink on<br>• 7: Reverse video on<br>• 8: Concealed on<br>• 22: Bold off<br>• 24: Underscore off<br>• 25: Blinking off<br>• 27: Reverse video off<br>• 30: Black foreground<br>• 31: Red foreground<br>• 32: Green foreground<br>• 33: Yellow foreground<br>• 34: Blue foreground<br>• 35: Magenta foreground<br>• 36: Cyan foreground<br>• 37: White foreground<br>• 40: Black background<br>• 41: Red background<br>• 42: Green background<br>• 43: Yellow background<br>• 44: Blue background<br>• 45: Magenta background<br>• 46: Cyan background<br>• 47: White background<br>**NOTE:** More than one parameter, separated by semicolons, may appear in this sequence. |

**Table 11: Character Attribute Sequences**

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| SFR | CSI > Ps m | | The private sequence Set Fading Rendition (SFR) specifies that a single attribute or several attributes are invoked in addition to the base attributes defined by the standard ANSI video attribute sequence Set Graphic Rendition (SGR). The display of recently updated fields may thus be enhanced in order to bring the user's attention to the modified area of the display.<br>The added attributes are cleared after a short period of time (usually 5 seconds). The length of application of these additional attributes is controlled by the display device. Specifically, fading allows the addition of any combination of the monochrome attributes normally supported by the display system. For color displays, fading also allows the modification of both the foreground and the background color.<br>Attributes are only added during the fade enhanced period. For example, a base attribute of UNDERSCORE may be enhanced by a fading BLINK. The result would be to display a blinking underscored character for the enhanced period and replace it with a steady underscored character when the enhanced period ends.<br>The foreground and the background colors may be individually replaced for the fade period. If a color is not modified by SFR, then the base color or colors is in effect. If either is replaced, even in the absence of a fading monochrome attribute, fading is invoked.<br>Thus, if base colors are set to blue foreground on a red background and a sequence is received to fade a white foreground, the display is white on red during the enhanced period. At the end of the enhanced period, the display returns to blue on red.<br>The supported parameter values of **Ps** are the same those listed in the preceding table for SGR.<br>The default is **Ps = 0**.<br>**NOTE:** More than one parameter may appear in this sequence, in which case they are separated by semicolons. |

**Table 11: Character Attribute Sequences (Continued)**

## 3.9 Character and Line Sizing

The following escape sequences to alter character cell dimensions are supported:

- DEC Private Sequences
- Rich Private Sequences

### 3.9.1 DEC Private Sequences

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| DECDHL | ESC # 3 | | This command changes the current line to double-height / double-width top half. |
| | ESC # 4 | | This command changes the current line to double-height / double-width bottom half. |
| DECSWL | ESC # 5 | | This command changes the current line to single-height / single-width. |
| DECDWL | ESC # 6 | | This command changes the current line to single-height / double-width. |

**Table 12: DEC Private Sequences**

### 3.9.2 Rich Private Sequences

The following sequences are NOT supported by DEC compatible terminals. These sequences have historically been called Rich Private Sequences. While they might still be in use, they are not consistent with the newer Reuters text encoding standards.

| SEQUENCE | DESCRIPTION |
|---|---|
| CSI > 1 Z | This Rich private sequence invokes double-height/single-width top half mode for all succeeding characters until modified by another character size command. |
| CSI > 2 Z | This Rich private sequence invokes double-height/single-width bottom half mode for all succeeding characters until modified by another character size command. |
| CSI > 3 Z | This Rich private sequence invokes double-height/double-width top half mode for all succeeding characters until modified by another character size command. |
| CSI > 4 Z | This Rich private sequence invokes double-height/double-width bottom half mode for all succeeding characters until modified by another character size command. |
| CSI > 5 Z | This Rich private sequence invokes single-height/single-width mode for all succeeding characters. |
| CSI > 6 Z | This Rich private sequence invokes single-height/double-width mode for all succeeding characters. |

**Table 13: Rich Private Sequences**

## 3.10 Miscellaneous Sequences

The following sections define all other supported sequences not grouped in the previous sections.

- Control Characters
- Controls

### 3.10.1 Control Characters

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| BELL | ^g | 0/7 | Generates the bell tone on the terminal keyboard if the bell is enabled. |
| DEL | | 7/15 | Delete characters are ignored when received. |

**Table 14: Control Characters**

### 3.10.2 Controls

| MNEMONIC | ANSI SEQUENCE(S) | OPTIONAL CODE(S) | DESCRIPTION |
|---|---|---|---|
| RIS | ESC c | | This command resets a device to its initial state, i.e., the state it has after it has been switched on. The screen (25 lines) and fading status is cleared, and the control structure is reset to its initial state (e.g., resets attributes, modes, etc). The scrolling region is reset to the default rows (1;24). |

**Table 15: Controls**

# Appendix A Programming Hints and Tips

Page objects contain 25 rows and 80 columns by default. The coordinates for each individual cell are 1-based. Therefore, the first cell would be at position (1,1). As stated earlier, it is often helpful to think of page data as being a two-dimensional array of data.

Document ID: ETAJ380REANS.240
Date of issue: April 2024