TeemTalk for Windows

API Programmer's Guide

USA
Neoware, Inc.
3200 Horizon Drive
King of Prussia, PA 19406
Tel: +1-610-277-8300
Fax: +1-610-771-4200
Email: info@neoware.com

UK
Neoware UK Ltd
The Stables, Cosgrove
Milton Keynes MK19 7JJ
Tel: +44 (0) 1908 267111
Fax: +44 (0) 1908 267112
Email: teem-info@neoware.com

TeemTalk
Software Support

Telephone: +1-610-277-8300
Web: http://www.neoware.com/support/

Software Version 5.0.1
October 2004

Part number: TT6PM5.0.1

Neoware UK Ltd,
The Stables, Cosgrove,
Milton Keynes, MK19 7JJ
Tel: +44 (0) 1908 267111
Fax: +44 (0) 1908 267112

The material in this manual is for information purposes only and is subject to change without notice. Neoware UK Ltd accepts no responsibility for any errors contained herein.

Trademarks

TeemTalk is a registered trademark of Neoware UK Ltd.

ADDS Viewpoint A2 is a trademark of Applied Digital Data Systems Inc.

AIX is a registered trademark of International Business Machines Corporation.

D100, D200 and D410 are trademarks of Data General.

Dataspeed is a registered trademark of AT&T.

DEC, VT52, VT100, VT131, VT220, VT300, VT320 and VT340 are registered trademarks of Digital Equipment Corporation.

Hazeltine is a trademark of Esprit Systems, Inc.

HP700/92, HP700/94, HP700/96, HP2392A and HP2622A are trademarks of Hewlett Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

Microsoft and MS-DOS are registered trademarks, and Windows and Excel are trademarks of Microsoft Corporation.

Prime is a registered trademark and PT250 is a trademark of Prime Computer, Inc.

Tandem, NonStop and LXN are trademarks of Tandem Computers Inc.

TeleVideo is a registered trademark, and TeleVideo 910, 910+, 925, 950 and 955 are trademarks of TeleVideo Systems, Inc.

WYSE is a registered trademark, and WY-50, WY-50+ and WY-60 are trademarks of Wyse Technology Inc.

All other product names are trademarks of their respective manufacturers.

Contents

Notes

1

Introduction

This chapter introduces the Application Programming Interface (API) and describes the scope of this manual.

What Is API?

The Application Programming Interface (API) enables the TeemTalk range of terminal emulators to be integrated into user written applications using a comprehensive set of commands. The API is compatible with the IBM HLLAPI interface, providing extensions to cater for all of the emulations supported by TeemTalk.

The API interface consists of a C library which the user links with an application program. The application program can then connect to up to 26 instances of any suitable TeemTalk emulator by invoking the API library. Each of these emulators executes on the current host computer and is free to connect to a target computer in exactly the same manner as a standard emulator.

The TeemTalk emulators for the API interface are similar to the standard TeemTalk emulators except that they have an additional communication channel to the API interface library. This interface channel is completely transparent to the user and the emulator can either be invoked from the command line as usual (with no connection to the API interface library), or from the TeemTalk API interface library (with the corresponding communication channel enabled). The user is free to specify any options that would normally be included on the command line for loading the TeemTalk emulator.

About This Programmer's Manual

This manual consists of the following chapters and appendices:

Chapter 1: ***Introduction***
Introduces the Application Program Interface.

Chapter 2: ***Overview***
Provides an overview of the API interface and describes how to create and use an API program.

Chapter 3: ***API Functions***
Describes all the supported API functions in detail.

Appendix A: ***Field Attribute Representation***
Describes how field attributes are represented in the data string returned by a query field attribute function.

Appendix B: ***Character Sets***
Shows some of the character set code tables supported by TeemTalk.

Appendix C: ***Host Command Summary***
Lists the host commands that are supported in each terminal emulation mode.

2

Overview

This chapter provides an overview of the API interface and describes how to create and use an API program.

The API Interface

The API can be used to simplify the creation of a local application which requires interaction with a Host system as a terminal. Typical uses might be to:

- Automate repetitive tasks, like automatic logon.
- Provide a "user-friendly" interface to a complex host application.
- Provide a single user interface to multiple host applications, on different host systems.
- Integrate existing host data into a new local application.

The API provides support for Citrix MetaFrame Password Manager.

The TeemTalk API is available via the **PCSHLL32.DLL** file which is compatible with the DLL provided with IBM's Personal Communications/3270 product. This DLL must be available in a directory in the standard Windows directory search path when TeemTalk and/or your application is first started. During installation the file **PCSHLL32.DLL** is copied into the same area as other TeemTalk DLLs.

TeemTalk will automatically allocate the next available short name to the session before starting up the API program. If you need to manually specify the short name, include the **-hllapi** command on the command line, followed by the short name in the range **A**-**Z**. For example:

-hllapiA

This may be necessary for some API programs to run.

Creating Your Own API Program

In order to use the API in your own program, you must use a function call to the DLL for each command you want to issue. This takes the following form:

```
__declspec(dllexport)
WORD WINAPI hllapi(
                    LPWORD    lpwFuncNum,
                    LPBYTE    lpDataString,
                    LPWORD    lpwDataStringLength,
                    LPWORD    lpwRetCode
                   );
```

where:

lpwFuncNum is a pointer to a WORD (16-bit unsigned) value containing the 'Function Number' parameter.

lpDataString is a pointer to the memory ('Data String' parameter) used to pass data to and from each HLLAPI function.

lpwStringLength is a pointer to a WORD (16-bit unsigned) value containing the 'Data String Length' parameter.

lpwRetCode is a pointer to to a WORD (16-bit unsigned) value containing the 'Presentation Space Position' parameter on entry to the function, and the 'Return Code' parameter on exit.

returns:

Return Code for the specific function, also in **lpwRetCode.**

The API is driven by function numbers, which identify the individual commands supported. The **hllapi()** function is prototyped in the **HAPIFUNC.H** file, which also contains definitions for all the Functions, Return Codes, etc. used by the API.

When you 'link' your application, you may include the **PCSHLL32.LIB** library file to IMPORT the **hllapi()** function needed from the DLL, or else specify it explicitly in the ***.DEF** file in the IMPORT section as:

PCSHLL32.hllapi

The Presentation Space

A presentation space (PS) is the portion of computer memory that is used to display information in the terminal emulation window. Each instance of TeemTalk (i.e. each session) has its own presentation space. The API supports up to 26 presentation spaces and interacts with these one at a time.

Identification

Each presentation space is identified by a unique name (a PSID) which is a single character in the range **A** through **Z**. Some functions require the PSID to be specified in a preceding call to **Connect Presentation Space** (1), while others include the PSID as part of their calling data string.

Making A Connection

The **Connect Presentation Space** (1) function is used to make an initial connection between your API program and an instance of TeemTalk, using a PSID to identify it. Future re-connects to an instance of the terminal emulator just require the PSID to be specified. The currently connected presentation space can be disconnected by the **Disconnect Presentation Space** (2) function, or you can disconnect all the presentation spaces launched by the API application by issuing **Reset System** (21).

Addressing

Some functions require a specific location within the presentation space to be specified in their calling data string. Presentation space addressing relates to the screen size of the current session. A PS position is specified as the offset into the presentation space working left to right, line by line from top to bottom, where position 1 is the top left corner of the screen display (row 1 and column 1) and the highest position number is the bottom right corner (e.g. 1920 for a display size of 24 rows by 80 columns). The size of the currently connected presentation space can be determined by a **Query Session Status** (22) call and examining bytes 12 to 15 of the returned data string.

ASCII Mnemonics

ASCII mnemonics are used by the **Send Keystring** (3) and **Get Key** (51) functions to represent keyboard functions or codes that may not be represented by ASCII values or an available key. For example, the keyboard for one session might not be capable of generating key codes that are required by another session, so ASCII mnemonics representing these codes would be included in the data string sent by the **Send Keystring** (3) function.

All defined keys are represented by either a 1-byte ASCII value from the 256-character ASCII character set (e.g. **a** for the **a** key), or a 2-, 4-, or 6-byte ASCII mnemonic for function keys (e.g. **@r** for the **Ctrl** key, **@A@T** for **Print Screen**).

The escape character is represented by the ASCII character @ (at) by default. You can change this to any other displayable ASCII character by using the **ESC=** option of the **Set Session Parameters** (9) function. The tables of mnemonics shown below use @ to represent the escape character.

Mnemonics for unshifted keys and the shift keys **Upper Shift**, **Alt** and **Ctrl** consist of the escape character followed by an abbreviation. Mnemonics for shifted keys consist of the mnemonic for the shift indicator followed by the mnemonic for the unshifted key. Shift indicators are represented by the following 2-byte ASCII mnemonics:

Upper Shift:	@S
Alt:	@A
Ctrl:	@r

These shift indicator mnemonics are always accompanied by a non-shift-indicator character or mnemonic, they are never sent or received by an application on their own.

The following tables show the mnemonics for keys defined with functions. Please note that the mnemonics are case sensitive. For example, **@N** would refer to the **New Line** key function, whereas **@n** would refer to the **PF23** key function. AID key mnemonics are indicated by an asterisk (*).

Standard IBM Keyboard Mapping

The following ASCII mnemonics are valid by default or when the **KEYSIBM** option is specified using the **Set Session Parameters** (9) function.

@B	Backtab	@L	Cursor Left	@Y	Caps Lock
@C	Clear *	@N	New Line	@Z	Cursor Right
@D	Delete	@P	Print	@s	Scroll Lock
@E	Enter *	@R	Reset	@t	Num Lock
@F	Erase EOF	@T	Tab	@@	@ (at) symbol
@I	Insert	@U	Cursor Up	@$	Alternate Cursor
@J	Jump	@V	Cursor Down	@<	Backspace Erase

*Note: The mnemonics for **Caps Lock**, **Scroll Lock** and **Num Lock** can be intercepted by the function **Get Key** (51), but they cannot be sent by **Send Keystring** (3).*

@0	Home	@7	PF7 *	@e	PF14 *	@l	PF21 *
@1	PF1 *	@8	PF8 *	@f	PF15 *	@m	PF22 *
@2	PF2 *	@9	PF9 *	@g	PF16 *	@n	PF23 *
@3	PF3 *	@a	PF10 *	@h	PF17 *	@o	PF24 *
@4	PF4 *	@b	PF11 *	@i	PF18 *	@x	PA1 *
@5	PF5 *	@c	PF12 *	@j	PF19 *	@y	PA2 *
@6	PF6 *	@d	PF13 *	@k	PF20 *	@z	PA3 *

@A@F	Erase Input	@A@b	Underscore
@A@H	System Request *	@A@y	Forward Word Tab
@A@J	Cursor Select *	@A@z	Backward Word Tab
@A@Q	Attention *	@S@x	Dup
@A@R	Device Cancel	@S@y	Field Mark
@A@T	Print Screen		

*Note: The mnemonic for the at symbol (@@) consists of the escape character followed by the literal @ symbol. If the escape character is represented by **%**, then the mnemonic would change to **%@**. The **ESC=** option of the **Set Session Parameters** (9) function is used to specify the character that represents escape.*

If you send either of the Print Screen mnemonics, place it at the end of the calling data string.

If you send the Device Cancel mnemonic, mnemonics are passed through with no error message. Local copy will not be affected.

Direct Keyboard Mapping

The following ASCII mnemonics are valid when the **KEYSKBD** option is specified using the **Set Session Parameters** (9) function.

@sp	Keypad 0	@@	@	@`	
@!	Keypad 1	@A	Alt	@a	F10
@"	Keypad 2	@B	Attn (122)	@b	F11
@#	Keypad 3	@C	Compose (LK250)	@c	F12
@$	Keypad 4	@D	Delete	@d	F13 (122)
@%	Keypad 5	@E	Enter	@e	F14 (122)
@&	Keypad 6	@F	Clear (122)	@f	F15 (122)
@'	Keypad 7	@G	Cursor Sel (122)	@g	F16 (122)
@(	Keypad 8	@H	Home	@h	F17 (122)
@)	Keypad 9	@I	Insert	@i	F18 (122)
@*	Keypad *	@J	ErInp (122)	@j	F19 (122)
@+	Keypad +	@K	ExSel (122)	@k	F20 (122)
@,	Keypad , (LK250/122)	@L	Cursor Left	@l	F21 (122)
@-	Keypad -	@M	Erase EOF (122)	@m	F22 (122)
@.	Keypad .	@N	Newline	@n	F23 (122)
@/	Keypad /	@O	Print (122)	@o	F24 (122)
@0		@P	Print Screen	@p	Pause
@1	F1	@Q	Play (122)	@q	Reset (122)
@2	F2	@R	Escape	@r	Control
@3	F3	@S	Shift	@s	Scroll Lock
@4	F4	@T	Tab	@t	Num Lock
@5	F5	@U	Cursor Up	@u	Enter (122)
@6	F6	@V	Cursor Down	@v	
@7	F7	@W	Zoom (122)	@w	
@8	F8	@X	Noname (122)	@x	Page Up
@9	F9	@Y	Caps Lock	@y	Page Down
@:	Hold (LK250)	@Z	Cursor Right	@z	End
@;	Print (LK250)	@[	PA1 (122)	@{	
@<	Backspace	@\	Backtab (122)	@|	
@=	Setup (LK250)	@]	PA2 (122)	@}	
@>	Datatalk (LK250)	@^	PA3 (122)	@~	
@?	Break (LK250)	@_	Home (122)		

Enhanced AT Keyboard Layout

These ASCII mnemonics are valid when the **KEYSKBD** option is specified using the **Set Session Parameters** (9) function. The characters must be preceded by the escape character (@ by default).

122-Key Keyboard Layout

These ASCII mnemonics are valid when the **KEYSKBD** option is specified using the **Set Session Parameters** (9) function. The characters must be preceded by the escape character (@ by default).

LK250 Keyboard Layout

These ASCII mnemonics are valid when the **KEYSKBD** option is specified using the **Set Session Parameters** (9) function. The characters must be preceded by the escape character (@ by default).

Notes

3

API Functions

This chapter describes all the API functions in detail.

Conventions

An API function is referred to by its function name which is followed by the function number in parentheses. Function descriptions are arranged alphabetically by name.

Each function description includes the following sections if applicable:

Prerequisite Calls

This section specifies the function(s) that must be called before the function described can take effect. If no prerequisite calls are required then *None* will be shown.

Supplied Parameters

This section lists and describes the parameters that must be defined for the function. If a parameter is not used then *NA* (not applicable) will be shown.

Returned Parameters

This section lists and explains the parameters that your program must receive following the function call.

Remarks

This section supplies technical information on the function and specifies any session options that may affect it.

Emulation Support Of Functions

The API functions described in this chapter can be used in all the terminal emulations supported by TeemTalk except MDIS Prism, with the following exceptions.

The following functions are only supported by the IBM emulations:

Get Key (51)
Send Keypress (103)

The functions listed below are not supported by VT52, ANSI-BBS, DG410, AT&T4410, Stratus V102, or the Viewdata modes:

Query Field Attribute (14)
Search Field (30)
Find Field Position (31)
Find Field Length (32)
Copy String To Field (33)
Copy Field To String (34)
Query Formatted Field Attribute (114)

These 'field' functions will only work in the emulations listed in the table below if the presentation space is formatted as specified:

Emulation	Considered Formatted When
VT100 **VT320 (7-bit)** **VT320 (8-bit)**	In Edit mode, plus ‘host characters protected’ and ‘NOT erase all’ set.
HP700/92	Format mode ON.
PT250	In Block mode, plus ‘logical attributes’ mode ON.
TA6530	In Block Protect mode, and a page is displayed.
IBM 3270	Presentation Space formatted.
IBM 5250	Connected.
IBM 3151	Presentation Space formatted.
Wyse Modes	In Block mode with Protect ON.

Function Descriptions

The API functions described on the following pages in alphabetical name order are listed below according to function number:

1	Connect Presentation Space
2	Disconnect Presentation Space
3	Send Keystring
4	Wait
5	Copy Presentation Space
6	Search Presentation Space
7	Query Cursor Location
8	Copy Presentation Space To String
9	Set Session Parameters
10	Query Sessions
11	Reserve
12	Release
13	Copy OIA
14	Query Field Attribute
15	Copy String To Presentation Space
18	Pause
20	Query System
21	Reset System
22	Query Session Status
23	Start Host Notification
24	Query Host Update
25	Stop Host Notification
30	Search Field
31	Find Field Position
32	Find Field Length
33	Copy String To Field
34	Copy Field To String
40	Set Cursor
50	Start Keystroke Intercept
51	Get Key
52	Post Intercept Status
53	Stop Keystroke Intercept
99	Convert Position / RowCol
103	Send Keypress
105	Copy Formatted Presentation Space
108	Copy Formatted Presentation Space To String
114	Query Formatted Field Attribute
150	Debug

CONNECT PRESENTATION SPACE (1)

This function establishes a connection between any API application program and the presentation space (PS), excluding it from use by any other API application program.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 1.

Data String: For the initial connection to a presentation space, this is a one character identifier (PSID) in the range **A** through **Z** (upper or lowercase).

For reconnecting to an existing presentation space, this is the one character identifier (PSID) which identifies the particular session as specified by the initial connection.

Data String Length: NA. (1 is implied.)

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful; the presentation space is connected and unlocked.
1	An invalid presentation space ID was specified.
4	Successful, but host input available.
5	Successful, but host PS locked.
9	A system error occurred.
11	Resource unavailable. The PSID is already being used by another API application.

CONVERT POSITION / ROWCOL (99)

This function converts the host presentation space positional value into the display row/column coordinates, or vice versa. Note that the cursor position remains unchanged.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 99.

Data String: The presentation space ID character plus **P** for convert position to row/column coordinates, or **R** for convert row/column coordinates to presentation space position.

Data String Length: Not applicable when **P** specified in the data string.
When **R** is specified, this is the row number between 1 and maximum.

PS Position: For **P**, this is the positional value between 1 and maximum.
For **R**, this is the column number between 1 and maximum.

Returned Parameters

Data String Length: For **P**, the row number between 1 and maximum.
For **R**, **0** if specified row number was invalid.

Return Code: The following codes are valid:

Code	Explanation
0	Specified position/row/column number invalid.
>0	This is the position or column.
9998	An invalid host presentation space ID was specified, or a system error occurred.
9999	Second character in the data string is neither **P** nor **R**.

Remarks

To determine how many rows and columns there are in the currently connected presentation space, use the **Query Session Status** (22) function.

COPY FIELD TO STRING (34)

This copies the contents of a specified field in the presentation space to a string. The function only applies to field formatted presentation spaces and can be used with both protected and unprotected fields. The location and length of a field can be found by using the **Find Field Position** (31) and **Find Field Length** (32) functions.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 34.

Data String: Predefined buffer to hold string (twice the length requested if the EAB option has been specified).

Data String Length: The number of characters to be copied.

PS Position: Any character position within the source field in the presentation space. (Note that the copy will always start at the beginning of the field.)

Returned Parameters

Data String: Characters from the source string in the presentation space. The function will copy up to the data string length specified or the end of the source field, whichever occurs first.

Data String Length: The number of characters actually copied.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Data string length 0 specified.
6	Incorrect field size specified. The returned data string may be truncated.
7	Invalid presentation space position.
9	A system error occurred.
24	The presentation space is unformatted.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Characters copied from the field are translated into 8-bit ASCII characters (ISO Latin-1) for the returned data string. This function is affected by the EAB/NOEAB and XLATE/NOXLATE session options. See **Set Session Parameters** (9).

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY FORMATTED PRESENTATION SPACE (105)

Copies the internal 4-byte representation of the currently connected formatted presentation space to a string.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 105.

Data String: Predefined buffer, at least four times the size of the presentation space.

Data String Length: Length of data string.

PS Position: NA.

Returned Parameters

Data String: Contents of the current presentation space.

Data String Length: Length of the data string returned.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
4	Successful, but host input available.
5	Successful, but keyboard locked.
9	A system error occurred.

Remarks

Returns 4-byte character/attributes as represented internally by the TeemTalk product emulator. The character/attribute value is emulation specific. Refer to the *Field Attribute Representation* appendix for details.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY FORMATTED PRESENTATION SPACE TO STRING (108)

Copies all or part of the internal 4-byte representation of the specified portion of the currently connected formatted presentation space to a string.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 108.

Data String: Predefined buffer to hold string.

Data String Length: Length of data string required.

PS Position: Position in presentation space to start copy.

Returned Parameters

Data String: Contents of the current presentation space.

Data String Length: Length of the data string returned.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Data string length 0 specified.
4	Successful, but host input available.
5	Successful, but keyboard locked.
7	Invalid presentation space position.
9	A system error occurred.

Remarks

Returns 4-byte character/attributes as represented internally by the TeemTalk product emulator. The character/attribute value is emulation specific. Refer to the *Field Attribute Representation* appendix for details.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY OIA (13)

This function returns a copy of the status line (Operator Information Area) contents for the currently connected presentation space.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 13.

Data String: Predefined buffer to hold string.

Data String Length: Length of the target data string (in bytes).
For the IBM 3270 and 5250 emulations this is 103 bytes.

PS Position: NA.

Returned Parameters

Data String: Status line (OIA) data string (see **Remarks** section).

Data String Length: Length of data string returned (in bytes).

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Error in specifying data string length. No OIA data returned.
4	OIA data returned, but presentation space is busy.
5	OIA data returned, but presentation space locked.
9	A system error occurred.

Remarks

The status line must be enabled (active) for this function to work. The following sections describe the data string returned for the IBM 3270 emulation, the IBM 5250 emulation, then for the other emulations.

IBM 3270 Emulation

In the IBM 3270 emulation, the requested Data String Length must be 103, and the returned Data String contains the following data:

Offset 0	**Emulation Mode** (3270 = **1**, hexadecimal character 31).
Offset 1-80	**OIA Image**.
Offset 81-102	**OIA Group Indicators**.

The OIA image is a copy of the information on the displayed status line, in IBM 3270 PC format, using the 3270 Host Presentation Space characters shown below.

	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	NUL	SP	0	&	à	ä	Á	Ä	a	q	A	Q			P	
x1	EM	=	1	–	è	ë	È	Ë	b	r	B	R	—	|	S	?
x2	FF	’	2	.	ì	ï	Ì	Ï	c	s	C	S		a	→	
x3	NL	”	3	,	ò	ö	Ò	Ö	d	t	D	T	_			
x4	STP	/	4	:	ù	ü	Ù	Ü	e	u	E	U				4
x5	CR	\	5	+	ã	â	Ã	Â	f	v	F	V		–		¯
x6		|	6	¬	õ	ê	Õ	Ê	g	w	G	W				—
x7		¦	7	¯	ÿ	î	Y	Î	h	x	H	X				
x8	>	?	8	°	à	ô	A	Ô	i	y	I	Y	←		µ	¿
x9	<	!	9		è	û	E	Û	j	z	J	Z			²	a
xA	[	$	ß	^	é	á	E	Á	k	æ	K	Æ			³	
xB	]	¢	§	~	ì	é	I	É	l	ø	L	Ø				
xC	)	£	#	¨	ò	í	O	Í	m	a	M	A	A			a
xD	(	¥	@	`	ù	ó	U	Ó	n	ç	N	Ç	B			
xE	}	Pts	%	´	ü	ú	Y	Ú	o	;	O	;				¡
xF	{	☼	_	ↄ	ç	ñ	C	Ñ	p	*	P	*				Not Supported

The OIA Group Indicators comprise bits representing the state of the connected session. Groups are divided by the host function they represent. The bits in each group are ordered so that the high-order bits represent the states of higher priority (bit 7 is high-order, hexadecimal 80). Therefore, if more than one state is active within a group, the state with the highest priority is the one represented in the OIA image for the session. The following list shows the meaning of the bits for each group indicator.

Group 1 (offset 81): Online and screen ownership

7-6 Reserved
5 SSCP-LU session owns screen
4 LU-LU session owns screen
3 Online and not owned
2 Subsystem ready
1-0 Reserved

Group 2 (offset 82): Character selection (not supported)

7-0 reserved

Group 3 (offset 83): Shift state

7 Reserved
6 Numeric
5-0 Reserved

Group 4 (offset 84): PSS group 1 (not supported)

7-0 Reserved

Group 5 (offset 85): Highlight group 1 (not supported)

7-0 Reserved

Group 6 (offset 86): Colour group 1 (not supported)

7-0 Reserved

Group 7 (offset 87): Insert

7 Insert mode
6-0 Reserved

Group 8 (offset 88-92): Input inhibited (5 bytes)

(offset 88)

7-4 Reserved
3 Program check
2-0 Reserved

(offset 89)

7 Device busy (OIA time)
6-4 reserved
3 Too much entered
2-0 Reserved

(offset 90)

7-4 Reserved
3 Wrong place
2-0 Reserved

(offset 91)

7-6 Reserved
5 System wait
4-0 Reserved

(offset 92)

7-0 Reserved

Group 9 (offset 93): PSS group 2 (not supported)

7-0 Reserved

Group 10 (offset 94): Highlight group 2 (not supported)

7-0 Reserved

Group 11 (offset 95): Colour group 2 (not supported)

7-0 Reserved

Group 12 (offset 96): Communication error reminder (not supported)

7-0 Reserved

Group 13 (offset 97): Printer status (not supported)

7-0 Reserved

Group 14 (offset 98): Graphics (not supported)

7-0 Reserved

Group 15 (offset 99): reserved

7-0 Reserved

Group 16 (offset 100): Autokey play/record status (not supported)

7-0 Reserved

Group 17 (offset 101): Autokey abort/pause status (not supported)

7-0 Reserved

Group 18 (offset 102): Enlarge state (not supported)

7-0 Reserved

IBM 5250 Emulation

In the IBM 5250 emulation, the requested Data String Length must be 103, and the returned Data String contains the following data:

Offset 0	**Emulation Mode** (5250 = **9**, hexadecimal character 39).
Offset 1-80	**OIA Image**.
Offset 81-102	**OIA Group Indicators**.

The OIA image is a copy of the information on the displayed status line.

The OIA Group Indicators comprise bits representing the state of the connected session. Groups are divided by the host function they represent. The bits in each group are ordered so that the high-order bits represent the states of higher priority (bit 7 is high-order, hexadecimal 80). Therefore, if more than one state is active within a group, the state with the highest priority is the one represented in the OIA image for the session. The following list shows the meaning of the bits for each group indicator.

Group 1 (offset 81): Online and screen ownership

7-5	Reserved
4	System available
3	Reserved
2	Subsystem ready
1-0	Reserved

Group 2 (offset 82): Character selection (not supported)

7-0	Reserved

Group 3 (offset 83): Shift state

7-0	Reserved

Group 4 (offset 84): PSS group 1 (not supported)

7-0	Reserved

Group 5 (offset 85): Highlight group 1 (not supported)

7-0	Reserved

Group 6 (offset 86): Colour group 1 (not supported)

7-0	Reserved

Group 7 (offset 87): Insert

7	Insert mode
6-0	Reserved

Group 8 (offset 88-92): Input inhibited (5 bytes)

(offset 88)

7-0	Reserved

(offset 89)

7-0 Reserved

(offset 90)

7-3 Reserved
2 Operator input error
1-0 Reserved

(offset 91)

7-6 Reserved
5 System wait
4-0 Reserved

(offset 92)

7-0 Reserved

Group 9 (offset 93): PSS group 2 (not supported)

7-0 Reserved

Group 10 (offset 94): Highlight group 2 (not supported)

7-0 Reserved

Group 11 (offset 95): Colour group 2 (not supported)

7-0 Reserved

Group 12 (offset 96): Communication error reminder (not supported)

7-1 Reserved
0 Message waiting

Group 13 (offset 97): Printer status (not supported)

7-0 Reserved

Group 14 (offset 98): Graphics (not supported)

7-0 Reserved

Group 15 (offset 99): reserved

7-0 Reserved

Group 16 (offset 100): Autokey play/record status (not supported)

7-0 Reserved

Group 17 (offset 101): Autokey abort/pause status (not supported)

7-0 Reserved

Group 18 (offset 102): Enlarge state (not supported)

7-0 Reserved

Other Emulations

The size and status of the OIA reported is emulation dependent as listed in the table below.

Offset 0 **Emulation Mode**
Offset 1 **Status Type**
Offset 2-> **Status Line(s)**

Emulation	Mode	Status Type	Status Line(s)
VT52 **VT100** **VT320 (7-bit)** **VT320 (8-bit)** **DG200/410** **ANSI BBS**	0 1 2 3 7 9	0 - None 1 - Status 2 - Host Writable	None 80 character status 80/132 characters
Viewdata 40 **Viewdata 80** **Viewdata Split**	4 5 6	0 - None	None
HP700/92	8	0 - None 1 - Aids 2 - Modes 3 - User Keys 4 - Marg/Cols/Tabs 5 - User Keys Define 6 - Device Control 7 - Device Modes 8 - Host Writable	None 2 rows of 80 columns (types 1 - 8)
TA6530	11	1 - Status 2 - Error	80 character status 80 character error
IBM 3151	13	0 - None 1 - Operator Message 2 - Host Writable 3 - PF Key	None 80/132 characters 80/132 characters 80/132 characters
Wyse	14 to 21	Lower 4 bits: 0 - Blank Status 1 - Standard Status 2 - Extended Status Upper 4 bits: 0 - No Label Line 1 - Unshifted Label 2 - Shifted Label	 80/132 characters 80/132 characters 80/132 characters 80/132 characters 80/132 characters 80/132 characters
AT&T 4410	22	1 - PF Keys	2 rows of 80 columns
Stratus V102	23	0 - None	None

COPY PRESENTATION SPACE (5)

Copies the entire contents of the connected presentation space to a data string.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 5.

Data String: Predefined buffer, which must be at least the size of the current presentation space, or twice that if the EAB option is specified.

Data String Length: NA. The length of the presentation space in implied.

PS Position: NA.

Returned Parameters

Data String: Contents of the presentation space.

Data String Length: Length of the data string returned (in bytes), or twice that if the EAB option is specified.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
4	Successful, but host input available.
5	Successful, but keyboard locked.
9	A system error occurred.

Remarks

Returns 8-bit ASCII characters (ISO Latin-1).

This function is affected by the ATTRIB/NOATTRIB, EAB/NOEAB and the XLATE/NOXLATE options of the **Set Session Parameters** (9) function.

If you want to copy only part of the presentation space, use **Copy Presentation Space To String** (8).

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY PRESENTATION SPACE TO STRING (8)

Copies all or part of the contents of the connected presentation space to a data string.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 8.

Data String: Predefined buffer to hold string (twice the length requested if the EAB option has been specified).

Data String Length: Length of data string required.

PS Position: Position in the presentation space to start the copy. This is the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

Returned Parameters

Data String: Contents of the current presentation space as requested.

Data String Length: Length of the data string returned in bytes.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Data string length 0, or PS position + (length - 1) greater than PS size.
4	Successful, but host input available.
5	Successful, but keyboard locked.
7	Invalid presentation space position.
9	A system error occurred.

Remarks

Returns 8-bit ASCII characters (ISO Latin-1). This function is affected by the ATTRIB/NOATTRIB, EAB/NOEAB and the XLATE/NOXLATE options of the **Set Session Parameters** (9) function.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY STRING TO FIELD (33)

Transfers a string of characters to a field in the currently connected presentation space.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 33.

Data String: String containing 8-bit ASCII displayable characters for the target field in the presentation space.

Data String Length: Length of data string (in bytes). This parameter is not applicable if in end of transmission (EOT) mode (see **Set Session Parameters** (9)).

PS Position: Any character position within the target field. (Note that the copy will always start at the beginning of the field.)

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Data string length 0 specified.
4	Function inhibited as target presentation space was busy.
5	Target field was protected or inhibited, or invalid data was sent.
6	Copy complete but data truncated.
7	Presentation space position invalid.
9	A system error occurred.
24	The presentation space is unformatted.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Only 8-bit ASCII displayable characters are supported. An attempt to write non-displayable characters (e.g. new line, carriage return or attribute values) will return error code **5** (invalid data was sent); the string will be copied up to but not including the invalid character.

The location of a character position within the target field is determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The first byte of the data string is always placed in the first character position of the target field. If the data string provided is longer than the target field, the data string is truncated to fit and a return code **6** (copy complete but data truncated) is returned.

End of transmission (EOT) mode (see **Set Session Parameters** (9)) is supported for all emulations.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

COPY STRING TO PRESENTATION SPACE (15)

Transfers characters to a specified position in the currently connected presentation space.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 15.

Data String: String containing 8-bit ASCII characters to be copied to the presentation space.

Data String Length: Length of the data string (in bytes). This parameter is not applicable if in end of transmission (EOT) mode (see **Set Session Parameters** (9)).

PS Position: Position in the presentation space to start the copy (for formatted presentation spaces only, otherwise this parameter is ignored).

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Data string length 0 specified.
4	Function inhibited as target presentation space was busy.
5	Target field was protected or inhibited, or invalid data was sent.
7	Presentation space position invalid.
9	A system error occurred.

Remarks

This function is faster than **Send Keystring** (3) and **Send Keypress** (103), but these functions provide additional support for special characters. The result is very similar to that achieved by **Send Keystring** (3), but bypasses the keyboard input checks for special keystrokes and macro expansion.

The PS position is determined by the offset into the formatted presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

In unformatted/non-block mode presentation spaces, the characters will be sent to the host computer (unless it is in local mode), not to the screen.

End of transmission (EOT) mode (see **Set Session Parameters** (9)) is supported for all emulations.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

DEBUG (150)

This causes all host data to be output to the Microsoft Debug window.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 150.

Data String: NA.

Data String Length: NA.

PS Position: Contains the following data:

Offset	Definition
2	Log in text.
4	Log in hexadecimal.
8	Log in DLL.
16	Log query session information.

DISCONNECT PRESENTATION SPACE (2)

Drops the connection between the API application program and the host presentation space. It also cancels the effect of the **Reserve** (11) function.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 2.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
8	Either USENEW is set and there is an existing session, or USEOLD is set and there isn't an existing session. (See the **Set Session Parameters** (9) function.)
9	A system error occurred.

Remarks

This function will cause **Stop Host Notification** (25) and **Stop Keystroke Intercept** (53) to be actioned if the relevant '**Start**' function has been called since the **Connect Presentation Space** (1) for this presentation space.

This function does not reset the session parameters to their default condition. Your API application must call the **Reset System** (21) function to reset to the defaults.

There is no need to disconnect from all instances of TeemTalk before exiting, but they will remain functional after the API application terminates.

After the function is actioned, the presentation space is available for use by a different application.

FIND FIELD LENGTH (32)

Returns the length of a target field in a field formatted presentation space. The function is valid for protected and unprotected fields. Note that you can use the **Find Field Position** (31) function to locate a particular field.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 32.

Data String: Two character field code as follows:

Code	Explanation
[space][space]	Current field.
T[space]	Current field.
N[space]	The Next field (protected or not).
NP	The Next Protected field.
NU	The Next Unprotected field.
P[space]	The Previous field (protected or not).
PP	The Previous Protected field.
PU	The Previous Unprotected field.

Data String Length: NA (2 implied).

PS Position: Any character position within the target field.

Returned Parameters

Data String Length:

0 If the return code is **28**, the field length is zero.
If the return code is **24**, the presentation space is unformatted.

>0 The number of characters from the beginning of the specified field up to the character preceding the next attribute byte.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.
7	Invalid presentation space specified.
9	A system error occurred.
24	The presentation space is unformatted, or no such field found.
28	Field length was **0**.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Use the **Find Field Position** (31) function to locate a particular field. The location of a character position within the target field is determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

FIND FIELD POSITION (31)

Returns the start position of a target field in a field-formatted presentation space. The function can be used for protected or unprotected fields.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 31.

Data String: Two character field code as follows:

Code	Explanation
[space][space]	Current field.
T[space]	Current field.
N[space]	The Next field (protected or not).
NP	The Next Protected field.
NU	The Next Unprotected field.
P[space]	The Previous field (protected or not).
PP	The Previous Protected field.
PU	The Previous Unprotected field.

Data String Length: NA (2 implied).

PS Position: Position within the presentation space to start the find.

Returned Parameters

Data String Length: Location of the first character position within the requested field, or **0** for return codes **24** and **28**.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.

Code	Explanation
7	Invalid presentation space specified.
9	A system error occurred.
24	The presentation space is unformatted, or no such field found.
28	Field length was **0**.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Presentation space positions are determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

GET KEY (51)

This is only supported by the IBM emulations. It will receive a keystroke from a session that has keystroke intercept enabled (see **Start Keystroke Intercept** (50)) so that it can either be processed, accepted or rejected by your API program. This function can be used to intercept a string by placing it in a loop.

Prerequisite Calls

Start Keystroke Intercept (50)

Supplied Parameters

Function Number: Must be 51.

Data String: Contains the following data:

Offset	Definition
0	One character ID for presentation space, or blank for current presentation space.
1-5	Blanks for holding symbol representation of keystroke.
6-7	Reserved.

Data String Length: NA (8 implied).

PS Position: NA.

Returned Parameters

Data String: Contains the following data:

Offset	Definition
0	One character ID for presentation space, or blank for current presentation space.
1	Character code as follows: **A** = for ASCII character. **M** = for keystroke mnemonic. **S** = for special shift (**Alt/Ctrl**) returned with other data.
2-7	Keystroke data.

In the following returned data string examples, the escape character is represented by the @ symbol. Any ASCII keystroke character can be specified as the escape character for keystroke mnemonics by using the **ESC=** option of the **Set Session Parameters** (9) function. Refer also to the *ASCII Mnemonics* section in the *Overview* chapter for a list of all supported mnemonics and their meaning.

BAg Indicates that the API program is connected to presentation space **B**, the keystrokes are ASCII (**A**), and the returned key is the character **g**.

BM@4 Indicates that the API program is connected to presentation space **B**, the keystrokes are mnemonic (**M**), and the returned key is **PF4**.

BS@A@4 Indicates that the API program is connected to presentation space **B**, the keystrokes are in special shift state (**S**), and the returned key is **Alt + PF4**.

BS@r@4 Indicates that the API program is connected to presentation space **B**, the keystrokes are in special shift state (**S**), and the returned key is **Ctrl + PF4**.

Data String Length: Length of the data string returned.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	An invalid presentation space was specified.
5	The AID keys only option was specified by **Start Keystroke Intercept** (50) and non-AID keys are inhibited by this session type.
8	**Start Keystroke Intercept** (50) has not been called.
9	A system error occurred.
20	An undefined key combination was typed.
25	The requested keystrokes were not available in the input queue.
31	Keystrokes were lost due to input queue overflow.

Remarks

The **Start Keystroke Intercept** (50) function specifies the size of the keystroke intercept buffer in which keystrokes are queued asynchronously. Each keystroke occupies three bytes in the buffer. If **Get Key** returns code **31**, either increase the size of the keystroke intercept buffer, or execute **Get Key** more frequently to retrieve keystrokes from the buffer so that space is made available.

Get Key will read keystrokes into the data area of your application. A WAIT option can be specified by **Set Session Parameters** (9) so that control is not returned to the API program until **Get Key** has intercepted a keystroke as defined by **Start Keystroke Intercept** (50).

If only AID keys (e.g. **Enter**) are required, as may be the case for field formatted sessions, set the **Start Keystroke Intercept** (50) option code to **D**.

The **Send Keystring** (3) function can be used to pass original keystrokes to the host-connected presentation space.

The type of keystroke data received by the host application when an operator types a key or shift key combination is as follows:

Key defined as an ASCII character

A 1-byte ASCII value corresponding to that character is received.

Key defined as a function

A 2-, 4-, or 6-byte ASCII mnemonic corresponding to that function is received. For example, if the **Delete** key is pressed, then **@D** is received.

Defined shift key combination

An ASCII character or the 2-, 4-, or 6-byte ASCII mnemonic corresponding to that function is received. For example, if the combination **Alt + z** is defined as delete, then **@D** is received when **Alt + z** is typed.

Single undefined key

Nothing is received. **Get Key** returns a return code **20**.

Undefined shift key combination

Depends on whether the unshifted state of the key is defined.

If the unshifted state is not defined, nothing is received. **Get Key** returns a return code **20**.

If the unshifted state is defined as an ASCII character, the ASCII mnemonic for the shift key followed by the character associated with the unshifted key is received. For example, if the **Ctrl** key is pressed with a key defined as lowercase **f**, **@rf** will be received.

If the unshifted state is defined as afunction, the ASCII mnemonic for the shift key followed by the ASCII mnemonic corresponding to the defined function is received. For example, if the **p** key is mapped as Print Screen, and **Ctrl + p** is undefined, when **Ctrl + p** is pressed, **@r@A@T** (the mnemonics for Ctrl and Print Screen) will be received, not **@rp** (the mnemonics for Ctrl and p).

PAUSE (18)

This causes the application to wait for a specified amount of time or until a host event occurs if **Start Host Notification** (23) has been called.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 18.

Data String: One character ID of the required presentation space, or blank for all presentation spaces that have **Start Host Notification** (23) enabled (if required for IPAUSE).

Data String Length: Pause duration in half-second increments. For example, a value of 120 will result in a pause duration of 60 seconds.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The wait duration has expired.
9	A system error occurred.
26	Presentation space or status line updated.

Remarks

The behaviour of this function depends on the session parameters FPAUSE and IPAUSE (see **Set Session Parameters** (9)).

If **Start Host Notification** (23) is not enabled, or the FPAUSE option is set in **Set Session Parameters** (9) (default), the **Pause** function will wait for the specified time and return code **0**.

If the IPAUSE option is set, **Pause** will return code **26** if the presentation space or status line has been updated. Determine this by **Query Host Update** (24) before calling **Pause** again, otherwise it will return code **26** continually.

POST INTERCEPT STATUS (52)

This informs the emulation that a keystroke obtained by **Get Key** (51) has been accepted or rejected, and will cause a warning beep to sound if a keystroke is rejected.

Prerequisite Calls

Start Keystroke Intercept (50)

Supplied Parameters

Function Number: Must be 52.

Data String: A 2-byte data string containing the following:

Offset	Definition
0	One character ID for presentation space, or blank for current presentation space.
1	**A** = for keystroke accepted. **R** = for keystroke rejected.

Data String Length: NA (2 implied).

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	An invalid presentation space was specified.
2	An invalid session option was specified.
8	**Start Keystroke Intercept** (50) has not been called for this presentation space.
9	A system error occurred.

QUERY CURSOR LOCATION (7)

This function indicates the current cursor position in the presentation space.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 7.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Data String Length: Presentation space positional value of the cursor.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
9	A system error occurred.

Remarks

To translate the presentation space positional value into a cursor location in rows and columns, use **Convert Position/RowCol** (99).

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

QUERY FIELD ATTRIBUTE (14)

Provides a copy of the attribute byte of the field containing the specified presentation space position.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 14.

Data String: NA.

Data String Length: NA.

PS Position: PS position of any byte in the target field.

Returned Parameters

Data String: NA.

Data String Length: If the screen is formatted and the emulation supports attribute bytes then the data string length contains the attribute byte as specified in the *Field Attribute Representation* appendix.

If the screen is unformatted or the emulation does not support attribute bytes then **0** is returned.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
7	Presentation space position invalid.
9	A system error occurred.
24	Unformatted presentation space or attribute bytes not supported by the emulation.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Presentation space positions are determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The current field attributes are determined by the bit setting of the attribute byte. Refer to the *Field Attribute Representation* appendix for a description of how field attributes are represented in the data string returned by this function.

See the **Query Formatted Field Attribute** (114) function for a more comprehensive function that returns the entire internally stored 4-byte field attribute.

QUERY FORMATTED FIELD ATTRIBUTE (114)

Provides a copy of the internal 4-byte representation of the attribute of the field containing the specified presentation space position.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 114.

Data String: Predefined buffer to hold string.

Data String Length: NA (4 bytes implied).

PS Position: PS position of any byte in the target field.

Returned Parameters

Data String: 4 bytes of character/attribute value.

Data String Length: Returns **4** if successful, otherwise **0**.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
7	Presentation space position invalid.
9	A system error occurred.
24	Unformatted presentation space.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T 4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Presentation space positions are determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The character/attribute value is emulation specific. Refer to the *Field Attribute Representation* appendix for details.

QUERY HOST UPDATE (24)

Determines whether the specified presentation space and/or the OIA (status line) has been updated since the last call to **Start Host Notification** (23) or **Query Host Update** (24).

Prerequisite Calls

Start Host Notification (23)

Supplied Parameters

Function Number: Must be 24.

Data String: One character ID for presentation space, or blank for the current presentation space.

Data String Length: NA (1 implied).

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	No updates since last call.
1	Invalid presentation space specified.
8	No prior **Start Host Notification** (23) for presentation space.
9	A system error occurred.
21	OIA updated.
22	Presentation space updated.
23	Presentation space and OIA updated.

Remarks

The presentation space does not have to be connected for this function to work, but a **Start Host Notification** (23) must have been issued to it beforehand.

If the presentation space has been terminated by the user (not by the API application) then return code **9** (system error) will be returned.

This function only supports reporting for both presentation space and OIA updates.

QUERY SESSION STATUS (22)

This function obtains session-specific information for a connected presentation space.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 22.

Data String: One character ID for presentation space, or blank for the current presentation space, plus 17 bytes for returned data.

Data String Length: 18 bytes.

PS Position: NA.

Returned Parameters

Data String: Contains the following returned data:

Offset	Definition
0	One character ID for presentation space.
1-8	Session long name (same as PSID).
9	D = IBM 3270 Display E = IBM 3270 Printer F = IBM 5250 Display T = Otherwise
10	0 (Hex 80 = IBM 3270 extended attribute supported).
11-12	Number of rows in presentation space (binary).
13-14	Number of columns in presentation space (binary)
15-16	IBM 3270 only: Host code page (binary).
17	Emulation mode as specified for **Copy OIA** (13).

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Invalid presentation space specified.
2	Invalid data string length.
9	A system error occurred.

Remarks

The number of rows and columns is zero for printer sessions.

QUERY SESSIONS (10)

This function returns details of all configured sessions.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 10.

Data String: Predefined buffer to hold string.

Data String Length: Minimum 12 times number of configured sessions.

PS Position: NA.

Returned Parameters

Data String: Contains the following 12 bytes of returned data for each configured session:

Offset	Definition
0	One character ID for presentation space.
1-8	Session long name (same as PSID).
9	Session type. (H = Host)
10-11	Size of presentation space (binary).

Data String Length: Number of configured sessions.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
2	Invalid data string length.
9	A system error occurred.

Remarks

If the return code is **0** or **2**, the data string length will indicate the number of configured sessions. This number can be used to determine the minimum data string length required.

A configured session is any session defined in the **TTHLLAPI.INI** file.

The size of the presentation space is zero for printer sessions.

QUERY SYSTEM (20)

Determines the level of API and TeemTalk support as well as other system related values.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 20.

Data String: Predefined buffer to hold 35 bytes.

Data String Length: NA (implied length is 35 bytes).

PS Position: NA.

Returned Parameters

Data String: Contains the following 35 bytes:

Offset	Definition
0	API version number.
1-**2**	API level number.
3-**8**	API date (MMDDYY).
9-**11**	Reserved.
12	Hardware **A** = Personal Computer
13	Program type **P** = IBM PC 3270
14-**15**	Reserved.
16-**17**	Emulator version for last connect, else **?**.
18	Emulator type: **2** = 320, **4** = 340, for last connect, else **?**.
19-**24**	Emulator date (YYMMDD) for last connect, else **?**.
25 -**34**	Reserved.

Data String Length: Returns **35** if successful, else **0**.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
9	A system error occurred.

RELEASE (12)

Unlocks the currently connected presentation space previously locked by the **Reserve** (11) function.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 12.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
9	A system error occurred.

Remarks

If you do not release the presentation space, it will remained locked until a **Disconnect Presentation Space** (2) or **Reset System** (21) is called.

If a presentation space is reserved when a API application is terminated, TeemTalk will be unable to process user input.

RESERVE (11)

This function locks the currently connected presentation space from change by the user via the keyboard or mouse.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 11.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
9	A system error occurred.

Remarks

Use the **Release** (12) function to unlock the presentation space. If you do not release the presentation space, it will remained locked until a **Disconnect Presentation Space** (2) or **Reset System** (21) is called.

If a presentation space is reserved when a API application is terminated, TeemTalk will be unable to process user input.

RESET SYSTEM (21)

This function reinitializes API to its starting state by defaulting all parameters and disconnecting any active sessions.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 21.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
9	A system error occurred.

Remarks

This function disconnects all presently active instances of TeemTalk.

It is advisable to call this function prior to terminating the API application. This will ensure all copies of TeemTalk are 'disconnected'.

SEARCH FIELD (30)

This function examines a field within a field-formatted presentation space for the occurence of a specified string. If the string is found, the presentation space position of the first character is returned. The function can be used for protected or unprotected fields.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 30.

Data String: Target string to search for in the presentation space (8-bit ASCII).

Data String Length: Length of the data string (in bytes). This parameter is not applicable in end of transmission (EOT) mode (see **Set Session Parameters** (9)).

PS Position: Position in the presentation space to start the search.

Returned Parameters

Data String Length: Presentation space position of the first character in the required data string, or **0** if not found.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.
7	Invalid presentation space specified.
9	A system error occurred.
24	The presentation space is unformatted or string not found.

Remarks

This function is not supported by VT52, ANSI-BBS, DG410, AT&T4410, Stratus V102, or the Viewdata modes. It is only valid when the presentation space is field formatted. Field formatting is emulation dependent as shown in the table on page 3-2.

Presentation space positions are determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The exact nature of the search within the field is determined by the following **Set Session Parameters** (9) options:

SRCHALL, SRCHFRWD	Searches the entire field that contains the PS position from the beginning to the end. The first instance of the required string is returned if it exists.
SRCHALL, SRCHBKWD	Searches the entire field that contains the PS position from the end to the beginning. The last instance of the required string is returned if it exists.
SRCHFROM, SRCHFRWD	The search begins at the PS position and continues to the end of the field until the specified string is found.
SRCHFROM, SRCHBKWD	The search begins at the PS position and continues to the start of the field until the specified string is found.

An attempt to search for non-displayable characters will not be successful and a return code **24** (string not found) will be issued.

SEARCH PRESENTATION SPACE (6)

Examines the connected presentation space for the occurrence of a specified string (characters only). If the string is found, the presentation space position of the first character is returned.

This function can be used to determine when the session is ready for input by searching for a required prompt.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 6.

Data String: Target string to search for in the presentation space (8-bit ASCII).

Data String Length: Length of the data string (in bytes). This parameter is not applicable in end of transmission (EOT) mode.

PS Position: Position in the presentation space to start the search if SRCHFROM is set by the **Set Session Parameters** (9) function, else NA.

Returned Parameters

Data String Length: Presentation space position of the first character in the required data string, or **0** if not found.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.
7	Host presentation space position invalid.
9	A system error occurred.
24	String not found.

Remarks

Presentation space positions are determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The exact nature of the search within the field is determined by the following **Set Session Parameters** (9) options:

SRCHALL, SRCHFRWD	Ignores the specified PS position and searches the entire presentation space from the beginning to the end. The first instance of the required string is returned if it exists.
SRCHALL, SRCHBKWD	Ignores the specified PS position and searches the entire presentation space from the end to the beginning. The last instance of the required string is returned if it exists.
SRCHFROM, SRCHFRWD	The search begins at the PS position and continues to the end of the presentation space until the specified string is found.
SRCHFROM, SRCHBKWD	The search begins at the PS position and continues to the start of the presentation space until the specified string is found.

An attempt to search for non-displayable characters will not be successful and a return code **24** (string not found) will be issued.

SEND KEYPRESS (103)

This is only supported by the IBM emulations. It imitates the pressing of an emulator macro key by the user (in TeemTalk internal format).

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 103.

Data String: Macro key number or virtual key name (refer to the TeemTalk for Windows User's Guide) as an 8-bit ASCII string.

Data String Length: Length of the data string. This parameter is not applicable if in end of transmission (EOT) mode.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.
4	Host session busy.
5	Input was inhibited.
9	A system error occurred.

Remarks

This function simulates the pressing of a single emulator macro key (the macro key numbers and virtual key names are defined in the TeemTalk for Windows User's Guide). In most cases this can also be achieved using **Send Keystring** (3).

If NORESET is requested by **Set Session Parameters** (9), a reset function is performed before the macro key. This will reset all states that can be reset except input-inhibited states. Note that soft keyboard locks will also be reset.

The effect on the emulator will be exactly as if the keystroke had occurred at the keyboard. If the keyboard is locked then a bell will sound. If the emulation is busy then the key macro will be placed on the keyboard queue and processed when the emulator becomes free.

The emulator will return once the keystroke string has been actioned, input was inhibited or host session was busy whenever these two conditions exist.

Error code **4** (host session busy) will be returned when the host is busy and the keyboard is locked.

SEND KEYSTRING (3)

Imitates the pressing of a series of keys by the user.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 3.

Data String: Series of keystrokes (maximum 255 bytes) specified as ASCII characters, virtual key names as listed in the TeemTalk.2000 User's Guide, or ASCII mnemonics (refer to the *Overview* chapter).

Data String Length: Length of the data string. This parameter is not applicable if in end of transmission (EOT) mode.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Parameter error.
4	Host session was busy.
5	Input was inhibited.
9	A system error occurred.

Remarks

Function keys can be represented by ASCII mnemonics as described in the *Overview* chapter. If an unrecognized mnemonic is sent, a beep and a return code rejecting the key may be sent.

If NORESET has not been requested by **Set Session Parameters** (9), a reset function is performed before the keystroke string. This will reset all states that can be reset except input-inhibited states. Note that soft keyboard locks will also be reset.

The effect on the TeemTalk product emulation will be exactly as if the keystrokes had occurred at the keyboard. If the keyboard is locked then the bell will sound for each keystroke. If the emulation is busy then the keystrokes will be placed on the keyboard queue and processed when the emulator becomes free. Error code **4** (host session busy) will be returned when the host is busy and the keyboard is locked.

SET CURSOR (40)

Sets the position of the cursor within the connected presentation space.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 40.

Data String: NA.

Data String Length: NA.

PS Position: Desired cursor position in the presentation space.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
7	Invalid presentation space position.
9	A system error occurred.

Remarks

A presentation space position is determined by the offset into the presentation space where position 1 is the top left corner of the screen display (row 1 and column 1), and the highest position number is the bottom right corner.

The IBM 5250 emulation supports a presentation space of 24 rows by 80 columns. When an error message is received from the host or when the operator presses the **SysReq** key, a 25th row is displayed. When row 25 is displayed, it is a valid area for this function.

SET SESSION PARAMETERS (9)

This function enables you to change certain default global session parameters. The parameters remain in effect until changed by this function or a **Reset System** (21) is issued.

Prerequisite Calls

None.

Supplied Parameters

Function Number: Must be 9.

Data String: Contains the session options that are to be changed (see the following pages), separated by commas or spaces.

Data String Length: Exact length of the data string in bytes (STREOT is not allowed).

PS Position: NA.

Returned Parameters

Data String Length: For return code 2, number of valid parameters set, else unchanged.

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
2	One or more parameters invalid.
9	A system error occurred.

Remarks

All correct parameters will be interpreted even if invalid parameters are present in the data string.

Session Options

The valid session options are as follows.

Data String Length

The following options specify how the data string length is determined.

STRLEN Default. An exact length is passed for all strings.

STREOT String lengths are not specifically encoded. They are terminated with an EOT character.

End Of Transmission Character

The following option specifies the end of transmission (EOT) character used to terminate a string when STREOT is set.

EOT=c "c" is the binary value of the EOT character, the default being binary zero. No spaces are to be included on either side of the equals sign. The space character is not a valid EOT character.

Extent Of Presentation Space Search

The following options specify how the presentation space is to be searched.

SRCHALL Default. A **Search Presentation Space** (6) call will search the entire presentation space, and **Search Field** (30) will search the entire field.

SRCHFROM If the session option SRCHFRWD is set, the search will start at the specified PS position and stop at the end of the field or presentation space. If SRCHBKWD is set, the search will start at the end of the presentation space or field and stop at the specified PS position.

Direction Of Search

The following options specify the direction in which a search is performed when the session option SRCHFROM is set.

SRCHFRWD Default. The search starts at the specified PS position and stops at the end of the field or presentation space.

SRCHBKWD The search starts at the specified PS position and stops at the specified PS position.

Pause Type

The following options determine the type of pause to use for the **Pause** (18) function.

FPAUSE Default. Pauses for the full length of time specified by the **Pause** (18) function.

IPAUSE Any host event can interrupt the pause once a **Start Host Notification** (23) call is made.

Reset Prior To Keystroke

The following options determine whether strings sent by the **Send Keystring** (3) function are automatically preceded with a reset.

AUTORESET Default. All strings sent by **Send Keystring** (3) are preceded with a reset.

NORESET Strings sent by **Send Keystring** (3) are not preceded with a reset.

Escape Character For Keystroke Mnemonics

The following option affects the **Send Keystring** (3) and **Get Key** (51) functions and specifies the literal character to be used to represent the escape character in keystroke mnemonics.

ESC=@ Specifies that @ represents the escape character (default). The character must immediately follow the equals sign, without a space.

Keyboard Mapping

The following options affect the **Send Keystring** (3) and **Get Key** (51) functions and determine whether standard IBM or direct keyboard mapping is supported. The *ASCII Mnemonics* section in chapter 2 lists the functions supported by each option.

KEYSIBM Default. Enable standard IBM mapping.

KEYSKBD Enable direct keyboard mapping.

Trace Facility

The following options enable or disable the trace facility that is used to debug the API program.

TROFF Default. Turns trace off.

TRON Turns trace on. All API functions are traced to 'stderr', which can be piped to a file if required.

Wait Period Characteristics

The following options affect the **Wait** (4) and **Get Key** (51) functions, but in different ways.

TWAIT For **Wait** (4), waits up to one minute before timing out on keyboard lock (default).

For **Get Key** (51), control is not returned to the API program until it has intercepted a normal or AID key as specified by the **Start Keystroke Intercept** (50) function.

LWAIT For **Wait** (4), waits forever on keyboard lock.

For **Get Key** (51), control is not returned to the API program until it has intercepted a normal or AID key as specified by the **Start Keystroke Intercept** (50) function.

NWAIT For **Wait** (4), checks status and return immediately without waiting.

For **Get Key** (51), returns return code 25 (keystrokes not available) in the fourth parameter if the keystroke intercept queue does not contain anything that matches the option specified by the **Start Keystroke Intercept** (50) function.

Attribute Byte Treatment

The following options determine how attribute bytes are treated.

NOATTRB Default. All bytes that do not have ASCII equivalents will be converted to spaces.

ATTRB All bytes that do not have ASCII equivalents will be passed as their original values.

Extended Attribute Byte Return

The following options determine whether **Copy Presentation Space** (5) and **Copy Presentation Space To String** (8) return extended attribute bytes (EABs).

NOEAB Default. Pass data only.

EAB Pass presentation space data with extended attribute bytes. Two bytes will be returned for each position in the presentation space. The first byte is the data character, the second is the extended attribute byte. The EAB returned is determined by the NOXLATE/XLATE setting, as described in the next section.

Extended Attribute Byte Translation

The following options determine the type of extended attribute byte (EAB) returned by **Copy Presentation Space** (5) and **Copy Presentation Space To String** (8).

Note that in all modes except IBM 3270 and IBM 5250, EABs are always returned as 0 (zero).

IBM 3270 EABs

NOXLATE Default. Standard EABs returned.

Bits	Meaning		
7 - 6	Visual attribute:	Hex 40	Blink
		80	Reverse video
		C0	Underline
5 - 3	Foreground colour:	0 Black	4 Green
		1 Blue	5 Turquoise
		2 Red	6 Yellow
		3 Pink	7 White
2 - 0	Reserved.		

XLATE EABs returned as PC CGA colours. The top four bits return the background colour, the bottom four bits return the foreground colour. The colours are as follows:

0 Black	6 Brown	12 Light red
1 Blue	7 White	13 Light magenta
2 Green	8 Grey	14 Yellow
3 Cyan	9 Light blue	15 High intensity white
4 Red	10 Light green	
5 Magenta	11 Light cyan	

IBM 5250 EABs

NOXLATE Default. Standard EABs returned.

Bits	Meaning
7	Reverse
6	Underscore
5	Blink
4	Column separators
3 - 0	Reserved

XLATE EABs returned as PC CGA colours. The top four bits return the background colour, the bottom four bits return the foreground colour. The colours are as follows:

0	Black	6	Brown	12	Light red
1	Blue	7	White	13	Light magenta
2	Green	8	Grey	14	Yellow
3	Cyan	9	Light blue	15	High intensity white
4	Red	10	Light green		
5	Magenta	11	Light cyan		

Non-Display (Hidden) Fields Return

The following options specify how non-display fields will be copied using **Copy Presentation Space** (5), **Copy Presentation Space To String** (8) or **Copy Field To String** (34).

DISPLAY Characters copied as for display fields.

NONDISPLAY Characters copied as spaces.

START HOST NOTIFICATION (23)

Begins the process by which **Pause** (18) and **Query Host Update** (24) can determine if the presentation space and/or status line (OIA) has been updated.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 23.

Data String: Contains the following data:

Offset	Definition
0	One character ID for presentation space, or blank for current presentation space.
1	**P** = for presentation space update only. **O** = for OIA update only. **B** = for presentation space and OIA updates.

Data String Length: NA (2 implied).

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Invalid parameter.
9	A system error occurred.

Remarks

Currently only supports reporting for both presentation space and OIA updates.

START KEYSTROKE INTERCEPT (50)

This function will intercept keystokes sent to a session by a terminal operator and store them until the keystroke queue overflows or the **Stop Keystroke Intercept** (53) or **Reset System** (21) functions are called. The intercepted keystokes can be:

- received through **Get Key** (51) and sent to the current or another session by **Send Keystring** (3).
- substituted with other keystrokes by **Send Keystring** (3).
- accepted or rejected by **Post Intercept Status** (52).
- used to initiate a process.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 50.

Data String: Contains the following data:

Offset	Definition
0	One character ID for presentation space, or blank for current presentation space.
1	An **option code** character **D** = for AID keystrokes only **L** = for all keystrokes.
2 - 5	The data at these positions is ignored.

Data String Length: The number of bytes in the keystroke intercept buffer, 256 are recommended. API allocates a minimum of 8 bytes for this buffer.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Not currently connected to a presentation space.
2	Invalid parameter.
4	Function inhibited as target presentation space was busy.
9	A system error occurred. **Release** (12) is being used.

Remarks

If **Get Key** (51) returns code **31**, either increase the size of the keystroke intercept buffer for this function, or execute **Get Key** (51) more frequently to retrieve keystrokes from the buffer so that space is made available. Note that each intercepted keystroke occupies three bytes in the buffer.

If option code D is provided, intercepted non-AID keys will be written to the intended presentation space, and only AID keys will be returned to the application.

Your API program must call the **Stop Keystroke Intercept** (53) function before exiting, otherwise this function will remain active and produce unexpected results.

STOP HOST NOTIFICATION (25)

Stops the process by which **Pause** (18) and **Query Host Update** (24) can determine if the presentation space and/or status line (OIA) has been updated.

Prerequisite Calls

Start Host Notification (23)

Supplied Parameters

Function Number: Must be 25.

Data String: One character ID for presentation space, or blank for current presentation space.

Data String Length: NA (1 implied).

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Invalid presentation space specified.
8	No prior **Start Host Notification** (23).
9	A system error occurred.

STOP KEYSTROKE INTERCEPT (53)

Stops the keystroke intercept process previously initiated by the **Start Keystroke Intercept** (50) function.

Prerequisite Calls

Start Keystroke Intercept (50)

Supplied Parameters

Function Number: Must be 53.

Data String: One character ID for presentation space, or blank for current presentation space.

Data String Length: NA (1 implied).

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful.
1	Invalid presentation space specified.
8	No prior **Start Keystroke Intercept** (50).
9	A system error occurred.

WAIT (4)

This function checks the host session for a wait condition. If the session is busy or the keyboard is locked, the API application will wait for the amount of time specified by the **Set Session Parameters** (9) function to see if the wait condition clears.

Prerequisite Calls

Connect Presentation Space (1)

Supplied Parameters

Function Number: Must be 4.

Data String: NA.

Data String Length: NA.

PS Position: NA.

Returned Parameters

Return Code: The following codes are valid:

Code	Explanation
0	The function was successful; the keyboard was unlocked and the emulator was not busy.
1	Not currently connected to a presentation space.
4	Time-out while still busy.
5	Keyboard locked.
9	A system error occurred.

Remarks

The **Wait** function can be used to allow functions such as **Send Keystring** (3) sufficient time to complete.

The behaviour of the **Wait** function is determined by the TWAIT, LWAIT or NWAIT setting specified by the **Set Session Parameters** (9) function.

If TWAIT is specified, the function waits for up to one minute for the emulator to become not busy and/or for an input-inhibited state to be cleared. If the emulator is busy and input is inhibited the function returns error code **4** (time-out while still busy).

If LWAIT is specified, the function waits indefinitely for the emulator to become not busy and for any input-inhibit state to be cleared. This option is not recommended as control does not return to the API application until the host is available.

If NWAIT is specified, the function returns immediately after checking the status of the TeemTalk product. Note that the return code **4** (time-out while still busy) is returned when the emulator is busy and in an input inhibit state.

Note that a successful return code **0** from a **Wait** function only indicates that the emulator is now free (not busy and unlocked) and not that a specific application has completed updating the screen. Use **Search Field** (30) or **Search Presentation Space** (6) combined with the **Wait** function to look for expected keyword prompts from the user application in order to determine the completion of a host update.

The 'busy' indication (on this and other functions) will be reported on ALL emulations while data is waiting to be processed from the host. Additionally, the 'busy' indication will be reported on the IBM 3270 emulation while XSYSTEM or XCLOCK is present, the IBM 5250 emulation while in a 'locked' state (awaiting host reply), and the HP700/92 emulation while in a 'transmit pending' state.

Wait is useful in waiting for the target computer to complete updating the presentation space after sending a command. The procedure relies on waiting for the presentation space to become busy indicating that the response is being received from the target computer. The API application should then wait for the presentation space to become not busy and finally search for an application specific prompt string or the presence of the cursor in a given location. The procedure should be repeated until the application specific prompt string is found in the presentation space or field specified.

The use of **Wait** in the above procedure reduces the processor load when waiting for the presentation space to become free.

A

Field Attribute Representation

This appendix describes how field attributes are represented in the data string returned by a query field attribute function.

Introduction

You can use one of three functions to provide a copy of the field attribute at the specified presentation space position:

Query Field Attribute (14)
Copy Formatted Presentation Space (105)
Query Formatted Field Attribute (114)

The following sections describe the meaning of the bytes returned by these functions.

Query Field Attribute

The information returned by the **Query Field Attribute** (14) function depends on the current terminal emulation.

VT Edit Mode:	Hex	Attribute
	01	Blink
	02	Reverse video
	04	Underline
	08	Bold
	10	Blank
	40	Mask for erasable character

HP700/92 Mode:	**Hex**	**Attribute**
	01	Blink
	02	Reverse video
	04	Underline
	08	Half intensity
	10	Security
	20	Line drawing
	40	Unprotected
	80	Protected

TA6530 Mode:	**Hex**	**Attribute**
	01	Modified data
	0E	Data type mask
	10	Auto tab disable
	20	Protected
	40	Upshift
	80	Field attribute

Wyse Modes:	**Hex**	**Attribute**
	01	Blink
	02	Reverse video
	04	Underline
	08	Half intensity
	10	Blank
	20	Erasable
	40	Field attribute

IBM 3270 Mode:	**Hex**	**Attribute**
	01	Modified data
	0C	Intensity mask:
		00 Display
		04 Display, intense
		08 Selectable
		0C Hidden
	10	Numeric
	20	Protected
	C0	Field attribute

IBM 3151 Mode:	**Hex**	**Attribute**
	01	Blink
	02	Reverse video
	04	Underline
	08	Bold
	10	Blank
	20	Visible rendition
	40	Protected
	80	Modified data

IBM 5250 Mode:	**Hex**	**Attribute**
	01	Modified data
	0E	Field type:
		00 Alphanumeric
		02 Alphabetic
		04 Numeric shift
		06 Numeric only
		08 Reserved
		0A Digits only
		0C Magnetic stripe reader data only
		0E Signed numeric
	10	High intensity
	20	Protected
	40	Displayed
	80	Field attribute

ICL 7561 Mode:	**Hex**	**Attribute**
	01	Blink
	02	Reverse video
	04	Underline
	08	Bold
	10	Blank
	20	Field attribute
	40	Protected
	80	Highlight attribute set

Formatted Field & Presentation Space

The **Query Formatted Field Attribute** (114) and **Copy Formatted Presentation Space** (105) functions are used to provide a comprehensive description of the field attribute at the specified position in the presentation space. The returned data string consists of four bytes which provide the following information:

Byte 1: Character displayed (or Viewdata special character).
Byte 2: Foreground and background colour index.
Byte 3: Generally indicates the display attribute (underline, flashing, etc.).
Byte 4: Generally indicates field formatting.

The following sections list the values applicable for each byte and their meaning.

Byte 1: Character Displayed

The first byte generally indicates the character displayed at the specified PS position. Most emulations use the ISO Latin-1 character set, exceptions are as follows:

HP700/92 Emulation:	HP Roman 8 character set
IBM 3270 Emulation:	EBCDIC character set
IBM 5250 Emulation:	EBCDIC character set
Viewdata Mode:	ASCII character set

Viewdata mode supports the following special characters in addition to the ASCII set:

Hex	Meaning	Hex	Meaning
01	Begin Alpha	18	Conceal
07	End Alpha	19	Contiguous Graphics
08	Begin Flashing	1A	Separated Graphics
09	Begin Steady	1C	Black
11	Begin Graphic	1D	New
17	End Graphic	1E	Hold
0C	Normal	1F	Release
0D	Double		

Byte 2: Colour Indices

The second byte indicates the background and foreground colours for the specified PS position. The top four bits (hex nibble) specify the background colour and the bottom four bits (hex nibble) specify the foreground colour. For example, a hex value of 1B specifies a blue background and bold green foreground.

The following hex values apply to all terminal emulations except IBM 3270.

Hex	Colour	Hex	Colour
0	Black	8	Bold Black
1	Blue	9	Bold Blue
2	Red	A	Bold Red
3	Green	B	Bold Green
4	Mauve	C	Bold Mauve
5	Cyan	D	Bold Cyan
6	Yellow	E	Bold Yellow
7	White	F	Bold White

The following hex values apply to the IBM 3270 terminal emulation.

Hex	Colour	Hex	Colour
0	Neutral 0	8	Black
1	Blue	9	Deep Blue
4	Green	C	Pale Green
5	Turquoise	D	Pale Turquoise
6	Yellow	E	Grey
7	Neutral 7	F	White

Byte 3: Display Attribute

The third byte generally indicates the display attribute (underline, flashing, etc.). The meaning of the byte returned depends on the terminal emulation.

VT, DG410, AT&T4410 & Stratus V102 Modes:

Hex	Display Attribute
01	Blink
02	Reverse video
04	Underline
08	Bold
10	Blank
40	Mask for erasable characters

HP700/92 Mode:

Hex	Display Attribute
01	Blink
02	Reverse video
04	Underline
08	Half intensity
10	Security
20	Line drawing
40	Unprotected
80	Protected

TA6530 Mode:	Hex	Display Attribute
	01	Half intensity
	02	Blink
	04	Reverse video
	08	Blank
	10	Underline
	20	Visual attribute start
	40	Field start

Wyse Modes:	Hex	Display Attribute
	01	Blink
	02	Reverse video
	04	Underline
	08	Half intensity
	10	Blank
	20	Erasable
	40	Field start

IBM 3270 Mode:	Hex	Display Attribute
	01	Blink
	02	Reverse video
	04	Underline
	07	Highlight mask

IBM 3151 Mode:	Hex	Display Attribute
	01	Blink
	02	Reverse video
	04	Underline
	08	Bold
	10	Blank
	20	Visible rendition
	40	Protected
	80	Modified data

IBM 5250 Mode:	Hex	Display Attribute
	01	Reverse video
	02	High intensity
	04	Underline
	08	Blink
	10	Column separator
	20	Visual attribute start
	40	Extended character set
	80	Input position

ICL 7561 Mode:	Hex	Display Attribute
	01	Blink
	02	Reverse video
	04	Underline
	08	Bold
	10	Blank
	20	Field attribute
	40	Protected
	80	Highlight attribute set

Byte 4: Field Formatting

The fourth byte generally indicates field formatting. The meaning of the byte returned depends on the terminal emulation.

VT, DG410, AT&T4410 & Stratus V102 Modes:

	Hex	Display Attribute
	07	Character font selection mask
	08	Double width
	10	Double height top
	20	Double height bottom
	80	Copy/cut/paste attribute

HP700/92 Mode:	Hex	Field Formatting
	01	Active row
	02	Start font
	04	Start protected/unprotected field
	08	Start video attribute field
	10	Start security field
	20	International font
	80	Copy/cut/paste attribute

TA6530 Mode:	Hex	Field Formatting
	01	Device type mask
	0E	Data type mask
	10	Auto tab disable
	20	Protected field
	40	Upshift
	80	Copy/cut/paste attribute

Wyse Modes:	Hex	Field Formatting
	80	Copy/cut/paste attribute

IBM 3270 Mode:	Hex	Field Formatting
	01	Modified data
	0C	Display/select/intensity mask
	10	Numeric field
	20	Protected field
	40	Field start
	80	Copy/cut/paste attribute

IBM 3151 Mode:	Hex	Field Formatting
	08	Field attribute
	10	Numeric field
	20	Default field attribute
	80	Copy/cut/paste attribute

IBM 5250 Mode:	Hex	Display Attribute
	07	Field type mask
	08	Modified data
	10	DUP enable
	20	Bypass field
	40	Field start
	80	Copy/cut/paste attribute

ICL 7561 Mode:	Hex	Display Attribute
	01	Font
	02	Word wrap
	04	Modified field
	08	Numeric validation
	10	Alpha validation
	20	Space validation
	40	Auto tab attribute
	80	Copy/cut/paste attribute

B

Character Sets

This appendix shows some of the tables of characters that are supported by TeemTalk.

Introduction

Each character set consists of a series of control characters and displayable characters. Displayable characters are alphanumeric, symbolic or graphic characters that can be displayed on the screen or printed by a hardcopy device. Control characters enable the terminal emulation or the printer to perform specific tasks, such as a line feed or carriage return. These will be actioned when received from the host or when TeemTalk is in local mode and they are entered from the keyboard.

> *Note: When the **Display Controls** option in the **Terminal Settings** dialog box is selected, a representation of most control characters received will be displayed on the screen instead of actioned.*

To enter a control character from the keyboard, first find the displayable character equivalent by adding 64 to the decimal value of the control character in the relevant character set table. For example, the control character **CR** (carriage return) has a decimal value of 13. Adding 64 makes 77 which is the decimal value of the displayable character **M**. When the **Ctrl** (control) key is held down and **Shift** + **M** is pressed, this will generate a **CR** code in local mode.

Some setup options require you to specify one or more control characters. A control character can be specified by typing **^** to represent the **Ctrl** key, immediately followed by the displayable character equivalent of the control character as described in the previous paragraph. For example, **^M**, represents **Ctrl** + **M**, which generates the control character **CR**.

Another way of specifying control characters is by entering the decimal value of the ASCII character. Decimal values are entered as three-digit numbers immediately preceded by an underscore character. Values with only two digits must be preceded by a zero. For example, the decimal value of **CR** is 13, so this would be entered as **_013**.

ASCII CHARACTER SET
(Multinational 7 Bit)

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$		$0\,0\,0\,1$		$0\,0\,1\,0$		$0\,0\,1\,1$		$0\,1\,0\,0$		$0\,1\,0\,1$		$0\,1\,1\,0$		$0\,1\,1\,1$	
0	0000	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	0001	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	0010	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	0011	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	0100	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	0101	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	0110	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	0111	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	1000	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	1001	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	1010	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	1011	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	1100	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	1101	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	1110	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	1111	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

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

The ASCII (American Standard Code for Information Interchange) character set will be selected when the **Keyboard Nationality** option in the **Emulation Settings** dialog box is set to **North American**, or the **Character Set Mode** option is set to **Multinational**. This table forms the first half of the Multinational character set, the second half of which may be the **DEC Additional** or one of the **ISO Latin Additional** sets (as determined by the **Preferred Char. Set** option in the **Terminal Settings** dialog box).

NATIONAL REPLACEMENT CHARACTERS

BINARY BIT 8	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	1	1	1	1	1	1
6	1	0	0	0	1	1	1	1	1	1	1	1
5	0	0	1	1	0	0	1	1	1	1	1	1
4	0	0	1	1	1	1	0	0	1	1	1	1
3	0	0	0	0	1	1	1	1	0	0	0	0
2	1	0	1	1	0	0	1	1	1	1	0	0
1	1	0	1	0	1	0	1	0	1	0	1	0
OCTAL	43	100	133	134	135	136	137	140	173	174	175	176
DECIMAL	35	64	91	92	93	94	95	96	123	124	125	126
HEXADECIMAL	23	40	5B	5C	5D	5E	5F	60	7B	7C	7D	7E
ASCII	#	@	[	\	]	^	_	`	{	|	}	~
British	£	@	[	\	]	^	_	`	{	|	}	~
Canadian	#	à	â	ç	ê	î	_	ô	é	ù	è	û
Danish Norwegian	#	Ä	Æ	Ø	Å	Ü	_	ä	æ	ø	å	ü
Dutch	£	¾	ij	½	|	^	_	`	¨	fl	¼	´
Finnish	#	@	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
French Belgian	£	à	°	ç	§	^	_	`	é	ù	è	¨
German	#	§	Ä	Ö	Ü	^	_	`	ä	ö	ü	ß
Italian	£	§	°	ç	é	^	_	ù	à	ò	è	ì
Portuguese	#	@	Ã	Ç	Õ	^	_	`	ã	ç	õ	~
Spanish	£	§	¡	Ñ	¿	^	_	`	°	ñ	ç	~
Swedish	#	É	Ä	Ö	Å	Ü	_	é	ä	ö	å	ü
Swiss French Swiss German	ù	à	é	ç	ê	î	è	ô	ä	ö	ü	û

This table shows the characters that replace certain ASCII characters when the **Character Set Mode** option in the **Emulation Settings** dialog box is set to **National**. The national character set consists of the ASCII set with the changed characters listed on the line for the selected keyboard nationality.

DEC ADDITIONAL CHARACTER SET
(Multinational 8 Bit)

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0_0$		$1\,0\,0_1$		$1\,0\,1_0$		$1\,0\,1_1$		$1\,1\,0_0$		$1\,1\,0_1$		$1\,1\,1_0$		$1\,1\,1_1$		8 7 BITS 6 5 4 3 2 1	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	0000	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	0001	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	0010	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	0011	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	0100	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	0101	5
SSA	206 134 86	**SPA**	226 150 96	ı	246 166 A6	¶	266 182 B6	**Æ**	306 198 C6	**Ö**	326 214 D6	**æ**	346 230 E6	**ö**	366 246 F6	0110	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	0111	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	1000	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	1001	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	1010	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	1011	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	1100	12
RI	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	1101	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	1110	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	1111	15

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This is one of three possible second halves of the Multinational character set (the first half is the ASCII character set and the other possible second halves are the ISO Latin-1 and ISO Latin-2 Additional character sets). These characters may be generated when TeemTalk is in **VT500 7** or **8** bit mode, the **Character Set Mode** option in the **Emulation Settings** dialog box is set to **Multinational**, and the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **DEC-MCS**.

ISO LATIN-1 ADDITIONAL CHARACTER SET
(Multinational 8 Bit)

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 8 7 6 5 / 4 3 2 1	ROW
	200 128 80	DCS	220 144 90	NBSP	240 160 A0	°	260 176 B0	À	300 192 C0	Ð	320 208 D0	à	340 224 E0	ð	360 240 F0	0000	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	0001	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	0010	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	0011	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	0100	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	0101	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	0110	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	×	327 215 D7	ç	347 231 E7	÷	367 247 F7	0111	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	1000	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	1001	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	1010	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	1011	11
PLU	214 140 8C	ST	234 156 9C	¬	254 172 AC	$^1/_4$	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1100	12
RI	215 141 8D	OSC	235 157 9D	—	255 173 AD	$^1/_2$	275 189 BD	Í	315 205 CD	Ý	335 221 DD	í	355 237 ED	ý	375 253 FD	1101	13
SS2	216 142 8E	PM	236 158 9E	®	256 174 AE	$^3/_4$	276 190 BE	Î	316 206 CE	Þ	336 222 DE	î	356 238 EE	þ	376 254 FE	1110	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	ÿ	377 255 FF	1111	15

KEY: ESC 33 (OCTAL) / 27 (DECIMAL) / 1B (HEXADECIMAL)

This is one of three possible second halves of the Multinational character set (the first half is the ASCII character set and the other possible second halves are the DEC Additional and ISO Latin-2 character sets). These characters may be generated when TeemTalk is in **VT500 7** or **8** bit mode, the **Character Set Mode** option in the **Emulation Settings** dialog box is set to **Multinational**, and the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **ISO Latin-1**.

ISO LATIN-2 ADDITIONAL CHARACTER SET
(Multinational 8 Bit)

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0_0$		$1\,0\,0_1$		$1\,0\,1_0$		$1\,0\,1_1$		$1\,1\,0_0$		$1\,1\,0_1$		$1\,1\,1_0$		$1\,1\,1_1$		8 7 BITS 6 5 4 3 2 1	ROW
	200 128 80	DCS	220 144 90	NBSP	240 160 A0	°	260 176 B0	Ŕ	300 192 C0	Đ	320 208 D0	ŕ	340 224 E0	ð	360 240 F0	0000	0
	201 129 81	PU1	221 145 91	Ą	241 161 A1	ą	261 177 B1	Á	301 193 C1	Ń	321 209 D1	á	341 225 E1	ń	361 241 F1	0001	1
	202 130 82	PU2	222 146 92	˘	242 162 A2	˛	262 178 B2	Â	302 194 C2	Ň	322 210 D2	â	342 226 E2	ň	362 242 F2	0010	2
	203 131 83	STS	223 147 93	Ł	243 163 A3	ł	263 179 B3	Ă	303 195 C3	Ó	323 211 D3	ă	343 227 E3	ó	363 243 F3	0011	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	0100	4
NEL	205 133 85	MW	225 149 95	Ľ	245 165 A5	ľ	265 181 B5	Ĺ	305 197 C5	Ő	325 213 D5	ĺ	345 229 E5	ő	365 245 F5	0101	5
SSA	206 134 86	SPA	226 150 96	Ś	246 166 A6	ś	266 182 B6	Ć	306 198 C6	Ö	326 214 D6	ć	346 230 E6	ö	366 246 F6	0110	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	ˇ	267 183 B7	Ç	307 199 C7	×	327 215 D7	ç	347 231 E7	÷	367 247 F7	0111	7
HTS	210 136 88		230 152 98	¨	250 168 A8	ş	270 184 B8	Č	310 200 C8	Ř	330 216 D8	č	350 232 E8	ř	370 248 F8	1000	8
HTJ	211 137 89		231 153 99	Š	251 169 A9	š	271 185 B9	É	311 201 C9	Ů	331 217 D9	é	351 233 E9	ů	371 249 F9	1001	9
VTS	212 138 8A		232 154 9A	Ş	252 170 AA	ş	272 186 BA	Ę	312 202 CA	Ú	332 218 DA	ę	352 234 EA	ú	372 250 FA	1010	10
PLD	213 139 8B	CSI	233 155 9B	Ť	253 171 AB	ť	273 187 BB	Ë	313 203 CB	Ű	333 219 DB	ë	353 235 EB	ű	373 251 FB	1011	11
PLU	214 140 8C	ST	234 156 9C	Ź	254 172 AC	ź	274 188 BC	E	314 204 CC	Ü	334 220 DC	ě	354 236 EC	ü	374 252 FC	1100	12
RI	215 141 8D	OSC	235 157 9D	SHY	255 173 AD	˝	275 189 BD	Í	315 205 CD	Ý	335 221 DD	í	355 237 ED	ý	375 253 FD	1101	13
SS2	216 142 8E	PM	236 158 9E	Ž	256 174 AE	ž	276 190 BE	Î	316 206 CE	Ţ	336 222 DE	î	356 238 EE	ţ	376 254 FE	1110	14
SS3	217 143 8F	APC	237 159 9F	Ż	257 175 AF	ż	277 191 BF	Ď	317 207 CF	ß	337 223 DF	ď	357 239 EF	˙	377 255 FF	1111	15

KEY: ESC | 33 (OCTAL) 27 (DECIMAL) 1B (HEXADECIMAL)

This is one of three possible second halves of the Multinational character set (the first half is the ASCII character set and the other possible second halves are the DEC Additional and ISO Latin-1 character sets). These characters may be generated when TeemTalk is in **VT500 7** or **8** bit mode, the **Character Set Mode** option in the **Emulation Settings** dialog box is set to **Multinational**, and the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **ISO Latin-2**.

ANSI 437 CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0\,0$		$1\,0\,0\,1$		$1\,0\,1\,0$		$1\,0\,1\,1$		$1\,1\,0\,0$		$1\,1\,0\,1$		$1\,1\,1\,0$		$1\,1\,1\,1$		8 7 BITS 6 5 4 3 2 1	ROW
Ç	200 128 80	É	220 144 90	á	240 160 A0		260 176 B0		300 192 C0		320 208 D0	α	340 224 E0	≡	360 240 F0	0000	0
ü	201 129 81	æ	221 145 91	í	241 161 A1		261 177 B1		301 193 C1		321 209 D1	ß	341 225 E1	±	361 241 F1	0001	1
é	202 130 82	Æ	222 146 92	ó	242 162 A2		262 178 B2		302 194 C2		322 210 D2	Γ	342 226 E2	≥	362 242 F2	0010	2
â	203 131 83	ô	223 147 93	ú	243 163 A3		263 179 B3		303 195 C3		323 211 D3	π	343 227 E3	≤	363 243 F3	0011	3
ä	204 132 84	ö	224 148 94	ñ	244 164 A4		264 180 B4		304 196 C4		324 212 D4	Σ	344 228 E4	⌠	364 244 F4	0100	4
à	205 133 85	ò	225 149 95	Ñ	245 165 A5		265 181 B5		305 197 C5		325 213 D5	σ	345 229 E5	⌡	365 245 F5	0101	5
å	206 134 86	û	226 150 96	ª	246 166 A6		266 182 B6		306 198 C6		326 214 D6	µ	346 230 E6	÷	366 246 F6	0110	6
ç	207 135 87	ù	227 151 97	º	247 167 A7		267 183 B7		307 199 C7		327 215 D7	τ	347 231 E7	≈	367 247 F7	0111	7
ê	210 136 88	ÿ	230 152 98	¿	250 168 A8		270 184 B8		310 200 C8		330 216 D8	Φ	350 232 E8	°	370 248 F8	1000	8
ë	211 137 89	Ö	231 153 99	⌐	251 169 A9		271 185 B9		311 201 C9		331 217 D9	Θ	351 233 E9	•	371 249 F9	1001	9
è	212 138 8A	Ü	232 154 9A	¬	252 170 AA		272 186 BA		312 202 CA		332 218 DA	Ω	352 234 EA	•	372 250 FA	1010	10
ï	213 139 8B	¢	233 155 9B	½	253 171 AB		273 187 BB		313 203 CB		333 219 DB	δ	353 235 EB	√	373 251 FB	1011	11
î	214 140 8C	£	234 156 9C	¼	254 172 AC		274 188 BC		314 204 CC		334 220 DC	∞	354 236 EC	n	374 252 FC	1100	12
ì	215 141 8D	¥	235 157 9D	¡	255 173 AD		275 189 BD		315 205 CD		335 221 DD	φ	355 237 ED	2	375 253 FD	1101	13
Ä	216 142 8E	₧	236 158 9E	«	256 174 AE		276 190 BE		316 206 CE		336 222 DE	∈	356 238 EE	■	376 254 FE	1110	14
Å	217 143 8F	ƒ	237 159 9F	»	257 175 AF		277 191 BF		317 207 CF		337 223 DF	∩	357 239 EF	BLANK FF	377 255 FF	1111	15

KEY: Ñ | 245 OCTAL | 165 DECIMAL | A5 HEXADECIMAL

This table forms the second half of the ANSI 437 character set, the first half being the ASCII character set.

The ANSI 437 character set is used when the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **Ansi** and the **ANSI Code Page** option is set to **437**.

ANSI 850 CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1_0 0_0$		$1_0 0_1$		$1_0 1_0$		$1_0 1_1$		$1_1 0_0$		$1_1 0_1$		$1_1 1_0$		$1_1 1_1$		8 7 BITS 6 5 4 3 2 1	ROW
Ç	200 128 80	É	220 144 90	á	240 160 A0	▒	260 176 B0	└	300 192 C0	ð	320 208 D0	Ó	340 224 E0	—	360 240 F0	0000	0
ü	201 129 81	æ	221 145 91	í	241 161 A1	▒	261 177 B1	┴	301 193 C1	Ð	321 209 D1	ß	341 225 E1	±	361 241 F1	0001	1
é	202 130 82	Æ	222 146 92	ó	242 162 A2	▓	262 178 B2	┬	302 194 C2	Ê	322 210 D2	Ô	342 226 E2	‗	362 242 F2	0010	2
â	203 131 83	ô	223 147 93	ú	243 163 A3	│	263 179 B3	├	303 195 C3	Ë	323 211 D3	Ò	343 227 E3	¾	363 243 F3	0011	3
ä	204 132 84	ö	224 148 94	ñ	244 164 A4	┤	264 180 B4	─	304 196 C4	È	324 212 D4	õ	344 228 E4	¶	364 244 F4	0100	4
à	205 133 85	ò	225 149 95	Ñ	245 165 A5	Á	265 181 B5	┼	305 197 C5	ı	325 213 D5	Õ	345 229 E5	§	365 245 F5	0101	5
å	206 134 86	û	226 150 96	ª	246 166 A6	Â	266 182 B6	ã	306 198 C6	Í	326 214 D6	µ	346 230 E6	÷	366 246 F6	0110	6
ç	207 135 87	ù	227 151 97	º	247 167 A7	À	267 183 B7	Ã	307 199 C7	Î	327 215 D7	Þ	347 231 E7	¸	367 247 F7	0111	7
ê	210 136 88	ÿ	230 152 98	¿	250 168 A8	©	270 184 B8	╚	310 200 C8	Ï	330 216 D8	Þ	350 232 E8	°	370 248 F8	1000	8
ë	211 137 89	Ö	231 153 99	®	251 169 A9	╣	271 185 B9	╔	311 201 C9	┘	331 217 D9	Ú	351 233 E9	¨	371 249 F9	1001	9
è	212 138 8A	Ü	232 154 9A	¬	252 170 AA	║	272 186 BA	╩	312 202 CA	┌	332 218 DA	Û	352 234 EA	•	372 250 FA	1010	10
ï	213 139 8B	ø	233 155 9B	½	253 171 AB	╗	273 187 BB	╦	313 203 CB	▓	333 219 DB	Ù	353 235 EB	1	373 251 FB	1011	11
î	214 140 8C	£	234 156 9C	¼	254 172 AC	╝	274 188 BC	╠	314 204 CC	▒	334 220 DC	ý	354 236 EC	3	374 252 FC	1100	12
ì	215 141 8D	Ø	235 157 9D	¡	255 173 AD	¢	275 189 BD	═	315 205 CD	¦	335 221 DD	Ý	355 237 ED	2	375 253 FD	1101	13
Ä	216 142 8E	×	236 158 9E	«	256 174 AE	¥	276 190 BE	╬	316 206 CE	Ì	336 222 DE	¯	356 238 EE	▮	376 254 FE	1110	14
Å	217 143 8F	ƒ	237 159 9F	»	257 175 AF	┐	277 191 BF	¤	317 207 CF	▀	337 223 DF	´	357 239 EF	BLANK FF	377 255 FF	1111	15

KEY: Ñ | 245 / 165 / A5 — OCTAL / DECIMAL / HEXADECIMAL

This table forms the second half of the ANSI 850 character set, the first half being the ASCII character set.

The ANSI 850 character set is used when the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **Ansi** and the **ANSI Code Page** option is set to **850**.

ANSI 858 CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0\,0$		$1\,0\,0\,1$		$1\,0\,1\,0$		$1\,0\,1\,1$		$1\,1\,0\,0$		$1\,1\,0\,1$		$1\,1\,1\,0$		$1\,1\,1\,1$		8 7 BITS 6 5 4 3 2 1	ROW
Ç	200 128 80	É	220 144 90	á	240 160 A0	▒	260 176 B0	└	300 192 C0	ð	320 208 D0	Ó	340 224 E0	—	360 240 F0	0000	0
ü	201 129 81	æ	221 145 91	í	241 161 A1	▒	261 177 B1	┴	301 193 C1	Ð	321 209 D1	ß	341 225 E1	±	361 241 F1	0001	1
é	202 130 82	Æ	222 146 92	ó	242 162 A2	▓	262 178 B2	┬	302 194 C2	Ê	322 210 D2	Ô	342 226 E2	‗	362 242 F2	0010	2
â	203 131 83	ô	223 147 93	ú	243 163 A3	│	263 179 B3	├	303 195 C3	Ë	323 211 D3	Ò	343 227 E3	¾	363 243 F3	0011	3
ä	204 132 84	ö	224 148 94	ñ	244 164 A4	┤	264 180 B4	─	304 196 C4	È	324 212 D4	õ	344 228 E4	¶	364 244 F4	0100	4
à	205 133 85	ò	225 149 95	Ñ	245 165 A5	Á	265 181 B5	┼	305 197 C5	€	325 213 D5	Õ	345 229 E5	§	365 245 F5	0101	5
å	206 134 86	û	226 150 96	ª	246 166 A6	Â	266 182 B6	ã	306 198 C6	Í	326 214 D6	µ	346 230 E6	÷	366 246 F6	0110	6
ç	207 135 87	ù	227 151 97	º	247 167 A7	À	267 183 B7	Ã	307 199 C7	Î	327 215 D7	Þ	347 231 E7	¸	367 247 F7	0111	7
ê	210 136 88	ÿ	230 152 98	¿	250 168 A8	©	270 184 B8	╚	310 200 C8	Ï	330 216 D8	Þ	350 232 E8	°	370 248 F8	1000	8
ë	211 137 89	Ö	231 153 99	®	251 169 A9	╣	271 185 B9	╔	311 201 C9	┘	331 217 D9	Ú	351 233 E9	¨	371 249 F9	1001	9
è	212 138 8A	Ü	232 154 9A	¬	252 170 AA	║	272 186 BA	╩	312 202 CA	┌	332 218 DA	Û	352 234 EA	•	372 250 FA	1010	10
ï	213 139 8B	ø	233 155 9B	½	253 171 AB	╗	273 187 BB	╦	313 203 CB	▉	333 219 DB	Ù	353 235 EB	1	373 251 FB	1011	11
î	214 140 8C	£	234 156 9C	¼	254 172 AC	╝	274 188 BC	╠	314 204 CC	▀	334 220 DC	ý	354 236 EC	3	374 252 FC	1100	12
ì	215 141 8D	Ø	235 157 9D	¡	255 173 AD	¢	275 189 BD	═	315 205 CD	¦	335 221 DD	Ý	355 237 ED	2	375 253 FD	1101	13
Ä	216 142 8E	×	236 158 9E	«	256 174 AE	¥	276 190 BE	╬	316 206 CE	Ì	336 222 DE	¯	356 238 EE	▉	376 254 FE	1110	14
Å	217 143 8F	ƒ	237 159 9F	»	257 175 AF	┐	277 191 BF	¤	317 207 CF	▉	337 223 DF	´	357 239 EF	BLANK FF	377 255 FF	1111	15

KEY: Ñ | 245 | 165 | A5 — OCTAL, DECIMAL, HEXADECIMAL

This table forms the second half of the ANSI 858 character set, the first half being the ASCII character set.

The ANSI 850 character set is used when the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **Ansi** and the **ANSI Code Page** option is set to **858**.

ANSI 1250 CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0\,0$		$1\,0\,0\,1$		$1\,0\,1\,0$		$1\,0\,1\,1$		$1\,1\,0\,0$		$1\,1\,0\,1$		$1\,1\,1\,0$		$1\,1\,1\,1$		8 7 BITS 6 5 4 3 2 1	ROW
	200 128 80	DCS	220 144 90	NBSP	240 160 A0	°	260 176 B0	Ŕ	300 192 C0	Đ	320 208 D0	ŕ	340 224 E0	ð	360 240 F0	0000	0
	201 129 81	PU1	221 145 91	Ą	241 161 A1	ą	261 177 B1	Á	301 193 C1	Ń	321 209 D1	á	341 225 E1	ń	361 241 F1	0001	1
	202 130 82	PU2	222 146 92	˘	242 162 A2	˛	262 178 B2	Â	302 194 C2	Ň	322 210 D2	â	342 226 E2	ň	362 242 F2	0010	2
	203 131 83	STS	223 147 93	Ł	243 163 A3	ł	263 179 B3	Ă	303 195 C3	Ő	323 211 D3	ă	343 227 E3	ő	363 243 F3	0011	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	0100	4
NEL	205 133 85	MW	225 149 95	Ľ	245 165 A5	ľ	265 181 B5	Ĺ	305 197 C5	Ő	325 213 D5	ĺ	345 229 E5	ő	365 245 F5	0101	5
SSA	206 134 86	SPA	226 150 96	Ś	246 166 A6	ś	266 182 B6	Ć	306 198 C6	Ö	326 214 D6	ć	346 230 E6	ö	366 246 F6	0110	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	ˇ	267 183 B7	Ç	307 199 C7	×	327 215 D7	ç	347 231 E7	÷	367 247 F7	0111	7
HTS	210 136 88		230 152 98	¨	250 168 A8	¸	270 184 B8	Č	310 200 C8	Ř	330 216 D8	č	350 232 E8	ř	370 248 F8	1000	8
HTJ	211 137 89		231 153 99	Š	251 169 A9	š	271 185 B9	É	311 201 C9	Ů	331 217 D9	é	351 233 E9	ů	371 249 F9	1001	9
VTS	212 138 8A		232 154 9A	Ş	252 170 AA	ş	272 186 BA	Ę	312 202 CA	Ú	332 218 DA	ę	352 234 EA	ú	372 250 FA	1010	10
PLD	213 139 8B	CSI	233 155 9B	Ť	253 171 AB	t’	273 187 BB	Ë	313 203 CB	Ű	333 219 DB	ë	353 235 EB	ű	373 251 FB	1011	11
PLU	214 140 8C	ST	234 156 9C	Ź	254 172 AC	ź	274 188 BC	Ě	314 204 CC	Ü	334 220 DC	ě	354 236 EC	ü	374 252 FC	1100	12
RI	215 141 8D	OSC	235 157 9D	SHY	255 173 AD	˝	275 189 BD	Í	315 205 CD	Ý	335 221 DD	í	355 237 ED	ý	375 253 FD	1101	13
SS2	216 142 8E	PM	236 158 9E	Ž	256 174 AE	ž	276 190 BE	Î	316 206 CE	Ţ	336 222 DE	î	356 238 EE	ţ	376 254 FE	1110	14
SS3	217 143 8F	APC	237 159 9F	Ż	257 175 AF	ż	277 191 BF	Ď	317 207 CF	ß	337 223 DF	d’	357 239 EF	·	377 255 FF	1111	15

KEY: ESC 33 27 1B — OCTAL / DECIMAL / HEXADECIMAL

This table forms the second half of the ANSI 1250 character set, the first half being the ASCII character set.

The ANSI 1250 character set is used when the **Preferred Char. Set** option in the **Terminal Settings** dialog box is set to **Ansi** and the **ANSI Code Page** option is set to **1250**.

DEC LINE DRAWING CHARACTER SET

COLUMN		0			1			2			3			4			5			6			7		
ROW	8 7 BITS 6 5 4 3 2 1	$0_0 0_0 0_0$			$0_0 0_0 1_1$			$0_0 1_1 0_0$			$0_0 1_1 1_1$			$0_1 0_0 0_0$			$0_1 0_1 0_1$			$0_1 1_0 0_0$			$0_1 1_1 1_1$		
0	0000	NUL	0 0 0	0	DLE	20 16 10	16	SP	40 32 20	32	0	60 48 30	48	@	100 64 40	64	P	120 80 50	80	◆	140 96 60	96	—	160 112 70	112
1	0001	SOH	1 1 1	1	DC1 XON	21 17 11	17	!	41 33 21	33	1	61 49 31	49	A	101 65 41	65	Q	121 81 51	81	▒	141 97 61	97	—	161 113 71	113
2	0010	STX	2 2 2	2	DC2	22 18 12	18	"	42 34 22	34	2	62 50 32	50	B	102 66 42	66	R	122 82 52	82	H_T	142 98 62	98	—	162 114 72	114
3	0011	ETX	3 3 3	3	DC3 XOFF	23 19 13	19	#	43 35 23	35	3	63 51 33	51	C	103 67 43	67	S	123 83 53	83	F_F	143 99 63	99	—	163 115 73	115
4	0100	EOT	4 4 4	4	DC4	24 20 14	20	$	44 36 24	36	4	64 52 34	52	D	104 68 44	68	T	124 84 54	84	C_R	144 100 64	100	├	164 116 74	116
5	0101	ENQ	5 5 5	5	NAK	25 21 15	21	%	45 37 25	37	5	65 53 35	53	E	105 69 45	69	U	125 85 55	85	L_F	145 101 65	101	┤	165 117 75	117
6	0110	ACK	6 6 6	6	SYN	26 22 16	22	&	46 38 26	38	6	66 54 36	54	F	106 70 46	70	V	126 86 56	86	°	146 102 66	102	┴	166 118 76	118
7	0111	BEL	7 7 7	7	ETB	27 23 17	23	´	47 39 27	39	7	67 55 37	55	G	107 71 47	71	W	127 87 57	87	±	147 103 67	103	┬	167 119 77	119
8	1000	BS	10 8 8	8	CAN	30 24 18	24	(	50 40 28	40	8	70 56 38	56	H	110 72 48	72	X	130 88 58	88	N_L	150 104 68	104	│	170 120 78	120
9	1001	HT	11 9 9	9	EM	31 25 19	25	)	51 41 29	41	9	71 57 39	57	I	111 73 49	73	Y	131 89 59	89	V_T	151 105 69	105	≤	171 121 79	121
10	1010	LF	12 10 A	10	SUB	32 26 1A	26	*	52 42 2A	42	:	72 58 3A	58	J	112 74 4A	74	Z	132 90 5A	90	┘	152 106 6A	106	≥	172 122 7A	122
11	1011	VT	13 11 B	11	ESC	33 27 1B	27	+	53 43 2B	43	;	73 59 3B	59	K	113 75 4B	75	[	133 91 5B	91	┐	153 107 6B	107	π	173 123 7B	123
12	1100	FF	14 12 C	12	FS	34 28 1C	28	,	54 44 2C	44	<	74 60 3C	60	L	114 76 4C	76	\	134 92 5C	92	┌	154 108 6C	108	≠	174 124 7C	124
13	1101	CR	15 13 D	13	GS	35 29 1D	29	-	55 45 2D	45	=	75 61 3D	61	M	115 77 4D	77	]	135 93 5D	93	└	155 109 6D	109	£	175 125 7D	125
14	1110	SO	16 14 E	14	RS	36 30 1E	30	.	56 46 2E	46	>	76 62 3E	62	N	116 78 4E	78	^	136 94 5E	94	┼	156 110 6E	110	·	176 126 7E	126
15	1111	SI	17 15 F	15	US	37 31 1F	31	/	57 47 2F	47	?	77 63 3F	63	O	117 79 4F	79		137 95 5F	95	¯	157 111 6F	111	DEL	177 127 7F	127

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This is a special DEC character set which is used by some applications.

DEC TECHNICAL CHARACTER SET (7 Bit)

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_0 0_0$		$0_0 1_1 1_1$		$0_1 0_0 0_0$		$0_1 0_1 1_1$		$0_1 1_0 0_0$		$0_1 1_1 1_1$	
0	0000	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	⎧	60 48 30	∴	100 64 40	∏	120 80 50	⌝	140 96 60	π	160 112 70
1	0001	SOH	1 1 1	DC1 XON	21 17 11	⌜	41 33 21	⎡	61 49 31	α	101 65 41	Ψ	121 81 51	α	141 97 61	ψ	161 113 71
2	0010	STX	2 2 2	DC2	22 18 12	⌐	42 34 22	∠	62 50 32	∞	102 66 42		122 82 52	ß	142 98 62	ρ	162 114 72
3	0011	ETX	3 3 3	DC3 XOFF	23 19 13	⎯	43 35 23	⎢	63 51 33	÷	103 67 43	Σ	123 83 53	χ	143 99 63	σ	163 115 73
4	0100	EOT	4 4 4	DC4	24 20 14	⌡	44 36 24	⎣	64 52 34	Δ	104 68 44		124 84 54	δ	144 100 64	τ	164 116 74
5	0101	ENQ	5 5 5	NAK	25 21 15	⌠	45 37 25	¬	65 53 35	∇	105 69 45		125 85 55	ε	145 101 65	ς	165 117 75
6	0110	ACK	6 6 6	SYN	26 22 16	⎮	46 38 26	⎦	66 54 36	Φ	106 70 46	√	126 86 56	φ	146 102 66	*f*	166 118 76
7	0111	BEL	7 7 7	ETB	27 23 17	⎡	47 39 27	⟩	67 55 37	Γ	107 71 47	Ω	127 87 57	γ	147 103 67	ω	167 119 77
8	1000	BS	10 8 8	CAN	30 24 18	⎣	50 40 28	⌡	70 56 38	~	110 72 48	Ξ	130 88 58	η	150 104 68	ξ	170 120 78
9	1001	HT	11 9 9	EM	31 25 19	⎤	51 41 29		71 57 39	≃	111 73 49	Υ	131 89 59	ι	151 105 69	υ	171 121 79
10	1010	LF	12 10 A	SUB	32 26 1A	⌋	52 42 2A		72 58 3A	Θ	112 74 4A	⊂	132 90 5A	θ	152 106 6A	ζ	172 122 7A
11	1011	VT	13 11 B	ESC	33 27 1B	⎛	53 43 2B		73 59 3B	×	113 75 4B	⊃	133 91 5B	κ	153 107 6B	←	173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C	⎝	54 44 2C	≤	74 60 3C	Λ	114 76 4C	∩	134 92 5C	λ	154 108 6C	↑	174 124 7C
13	1101	CR	15 13 D	GS	35 29 1D	⎞	55 45 2D	≠	75 61 3D	⇔	115 77 4D	∪	135 93 5D		155 109 6D	→	175 125 7D
14	1110	SO	16 14 E	RS	36 30 1E	⎠	56 46 2E	≥	76 62 3E	⇒	116 78 4E	∧	136 94 5E	ν	156 110 6E	↓	176 126 7E
15	1111	SI	17 15 F	US	37 31 1F	⎟	57 47 2F	∫	77 63 3F	≡	117 79 4F	∨	137 95 5F	∂	157 111 6F	DEL	177 127 7F

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This is a special DEC character set used by some applications.

DEC TECHNICAL CHARACTER SET (8 Bit)

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0_0$		$1\,0\,0_1$		$1\,0\,1_0$		$1\,0\,1_1$		$1\,1\,0_0$		$1\,1\,0_1$		$1\,1\,1_0$		$1\,1\,1_1$		8 7 BITS 6 5 4 3 2 1	ROW
	200 128 80	DCS	220 144 90		240 160 A0	⎫	260 176 B0	∴	300 192 C0	∏	320 208 D0	⌝	340 224 E0	π	360 240 F0	0000	0
	201 129 81	PU1	221 145 91	⌟	241 161 A1	⎧	261 177 B1	α	301 193 C1	Ψ	321 209 D1		341 225 E1	ψ	361 241 F1	0001	1
	202 130 82	PU2	222 146 92	⌜	242 162 A2	∠	262 178 B2	∞	302 194 C2		322 210 D2	ß	342 226 E2	ρ	362 242 F2	0010	2
	203 131 83	STS	223 147 93	⎯	243 163 A3	⎩	263 179 B3	÷	303 195 C3	Σ	323 211 D3	χ	343 227 E3	σ	363 243 F3	0011	3
IND	204 132 84	CCH	224 148 94	⌠	244 164 A4	⁄	264 180 B4	Δ	304 196 C4		324 212 D4	δ	344 228 E4	τ	364 244 F4	0100	4
NEL	205 133 85	MW	225 149 95	⌡	245 165 A5	⎫	265 181 B5	∇	305 197 C5		325 213 D5	ε	345 229 E5	ʕ	365 245 F5	0101	5
SSA	206 134 86	SPA	226 150 96	⎮	246 166 A6	⎭	266 182 B6	Φ	306 198 C6	√	326 214 D6	φ	346 230 E6	*f*	366 246 F6	0110	6
ESA	207 135 87	EPA	227 151 97	⎡	247 167 A7	⟩	267 183 B7	Γ	307 199 C7	Ω	327 215 D7	γ	347 231 E7	ω	367 247 F7	0111	7
HTS	210 136 88		230 152 98	⎣	250 168 A8	ʕ	270 184 B8	∼	310 200 C8	Ξ	330 216 D8	η	350 232 E8	ξ	370 248 F8	1000	8
HTJ	211 137 89		231 153 99	⎤	251 169 A9		271 185 B9	≃	311 201 C9	Υ	331 217 D9	ι	351 233 E9	υ	371 249 F9	1001	9
VTS	212 138 8A		232 154 9A	⎦	252 170 AA		272 186 BA	Θ	312 202 CA	⊂	332 218 DA	θ	352 234 EA	ζ	372 250 FA	1010	10
PLD	213 139 8B	CSI	233 155 9B	⎛	253 171 AB		273 187 BB	×	313 203 CB	⊃	333 219 DB	κ	353 235 EB	←	373 251 FB	1011	11
PLU	214 140 8C	ST	234 156 9C	⎜	254 172 AC	≤	274 188 BC	Λ	314 204 CC	∩	334 220 DC	λ	354 236 EC	↑	374 252 FC	1100	12
RI	215 141 8D	OSC	235 157 9D	⎝	255 173 AD	≠	275 189 BD	⇔	315 205 CD	∪	335 221 DD		355 237 ED	→	375 253 FD	1101	13
SS2	216 142 8E	PM	236 158 9E	⎞	256 174 AE	≥	276 190 BE	⇒	316 206 CE	∧	336 222 DE	ν	356 238 EE	↓	376 254 FE	1110	14
SS3	217 143 8F	APC	237 159 9F	⎠	257 175 AF	∫	277 191 BF	≡	317 207 CF	∨	337 223 DF	∂	357 239 EF		377 255 FF	1111	15

KEY: ESC | 33 27 1B | OCTAL DECIMAL HEXADECIMAL

This is a special DEC character set used by some applications.

DG 410/412 WORD PROCESSING, MATH & GREEK ALPHABET CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_1 0_0$		$0_0 1_1 1_1$		$0_1 0_0 0_0$		$0_1 0_0 1_1$		$0_1 1_0 0_0$		$0_1 1_1 1_1$	
0	0000	**NUL**	0 0 0	**DLE**	20 16 10	**SP**	40 32 20	**0**	60 48 30	**!!**	100 64 40	π	120 80 50	⊢	140 96 60	**0**	160 112 70
1	0001	**SOH**	1 1 1	**DC1** **XON**	21 17 11	⌜	41 33 21	**1**	61 49 31	∝	101 65 41	ρ	121 81 51	◆	141 97 61	**1**	161 113 71
2	0010	**STX**	2 2 2	**DC2**	22 18 12	⌊	42 34 22	**2**	62 50 32	ß	102 66 42	σ	122 82 52	▶	142 98 62	**2**	162 114 72
3	0011	**ETX**	3 3 3	**DC3** **XOFF**	23 19 13	⌝	43 35 23	**3**	63 51 33	γ	103 67 43	τ	123 83 53	▷	143 99 63	**3**	163 115 73
4	0100	**EOT**	4 4 4	**DC4**	24 20 14	⌟	44 36 24	**4**	64 52 34	δ	104 68 44	υ	124 84 54	◀	144 100 64	**4**	164 116 74
5	0101	**ENQ**	5 5 5	**NAK**	25 21 15	ƒ	45 37 25	**5**	65 53 35	ε	105 69 45	φ	125 85 55	▲	145 101 65	**5**	165 117 75
6	0110	**ACK**	6 6 6	**SYN**	26 22 16	∽	46 38 26	**6**	66 54 36	ζ	106 70 46	χ	126 86 56	▼	146 102 66	**6**	166 118 76
7	0111	**BEL**	7 7 7	**ETB**	27 23 17	∂	47 39 27	**7**	67 55 37	η	107 71 47	ψ	127 87 57	☒	147 103 67	**7**	167 119 77
8	1000	**BS**	10 8 8	**CAN**	30 24 18	⌃	50 40 28	**8**	70 56 38	θ	110 72 48	ω	130 88 58	⧗	150 104 68	**8**	170 120 78
9	1001	**HT**	11 9 9	**EM**	31 25 19	∠	51 41 29	**9**	71 57 39	ι	111 73 49	Ω	131 89 59	⧗	151 105 69	**9**	171 121 79
10	1010	**LF**	12 10 A	**SUB**	32 26 1A	⌠	52 42 2A	≠	72 58 3A	Κ	112 74 4A	Δ	132 90 5A	⧖	152 106 6A	H_F	172 122 7A
11	1011	**VT**	13 11 B	**ESC**	33 27 1B	⌡	53 43 2B	⌜	73 59 3B	λ	113 75 4B	¶	133 91 5B	⧖	153 107 6B	↑	173 123 7B
12	1100	**FF**	14 12 C	**FS**	34 28 1C	⌿	54 44 2C	⇤	74 60 3C	⇊	114 76 4C	B_P	134 92 5C	◺	154 108 6C	→	174 124 7C
13	1101	**CR**	15 13 D	**GS**	35 29 1D	⌿	55 45 2D	⇤	75 61 3D	ν	115 77 4D	B_E	135 93 5D	◿	155 109 6D	←	175 125 7D
14	1110	**SO**	16 14 E	**RS**	36 30 1E	∞	56 46 2E	⇥	76 62 3E	ξ	116 78 4E	F_N	136 94 5E	◣	156 110 6E	↓	176 126 7E
15	1111	**SI**	17 15 F	**US**	37 31 1F	∝	57 47 2F	•	77 63 3F	◇	117 79 4F	F_E	137 95 5F	◢	157 111 6F	**DEL**	177 127 7F

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

DG 410/412 LINE DRAWING CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 6 5 BITS 4 3 2 1	$0\,0\,0\,0$	$0\,0\,0\,1$	$0\,0\,1\,0$	$0\,0\,1\,1$	$0\,1\,0\,0$	$0\,1\,0\,1$	$0\,1\,1\,0$	$0\,1\,1\,1$
0	0000	0 0 0	20 16 10	40 32 20	60 48 30	100 64 40	120 80 50	140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	61 49 31	101 65 41	121 81 51	141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	62 50 32	102 66 42	122 82 52	142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	63 51 33	103 67 43	123 83 53	143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	64 52 34	104 68 44	124 84 54	144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	65 53 35	105 69 45	125 85 55	145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	66 54 36	106 70 46	126 86 56	146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	67 55 37	107 71 47	127 87 57	147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	70 56 38	110 72 48	130 88 58	150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	71 57 39	111 73 49	131 89 59	151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY: ÷ | 75 61 3D | OCTAL DECIMAL HEXADECIMAL

PT250 ADDITIONAL CHARACTER SET

8	9	10	11	12	13	14	15	COLUMN
$1_0 0_0$	$1_0 0_1$	$1_0 1_0$	$1_0 1_1$	$1_1 0_0$	$1_1 0_1$	$1_1 1_0$	$1_1 1_1$	8 7 BITS 6 5 4 3 2 1 / ROW
200 128 80	220 144 90	240 160 A0	260 176 B0	300 192 C0	320 208 D0	340 224 E0	360 240 F0	0000 / 0
201 129 81	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	0001 / 1
202 130 82	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	0010 / 2
203 131 83	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	0011 / 3
204 132 84	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	0100 / 4
205 133 85	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	0101 / 5
206 134 86	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	0110 / 6
207 135 87	227 151 97	§ 247 167 A7	• 267 183 B7	Ç 307 199 C7	× 327 215 D7	ç 347 231 E7	÷ 367 247 F7	0111 / 7
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	1000 / 8
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	1001 / 9
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	1010 / 10
213 139 8B	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	1011 / 11
214 140 8C	234 156 9C	¬ 254 172 AC	1/4 274 188 BC	Ì 314 204 CC	Ü 334 220 DC	ì 354 236 EC	ü 374 252 FC	1100 / 12
215 141 8D	235 157 9D	– 255 173 AD	1/2 275 189 BD	Í 315 205 CD	Ý 335 221 DD	í 355 237 ED	ý 375 253 FD	1101 / 13
216 142 8E	236 158 9E	® 256 174 AE	3/4 276 190 BE	Î 316 206 CE	Þ 336 222 DE	î 356 238 EE	þ 376 254 FE	1110 / 14
217 143 8F	237 159 9F	— 257 175 AF	¿ 277 191 BF	Ï 317 207 CF	ß 337 223 DF	ï 357 239 EF	ÿ 377 255 FF	1111 / 15

KEY: ESC | 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

These characters can normally be displayed in PT250 mode by holding down the **Alt** + **E** keys and pressing the equivalent standard (ASCII) character key.

The equivalent standard character key is generally found by subtracting 128 from the decimal value of the Additional character, then looking up the character with the resulting decimal number in the ASCII character set. Exceptions to this rule are the standard characters **"**, **6**, **<**, **=**, [and]. **6** should be swapped with ", **[** and **]** should be swapped with **<** and **=**, and vice versa.

PT250 LINE DRAWING CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$	$0\,0\,0\,1$	$0\,0\,1\,0$	$0\,0\,1\,1$	$0\,1\,0\,0$	$0\,1\,0\,1$	$0\,1\,1\,0$	$0\,1\,1\,1$
0	0000	0 0 0	20 16 10	40 32 20	60 48 30	100 64 40	120 80 50	140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	61 49 31	101 65 41	121 81 51	141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	62 50 32	102 66 42	122 82 52	142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	63 51 33	103 67 43	123 83 53	143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	64 52 34	104 68 44	124 84 54	144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	65 53 35	105 69 45	125 85 55	145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	66 54 36	106 70 46	126 86 56	146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	67 55 37	107 71 47	127 87 57	147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	70 56 38	110 72 48	130 88 58	150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	71 57 39	111 73 49	131 89 59	151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY: 

OCTAL
DECIMAL
HEXADECIMAL

This is a special PT250 character set which is used by some applications.

PT250 BLOCK GRAPHICS CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 6 5 BITS 4 3 2 1	$0\,0\,0\,0$	$0\,0\,0\,1$	$0\,0\,1\,0$	$0\,0\,1\,1$	$0\,1\,0\,0$	$0\,1\,0\,1$	$0\,1\,1\,0$	$0\,1\,1\,1$
0	0000	0 0 0	20 16 10	40 32 20	60 48 30	100 64 40	120 80 50	140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	61 49 31	101 65 41	121 81 51	141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	62 50 32	102 66 42	122 82 52	142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	63 51 33	103 67 43	123 83 53	143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	64 52 34	104 68 44	124 84 54	144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	65 53 35	105 69 45	125 85 55	145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	66 54 36	106 70 46	126 86 56	146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	67 55 37	107 71 47	127 87 57	147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	70 56 38	110 72 48	130 88 58	150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	71 57 39	111 73 49	131 89 59	151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY: 

This is a special PT250 character set which is used by some applications.

VIEWDATA ALPHANUMERIC CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$		$0\,0\,0\,1$		$0\,0\,1\,0$		$0\,0\,1\,1$		$0\,1\,0\,0$		$0\,1\,0\,1$		$0\,1\,1\,0$		$0\,1\,1\,1$	
0	0000	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	0001	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	0010	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	0011	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	0100	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	0101	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	0110	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	0111	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	1000	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	1001	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	1010	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	1011	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	¼	173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	½	134 92 5C	l	154 108 6C	||	174 124 7C
13	1101	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	¾	175 125 7D
14	1110	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	1111	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	■	177 127 7F

KEY: ESC (33 / 27 / 1B) — OCTAL / DECIMAL / HEXADECIMAL

This character set is used in Viewdata mode.

VIEWDATA GRAPHICS CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$		$0\,0\,0\,1$		$0\,0\,1\,0$		$0\,0\,1\,1$		$0\,1\,0\,0$		$0\,1\,0\,1$		$0\,1\,1\,0$		$0\,1\,1\,1$	
0	0000	NUL	0 0 0	DLE	20 16 10		40 32 20		60 48 30	@	100 64 40	P	120 80 50		140 96 60		160 112 70
1	0001	SOH	1 1 1	DC1 XON	21 17 11		41 33 21		61 49 31	A	101 65 41	Q	121 81 51		141 97 61		161 113 71
2	0010	STX	2 2 2	DC2	22 18 12		42 34 22		62 50 32	B	102 66 42	R	122 82 52		142 98 62		162 114 72
3	0011	ETX	3 3 3	DC3 XOFF	23 19 13		43 35 23		63 51 33	C	103 67 43	S	123 83 53		143 99 63		163 115 73
4	0100	EOT	4 4 4	DC4	24 20 14		44 36 24		64 52 34	D	104 68 44	T	124 84 54		144 100 64		164 116 74
5	0101	ENQ	5 5 5	NAK	25 21 15		45 37 25		65 53 35	E	105 69 45	U	125 85 55		145 101 65		165 117 75
6	0110	ACK	6 6 6	SYN	26 22 16		46 38 26		66 54 36	F	106 70 46	V	126 86 56		146 102 66		166 118 76
7	0111	BEL	7 7 7	ETB	27 23 17		47 39 27		67 55 37	G	107 71 47	W	127 87 57		147 103 67		167 119 77
8	1000	BS	10 8 8	CAN	30 24 18		50 40 28		70 56 38	H	110 72 48	X	130 88 58		150 104 68		170 120 78
9	1001	HT	11 9 9	EM	31 25 19		51 41 29		71 57 39	I	111 73 49	Y	131 89 59		151 105 69		171 121 79
10	1010	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	1011	VT	13 11 B	ESC	33 27 1B		53 43 2B		73 59 3B	K	113 75 4B	←	133 91 5B		153 107 6B		173 123 7B
12	1100	FF	14 12 C	FS	34 28 1C		54 44 2C		74 60 3C	L	114 76 4C	½	134 92 5C		154 108 6C		174 124 7C
13	1101	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	1110	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	1111	SI	17 15 F	US	37 31 1F		57 47 2F		77 63 3F	O	117 79 4F	#	137 95 5F		157 111 6F		177 127 7F

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This character set is used in Viewdata mode.

WYSE NATIVE MODE CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0000		0 0 0	┬	20 16 10		40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0001	SH	1 1 1	└	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	0010	SX	2 2 2	┌	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	0011	EX	3 3 3	┐	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	0100	ET	4 4 4	├	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	0101	EQ	5 5 5	┘	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	0110	AK	6 6 6	│	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	0111	BL	7 7 7	■	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	1000	BS	10 8 8	┼	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	1001	HT	11 9 9	┤	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	1010	LF	12 10 A	─	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	1011	VT	13 11 B	▒	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1100	FF	14 12 C	═	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1101	CR	15 13 D	┴	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1110	SO	16 14 E	╫	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1111	SI	17 15 F	▓	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	▓	177 127 7F

KEY: ┴ 35 29 1D — OCTAL, DECIMAL, HEXADECIMAL

This character set is used by all Wyse emulations and associated emulations. In WY-60 mode this is the default primary character set.

WY-60 MULTINATIONAL (CODE PAGE 437) CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1_0 0_0$		$1_0 0_1$		$1_0 1_0$		$1_0 1_1$		$1_1 0_0$		$1_1 0_1$		$1_1 1_0$		$1_1 1_1$		8 7 BITS 6 5 4 3 2 1	ROW
Ç	200 128 80	É	220 144 90	á	240 160 A0		260 176 B0	└	300 192 C0	╨	320 208 D0	α	340 224 E0	≡	360 240 F0	0000	0
ü	201 129 81	æ	221 145 91	í	241 161 A1		261 177 B1	┴	301 193 C1	╤	321 209 D1	ß	341 225 E1	±	361 241 F1	0001	1
é	202 130 82	Æ	222 146 92	ó	242 162 A2		262 178 B2	┬	302 194 C2	╥	322 210 D2	Γ	342 226 E2	≥	362 242 F2	0010	2
â	203 131 83	ô	223 147 93	ú	243 163 A3		263 179 B3	├	303 195 C3	╙	323 211 D3	π	343 227 E3	≤	363 243 F3	0011	3
ä	204 132 84	ö	224 148 94	ñ	244 164 A4		264 180 B4	─	304 196 C4	╘	324 212 D4	Σ	344 228 E4	⌠	364 244 F4	0100	4
à	205 133 85	ò	225 149 95	Ñ	245 165 A5		265 181 B5	┼	305 197 C5	╒	325 213 D5	σ	345 229 E5	⌡	365 245 F5	0101	5
å	206 134 86	û	226 150 96	ª	246 166 A6		266 182 B6	╞	306 198 C6	╓	326 214 D6	µ	346 230 E6	÷	366 246 F6	0110	6
ç	207 135 87	ù	227 151 97	º	247 167 A7		267 183 B7	╟	307 199 C7	╫	327 215 D7	τ	347 231 E7	≈	367 247 F7	0111	7
ê	210 136 88	ÿ	230 152 98	¿	250 168 A8		270 184 B8	╚	310 200 C8	╪	330 216 D8	Φ	350 232 E8	○	370 248 F8	1000	8
ë	211 137 89	Ö	231 153 99	⌐	251 169 A9		271 185 B9	╔	311 201 C9	┘	331 217 D9	Θ	351 233 E9	●	371 249 F9	1001	9
è	212 138 8A	Ü	232 154 9A	¬	252 170 AA		272 186 BA	╩	312 202 CA	┌	332 218 DA	Ω	352 234 EA	•	372 250 FA	1010	10
ï	213 139 8B	¢	233 155 9B	½	253 171 AB		273 187 BB	╦	313 203 CB	■	333 219 DB	δ	353 235 EB	√	373 251 FB	1011	11
î	214 140 8C	£	234 156 9C	¼	254 172 AC		274 188 BC	╠	314 204 CC	▄	334 220 DC	∞	354 236 EC	n	374 252 FC	1100	12
ì	215 141 8D	¥	235 157 9D	¡	255 173 AD		275 189 BD	═	315 205 CD	▌	335 221 DD	φ	355 237 ED	2	375 253 FD	1101	13
Ä	216 142 8E	₧	236 158 9E	«	256 174 AE		276 190 BE	╬	316 206 CE	▐	336 222 DE	∈	356 238 EE	■	376 254 FE	1110	14
Å	217 143 8F	ƒ	237 159 9F	»	257 175 AF		277 191 BF	╧	317 207 CF	▀	337 223 DF	∩	357 239 EF	BLANK FF	377 255 FF	1111	15

KEY: Ñ 245 165 A5 — OCTAL, DECIMAL, HEXADECIMAL

This Wyse character set is only available in WY-60 mode and is the secondary character set when the **Code Page** option in the **Wyse Settings** dialog box is set to **437**.

WY-60 MULTINATIONAL (CODE PAGE 850) CHARACTER SET

8		9		10		11		12		13		14		15		COLUMN	
$1\,0\,0\,0$		$1\,0\,0\,1$		$1\,0\,1\,0$		$1\,0\,1\,1$		$1\,1\,0\,0$		$1\,1\,0\,1$		$1\,1\,1\,0$		$1\,1\,1\,1$		$8\,7$ BITS $6\,5\,4\,3\,2\,1$	ROW
Ç	200 128 80	É	220 144 90	á	240 160 A0	▒	260 176 B0	└	300 192 C0	ð	320 208 D0	Ó	340 224 E0	—	360 240 F0	0000	0
ü	201 129 81	æ	221 145 91	í	241 161 A1	▒	261 177 B1	┴	301 193 C1	Ð	321 209 D1	ß	341 225 E1	±	361 241 F1	0001	1
é	202 130 82	Æ	222 146 92	ó	242 162 A2	▓	262 178 B2	┬	302 194 C2	Ê	322 210 D2	Ô	342 226 E2	‗	362 242 F2	0010	2
â	203 131 83	ô	223 147 93	ú	243 163 A3	│	263 179 B3	├	303 195 C3	Ë	323 211 D3	Ò	343 227 E3	¾	363 243 F3	0011	3
ä	204 132 84	ö	224 148 94	ñ	244 164 A4	┤	264 180 B4	─	304 196 C4	È	324 212 D4	õ	344 228 E4	¶	364 244 F4	0100	4
à	205 133 85	ò	225 149 95	Ñ	245 165 A5	Á	265 181 B5	┼	305 197 C5	ı	325 213 D5	Õ	345 229 E5	§	365 245 F5	0101	5
å	206 134 86	û	226 150 96	ª	246 166 A6	Â	266 182 B6	ã	306 198 C6	Í	326 214 D6	µ	346 230 E6	÷	366 246 F6	0110	6
ç	207 135 87	ù	227 151 97	º	247 167 A7	À	267 183 B7	Ã	307 199 C7	Î	327 215 D7	Þ	347 231 E7	ڃ	367 247 F7	0111	7
ê	210 136 88	ÿ	230 152 98	¿	250 168 A8	©	270 184 B8	╚	310 200 C8	Ï	330 216 D8	Þ	350 232 E8	○	370 248 F8	1000	8
ë	211 137 89	Ö	231 153 99	®	251 169 A9	╣	271 185 B9	╔	311 201 C9	┘	331 217 D9	Ú	351 233 E9	▪▪	371 249 F9	1001	9
è	212 138 8A	Ü	232 154 9A	¬	252 170 AA	║	272 186 BA	╩	312 202 CA	┌	332 218 DA	Û	352 234 EA	•	372 250 FA	1010	10
ï	213 139 8B	ø	233 155 9B	½	253 171 AB	╗	273 187 BB	╦	313 203 CB	▮	333 219 DB	Ù	353 235 EB	1	373 251 FB	1011	11
î	214 140 8C	£	234 156 9C	¼	254 172 AC	╝	274 188 BC	╠	314 204 CC	▬	334 220 DC	ý	354 236 EC	3	374 252 FC	1100	12
ì	215 141 8D	Ø	235 157 9D	¡	255 173 AD	¢	275 189 BD	═	315 205 CD	¦	335 221 DD	Ý	355 237 ED	2	375 253 FD	1101	13
Ä	216 142 8E	×	236 158 9E	«	256 174 AE	¥	276 190 BE	╬	316 206 CE	Ì	336 222 DE	—	356 238 EE	▮	376 254 FE	1110	14
Å	217 143 8F	ƒ	237 159 9F	»	257 175 AF	┐	277 191 BF	¤	317 207 CF	▮	337 223 DF	´	357 239 EF	BLANK FF	377 255 FF	1111	15

KEY: Ñ 245 / 165 / A5 — OCTAL / DECIMAL / HEXADECIMAL

This Wyse character set is only available in WY-60 mode and is the secondary character set when the **Code Page** option in the **Wyse Settings** dialog box is set to **850**.

WY-60 PC EQUIVALENT CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0000		0 0 0	▶	20 16 10		40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0001	☺	1 1 1	◀	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	0010	☻	2 2 2	↕	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	0011	♥	3 3 3	!!	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	0100	♦	4 4 4	¶	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	0101	♣	5 5 5	§	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	0110	♠	6 6 6	▬	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	0111	•	7 7 7	↨	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	1000	◘	10 8 8	↑	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	1001	○	11 9 9	↓	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	1010	◙	12 10 A	→	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	1011	♂	13 11 B	←	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1100	♀	14 12 C	∟	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1101	♪	15 13 D	↔	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1110	♫	16 14 E	▲	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1111	☼	17 15 F	▼	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	▒	177 127 7F

KEY: ← 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

WY-60 STANDARD ASCII CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0000		0 0 0	DL	20 16 10		40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0001	SH	1 1 1	D1	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	0010	SX	2 2 2	D2	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	0011	EX	3 3 3	D3	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	0100	ET	4 4 4	D4	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	0101	EQ	5 5 5	NK	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	0110	AK	6 6 6	SY	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	0111	BL	7 7 7	EB	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	1000	BS	10 8 8	CN	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	1001	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	1010	LF	12 10 A	SB	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	1011	VT	13 11 B	EC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1100	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	1101	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	1110	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	1111	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		177 127 7F

KEY: EC | 33 OCTAL, 27 DECIMAL, 1B HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

WY-60 STANDARD ANSI CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_0 0_0$		$0_0 1_1 0_0$		$0_1 0_0 0_0$		$0_1 0_1 0_1$		$0_1 1_0 0_0$		$0_1 1_1 0_1$	
0	0000		0 0 0	⎺	20 16 10		40 32 20	**0**	60 48 30	**@**	100 64 40	**P**	120 80 50	`	140 96 60	**p**	160 112 70
1	0001	◆	1 1 1	⎺	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	0010	▮	2 2 2	⎺	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	0011	H_T	3 3 3	⎽	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	0100	F_F	4 4 4	⎽	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	0101	C_R	5 5 5	├	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	0110	L_F	6 6 6	┤	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	0111	°	7 7 7	┴	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	1000	±	10 8 8	┬	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	1001	N_L	11 9 9	│	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	1010	V_T	12 10 A	≤	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	1011	┘	13 11 B	≥	33 27 1B	**+**	53 43 2B	**;**	73 59 3B	**K**	113 75 4B	**[**	133 91 5B	**k**	153 107 6B	**{**	173 123 7B
12	1100	┐	14 12 C	π	34 28 1C	**,**	54 44 2C	**<**	74 60 3C	**L**	114 76 4C	****	134 92 5C	**l**	154 108 6C	**|**	174 124 7C
13	1101	┌	15 13 D	≠	35 29 1D	**-**	55 45 2D	**=**	75 61 3D	**M**	115 77 4D	**]**	135 93 5D	**m**	155 109 6D	**}**	175 125 7D
14	1110	└	16 14 E	£	36 30 1E	**.**	56 46 2E	**>**	76 62 3E	**N**	116 78 4E	**^**	136 94 5E	**n**	156 110 6E	**~**	176 126 7E
15	1111	┼	17 15 F	·	37 31 1F	**/**	57 47 2F	**?**	77 63 3F	**O**	117 79 4F	_	137 95 5F	**o**	157 111 6F	▮	177 127 7F

KEY: ≥ | 33 OCTAL | 27 DECIMAL | 1B HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

WY-60 GRAPHICS 1
CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 BITS 6 5 4 3 2 1	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1
0	0000	0 0 0	20 16 10	40 32 20	0 60 48 30	100 64 40	120 80 50	0 140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	1 61 49 31	101 65 41	121 81 51	1 141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	2 62 50 32	102 66 42	122 82 52	2 142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	3 63 51 33	103 67 43	123 83 53	3 143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	4 64 52 34	104 68 44	124 84 54	4 144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	5 65 53 35	105 69 45	125 85 55	5 145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	6 66 54 36	106 70 46	126 86 56	6 146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	7 67 55 37	107 71 47	127 87 57	7 147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	8 70 56 38	110 72 48	130 88 58	8 150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	9 71 57 39	111 73 49	131 89 59	9 151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY:  156 OCTAL, 110 DECIMAL, 6E HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

WY-60 GRAPHICS 2
CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$	$0\,0\,0\,1$	$0\,0\,1\,0$	$0\,0\,1\,1$	$0\,1\,0\,0$	$0\,1\,0\,1$	$0\,1\,1\,0$	$0\,1\,1\,1$
0	0000	0 0 0	20 16 10	40 32 20	60 48 30	100 64 40	120 80 50	140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	61 49 31	101 65 41	121 81 51	141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	62 50 32	102 66 42	122 82 52	142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	63 51 33	103 67 43	123 83 53	143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	64 52 34	104 68 44	124 84 54	144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	65 53 35	105 69 45	125 85 55	145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	66 54 36	106 70 46	126 86 56	146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	67 55 37	107 71 47	127 87 57	147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	70 56 38	110 72 48	130 88 58	150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	71 57 39	111 73 49	131 89 59	151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY: ┘ 114 / 76 / 4C — OCTAL / DECIMAL / HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

WY-60 GRAPHICS 3
CHARACTER SET

COLUMN		0	1	2	3	4	5	6	7
ROW	8 7 BITS 6 5 4 3 2 1	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1
0	0000	0 0 0	20 16 10	40 32 20	60 48 30	100 64 40	120 80 50	140 96 60	160 112 70
1	0001	1 1 1	21 17 11	41 33 21	61 49 31	101 65 41	121 81 51	141 97 61	161 113 71
2	0010	2 2 2	22 18 12	42 34 22	62 50 32	102 66 42	122 82 52	142 98 62	162 114 72
3	0011	3 3 3	23 19 13	43 35 23	63 51 33	103 67 43	123 83 53	143 99 63	163 115 73
4	0100	4 4 4	24 20 14	44 36 24	64 52 34	104 68 44	124 84 54	144 100 64	164 116 74
5	0101	5 5 5	25 21 15	45 37 25	65 53 35	105 69 45	125 85 55	145 101 65	165 117 75
6	0110	6 6 6	26 22 16	46 38 26	66 54 36	106 70 46	126 86 56	146 102 66	166 118 76
7	0111	7 7 7	27 23 17	47 39 27	67 55 37	107 71 47	127 87 57	147 103 67	167 119 77
8	1000	10 8 8	30 24 18	50 40 28	70 56 38	110 72 48	130 88 58	150 104 68	170 120 78
9	1001	11 9 9	31 25 19	51 41 29	71 57 39	111 73 49	131 89 59	151 105 69	171 121 79
10	1010	12 10 A	32 26 1A	52 42 2A	72 58 3A	112 74 4A	132 90 5A	152 106 6A	172 122 7A
11	1011	13 11 B	33 27 1B	53 43 2B	73 59 3B	113 75 4B	133 91 5B	153 107 6B	173 123 7B
12	1100	14 12 C	34 28 1C	54 44 2C	74 60 3C	114 76 4C	134 92 5C	154 108 6C	174 124 7C
13	1101	15 13 D	35 29 1D	55 45 2D	75 61 3D	115 77 4D	135 93 5D	155 109 6D	175 125 7D
14	1110	16 14 E	36 30 1E	56 46 2E	76 62 3E	116 78 4E	136 94 5E	156 110 6E	176 126 7E
15	1111	17 15 F	37 31 1F	57 47 2F	77 63 3F	117 79 4F	137 95 5F	157 111 6F	177 127 7F

KEY:  114 OCTAL, 76 DECIMAL, 4C HEXADECIMAL

This Wyse character set is only available in WY-60 mode.

SIEMENS 97801 INTERNATIONAL A CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0\,0\,0\,0$		$0\,0\,0\,1$		$0\,0\,1\,0$		$0\,0\,1\,1$		$0\,1\,0\,0$		$0\,1\,0\,1$		$0\,1\,1\,0$		$0\,1\,1\,1$	
0	0000	**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	0001	**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	0010	**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	0011	**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	0100	**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	0101	**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	0110	**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	0111	**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	1000	**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	1001	**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	1010	**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	1011	**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	1100	**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	1101	**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	1110	**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	1111	**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

KEY: **ESC** 33 / 27 / 1B — OCTAL / DECIMAL / HEXADECIMAL

SIEMENS 97801 INTERNATIONAL CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_1 0_0$		$0_0 1_1 1_1$		$0_1 0_0 0_0$		$0_1 0_0 1_1$		$0_1 1_1 0_0$		$0_1 1_1 1_1$	
0	0000	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	0001	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	0010	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	0011	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	0100	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	0101	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	0110	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	0111	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	1000	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	1001	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	1010	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	1011	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	1100	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	1101	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	1110	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	1111	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

KEY: ESC 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

SIEMENS 97801 GERMAN CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_0 0_0$		$0_0 1_1 1_1$		$0_1 0_0 0_0$		$0_1 0_1 1_1$		$0_1 1_0 0_0$		$0_1 1_1 1_1$	
0	0000	**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	0001	**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	0010	**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	0011	**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	0100	**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	0101	**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	0110	**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	0111	**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	1000	**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	1001	**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	1010	**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	1011	**VT**	13 11 B	**ESC**	33 27 1B	**+**	53 43 2B	**;**	73 59 3B	**K**	113 75 4B	**Ä**	133 91 5B	**k**	153 107 6B	**ä**	173 123 7B
12	1100	**FF**	14 12 C	**FS**	34 28 1C	**,**	54 44 2C	**<**	74 60 3C	**L**	114 76 4C	**Ö**	134 92 5C	**l**	154 108 6C	**ö**	174 124 7C
13	1101	**CR**	15 13 D	**GS**	35 29 1D	**-**	55 45 2D	**=**	75 61 3D	**M**	115 77 4D	**Ü**	135 93 5D	**m**	155 109 6D	**ü**	175 125 7D
14	1110	**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	**ß**	176 126 7E
15	1111	**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

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

SIEMENS 97801 EURO CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 6 5 BITS 4 3 2 1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0000	**NUL**	0 0 0	**DLE**	20 16 10	**SP**	40 32 20	**è**	60 48 30	**ň**	100 64 40	**ş**	120 80 50	**Å**	140 96 60	**©**	160 112 70
1	0001	**SOH**	1 1 1	**DC1** **XON**	21 17 11	**à**	41 33 21	**é**	61 49 31	**ń**	101 65 41	**ß**	121 81 51	**Æ**	141 97 61	**Ω**	161 113 71
2	0010	**STX**	2 2 2	**DC2**	22 18 12	**á**	42 34 22	**ê**	62 50 32	**ñ**	102 66 42	**ţ**	122 82 52	**Đ**	142 98 62	**µ**	162 114 72
3	0011	**ETX**	3 3 3	**DC3** **XOFF**	23 19 13	**â**	43 35 23	**ë**	63 51 33	**ò**	103 67 43	**ť**	123 83 53	**İ**	143 99 63	**°**	163 115 73
4	0100	**EOT**	4 4 4	**DC4**	24 20 14	**ä**	44 36 24	**ě**	64 52 34	**ó**	104 68 44	**ù**	124 84 54	**1J**	144 100 64	**Ç**	164 116 74
5	0101	**ENQ**	5 5 5	**NAK**	25 21 15	**å**	45 37 25	**ę**	65 53 35	**ô**	105 69 45	**ú**	125 85 55	**ŧ**	145 101 65	**P_t**	165 117 75
6	0110	**ACK**	6 6 6	**SYN**	26 22 16	**ą**	46 38 26	**ğ**	66 54 36	**ö**	106 70 46	**û**	126 86 56	**Ø**	146 102 66	**π**	166 118 76
7	0111	**BEL**	7 7 7	**ETB**	27 23 17	**ã**	47 39 27	**ı**	67 55 37	**õ**	107 71 47	**ü**	127 87 57	**Œ**	147 103 67	**˘**	167 119 77
8	1000	**BS**	10 8 8	**CAN**	30 24 18	**ă**	50 40 28	**î**	70 56 38	**ø**	110 72 48	**ů**	130 88 58	**Þ**	150 104 68	**´**	170 120 78
9	1001	**HT**	11 9 9	**EM**	31 25 19	**æ**	51 41 29	**ì**	71 57 39	**ő**	111 73 49	**ű**	131 89 59	**Ä**	151 105 69	**˝**	171 121 79
10	1010	**LF**	12 10 A	**SUB**	32 26 1A	**ç**	52 42 2A	**ĭ**	72 58 3A	**œ**	112 74 4A	**ý**	132 90 5A	**Ö**	152 106 6A	**Ñ**	172 122 7A
11	1011	**VT**	13 11 B	**ESC**	33 27 1B	**č**	53 43 2B	**ï**	73 59 3B	**þ**	113 75 4B	**ÿ**	133 91 5B	**Ü**	153 107 6B	**˩**	173 123 7B
12	1100	**FF**	14 12 C	**FS**	34 28 1C	**ć**	54 44 2C	**ıj**	74 60 3C	**ř**	114 76 4C	**ž**	134 92 5C	**§**	154 108 6C	**˫**	174 124 7C
13	1101	**CR**	15 13 D	**GS**	35 29 1D	**đ**	55 45 2D	**ĺ**	75 61 3D	**ŕ**	115 77 4D	**ź**	135 93 5D	**$**	155 109 6D	**~**	175 125 7D
14	1110	**SO**	16 14 E	**RS**	36 30 1E	**ď**	56 46 2E	**ł**	76 62 3E	**š**	116 78 4E	**ż**	136 94 5E	**£**	156 110 6E	**˯**	176 126 7E
15	1111	**SI**	17 15 F	**US**	37 31 1F	**ď**	57 47 2F	**ľ**	77 63 3F	**ś**	117 79 4F	**É**	137 95 5F	**®**	157 111 6F	**DEL**	177 127 7F

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

SIEMENS 97801 BRACKETS CHARACTER SET

COLUMN		0		1		2		3		4		5		6		7	
ROW	8 7 BITS 6 5 4 3 2 1	$0_0 0_0 0_0$		$0_0 0_0 1_1$		$0_0 1_0$		$0_0 1_1$		$0_1 0_0$		$0_1 0_1$		$0_1 1_0$		$0_1 1_1$	
0	0000	**NUL**	0 0 0	**DLE**	20 16 10	**SP**	40 32 20		60 48 30		100 64 40		120 80 50		140 96 60		160 112 70
1	0001	**SOH**	1 1 1	**DC1** **XON**	21 17 11		41 33 21		61 49 31		101 65 41		121 81 51		141 97 61		161 113 71
2	0010	**STX**	2 2 2	**DC2**	22 18 12		42 34 22		62 50 32		102 66 42		122 82 52		142 98 62		162 114 72
3	0011	**ETX**	3 3 3	**DC3** **XOFF**	23 19 13		43 35 23		63 51 33		103 67 43		123 83 53		143 99 63		163 115 73
4	0100	**EOT**	4 4 4	**DC4**	24 20 14		44 36 24		64 52 34		104 68 44		124 84 54		144 100 64		164 116 74
5	0101	**ENQ**	5 5 5	**NAK**	25 21 15		45 37 25		65 53 35		105 69 45		125 85 55		145 101 65		165 117 75
6	0110	**ACK**	6 6 6	**SYN**	26 22 16		46 38 26		66 54 36		106 70 46		126 86 56		146 102 66		166 118 76
7	0111	**BEL**	7 7 7	**ETB**	27 23 17		47 39 27		67 55 37		107 71 47		127 87 57		147 103 67		167 119 77
8	1000	**BS**	10 8 8	**CAN**	30 24 18		50 40 28		70 56 38		110 72 48		130 88 58		150 104 68		170 120 78
9	1001	**HT**	11 9 9	**EM**	31 25 19		51 41 29		71 57 39		111 73 49		131 89 59		151 105 69		171 121 79
10	1010	**LF**	12 10 A	**SUB**	32 26 1A		52 42 2A		72 58 3A		112 74 4A		132 90 5A		152 106 6A		172 122 7A
11	1011	**VT**	13 11 B	**ESC**	33 27 1B		53 43 2B		73 59 3B		113 75 4B		133 91 5B		153 107 6B		173 123 7B
12	1100	**FF**	14 12 C	**FS**	34 28 1C		54 44 2C		74 60 3C		114 76 4C		134 92 5C		154 108 6C		174 124 7C
13	1101	**CR**	15 13 D	**GS**	35 29 1D		55 45 2D		75 61 3D		115 77 4D		135 93 5D		155 109 6D		175 125 7D
14	1110	**SO**	16 14 E	**RS**	36 30 1E		56 46 2E		76 62 3E		116 78 4E		136 94 5E		156 110 6E		176 126 7E
15	1111	**SI**	17 15 F	**US**	37 31 1F		57 47 2F		77 63 3F		117 79 4F		137 95 5F		157 111 6F	**DEL**	177 127 7F

KEY: **ESC** 33 27 1B — OCTAL, DECIMAL, HEXADECIMAL

IBM 3270 & IBM 5250 ENGLISH (U.S.), CANADIAN BILINGUAL & NETHERLANDS CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	**SP**	**&**	**-**	**ø**	**Ø**	°	**µ**	**^**	**{**	**}**	****	**0**
-1	**RSP**	**é**	**/**	**É**	**a**	**j**	**~**	**£**	**A**	**J**	**÷**	**1**
-2	**â**	**ê**	**Â**	**Ê**	**b**	**k**	**s**	**¥**	**B**	**K**	**S**	**2**
-3	**ä**	**ë**	**Ä**	**Ë**	**c**	**l**	**t**	**·**	**C**	**L**	**T**	**3**
-4	**à**	**è**	**À**	**È**	**d**	**m**	**u**	**©**	**D**	**M**	**U**	**4**
-5	**á**	**í**	**Á**	**Í**	**e**	**n**	**v**	**§**	**E**	**N**	**V**	**5**
-6	**ã**	**î**	**Ã**	**Î**	**f**	**o**	**w**	**¶**	**F**	**O**	**W**	**6**
-7	**å**	**ï**	**Å**	**Ï**	**g**	**p**	**x**	¼	**G**	**P**	**X**	**7**
-8	**ç**	**ì**	**Ç**	**Ì**	**h**	**q**	**y**	½	**H**	**Q**	**Y**	**8**
-9	**ñ**	**ß**	**Ñ**	**`**	**i**	**r**	**z**	¾	**I**	**R**	**Z**	**9**
-A	**¢**	**!**	¦	**:**	«	ª	**¡**	**[**	**SHY**	¹	²	³
-B	**.**	**$**	**,**	**#**	»	º	**¿**	**]**	**ô**	**û**	**Ô**	**Û**
-C	**<**	*****	**%**	**@**	**ð**	**æ**	**Ð**	¯	**ö**	**ü**	**Ö**	**Ü**
-D	**(**	**)**	**_**	**´**	**ý**	¸	**Ý**	¨	**ò**	**ù**	**Ò**	**Ù**
-E	**+**	**;**	**>**	**=**	**þ**	**Æ**	**þ**	**´**	**ó**	**ú**	**Ó**	**Ú**
-F	**|**	¬	**?**	**"**	**±**	¤	®	×	**õ**	**ÿ**	**Õ**	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
ENGLISH (U.K.) CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	**SP**	**&**	**-**	**ø**	**Ø**	**°**	**µ**	**¢**	**{**	**}**	****	**0**
-1	**RSP**	**é**	**/**	**É**	**a**	**j**	**-**	**[**	**A**	**J**	**÷**	**1**
-2	**â**	**ê**	**Â**	**Ê**	**b**	**k**	**s**	**¥**	**B**	**K**	**S**	**2**
-3	**ä**	**ë**	**Ä**	**Ë**	**c**	**l**	**t**	**·**	**C**	**L**	**T**	**3**
-4	**à**	**è**	**À**	**È**	**d**	**m**	**u**	**©**	**D**	**M**	**U**	**4**
-5	**á**	**í**	**Á**	**Í**	**e**	**n**	**v**	**§**	**E**	**N**	**V**	**5**
-6	**ã**	**î**	**Ã**	**Î**	**f**	**o**	**w**	**¶**	**F**	**O**	**W**	**6**
-7	**å**	**ï**	**Å**	**Ï**	**g**	**p**	**x**	**¼**	**G**	**P**	**X**	**7**
-8	**ç**	**ì**	**Ç**	**Ì**	**h**	**q**	**y**	**½**	**H**	**Q**	**Y**	**8**
-9	**ñ**	**ß**	**Ñ**	**`**	**i**	**r**	**z**	**¾**	**I**	**R**	**Z**	**9**
-A	**$**	**!**	**¦**	**:**	**«**	**ª**	**¡**	**^**	**SHY**	**¹**	**²**	**³**
-B	**.**	**£**	**,**	**#**	**»**	**º**	**¿**	**]**	**ô**	**û**	**Ô**	**Û**
-C	**<**	*****	**%**	**@**	**ð**	**æ**	**Ð**	**~**	**ö**	**ü**	**Ö**	**Ü**
-D	**(**	**)**	**_**	**´**	**ý**	**¸**	**Ý**	**¨**	**ò**	**ù**	**Ò**	**Ù**
-E	**+**	**;**	**>**	**=**	**þ**	**Æ**	**Þ**	**´**	**ó**	**ú**	**Ó**	**Ú**
-F	**|**	**¬**	**?**	**"**	**±**	**¤**	**®**	**×**	**õ**	**ÿ**	**Õ**	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
DANISH & NORWEGIAN CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	¦	@	°	µ	¢	æ	å	\	0
-1	RSP	é	/	É	a	j	ü	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	§	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	}	ï	$	Ï	g	p	x	¼	G	P	X	7
-8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
-A	#	¤	ø	:	«	ª	¡	¬	SHY	¹	²	³
-B	.	Å	,	Æ	»	º	¿	ǀ	ô	û	Ô	Û
-C	<	*	%	Ø	ð	{	Ð	¯	ö	ü	Ö	Ü
-D	(	)	_	´	ý	¸	Ý	¨	ò	ù	Ò	Ù
-E	+	;	>	=	þ	[	Þ	´	ó	ú	Ó	Ú
-F	!	^	?	"	±	]	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
FRENCH CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	[	`	¢	é	è	ç	0
-1	RSP	{	/	É	a	j	¨	#	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	@	}	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	]	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	1/4	G	P	X	7
-8	\	ì	Ç	Ì	h	q	y	1/2	H	Q	Y	8
-9	ñ	ß	Ñ	µ	i	r	z	3/4	I	R	Z	9
-A	°	§	ù	:	«	ª	¡	¬	SHY	¹	²	³
-B	.	$	,	£	»	º	¿	l	ô	û	Ô	Û
-C	<	*	%	à	ð	æ	Ð	¯	ö	ü	Ö	Ü
-D	(	)	_	´	ý	¸	Ý	~	ò	¦	Ò	Ù
-E	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
-F	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
GERMAN & AUSTRIAN CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	°	µ	¢	ä	ü	Ö	0
-1	RSP	é	/	É	a	j	ß	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	{	ë	[	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	@	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	~	Ñ	`	i	r	z	¾	I	R	Z	9
-A	Ä	Ü	ö	:	«	ª	¡	¬	SHY	¹	²	³
-B	.	$	,	#	»	º	¿	‖	ô	û	Ô	Û
-C	<	*	%	§	ð	æ	Ð	¯	¦	}	\	]
-D	(	)	_	´	ý	¸	Ý	¨	ò	ù	Ò	Ù
-E	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
-F	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
ITALIAN CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	[	µ	¢	à	è	ç	0
-1	RSP	]	/	É	a	j	ì	#	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	{	}	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	@	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	1/4	G	P	X	7
-8	\	~	Ç	Ì	h	q	y	1/2	H	Q	Y	8
-9	ñ	ß	Ñ	ù	i	r	z	3/4	I	R	Z	9
-A	°	é	ò	:	«	ª	¡	¬	SHY	¹	²	³
-B	.	$	,	£	»	º	¿	ǀ	ô	û	Ô	Û
-C	<	*	%	§	ð	æ	Ð	¯	ö	ü	Ö	Ü
-D	(	)	_	´	ý	¸	Ý	¨	¦	`	Ò	Ù
-E	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
-F	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
SPANISH CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	°	µ	¢	{	}	\	0
-1	RSP	é	/	É	a	j	¨	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	§	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	¦	ß	#	`	i	r	z	¾	I	R	Z	9
-A	[	]	ñ	:	«	ª	¡	^	SHY	¹	²	³
-B	.	$	,	Ñ	»	º	¿	!	ô	û	Ô	Û
-C	<	*	%	@	ð	æ	Ð	¯	ö	ü	Ö	Ü
-D	(	)	_	´	ý	¸	Ý	~	ò	ù	Ò	Ù
-E	+	;	>	=	þ	Æ	þ	´	ó	ú	Ó	Ú
-F			¬	?	"	±	¤	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250
SWEDISH & FINNISH CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	**SP**	**&**	**-**	**ø**	**Ø**	**°**	**µ**	**¢**	**ä**	**å**	**É**	**0**
-1	**RSP**	**`**	**/**	****	**a**	**j**	**ü**	**£**	**A**	**J**	**÷**	**1**
-2	**â**	**ê**	**Â**	**Ê**	**b**	**k**	**s**	**¥**	**B**	**K**	**S**	**2**
-3	**{**	**ë**	**#**	**Ë**	**c**	**l**	**t**	**·**	**C**	**L**	**T**	**3**
-4	**à**	**è**	**À**	**È**	**d**	**m**	**u**	**©**	**D**	**M**	**U**	**4**
-5	**á**	**í**	**Á**	**Í**	**e**	**n**	**v**	**[**	**E**	**N**	**V**	**5**
-6	**ã**	**î**	**Ã**	**Î**	**f**	**o**	**w**	**¶**	**F**	**O**	**W**	**6**
-7	**}**	**ï**	**$**	**Ï**	**g**	**p**	**x**	**¼**	**G**	**P**	**X**	**7**
-8	**ç**	**ì**	**Ç**	**Ì**	**h**	**q**	**y**	**½**	**H**	**Q**	**Y**	**8**
-9	**ñ**	**ß**	**Ñ**	**é**	**i**	**r**	**z**	**¾**	**I**	**R**	**Z**	**9**
-A	**§**	**¤**	**ö**	**:**	**«**	**ª**	**¡**	**¬**	**SHY**	**¹**	**²**	**³**
-B	**.**	**Å**	**,**	**Ä**	**»**	**º**	**¿**	**|**	**ô**	**û**	**Ô**	**Û**
-C	**<**	*****	**%**	**Ö**	**ð**	**æ**	**Ð**	**¯**	**¦**	**~**	**@**	**Ü**
-D	**(**	**)**	**_**	**´**	**ý**	**¸**	**Ý**	**¨**	**ò**	**ù**	**Ò**	**Ù**
-E	**+**	**;**	**>**	**=**	**þ**	**Æ**	**Þ**	**´**	**ó**	**ú**	**Ó**	**Ú**
-F	**!**	**^**	**?**	**"**	**±**	**]**	**®**	**×**	**õ**	**ÿ**	**Õ**	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM 3270 & IBM 5250 BELGIAN & SWISS-FRENCH/GERMAN CHARACTER SET

HEX	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	°	µ	¢	{	}	\	0
-1	RSP	é	/	É	a	j	~	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	§	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
-A	[	]	¦	:	«	ª	¡	¬	SHY	¹	²	³
-B	.	$	,	#	»	º	¿	ǀ	ô	û	Ô	Û
-C	<	*	%	@	ð	æ	Ð	¯	ö	ü	Ö	Ü
-D	(	)	_	´	ý	¸	Ý	¨	ò	ù	Ò	Ù
-E	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
-F	!	^	?	"	±	¤	®	×	õ	ÿ	Õ	

Legend: **RSP** required space, **SHY** syllable hyphen, **SP** space.

IBM EBCDIC CODEPAGES SUPPLIED

Codepage	Type	Countries
37	SBCS	USA, Canada, Netherlands, Portugal, Brazil
273	SBCS	Austria, Germany
274	SBCS	Belgium (old)
277	SBCS	Denmark, Norway
278	SBCS	Finland, Sweden
280	SBCS	Italy
284	SBCS	Spain, Latin America (Spanish)
285	SBCS	UK
290	SBCS	Japanese - Katakana
297	SBCS	France
300	DBCS	Japanese - Kanji
420	SBCS	Arabic
424	SBCS	Hebrew New Code
500	SBCS	(Latin 1) Belgium, Canada, Switzerland
803	SBCS	Hebrew Old Code
833	SBCS	Korean
834	DBCS	Korean
835	DBCS	Traditional Chinese
836	SBCS	Simplified Chinese
837	DBCS	Simplified Chinese
838	SBCS	Thai
870	SBCS	(Latin 2) Czech, Slovak, Polish
871	SBCS	Icelandic
875	SBCS	Greek
880	SBCS	Cyrillic
905	SBCS	(Latin 3) Turkish (old)
1025	SBCS	Russian Cyrillic
1026	SBCS	(Latin 5) Turkish
1027	SBCS	Japanese - Latin extended
1140	SBCS	[EURO] USA, Canada, Netherlands, Portugal, Brazil
1141	SBCS	[EURO] Austria, Germany
1142	SBCS	[EURO] Denmark, Norway
1143	SBCS	[EURO] Finland, Sweden
1144	SBCS	[EURO] Italy
1145	SBCS	[EURO] Spain, Latin America
1146	SBCS	[EURO] UK
1147	SBCS	[EURO] France
1148	SBCS	[EURO] Belgium, Canada, Switzerland
1149	SBCS	[EURO] Icelandic

C

Host Command Summary

This appendix lists the host commands that are valid in each terminal emulation mode. TeemTalk-specific commands are listed at the back.

The following conventions are used in this command list. Spaces in a command are for clarity only and are not to be entered as part of the command. A space character that is part of the command will be shown as SP. An asterisk (*) in a command indicates the location of one or more parameters except otherwise indicated next to the command.

Note that IBM 3270 and IBM 5250 emulation commands are not included because of their complexity. Refer to the manuals supplied with these terminals for the host commands that are supported.

VT52 Emulation

CHARACTER SET SELECTION

Invoke G0 character set	SI
Invoke G1 character set	SO
Select G0 character set	ESC G
Select Line Drawing character set	ESC F

CURSOR

Direct cursor addressing (1 to 96 = SP to DEL)	ESC Y *line *column
Insert FF character & advance cursor	FF
Line feed	LF
Move cursor down one line	ESC B
Move cursor down one line	VT
Move cursor home	ESC H
Move cursor one column left	BS
Move cursor one column left	ESC D
Move cursor one column right	ESC C
Move cursor to left margin of current line	CR
Move cursor to next tab stop	HT
Move cursor up one line	ESC A
Reverse line feed	ESC I

TEXT ERASURE

Erase text to end of line	ESC K
Erase text to end of screen	ESC J

GENERAL

Cancel current ESC sequence & display error	CAN
Sound audible tone	BEL

MODE SELECTION

Select numeric keypad application mode	ESC =
Select numeric keypad normal mode	ESC >
Select VT100 mode	ESC <

PRINTING

Auto print off	ESC _
Auto print on	ESC ^
Print controller off	ESC X
Print controller on	ESC W

REPORTS

Request mode identification report	ESC Z
Send terminal emulation mode report	ESC # ! 0

ANSI VT100 Emulation

CHARACTER ATTRIBUTES

Assign * attribute(s) to following characters — ESC [* m

Default attributes	0	Blue foreground	34	White background	50
Bold on	1	Magenta foreground	35	Red background	51
Dim White	2	Cyan foreground	36	Red background	52
Underline on	4	White foreground	37	Yellow background	53
Flashing on	5	'Normal' foreground	39	Blue background	54
Inverse video on	7	White background	40	Magenta background	55
Half intensity off	22	Red background	41	Cyan background	56
Underline off	24	Red background	42	White background	57
Flashing off	25	Yellow background	43	'Normal' background	59
Inverse video off	27	Blue background	44		
Black foreground	30	Magenta background	45		
Red foreground	31	Cyan background	46		
Green foreground	32	White background	47		
Yellow foreground	33	'Normal' background	49		

Deselect underline character mode	ESC [< 1 h
Double width & height (top half) characters	ESC # 3
Double width & height (bottom half) characters	ESC # 4
Double width single height characters	ESC # 6
Select underline character mode	ESC [< 1 l
Single width & height (normal) characters	ESC # 5

CHARACTER SET SELECTION

Assign G0 label to * character set	ESC (*
Assign G1 label to * character set	ESC) *

ASCII / N.American	B	Italian	Y
British	A	Danish / Norwegian	' *or* E *or* 6
Dutch	4	Portuguese	%6
Finnish	5 *or* C	Spanish	Z
French	R	Swedish	7 *or* H
French Canadian	9 *or* Q	Swiss	=
German	K	Line Drawing	0

Assign G0 labelled set to 7 bit codes	SI
Assign G1 labelled set to 7 bit codes	SO

CURSOR

Cursor on	ESC [? 25 h
Cursor off	ESC [? 25 l
Clear tab stops (0 = cursor position, 2 or 3 = all)	ESC [* g
Deselect auto carriage return	ESC [20 l
Disable cursor	ESC [? 50 l
Disable cursor autowrap	ESC [? 7 l
Enable cursor	ESC [? 50 h
Enable cursor autowrap	ESC [? 7 h
Index cursor (move down one line)	ESC D
Insert FF character & advance cursor	FF
Line feed	LF
Move cursor down * lines	ESC [* B
Move cursor down one line	VT
Move cursor left * columns	ESC [* D
Move cursor one column left	BS
Move cursor right * columns	ESC [* C
Move cursor to beginning of next line	ESC E
Move cursor to left margin of current line	CR
Move cursor to line (*l) column (*c)	ESC [*l ; *c H
Move cursor to line (*l) column (*c)	ESC [*l ; *c f
Move cursor to next tab stop	HT
Move cursor up * lines	ESC [* A
Reverse index cursor (move up one line)	ESC M
Select absolute origin mode	ESC [? 6 l
Select auto carriage return	ESC [20 h
Select relative origin mode	ESC [? 6 h
Set tab stop at current cursor position	ESC H
Tab cursor backward * tabs	ESC [* Z
Tab cursor forward * tabs	ESC [* I

DISPLAY

Scroll display down * lines	ESC [* T
Scroll display up * lines	ESC [* S
Select 80 column display mode	ESC [? 3 l
Select 132 column display mode	ESC [? 3 h
Select invisible display	ESC [? 75 l
Select normal colour display mode	ESC [? 5 l
Select reverse colour display mode	ESC [? 5 h
Select thin-line graphics in ANSI BBS mode	ESC 10 m
Select visible display	ESC [? 75 h
Set top (*t) and bottom (*b) margin positions	ESC [*t ; *b r

EDITING

Delete * characters from cursor position right	ESC [* P
Delete * lines from cursor position down	ESC [* M
Enable all characters to be erased	ESC [6 h
Enable erasure of unprotected characters only	ESC [6 l
End protected area	ESC W
Erase * characters & attributes from cursor right	ESC [* X
Erase line portion (0 = from, 1 = to, 2 = all)	ESC [* K
Erase screen portion (0 = from, 1 = to, 2 = all)	ESC [* J
Insert * blank lines	ESC [* L
Insert * space characters	ESC [* @
Protect characters with * attribute(s)	ESC [* }
Select insert mode	ESC [4 h
Select replace mode	ESC [4 l
Start protected area	ESC V

GENERAL OPERATION

Cancel current ESC sequence & display error	CAN
Local echo mode on	ESC [12 l
Local echo mode off	ESC [12 h
Reset features (* = * in ESC [* h commands)	ESC [* l
Reset terminal emulation	ESC c
Restore saved features	ESC 8
Save features (char. set, attrib's, cursor, origin)	ESC 7
Select ReGIS mode at command level	ESC P 1 p
Select ReGIS mode at command level (cmds disp)	ESC P 3 p
Select ReGIS mode at level last exited	ESC P p
Select ReGIS mode at level last exited	ESC P 0 p
Select ReGIS mode at level last exited (cmds disp)	ESC P 2 p
Select VT52 mode	ESC [? 2 l
Select VT100 mode from VT320 mode	ESC [61 " p
Select VT300 7 bit mode	ESC [62 ; 1 " p
Select VT300 8 bit mode	ESC [62 " p
Select VT300 8 bit mode (* can be 0 or 2)	ESC [62 ; * " p
Select Viewdata mode (1 = 40 col, 2 = 80, 3 = split)	ESC [? 7 ; * z
Select Wyse 60 mode	ESC [42 h
Set features (* = * in ESC [* h commands)	ESC [* h
Soft reset	ESC [! p
Sound audible tone	BEL
Transmit rate limited to 150-180 cps	ESC [? 73 h
Transmit rate unlimited	ESC [? 73 l

KEYBOARD

Backspace key performs backspace only	ESC [? 67 h
Backspace key performs delete	ESC [? 67 l
Data processing keys mode	ESC [? 68 h
Disable key autorepeat	ESC [? 8 l
Disable keyboard	ESC `
Disable keyboard input	ESC [2 h
Enable key autorepeat	ESC [? 8 h
Enable keyboard	ESC b
Enable keyboard input	ESC [2 l
Select cursor key application mode	ESC [? 1 h
Select cursor key normal mode	ESC [? 1 l
Select keypad application mode	ESC =
Select keypad numeric mode	ESC >
Typewriter keys mode	ESC [? 68 l

LOCAL EDITING

Data block for transmission is cursor line	ESC [? 11 h
Data block for transmission is page	ESC [? 11 l
Disable transmission of protected areas	ESC [1 l
Edit key changes mode immediately	ESC [? 16 h
Edit key waits for host to enable mode change	ESC [? 16 l
Enable all selected areas to be transmitted	ESC [15 h
Enable only cursor area to be transmitted	ESC [15 l
Enable transmission of all characters	ESC [17 h
Enable transmission of protected areas	ESC [1 h
Enable transmission of selected characters only	ESC [17 l
End of block indicator character(s)	ESC [* |
(0 = no, 1 = FF, 2 = ETX, 3 = EOT, 4 = CR, 5 = DC3)	
End selected area	ESC G
Enter edit mode	ESC [? 10 h
Enter interactive mode	ESC [? 10 l
Function according to ANSI rules	ESC [? 53 l
Function as VT131 terminal	ESC [? 53 h
Line termination characters (ASCII decimal)	ESC [? * ' s
Space compression mode off	ESC [? 13 l
Space compression mode on	ESC [? 13 h
Start selected area	ESC F
Transmission occurs immediately	ESC [? 14 h
Transmission waits for host	ESC [? 14 l
Transmit block of data	ESC 5
Transmit scrolling region	ESC [16 h
Transmit VT131 or ANSI partial page	ESC [16 l

PRINTING

Auto print off	ESC [? 4 i
Auto print on	ESC [? 5 i
Form feed at end of print	ESC [? 18 h
No form feed at end of print	ESC [? 18 l
Print controller on	ESC [5 i
Print controller off	ESC [4 i

Print cursor line	ESC [? 1 i
Print page	ESC [i
Print page prints complete page	ESC [? 19 h
Print page prints scrolling region only	ESC [? 19 l

REPORTS

Report compatibility level	ESC [> c
Report cursor position	ESC [6 n
Report keyboard nationality	ESC [? 26 n
Report operating status	ESC [5 n
Report terminal emulation mode	ESC # ! 0
Report VT terminal identity	ESC [0 c
Report VT terminal identity	ESC [c
Report VT terminal identity	ESC Z

ANSI VT500 Emulation

When running the VT500 7 or 8 bit emulation, the following commands will be executed in addition to those listed for ANSI VT100.

CHARACTER ATTRIBUTES

Assign * attribute(s) to following characters				ESC [* m
Attributes:	Invisible	8	Flashing off	25
	Bold off	22	Reverse video off	27
	Underline off	24	Invisible off	28

Non-erase attribute on	ESC [1 " q
Non-erase attribute off (* = 0 or 2)	ESC [* " q

CHARACTER SET SELECTION

Assign G2 label to * character set		ESC * * (second * is parameter)
Assign G3 label to * character set		ESC + *
Extra character sets:	DEC Additional	%5
	ISO Latin-1 Additional	A
	DOC-18 (optional)	D

Assign G1 labelled set to 8 bit codes	ESC ~
Assign G2 labelled set to 7 bit codes	ESC n
Assign G2 labelled set to 7 bit codes for 1 character	ESC N
Assign G2 labelled set to 8 bit codes	ESC }
Assign G3 labelled set to 7 bit codes	ESC o
Assign G3 labelled set to 7 bit codes for 1 character	ESC O
Assign G3 labelled set to 8 bit codes	ESC |
Clear redefinable character set	ESC P 1 ; 1 ; 2 { SP @ ESC \
Load redefinable character set	ESC P * { * ESC \
Preferred Additional set is DEC	ESC P 0 ! u %5 ESC \
Preferred Additional set is ISO Latin-1	ESC P 1 ! u A ESC \
Select Multinational character set mode	ESC [? 42 l
Select National character set mode	ESC [? 42 h

DISPLAY

Display host-writable status line	ESC [2 $ ~
Display indicator status line	ESC [1 $ ~
Display no status line	ESC [0 $ ~
Send data to main display	ESC [0 $ }
Send data to status line	ESC [1 $ }
Display time on status line (*h = hour (24) *m = minutes)	ESC [*h ; *m , p
Display controls on	ESC [3 h
Display controls off	ESC [3 l
Rectangular fill (*c = fill character, *sr = start row *sc = start column, *er = end row, *ec = end column	ESC [*c ; *sr ; *sc ; *er ; *ec $ z

EDITING

Selective erase line (0 = from, 1 = to, 2 = all)	ESC [? * K
Selective erase screen (0 = from, 1 = to, 2 = all)	ESC [? * J

GENERAL OPERATION

Enter HP700/92 mode (0 = zero)	ESC & k 0 \
Select C1 7 bit control mode	ESC SP F
Select C1 8 bit control mode	ESC SP G

KEYBOARD

Set national keyboard — ESC [2 ; * space }

* =			
1	American	14	French
2	British	15	Spanish
3	Belgian	16	Portuguese
4	Canadian	19	Hebrew
5	Danish	22	Greek
6	Finnish	29	Turkish
7	German	31	Hungarian
8	Dutch	33	Slovak
9	Italian	34	Czech
10	Swiss French	35	Polish
11	Swiss German	36	Romanian
12	Swedish	38	Serbian
13	Norwegian	39	Russian

Select application keypad mode	ESC [? 66 h
Select numeric keypad mode	ESC [? 66 l
Set DEC keyboard mode	ESC [? 99 h
Reset DEC keyboard mode	ESC [? 99 l

PF KEYS

Clear all PF keys	ESC P 0 ; 1 | ESC \
Lock PF keys	ESC P 1 ; 0 | ESC \
Program a PF key	ESC P * ; * | * / * ESC \

PRINTING

Enable bidirectional printing	ESC [7 i
Disable bidirectional printing	ESC [6 i
Enable printer/tablet to talk directly to host (one way)	ESC [? 9 i
Disable printer/tablet from talking directly to host	ESC [? 8 i
Destination port (only first * is parameter: 0 = None, 1 = LPT 1/2, 2 = COM 1, 3 = COM 2, etc.)	ESC [* * u

REPORTS

Request colour table report	ESC [2 $ u
Request control function settings	ESC P $ q
Request cursor information report	ESC [1 $ w
Request emulation state report	ESC [1 $ u
Request locator device port status	ESC [? 55 n
Request locator device type	ESC [? 56 n
Request mode settings	ESC [* $ p
Request tab stop report	ESC [2 $ w
Request user-preferred Additional set	ESC [& u
Restore colour table	ESC P 2 $ p
Restore cursor information	ESC P 1 $ t
Restore emulation state	ESC P 1 $ p
Restore tab stops	ESC P 2 $ t

ANSI VT420 Emulation

When TeemTalk is running the VT420 emulation, the following commands will be executed in addition to those listed for ANSI VT100 and VT500.

CURSOR MOVEMENT & PANNING

Back index	ESC 6
Forward index	ESC 9
Pan down (*l = number of lines)	ESC [*l S
Pan up (*l = number of lines)	ESC [*l T
Vertical cursor coupled mode	ESC [? 61 h
Vertical cursor uncoupled mode	ESC [? 61 l
Page cursor coupled mode	ESC [? 64 h
Page cursor uncoupled mode	ESC [? 64 l

EDITING

Delete column(s) (*c = number of columns to delete)	ESC [*c ' ~
Insert column(s) (*c = number of columns to insert)	ESC [*c ' }

GENERAL OPERATION

Secure reset (*n = any number in range 0 - 16383)	ESC [*n + p
Secure reset confirmation (*n = number in range 0 - 16383)	ESC [*n * q

MACROS

Define macro — ESC P *n ; *d ; *e ! z D...D ESC \

*n = Macro ID number	0-63
*d = Delete all macros	0
Delete current macro	1
*e = Encoding format for macro text:	
Standard ASCII characters	0
Hex pairs for each ASCII character	1
Control data string	D...D
Repeat sequence introducer	!

Invoke macro (*n = macro ID number) — ESC [*n * z

KEYBOARD

Enable local functions — ESC [*n ; *c ; ... *n ; *c + q

*n = Function number:		*c = Control performed:	
All local functions	0	Factory default	0
Local copy & paste	1	Enable local function	1
Local panning	2	Disable local function	2
Local window resize	3		

Local function key control — ESC [*k ; *f ; ... *k ; *f * }

*k = Function key number:

All local function keys	0
F1 or Hold	1
F2 or Print	2
F3 or Set-Up	3
F4 or Session	4

*f = Function performed:

Factory default	0
Local function	1
Send key sequence	2
Disable key	3

Select modifier key reporting — ESC [*k ; *c ... *k ; *c + r

*k = Key number:

All keys	0
Left Shift	1
Right Shift	2
Lock key	3
Ctrl key	4
Left Alt Function	5
Right Alt Function	6
Left Compose Char	7
Right Compose Char	8

*c = Control performed:

Factory default	0
Modifier function	1
Extended keyboard report	2
Key disabled	3

PAGE MEMORY

Set lines per page — ESC [*l t

Session:	Dual	Single	
*l =	3 pages	6 pages	24
	2 pages	5 pages	25
	2 pages	4 pages	36
	1 page	3 pages	48
	1 page	2 pages	72
	-	1 page	144

Set left & right margins (*l = left column, *r = right) — ESC [*l ; *r s
Vertical split screen mode - L&R margins can be changed — ESC [? 69 h
Vertical split screen mode - L&R margins cannot be changed — ESC [? 69 l
Move cursor to page *n at same position — ESC [*n SP P

RECTANGULAR AREA OPERATIONS

Copy rectangular area — ESC [*t ; *l ; *b ; *r ; *s ; *dt ; *dl ; *dp $ v

*t	Top-line border	*s	Source page number
*l	Left-column border	*dt	Destination top-line border
*b	Bottom-line border	*dl	Destination left-column border
*r	Right-column border	*dp	Destination page number

Erase rectangular area — ESC [*t ; *l ; *b ; *r $ z

*t	Top-line border	*b	Bottom-line border
*l	Left-column border	*r	Right-column border

Fill rectangular area — ESC [*f ; *t ; *l ; *b ; *r $ x

*f Decimal code of fill character
*t Top-line border
*l Left-column border
*b Bottom-line border
*r Right-column border

Selective erase rectangular area — ESC [*t ; *l ; *b ; *r $ {

*t Top-line border
*l Left-column border
*b Bottom-line border
*r Right-column border

Select attribute change extent — ESC [*c * x

*c = character positions affected:
Stream of character positions — 0 or 1
Rectangular area of character positions — 2

Change attributes in rectangular area — ESC [*t ; *l ; *b ; *r *a $ r

*t Top-line border
*l Left-column border
*b Bottom-line border
*r Right-column border
*a Visual character attributes

Reverse attributes in rectangular area — ESC [*t ; *l ; *b ; *r *a $ t

*t Top-line border
*l Left-column border
*b Bottom-line border
*r Right-column border
*a Visual character attributes

VT420 REPORTS

Tertiary device attribute request — ESC [= c *or* ESC [= 0 c
Request extended cursor position report — ESC [? 6 n
Request checksum of rectangular area — ESC [*id ; *p ; *t ; *l ; *b ; *r * y

*id Request label
*p Page number
*t Top-line border
*l Left-column border
*b Bottom-line border
*r Right-column border

Request macro space report — ESC [? 62 n
Request memory checksum report (*l = request label) — ESC [? 63 ; *l n
Request multiple session status report — ESC [? 85 n
Request window report — ESC [" v

ANSI VT510 Emulation

When TeemTalk is running the VT510 emulation, the following commands will be executed in addition to those listed for ANSI VT100 and VT500.

USER-DEFINED KEYS

Download definitions for user-defined keys DCS *c ; *l ; *m | D...D ST

*c	0 *or* none	Clear all keys before loading new values (0 is default).
	1	Load new UDK values; clear old values only when redefined.
*l	0 *or* none	Lock the keys.
	1	Do not lock the keys against future redefinition.
*m	0, 2 *or* none	Defines the shifted function key.
	1	Defines the unshifted function key.
	3	Defines the alternate unshifted function key.
	4	Defines the alternate shifted function key.
|	(vertical bar)	The final character. Key definition strings follow and these are terminated by ST.

D...D are the key definition strings in the following format:

Key1/UDS/UDS Direction;Key2/UDS/UDS Direction;.....;

Key# Is the key selection number of the key to be defined as listed below:

F1	11	F5	15	F9	20
F2	12	F6	17	F10	21
F3	13	F7	18	F11	23
F4	14	F8	19	F12	24

/ Is a delimiter.

UDS Is the user-defined string consisting of hex pairs in the following ranges:

3/0 through 3/9 (0 through 9)
4/1 through 4/6 (A through F)
6/1 through 6/6 (a through f)

For example, the hex encoding for "PRINT" would be as follows: 5 0 5 2 4 9 4 E 5 4

This enables you to use any of the 256 character codes in the key string. You can enter key definition strings in any order.

UDS Direction Specifies the transmission direction:

0 *or* none	Normal (Host and/or terminal) default
1	Local (Terminal only)
2	Remote (Host only)

PROGRAMMING ALPHANUMERIC KEYS

Program Alphanumeric Key(s) DCS " y D...D ST

where the data string D...D format is as follows:

Key1/Hex Code String/Function #/UDS/UDS Direction;Key2/...;

Key# Is the key station number of the key to be programmed as listed below:

Esc	110	F10	121	Page Up	85
F1	112	F11	122	Page Down	86
F2	113	F12	123	Cursor Left	79
F3	114	Print Screen	124	Cursor Right	89
F4	115	Scroll Lock	125	Cursor Up	83
F5	116	Pause	126	Cursor Down	84
F6	117	Insert	75	Keypad Enter	108
F7	118	Delete	76	Return	43
F8	119	Home	80	Backspace	15
F9	120	End	81	Tab	16

/ Is a delimiter.

Hex Code String Specifies the hex code of the character to be transmitted with each of the four modifier states in the following order: Unshifted, Shifted, Alternate Shifted (Shift 2), and Control (if omitted, use default). Use a period "." as a place holder for an undefined modifier combination.

The hex code represents a valid code in the current 7-bit or 8-bit character set.

Use a minus "-" preceding the hex representation of a diacritical sign if a diacritical mark is to be defined.

Function # Is a number associated with a local function as listed below. For example, function number "0" makes the key or key/modifier inoperative. Function number "100" indicates a user-defined sequence (UDS), and a UDS direction is defined following the slash "/" delimiter.

0	No Function	91	Backspace	93	Escape
100	UDS	92	Cancel	94	Delete

UDS Is the user-defined sequence specified in Hex format. Each hex code in this string represents a value transmitted by the defined key combined with the modifiers.

UDS Direction Specifies the transmission direction:

0 *or* none	Normal (Host and/or terminal) default
1	Local (Terminal only)
2	Remote (Host only)

PROGRAMMING FUNCTION KEYS

Program Function Key(s) DCS " x D...D ST

where the data string D...D format is as follows:

Key1/Mod1/Function #/UDS/UDS Direction;
Key2/Mod2/Function #/UDS/UDS Direction;.....;

Key# Is the key station number of the key to be programmed as listed below:

Esc	110	F10	121	Page Up	85
F1	112	F11	122	Page Down	86
F2	113	F12	123	Cursor Left	79
F3	114	Print Screen	124	Cursor Right	89
F4	115	Scroll Lock	125	Cursor Up	83
F5	116	Pause	126	Cursor Down	84
F6	117	Insert	75	Keypad Enter	108
F7	118	Delete	76	Return	43
F8	119	Home	80	Backspace	15
F9	120	End	81	Tab	16

/ Is a delimiter.

Mod# Is an integer that specifies the modifier key that is to be pressed at the same time as the defined key:

0 *or* none	Normal	5	Control
1	Normal	6	Shift+Control
2	Shift	7	Alt+Control
3	Alt	8	Alt+Control+Shift
4	Alt+Shift		

Function # Is a number associated with a local function as listed below. For example, function number "0" makes the key or key/modifier inoperative. Function number "100" indicates a user-defined sequence (UDS), and a UDS direction is defined following the slash "/" delimiter.

0	No Function	91	Backspace	93	Escape
100	UDS	92	Cancel	94	Delete

UDS Is the user-defined sequence specified in Hex format. Each hex code in this string represents a value transmitted by the defined key combined with the modifiers.

UDS Direction Specifies the transmission direction:

0 *or* none	Normal (Host and/or terminal) default
1	Local (Terminal only)
2	Remote (Host only)

ANSI VT520 Emulation

When running the VT520 emulation, the following commands will be executed in addition to those listed for ANSI VT100 and VT500. Note that an * (asterisk) before a command description indicates the command is accepted but not actioned.

Sessions

* Enable session	CSI & x
* Session page memory (*ps# = pages per session 1-4)	CSI *ps1 ; *ps2 ; *ps3 ; *ps4 ; , x
* Update session	CSI * , y
* = Only when active	1
When available	2
At regular intervals	3

Window Management

Auto reset mode enabled	CSI ? 98 h
Auto reset mode disabled	CSI ? 98 l
Framed windows mode enabled	CSI ? 111 h
Framed windows mode disabled	CSI ? 111 l
* Set icon name (12 characters max.)	OSC 2 L ; *name* ST
Set window title (30 characters max.)	OSC 2 1 ; *name* ST

Audible Attributes

* Margin bell volume	CSI * SP u
* = Off	none, 0, 1
Low	2, 3, 4
High	5, 6, 7, 8
* Warning bell volume	CSI * SP t
* = Off	1
Low	2, 3, 4
High	none, 0, 5, 6, 7, 8
* Play sound	CSI *v , *d ; *n , ~
*v = Volume	0 - 7 (0 = silent)
*d = Duration	0 - 255 (1/32nd of a sec)

*n = Note	0	Silent	9	G#5	18	F6
	1	C5	10	A5	19	F#6
	2	C#5	11	A#5	20	G6
	3	D5	12	B5	21	G#6
	4	D#5(Eb)	13	C6	22	A6
	5	E5	14	C#6	23	A#6
	6	F5	15	D6	24	B6
	7	F#5	16	D#6	25	C7
	8	G5	17	E6		

Colour Selection

Assign colour — CSI *i *f *b , |

*i = Item	Normal text	1
	Window frame	2 (not actioned)
*f = Foreground colour index		0 - 15
*b = Background colour index		0 - 15

Alternate text colour — CSI *a *f *b , }

*a = Attribute	0	Normal text	8	Reverse underline
	1	Bold	9	Reverse blink
	2	Reverse	10	Underline blink
	3	Underline	11	Bold reverse underline
	4	Blink	12	Bold reverse blink
	5	Bold reverse	13	Bold underline blink
	6	Bold underline	14	Reverse underline blink
	7	Bold blink	15	Bold reverse underline blink

*f = Foreground colour index	0 - 15
*b = Background colour index	0 - 15

Alternate text colour blink enabled	CSI ? 115 h
Alternate text colour blink disabled	CSI ? 115 l
Alternate text colour underline enabled	CSI ? 114 h
Alternate text colour underline disabled	CSI ? 114 l
Bold and blink foreground & background enabled	CSI ? 116 h
Bold and blink foreground only enabled	CSI ? 116 l
Colour table request (* value: 1 = HLS, 2 = RGB)	CSI 2 ; * $ u
Colour table reply to host (*...* = parameter group)	DCS 2 $ s *...* ST
Colour table restore format (*...* = parameter group)	DCS 2 $ p *...* ST

... = group of 5 parameters: *n ; *s ; *1 ; *2 ; *3 /

*n = Colour number: 0 - 255
*s = Colour coordinate system: 0 = illegal, 1 = HLS, 2 = RGB

*1 = HLS hue: 0 - 360	or	RGB red: 0 - 100
*2 = HLS lightness: 0 - 100	or	RGB green: 0 - 100
*3 = HLS saturation: 0 - 100	or	RGB blue: 0 - 100

Select colour look-up table — CSI *) {

* = 0	Mono
1	Alternate colour (use text attributes)
2	Alternate colour
3	ANSI SGR colour

Text Processing

Cursor backward tabulation (* = no. of active tab positions)	CSI * Z
Cursor horizontal absolute (* = no. of active char. positions)	CSI * G
Cursor horizontal forward tab (* = no. of active tab positions)	CSI * I
Cursor next line (* = number of active position)	CSI * E
Cursor previous line (* = number of active position)	CSI * F
Save cursor position (SCO)	CSI s

Restore saved cursor (SCO)		CSI u
Set cursor style		CSI * SP q
	* = Blinking block	none, 0, 1
	Steady block	2
	Blinking underline	3
	Steady underline	4
Right-to-left copy enabled		CSI ? 96 h
Right-to-left copy disabled		CSI ? 96 l
Right-to-left mode enabled		CSI ? 34 h
Right-to-left mode disabled		CSI ? 34 l
Vertical line position absolute		CSI * d
Vertical line position relative		CSI * e
Set tab every 8 columns		CSI ? 5 W
* Set scroll speed		CSI * SP p
	* = Smooth 2	none, 0, 1, 2, 3
	Smooth 4	4, 5, 6, 7, 8
	Jump	9
Clear screen on column change		CSI ? 95 l
Do not clear screen on column change		CSI ? 95 h

Character Sets

Assign user-preferred supplemental character set — DCS *s ! u *c ST

*s = 94-character set 0
96-character set 1

*c = character set as follows:

Default 94-Character Set		Default 96-Character Set	
% 5	DEC Supplemental	A	ISO Latin-1 Supplemental
" ?	DEC Greek	B	ISO Latin-2 Supplemental
" 4	DEC Hebrew	F	ISO Greek Supplemental
% 0	DEC Turkish	H	ISO Hebrew Supplemental
& 4	DEC Cyrillic	L	ISO Latin-Cyrillic
<	User-pref. Supplemental	M	ISO Latin-5 Supplemental
		<	User-pref. Supplemental

* Down-line load allocation — CSI * , z

* = One each 1
Two each S1 & S2 2

Select character set — ESC *g *c

*g = G-set as follows:

94-Character Set		96-Character Set	
(	G0	-	G1
)	G1	.	G2
*	G2	/	G3
+	G3		

*c = character set as follows:

Default 94-Character Set		Default 96-Character Set	
% 4	Cyrillic (DEC)	A	ISO Latin-1 Supplemental
" ?	Greek (DEC)	B	ISO Latin-2 Supplemental
" >	Greek NRCS	F	ISO Greek Supplemental
" 4	Hebrew (DEC)	H	ISO Hebrew Supplemental
% =	Hebrew NRCS	M	ISO Latin-5 Supplemental
% 6	Portuguese NRCS	L	ISO Latin-Cyrillic
& 5	Russian NRCS		
% 3	SCS NRCS		
% 0	Turkish (DEC)		
% 2	Turkish NRCS		

Select zero symbol		CSI * , {
	* = Oval zero	1
	Zero with slash	2
	Zero with dot	3
Request user-preferred supplemental set		CSI & u

Keyboard

Auto repeat rate		CSI * - p
	* = Off	0 - 5
	Slow (10 cps)	6 - 15
	Fast (30 cps)	16 - 30
Copy key default (*...* = Key-source/Key-dest;K-s/K-d...;)		DCS " z *...* ST
Define function key (SCO)		ESC Q *Fn string*
Extended keyboard report		APC : *ppp mm* ST
Hebrew encoding mode set (8-bit char.)		CSI ? 36 h
Hebrew encoding mode reset (7-bit char.)		CSI ? 36 l
Keyboard language selection		CSI *t ; *l SP }
	*t = keyboard type:	
	DEC keyboard layout	0, 1
	Enhanced PC layout	2

*l = keyboard language as follows:

Keyboard language (VT/PC)		Keyboard language (VT/PC)	
none, 0 or 1	North American	15	Spanish
		16	Portuguese
2	British	19	Hebrew
3	Belgian	22	Greek
4	French Canadian	28	Canadian English
5	Danish	29	Turkish Q/Turkish
6	Finnish	30	Turkish F/Turkish
7	German	31	Hungarian
8	Dutch	33	Slovak
9	Italian	34	Czech
10	Swiss-French	35	Polish
11	Swiss-German	36	Romanian

12	Swedish	38	SCS
13	Norwegian	39	Russian
14	French	40	Latin-American

Key position mode set (key position)	CSI ? 81 h
Key position mode reset (char. codes)	CSI ? 81 l
North American/Greek selection set (North American)	CSI ? 57 h
North American/Greek selection reset (Greek keyboard)	CSI ? 57 l
Program alphanumeric key (for *...* see below)	DCS " y *...* ST
Program function key (for *...* see below)	DCS " x *...* ST

... = *Key# / Mod# / Func# / UDS / Dir ; Key# / Mod# / Func# / UDS / Dir ;*

Key#	VT	PC Keyboard
11	F1	F1
12	F2	F2
13	F3	F3
14	F4	F4
15	F5	F5
17	F6	F6
18	F7	F7
19	F8	F8
20	F9	F9
21	F10	F10
23	F11	F11
24	F12	F12
25	F13	
26	F14	
28	F15	
29	F16	
31	F17	
32	F18	
33	F19	
34	F20	

Mod#	Modifier Key
0, none	Normal
1	Normal
2	Shift
3	Alt
4	Alt + Shift
5	Control
6	Shift + Ctrl
7	Alt + Ctrl
8	Alt + Ctrl + Shift

Func# = Local function number

UDS = User-defined sequence

Dir	UDS Direction
0, none	Normal
1	Local
2	Remote

Program key action	CSI * + z
* = None	none, 0
Lock keys & modifiers	1
Restore factory defaults	2
Recall key definitions	3

Program key free memory report (*t = total bytes, *f = free)	CSI *t ; *f + y
Request program key free memory	CSI + x
Request key definition	CSI *n ; *m , w

*n = Key station number

*m = Modifier key:	
Normal	0, none, 1
Shift	2
Alt	3
Alt + Shift	4
Control	5
Shift + Ctrl	6
Alt + Ctrl	7
Alt + Ctrl + Shift	8

Request key type (* = decimal key station number) CSI * , u
Report key type CSI *n ; *t , v

*n = decimal key station number

*t = key type: Alphanumeric key 0
Function key 1

Report function key definition (for *...* see below) DCS " } *...* ST
Report modifiers/key state (for *...* see below) DCS " ~ *...* ST

... = *Key# / Mod# / Func# / UDS / Dir* (except no *Func#* for function key report)

Key#	VT	PC Keyboard
11	F1	F1
12	F2	F2
13	F3	F3
14	F4	F4
15	F5	F5
17	F6	F6
18	F7	F7
19	F8	F8
20	F9	F9
21	F10	F10
23	F11	F11
24	F12	F12
25	F13	
26	F14	
28	F15	
29	F16	
31	F17	
32	F18	
33	F19	
34	F20	

Mod#	Modifier Key
0, none	Normal
1	Normal
2	Shift
3	Alt
4	Alt + Shift
5	Control
6	Shift + Ctrl
7	Alt + Ctrl
8	Alt + Ctrl + Shift

Func# = Local function number

UDS = User-defined sequence

Dir	UDS Direction
0, none	Normal
1	Local
2	Remote

* Set key click volume CSI * SP r

* = Off 1
Low 2, 3, 4
High none, 0, 5, 6, 7, 8

* Set lock key style CSI * SP v

* = Caps lock none, 0, 1
Shift lock 2
Reverse lock 3

* Load LEDs CSI * q

* = Clear all LEDs 0
Light Num lock 1
Light Caps lock 2
Light Scroll lock 3
Extinguish Num lock 21
Extinguish Caps lock 22
Extinguish Scroll lock 23

Printer

Media copy - send screen data		CSI 2 i
Select digital printed data type		CSI *) p
	* = Print National only	none, 0, 1
	National & Line Drawing	2
	Print Multinational	3
	Print all characters	4
* Select printer type		CSI * $ s
	* = DEC ANSI	none, 0, 1
	IBM ProPrinter	2
	DEC + IBM	3

* Select ProPrinter character set — CSI *c * p

*c =	PC Greek	210	PC Portuguese	860
	PC Spanish	220	PC Hebrew	862
	PC International	437	PC French-Canada	863
	PC Multilingual	850	PC Danish-Norway	865
	PC Slavic (Latin 2)	852	PC Cyrillic	866
	PC Turkish	857		

Terminal Management

* Select Setup language		CSI * p
	* = English	none, 0, 1
	French	2
	German	3
	Spanish	4
	Italian	5
Load time of day (*h = hour 00 - 23, *m = minutes 00 - 59)		CSI *h ; *m , p
* CRT saver timing		CSI * - q
	* = Never	0
	5 min	5
	15 min	15
	30 min	30
	60 min	60
* Energy saver timing		CSI * - r
	* = Never	0
	5 min	5
	15 min	15
	30 min	30
Report terminal ID (*...* = 4 hexadecimal pairs)		DCS ! | *...* ST
Set terminal unit ID (*...* = 4 hexadecimal pairs)		DCS ! { *...* ST

Enter/exit VT PC Term mode / select ASCII or scancodes — CSI ? *m ; *c r

*m = VT mode or SCO ASCII/Scancodes as follows:

VT Emulation Mode	SCO Console Mode	
VT Mode	ASCII characters	none, 0
VT PCTerm Mode	Scancodes	1

*c = character set as follows:

VT Emulation Mode		SCO Console Mode	
210	PC Greek	none, 0	No change
220	PC Spanish	1	PC Multilingual
437	PC International	2	PC International
850	PC Multilingual	3	PC Danish/Norwegian
852	PC Slavic (Latin-2)	4	PC Spanish
857	PC Turkish	5	PC Portuguese
860	PC Portuguese	6	DEC Supplemental
862	PC Hebrew	7	ISO Latin-1
863	PC French-Canadian		
865	PC Danish		
866	PC Cyrillic		

Select conformance (operating) level — CSI *l ; *b ; " p

*l = level:	1 - VT100	none, 0, 1
	5 - VT520/525	2, 3, 4, 5
*b = data bits:	8-bit controls	none, 0, 2
	7-bit controls	1

Terminal mode emulation — CSI * SP ~

* = terminal mode:	VT520/525 (VT level 5)	1
	VT100	2
	VT52	3
	VT PCTerm	4
	WYSE 160/60	5
	WYSE PCTerm	6
	WYSE 50/50+	7
	WYSE 150/120	8
	TVI 950	9
	TVI 925	10
	TVI 910+	11
	ADDS A2	12
	SCO Console	13
	WYSE 325	14

Auto answerback mode set — CSI ? 100 h
Auto answerback mode reset — CSI ? 100 l
Conceal answerback message mode set — CSI ? 101 h
Conceal answerback message mode reset — CSI ? 101 l
Load answerback message (*...* = data string of hex pairs) — CSI 1 v *...* ST
Load banner message (*...* = string of up to 30 chars.) — CSI *e r *...* ST

*e = message encoding:	ASCII hexadecimal pairs	1
	Text as VT default char. set	2, none, 0

Status line type — CSI * $ ~

* =	No status line	0
	Indicator status line	1
	Host-writable status line	2

Communications

Select communication port — CSI *p ; *h * u

*p = printer port:

None	0	Comm1	1
Centronics	1	Comm2	2
Comm3	4	Comm3	3

*h = host port:

(note that the last * is part of the command)

Select communication speed — CSI *l ; *s * r

*l = comm line:

Host transmit	none, 0, 1
Host receive	2
Printer	3
Modem Hi	4
Modem Lo	5

*s = speed:

Use default	none, 0
300	1
600	2
1200	3
2400	4
4800	5
9600	6
19200	7
38400	8
57600	9
76800	10
115200	11

(note that the last * is part of the command)

Set port parameters — CSI *p ; *d ; *y ; *s + w

*p = port:

Comm port	none, 0, 1
Printer port	2

*d = data:

8 bits	1, 0, none
7 bits	2

*y = parity:

No parity	none, 0, 1
Even	2
Odd	3
Even	4
Odd, unchecked	5
Mark	6
Space	7

*s = stop bits:

1 bit	none, 0, 1
2 bits	2

Set transmit rate limit — CSI *k ; *s " u

*k = key type:	All keys	none, 0, 1
	Graphic key	2
	Function key	3

*s = characters per second:	150 cps	none, 0, 1
	50 cps	2
	30 cps	3

Transmit rate limiting mode set (limited) CSI ? 73 h
Transmit rate limiting mode reset (unlimited) CSI ? 73 l

Terminal Synchronization

Disconnect delay time CSI * $ q

* = delay time:	None	1
	60 ms	2
	2 seconds	none, 0, 3

Flow control CSI *p ; *d ; *f ; *t * s

*p = port type:

Comm port	none, 0, 1
Printer port	2

*d = direction:

Transmit	1, 0, none
Receive	2
Tx & Receive	3

*f = flow control type:

XON/XOFF or XPC	1, 0, none
DTR	2
Both	3
None	4

*t = flow control threshold:

Low (64)	none, 0, 1
High (768)	2

(note that the last * is part of the command)

Null mode set (ignore NUL) CSI ? 102 h
Null mode reset (accept NUL) CSI ? 102 l

AIXTerm Emulation

All commands are supported in both VT100 and HFT (High Function Terminal) modes unless indicated otherwise.

SINGLE BYTE CONTROLS

Bell	BEL
Backspace	BS
Horizontal tab	HT
Linefeed	LF
Vertical tab	VT
Form feed	FF
Carriage return	CR
Shift out	SO
Shift in	SI
Device control 1	DC1
Device control 3	DC3
Cancel	CAN
Substitute	SUB
Escape	ESC

CHARACTER ATTRIBUTES

Assign * attribute(s) to following characters	ESC [* m
Normal	0
Bold	1
Underscore	4
Blink (appears bold)	5
Reverse	7
Invisible (HFT)	8
Foreground colours (HFT)	30...37
Background colours (HFT)	40...47
Foreground colours (HFT)	90...97
Background colours (HFT)	100...107

CHARACTER SET SELECTION

United Kingdom character set G0 (VT100)	ESC (A
United Kingdom character set G1 (VT100)	ESC) A
United Kingdom character set G2 (VT100)	ESC * A
United Kingdom character set G3 (VT100)	ESC + A
ASCII (USASCII) character set G0 (VT100)	ESC (B
ASCII (USASCII) character set G1 (VT100)	ESC) B
ASCII (USASCII) character set G2 (VT100)	ESC * B
ASCII (USASCII) character set G3 (VT100)	ESC + B
Special graphics character set G0 (VT100)	ESC (0
Special graphics character set G1 (VT100)	ESC) 0
Special graphics character set G2 (VT100)	ESC * 0
Special graphics character set G3 (VT100)	ESC + 0
Single shift G2 (VT100)	ESC N

Single shift G3 (VT100)	ESC O
Set G0 character set (HFT)	ESC (<
Set G1 character set (HFT)	ESC) <
Lock shift G2 (VT100)	ESC n
Lock shift G3 (VT100)	ESC o

COLOUR

Set foreground & background colour — ESC [2 ; *fg ; *bg m

*fg and *bg are integers as listed below:

Dull	Colour	Bold
0	Black	8
1	Blue	9
2	Green	10
3	Cyan	11
4	Red	12
5	Magenta	13
6	Yellow	14
7	White	15

Set inverse & foreground & background colour (* as above)	ESC [7 ; *fg ; *bg ; m
Set normal foreground colour (* = integer as above)	ESC [= * F
Set normal background colour (* = integer as above)	ESC [= * G
Set reverse foreground colour (* = integer as above)	ESC [= * H
Set reverse background colour (* = integer as above)	ESC [= * I
Set graphic foreground colour (* = integer as above)	ESC [= * J
Set graphic background colour (* = integer as above)	ESC [= * K
Request current colour attribute (* = integer as above)	ESC [= * M

CURSOR

Clear tab stop — ESC [* g

Clear horizontal tab stop at active position	0
Vertical tab at cursor line (HFT)	1
Horizontal tabs on line (HFT)	2
All horizontal tabs	3
All vertical tabs (HFT)	4

Vertical tab stop (HFT)	ESC I
Cursor backward * tabs	ESC [* Z
Cursor horizontal absolute	ESC [* G
Cursor forward * tabs (HFT)	ESC [* I
Cursor tab stop control (HFT)	ESC [* W
Cursor down * lines (HFT)	ESC [* E
Cursor up * lines	ESC [* F
Horizontal tab stop	ESC H
Index cursor (move down one line)	ESC D
Reverse index cursor (move up one line)	ESC M
Move cursor to beginning of next line	ESC E
Move cursor left * columns	ESC [* D
Move cursor down * lines	ESC [* B
Move cursor right * columns	ESC [* C
Move cursor to line (*l) column (*c)	ESC [*l ; *c f

Move cursor to line (*l) column (*c)	ESC [*l ; *c H
Move cursor up * lines	ESC [* A
Move cursor up * tabs (HFT)	ESC [* Y
Restore cursor position	ESC [u
Save cursor position	ESC [s

DISPLAY

Restore cursor & attributes	ESC 8
Save cursor & attributes	ESC 7
Scroll display down * lines (HFT)	ESC [* T
Scroll display up * lines	ESC [* S
Scroll left * columns (HFT)	ESC [* SP @
Scroll right * columns (HFT)	ESC [* SP A
Select screen direction (HFT)	ESC [* 1 ; 1 S
Left-to-right, set to Latin keyboard	0
Right-to-left, set to National keyboard	1
Screen alignment display	ESC # 8
Set top & bottom margins	ESC [*t ; *b r
Erase status line	ESC [? E
Return from status line	ESC [? F
Hide status line	ESC [? H
Show status line	ESC [? S
Go to column * of status line	ESC [? * T
Set text parameters	ESC] * ; *t \007
Change window name & title to *t	0
Sets only the icon name	1
Sets only the title name	2

EDITING

Delete * characters from cursor position right	ESC [* P
Delete * lines from cursor position down	ESC [* M
Erase * characters from cursor right	ESC [* X
Erase area (* is one of the following)	ESC [* O
Erase to end of area	0
Erase from area start	1
Erase entire area	2
Erase display (* is one of the following)	ESC [* J
Erase to end of display	0
Erase from display start	1
Erase entire display	2
Erase field (* is one of the following)	ESC [* N
Erase to end of field	0
Erase from field start	1
Erase entire field	2
Erase line (* is one of the following)	ESC [* K
Erase to end of line	0
Erase from line start	1
Erase entire line	2

Insert * blank lines	ESC [* L
Insert * space characters	ESC [* @

GENERAL OPERATION

Lock shift G2	ESC n
Lock shift G3	ESC o
Reset to initial state	ESC c
ANSI specified modes	ESC [*;...;* h
IRM insert mode	4
SRM send/rec mode (HFT)	12
TSM tab stop mode (HFT)	18
LNM linefeed/newline	20
XTERM private modes	ESC [? *;...;* h
132/80 column mode	40
Scrollbar hide/show	42
Save scroll text on/off	43
Margin bell on/off	44
Reverse wraparound on/off	45
Screen buffer alternate/normal	47
Status line reverse/normal	48
Scroll mode page/normal	49
Other private modes	ESC [? *;...;* h
Normal/application cursor (VT100)	1
80/132 columns	3
Reverse/normal video	5
Origin/normal	6
Autowrap on/off	7
Autorept on/off	8
CNM CR-NL (HFT)	21
Reset mode, ANSI specified modes	ESC [* ;...; *
Reset mode, other private modes & XTERM private modes	ESC [? *;...;* l
Restore mode, other private modes & XTERM private modes	ESC [? *;...;* r
Save mode, other private modes & XTERM private modes	ESC [? *;...;* s
Ignore everything between ESC - P and ESC \.	
AIXTerm works as normal after ESC \	ESC - P ... ESC \

KEYBOARD & MOUSE

Select numeric keypad application mode (VT100)	ESC =
Select numeric keypad normal mode (VT100)	ESC >
Disable manual input (HFT)	ESC `
Enable manual input (HFT)	ESC b

REPORTS

Device status report (* is one of the following)	ESC [* n
Response from VT100: ready	0
Command from host: please report status	5
Command from host: report active position	6
Keyboard status information	ESC [* p

PF key report (HFT)	ESC [* q
Report cursor position	ESC [*l ; *c R
Report terminal identity (host to VT100)	ESC [c
Report terminal identity (host to VT100)	ESC [0 c
Terminal response (VT100 to host)	ESC [? 1 ; 2 c

AT&T 4410 Emulation

CHARACTER ATTRIBUTES

Normal	ESC [0 m
Dimmed	ESC [2 m
Underline	ESC [4 m
Flashing	ESC [5 m
Reverse video	ESC [7 m
Blanked	ESC [8 m

CURSOR

Cursor down one line or * lines	ESC [B *or* ESC [* B
Cursor left one column or * columns	ESC [D *or* ESC [* D
Cursor right one column or * columns	ESC [C *or* ESC [* C
Cursor up one line or * lines	ESC [A *or* ESC [* A
Cursor to line 1, column 1	ESC [H
Origin mode	ESC [? 6 h
Origin mode reset	ESC [? 6 l
Move cursor to line (*l) and column (*c)	ESC [*l ; *c f
Move cursor to line (*l) and column (*c)	ESC [*l ; *c H
Save cursor location	ESC 7
Restore cursor location	ESC 8
Reverse index	ESC M

DISPLAY

132 column display	ESC [? 3 h
80 column display	ESC [? 3 l
Clear all	ESC [2 J
Clear to cursor	ESC [1 J
Clear to end of display	ESC [J
Clear to end of line	ESC [K *or* ESC [0 K
Clear line to cursor	ESC [1 K
Clear entire line	ESC [2 K
Download label entry (*1 = 1-8, *2 = no. of chrs in def.)	ESC [*1 ; *2 q label definition
Reverse video screen	ESC [5 ? h
Reverse video screen reset	ESC [5 ? l
Scrolling region	ESC [*1 ; *2 r

EDITING

Delete character or * characters	ESC [P *or* ESC [* P
Delete line or * lines	ESC [M *or* ESC [* M
Insert character or * characters	ESC [@ *or* ESC [* @
Insert line or * lines	ESC [L *or* ESC [* L

GENERAL OPERATION

Reset terminal	ESC c
Report status	ESC [5 n
Test passed	ESC [0 n

BQ 3107 Emulation

CHARACTER DISPLAY

Blank area rendition	~
Blink area rendition	^
Select graphic rendition	ESC [* ; * ; * m
Select G0, national or ASCII character set	SI
Select G1 graphic symbols character set	SO
Select G2 accented characters and special symbols set	ESC E

CURSOR

Cursor addressing (1 to 80 = SP to g	DC3 *line *column
Cursor position is end of next transmitted text	ESC U
Cursor position is start of next transmitted text	ESC T
Cursor On (* = 0) or Off (* = 1)	ESC [* r
Move cursor to left margin of current line	CR
Move cursor one column left	BS
Move cursor one column right	DC2
Move cursor to begining of partition or field	DC4
Move cursor down one line	LF
Move cursor up one line	DC1
Move cursor to next tab stop	HT
Set tab stop	ESC 1
Cancel tab stop	ESC 2

ERASE

Active partition erasure	FF
Erase in field (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* N
Erase in line (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* K
Erase in partition (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* J

FIELDS

Start of a fixed field	FS
Start of a variable field	GS *
Repeat fields	RS X

GENERAL

Buffer character repetition (* = @ to DEL)	US *
Define area qualification	ESC [* o
Format mode	ESC M
Normal mode	ESC N
Message waiting	ESC R
Set mode (? = SDP mode, ? < extended mode)	ESC [? = ; ? < ; ? 0 h
Reset mode (? = VIP mode, ? < 94 character mode)	ESC [? = ; ? < ; ? 0 l
Reset both partitions	ESC c
Select current partition (0 = partition 0, 1 = partition 1)	ESC [* v
Warning (bell)	BEL

DG 410/412 Emulation

DG Mode

In the following command sequences, <n> represents a 1-byte argument, <nn> represents a 2-byte argument, and <nnn> represents a 3-byte argument. Bytes are entered as ASCII characters. Only the four least significant bits are used in each byte.

Commands that are D100/200 compatible are indicated by an ◆.

CHARACTER ATTRIBUTES

Change attributes (<count><on><off>)	RS F N <nnn><n><n>
◆ Blink enable	ETX
◆ Blink disable	EOT
◆ Blink on	SO
◆ Blink off	SI
◆ Dim on	FS
◆ Dim off	GS
Protect enable	RS F V
Protect disable	RS F W
Protect on	RS F L
Protect off	RS F M
◆ Reverse video on	RS D *or* SYN
◆ Reverse video off	RS E *or* STX
◆ Underscore on	DC4
◆ Underscore off	NAK

CURSOR

◆ Carriage return	CR
◆ Cursor left	EM
◆ Cursor right	CAN
◆ Cursor down	SUB
◆ Cursor up	ETB
◆ New line	LF
Read screen address	RS F b
Write screen address (<column><row>)	RS F P <nn><nn>
◆ Read window address	ENQ
◆ Write window address	DLE *column *row
Screen home	RS F G
◆ Window home	BS
Set cursor type	RS F Q <nn>

DISPLAY

◆ Roll enable	DC2
◆ Roll disable	DC3

Scroll down	RS I
Scroll up	RS H
Scroll left	RS F C <n>
Scroll right	RS F D <n>
Select compressed spacing	RS F K
Select normal spacing	RS F J
Set margins	RS F X <nn><nn>
Set alternate margins	RS F Y <nn><nn><nn>
Set scroll rate	RS F T <n>
Set windows (<rows><0 = 81 cols, 1 = 135 cols>)	RS F B <nn><n>...
Show columns	RS F <nn><nn>
Horizontal scroll enable	RS F ^
Horizontal scroll disable	RS F]
Restore normal margins	RS F Z

EDITING

Delete character	RS K
Delete line	RS F I
Delete line between margins	RS F \
◆ Erase window	FF
Erase screen	RS F E
Erase unprotected	RS F F
◆ Erase to end of line	VT
Insert character	RS J
Insert line	RS F H
Insert line between margins	RS F [

GENERAL

◆ Bell	BEL
Read horizontal scroll offset	RS F O
◆ Read model ID	RS C
Reset	RS F A
Select ANSI mode	RS F @
Select 7/8 bit operation (0 = 7-bit, 1 = 8-bit)	RS F U *bit
Select character set	RS F S <nn>
Set keyboard language (0 = match native language, 1 = US ASCII & DG International)	RS F f *
Shift in	RS N
Shift out	RS O

PRINTING

Form bit dump	RS F ? 6
Window bit dump	RS F ? 5
◆ Print form	SOH
Print pass through on	RS F `
Print pass through off	RS F a
◆ Print window	DC1

DG410 ANSI Standard Mode

CHARACTER ATTRIBUTES

Change attributes	CSI *count ; *on ; *off q

CURSOR

Backspace	BS
Carriage return	CR
Cursor left * columns	CSI * D
Cursor right * columns	CSI * C
Cursor down * lines	CSI * B
Cursor up * lines	CSI * A
Cursor position (line ; column)	CSI *l ; *c f
Cursor position (line ; column)	CSI *l ; *c H
Form feed	FF
Index	ESC D
Insert * space characters	CSI * @
Insert * lines	CSI * L
New line	LF
Next line	ESC E

EDITING

Delete * character(s) from cursor	CSI * P
Delete * lines from & including cursor line	CSI * M
End protected area	ESC w
Erase line portion (0 = from cursor, 1 = to, 2 = all)	CSI * K
Erase window portion (0 = from cursor, 1 = to, 2 = all)	CSI * J

GENERAL

Bell	BEL
Device status report	CSI 6 n

PRINTING

Media copy	CSI i

HP 700-92/96 Emulation

CHARACTER ATTRIBUTES

Assign * attribute(s) to following characters	ESC & d *
Assign invisible & * attribute(s) to following chars.	ESC & d s *

Dim	H	Dim, Flash & Underline	M
Flashing	A	Dim, Under & Inverse	N
Inverse video	B	Dim, Flash, Und. & Inv.	O
Invisible	S	Flash & Inverse	C
Underline	D	Flash & Underline	E
Dim & Flash	I	Flash, Inverse & Under	G
Dim & Inverse	J	Inverse & Underline	F
Dim & Underline	L	Attributes off	@
Dim, Flash & Inverse	K		

CHARACTER SETS

Select ROMAN 8 (normal) character set	SI
Select Line Drawing character set	SO

CURSOR

Clear all tab stops	ESC 3
Clear tab stop at current cursor position	ESC 2
Cursor relative addressing (column)	ESC & +/- * C
Cursor relative addressing (column/line)	ESC & a +/- *c c +/- *l R
Cursor relative addressing (line)	ESC & a +/- * R
Cursor relative addressing (line/column)	ESC & a +/- *l r +/- *c C
Cursor sensing absolute	ESC a
Cursor sensing relative	ESC `
End of line wrap inhibited	ESC & s 1 C
End of line wrap enabled	ESC & s 0 C
Memory absolute addressing (column)	ESC & a * C
Memory absolute addressing (column/line)	ESC & a *c c *l R
Memory absolute addressing (line)	ESC & a * R
Memory absolute addressing (line/column)	ESC & a *l r *c C
Move cursor home down	ESC F
Move cursor home up	ESC H *or* ESC h
Move cursor one column left	ESC D
Move cursor one column right	ESC C
Move cursor one line down	ESC B
Move cursor one line up	ESC A
Move cursor to left margin	ESC G
Move cursor to next tab stop	ESC I (uppercase i)
Move cursor to previous tab stop	ESC i
Screen relative addressing (column)	ESC & a * C
Screen relative addressing (column/line)	ESC & a *c c *l Y
Screen relative addressing (line)	ESC & a * Y
Screen relative addressing (line/column)	ESC & a *l y *c C
Set tab stop at current cursor position	ESC 1

DISPLAY

80 column display	ESC & w 6 f 80 X
132 column display	ESC & w 6 f 132 X
Clear all margins	ESC 9
Display next page	ESC U
Display previous page	ESC V
Format mode off	ESC X
Format mode on	ESC W
Jump scroll enabled	ESC & k 0 [
Memory lock disabled	ESC m
Memory lock enabled	ESC l
Scroll down one line	ESC T
Scroll up one line	ESC S
Select invisible display	ESC & w 13 F
Select visible display	ESC & w 12 F
Set left margin	ESC 4
Set right margin	ESC 5
Smooth scroll enabled	ESC & k 1 [
Define colour pair	ESC & v *parameters

*parameters can be one or more of the following:

<0/1> m	RGB (0) or HSL (1) colour specification method
<decimal> a	Red or Hue colour value for foreground
<decimal> b	Green or Saturation colour value for foreground
<decimal> c	Blue or Luminosity colour value for foreground
<decimal> x	Red or Hue colour value for background
<decimal> y	Green or Saturation colour value for background
<decimal> z	Blue or Luminosity colour value for background
<0 - 7> i	Colour pair # to be initialized
<0 - 7> s	Colour pair # to be selected
<0 - 7> ^	Colour pair definition status

Assign colours to colour pair (* = decimal #)	ESC & v * a * b * c * x * y * z <colour pair #> i
Select an alpha colour pair (* = 0 - 7)	ESC & v * S

EDITING

Clear display memory from cursor	ESC J
Clear line or field from cursor	ESC K
Delete character	ESC P
Delete line	ESC M
Insert character mode	ESC Q
Insert line	ESC L
Modify all mode disabled	ESC & k 0 M
Modify all mode enabled	ESC & k 1 M
Replace character mode	ESC R
Unprotected field end	ESC]
Unprotected field start	ESC [

FUNCTION KEYS

Begin user key definition mode	ESC j
Default definition for f1 key	ESC p
Default definition for f2 key	ESC q
Default definition for f3 key	ESC r
Default definition for f4 key	ESC s
Default definition for f5 key	ESC t
Default definition for f6 key	ESC u
Default definition for f7 key	ESC v
Default definition for f8 key	ESC w
Define f-key	ESC & f *
Disable User System & Menu and label Modes	ESC & j S
Enable User System & Menu and label Modes	ESC & j R
Enable f-keys & remove labels and status line	ESC & j @
Enable & display Modes labels	ESC & j A
Enable & display user function key labels	ESC & j B
End user key definition mode	ESC k
Replace key labels with character string	ESC & j *n L *s
Restore labels	ESC & j C

GENERAL OPERATION

7 bits & existing parity	ESC & k 0 I
8 bits & no parity	ESC & k 1 I
Audible tone disabled	ESC & k 0 D
Audible tone enabled	ESC & k 1 D
Check parity no	ESC & s 0 Z
Check parity yes	ESC & s 1 Z
Configuration menus locked	ESC & q 1 L
Configuration menus unlocked	ESC & q 0 L
Data speed high	ESC & s 1 X
Data speed low	ESC & s 0 X
DC2 not inhibited	ESC & s 0 H
DC2 inhibited	ESC & s 1 H
Delay one second	ESC @
Display functions mode disabled	ESC Z
Display functions mode enabled	ESC Y
Enter VT220 7-bit mode (25 line display, keypad F1 - F4 mapped to top left F1 - F4 keys, F5 - F12 as per VT220)	ESC & k 1 \
Hand shake not inhibited	ESC & s 0 G
Hand shake inhibited	ESC & s 1 G
Hard reset	ESC E
Local echo disabled	ESC & k 0 L
Local echo enabled	ESC & k 1 L
Local mode disabled (only for user f-key)	ESC & k 1 R
Local mode enabled (only for user f-key)	ESC & k 0 R
Modem disconnect	ESC f
Self test	ESC z
Soft reset	ESC g

KEYBOARD

Auto keyboard lock off	ESC & k 0 K
Auto keyboard lock on	ESC & k 1 K
Auto line feed mode off	ESC & k 0 A
Auto line feed mode on	ESC & k 1 A
Caps lock off	ESC & k 0 C
Caps lock on	ESC & k 1 C
Caps mode off	ESC & k 0 P
Caps mode on	ESC & k 1 P
Function key codes local only	ESC & s 0 A
Function key codes transmitted to host	ESC & s 1 A
Lock keyboard	ESC c
Space overwrite latch disabled	ESC & s 0 B
Space overwrite latch enabled	ESC & s 1 B
SPOW latch off	ESC & k 0 N
SPOW latch on	ESC & k 1 N
Unlock keyboard	ESC b

LOCAL EDITING MODE

Data block for transmission is cursor line	ESC & s 0 D
Data block for transmission is page	ESC & s 1 D
Enter edit mode	ESC & k 1 B
Enter interactive mode	ESC & k 0 B
Transmit block of text to host	ESC d

PRINTING

Copy all to printer	ESC & p M *or* ESC & p 0 M
Copy display memory to printer	ESC 0
Copy line to printer	ESC & p B *or* ESC & p 0 B
Copy page to printer	ESC & p F *or* ESC & p 0 F
Data transfer host to printer (* = 1-256)	ESC & p * W
Disable logging	ESC & p 13 C
Enable bottom logging	ESC & p 11 C
Enable top logging	ESC & p 12 C
Execute form feed	ESC & p 4 u 0 C
Execute * line feeds	ESC & p 1 * 4 u 1 C
Record mode on (* is optional)	ESC & p * 20 C
Select external device	ESC & p 4 D
Transfer display escape sequences no	ESC & s 0 N
Transfer display escape sequences yes	ESC & s 1 N

REPORTS

Request cursor position report disabled	ESC & x 0 C
Request cursor position report enabled	ESC & x 1 C
Request external printer status report	ESC & p 4 ^
Request primary terminal status report	ESC ^
Request secondary terminal status report	ESC ~
Request terminal identity report	ESC * s ^
Request application program name report	ESC * s 12347 ^

IBM 3151 Emulation

CHARACTER SET SELECTION

Select G0	SI
Select G1	SO
Select character set G0	ESC < *set
Select character set G1	ESC > *set

CURSOR

Backspace	BS
Carriage return, new line or LTA	CR
Cursor down	ESC B
Cursor left	ESC D
Cursor right	ESC C
Cursor up	ESC A
Cursor home	ESC H
Index	ESC SP M
Insert cursor	ESC Z
Line feed	VT
Line feed or new line	LF
Line feed or erase input	FF
Next line	ESC M
Page down AID	ESC ! B
Reverse index	ESC ! M
Read cursor address	ESC 5
Set buffer address	ESC X *row *column
Set cursor address	ESC Y *row *column
Reset buffer address mode	ESC SP Z
Tab	HT
Back tab	ESC 2
Set column tab	ESC 0
Clear column tab	ESC 1
Clear all column tabs	ESC SP 1

DISPLAY

Clear all	ESC ! L
Clear page	ESC L
Create viewport	ESC SP r *...*
Disable default field attribute	ESC ' ;
Disable field attribute visible renditions	ESC , ;
Disable host protect	ESC & ;
Disable OIA divide line	ESC * ;
Disable host protect	ESC & ;
Disable OIA divide line	ESC * ;
Disable partition separate line	ESC + ;
Disable read unprotected field	ESC % ;
Display machine status	ESC # :
Enable default field attribute	ESC ' :
Enable field attribute visible renditions	ESC , :

Enable host protect	ESC & :
Enable OIA divide line	ESC * :
Enable read unprotected field	ESC % :
Host message write	ESC = *message ESC =
Host message display	ESC # ;
Set character attribute	ESC 4 *attrib *operation
Set field attribute	ESC 3 *attrib *attrib *operation
Set field attribute response	ESC 3 *attrib *attrib

Create 1 viewport 24 rows x 80 columns	ESC SP r ! ! SP 8 " P
Create 1 viewport 25 rows x 80 columns	ESC SP r ! ! SP 9 " P
Create 1 viewport 28 rows x 80 columns	ESC SP r ! ! SP " D
Create 1 viewport 24 rows x 132 columns	ESC SP r ! ! SP 8 $ D
Create 1 viewport 25 rows x 132 columns	ESC SP r ! ! SP 9 $ D
Create 1 viewport 28 rows x 132 columns	ESC SP r ! ! SP $ D
Create 2 viewports in 80 column screen	ESC SP r " ! SP *r1 " 0 " SP *r2 " P
Create 2 viewports in 132 column screen	ESC SP r " ! SP *r1 $ $ " SP *r2 $ D
Create 3 viewports in 80 column screen	ESC SP r # ! SP *r1 " 0 " SP *r2 " 0 # SP *r3 " P
Create 3 viewports in 132 column screen	ESC SP r # ! SP *r1 $ $ " SP *r2 $ $ # SP *r3 $ D

*The ASCII character values of *r1, *r2 and *r3 are listed below. When creating two viewports, *r1 + *r2 must be 24 or 25. When creating three viewports, *r1 + *r2 + *r3 must be 24 or 25.*

Rows	ASCII	Rows	ASCII	Rows	ASCII	Rows	ASCII
1	!	8	(	14	.	20	4
2	"	9	)	15	/	21	5
3	#	10	*	16	0	22	6
4	$	11	+	17	1	23	7
5	%	12	,	18	2	24	8
6	&	13	-	19	3	25	9
7	'						

Select active partition command/response *(not model 11)*	ESC ! q *
Select host partition command/response *(not model 11)*	ESC SP q *
* A = Viewport 1, B = Viewport 2, C = Viewport 3	
Jump partition command/response *(not model 11)*	ESC " A
Enable partition separator line *(not model 11)*	ESC + :
Disable partition separator line *(not model 11)*	ESC + ;

EDITING

Delete character	ESC Q
Delete line	ESC O
Erase to end of field/line	ESC I
Erase to end of page	ESC J
Erase input	ESC K
Insert character	ESC P *character
Insert line	ESC N

GENERAL OPERATION

Bell	BEL
Cancel	ESC S or CAN
Disable write Null	ESC " ;
Enable write Null	ESC " :
Line turnaround character (if selected)	ETX, CR, EOT or DC3

Read status	ESC 6
Enter transparent mode (control characters displayed)	DLE STX
Exit transparent mode	DLE ETX
Mandatory disconnect	DLE EOT
Begin pass-through data stream	DLE DC2
End pass-through data stream	DLE DC4
Restart transmission (pacing)	DC1 (XON)
Stop transmission (pacing)	DC3 (XOFF)
Begin outbound trace	ESC SP :
End outbound trace	ESC SP ;
Reset to initial state	ESC SP S
Reset keyboard lock & keep MDT bit	ESC ! S
Read control 1	ESC SP 7
Read control 2	ESC ! 7
Read control 3	ESC " 7
Read control 4	ESC # 7
Read control 5	ESC $ 7
Read control 6	ESC % 7
Read control 7	ESC & 7
Read model	ESC SP 6
Read model (extended)	ESC " 6
Read terminal ID	ESC ! 6
Read all	ESC # 8
Read line/send line	ESC ! 8
Read message/send message	ESC SP 8
Read page/send page	ESC 8
Write send mark	ESC E
Set control 1	ESC SP 9 *mode1 *mode2 *op
Set control 2	ESC ! 9 *mode1 *mode2 *op
Set control 3	ESC " 9 *mode1 *mode2 *op
Set control 4	ESC # 9 *mode *operation
Set control 5	ESC $ 9 *m1 *m2 *m3 *m4 *op
Set control 6	ESC % 9 *m1 *m2 *m3 *m4 *op
Set control 7	ESC & 9 *m1 *m2 *m3 *op

KEYBOARD

Disable print key attention	ESC) ;
Disable reset key attention	ESC (;
Enable print key attention	ESC) :
Enable reset key attention	ESC (:
Keyboard lock	ESC :
Keyboard unlock	ESC ;
Load programmable function key	ESC ! = *fn *fnx *ff *fp ESC =
Set all default function keys	ESC SP t
Set default function key	ESC t *key

PRINTING

Print line	ESC U
Print message	ESC V
Print screen	ESC SP W
Print viewport	ESC W

MDIS Prism-8 & Prism-9 Emulations

CHARACTER SET SELECTION

Assign G0 label to * character set	ESC (*
Assign G1 label to * character set	ESC) *
Assign G2 label to * character set	ESC * * (second * is parameter)
Assign G3 label to * character set	ESC + *
Assign G0 labelled set to 7 bit codes	SI
Assign G1 labelled set to 7 bit codes	SO
Assign G2 labelled set to 7 bit codes	ESC n
Assign G3 labelled set to 7 bit codes	ESC o
Assign G1 labelled set to 8 bit codes	ESC ~
Assign G2 labelled set to 8 bit codes	ESC }
Assign G3 labelled set to 8 bit codes	ESC |
Assign G1 labelled set to 7 bit codes for 1 character	ESC R
Assign G2 labelled set to 7 bit codes for 1 character	ESC N
Assign G3 labelled set to 7 bit codes for 1 character	ESC O
Load redefinable character set	ESC P *b ; *c ; *e % q S1 ;....Sn ESC \

COMMUNICATIONS

Transmission on	XON
Transmission off	XOFF
Select 7-bit C1 code transmission	ESC SP F
Select 8-bit C1 code transmission	ESC SP G

CURSOR

Cursor off	Hex E4
Cursor on	Hex E2
Move cursor * lines up (default 1)	ESC [* A
Move cursor * lines down (default 1)	ESC [* B
Move cursor * columns right (default 1)	ESC [* C
Move cursor * columns left (default 1)	ESC [* D
Move cursor * lines down and to start of line (default 1)	ESC [* E
Move cursor * lines up and to start of line (default 1)	ESC [* F
Move cursor to specified line (*l) and column (*c) on page	ESC [*l ; *c ; H
Move cursor to specified line (*l) and column (*c) on page	ESC [*l ; *c ; f
Move cursor to absolute column number * position	ESC [* G
Move cursor to absolute column number * position	ESC [* `
Horizontal position relative	ESC [* a
Move cursor * columns left (default 1)	ESC [* j
Move cursor to absolute line number * position	ESC [* d
Move cursor * lines down (default 1)	ESC [* e
Move cursor * lines up (default 1)	ESC [* k
Move cursor * columns right (default 1)	ESC [* o
Line addressing relative to top line of page	ESC [< 6 l
Line addressing relative to scroll region	ESC [< 6 h
Make cursor invisible	ESC [< 4 l
Make cursor visible	ESC [< 4 h
Set tab stop at current cursor position	ESC H

Set tab stops at specified * column numbers	ESC [*c ; *c ; .... *c SP N
Clear tab stops (0 = at cursor position, 2 & 3 = all on page)	ESC [* g
Set tab stop at current cursor position	ESC [0 W
Clear tab stops (2 = at cursor position, 4 & 5 = all on page)	ESC [* W
Move cursor to next tab stop	HT
Move cursor to next tab stop	ESC [* I
Move cursor to previous tab stop	ESC [* Z

DISPLAY

Status line off	Hex E5
Status line on	Hex E3

Assign * attribute(s) to following characters (default 0) — ESC [* % {

Attributes:				
	Flashing	1	Bold	8
	Underline	2	Protected	32
	Inverse video	4		

Select graphic rendition — ESC [Pa;...Pa m

Attributes:				
	Clear all	0	Normal intensity	22
	Bold	1	Underline off	24
	Underline	4	Flashing off	25
	Flashing	5	Inverse video off	27
	Inverse video	7		

Assign * line attributes to current line — ESC [* % l

Attributes:	
Single width, single height (default)	0
Double width	1
Double height & width top half	2
Double height & width bottom half	3

No attributes	ETX SP
Bold attribute	ETX A
Flashing attribute	ETX B
Bold & Flashing attributes	ETX C
Inverse attribute	ETX D
Bold & Inverse attributes	ETX E
Flashing & Inverse attributes	ETX F
Bold, Flashing & Inverse attributes	ETX G
Blank attribute	ETX H
Bold & Blank attributes	ETX I
Flashing & Blank attributes	ETX J
Bold, Flashing & Blank attributes	ETX K
Inverse & Blank attributes	ETX L
Bold, Inverse & Blank attributes	ETX M
Flashing, Inverse & Blank attributes	ETX N
Bold, Flashing, Invers & Blank attributes	ETX O
Underline attribute	ETX P
Bold & Underline attributes	ETX Q
Flashing & Underline attributes	ETX R
Bold, Flashing & Underline attributes	ETX S
Inverse & Underline attributes	ETX T
Bold, Inverse & Underline attributes	ETX U
Flashing & Underline attributes	ETX V

Bold, Flashing, Inverse & Underline attributes	ETX W
Blank & Underline attributes	ETX X
Bold, Blank & Underline attributes	ETX Y
Flashing, Blank & Underline attributes	ETX Z
Bold, Flashing, Blank & Underline attributes	ETX [
Inverse, Blank & Underline attributes	ETX \
Bold, Inverse & Underline attributes	ETX]
Underline attribute	ETX 0
Bold & Underline attributes	ETX 1
Flashing & Underline attributes	ETX 2
Bold, Flashing & Underline attributes	ETX 3
Inverse & Underline attributes	ETX 4
Bold, Inverse & Underline attributes	ETX 5
Flashing, Inverse & Underline attributes	ETX 6
Bold, Flashing, Inverse & Underline attributes	ETX 7
Blank & Underline attributes	ETX 8
Inverse, Blank & Underline attributes	ETX /
Flashing, Inverse, Blank & Underline attributes	ETX ^
Flashing, Inverse, Blank & Underline attributes	ETX _

EDITING

Insert * number of blank characters from cursor position	ESC [* @
Delete * number of characters from cursor position	ESC [* P
Insert * number of blank lines from cursor position	ESC [* L
Delete * number of lines from cursor position	ESC [* M
Replace character mode	ESC [4 l
Insert character mode	ESC [4 h
Blank * number of columns from cursor position	ESC [* X
Erase in field (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* N
Erase in line (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* K
Erase in page (0 = from cursor, 1 = to cursor, 2 = all)	ESC [* J
Enable erasure of unprotected characters only	ESC [6 l
Enable all characters to be erased, including protected	ESC [6 h
Erasing characters clears display attributes to default	ESC [< 5 l
Erasing characters retains display attributes	ESC [< 5 h

ENCODING DATA

Escape sequence introducer	ESC
Control sequence introducer	ESC [
Device control string	ESC P
String terminator	ESC \
Cancel	CAN
Abort control sequence & display error	SUB
Set mode	ESC [*m ; *m ; ... *m h
Reset mode	ESC [*m ; *m ; ... *m l

FORMATTING DATA

Move cursor to left margin of current line	CR
Move cursor down one line	LF
Move cursor down one line	ESC D
Move cursor down one line	VT

Move cursor to start of next line	ESC E
Move cursor up one line	ESC M
Move cursor to start of new page	FF
Move cursor one column left	BS
Repeat last displayable character * times (default 1)	ESC [* b

KEYBOARD

Key presses generate codes which are sent to host	ESC [2 l
Key presses are ignored (except Break)	ESC [2 h
Define function key	ESC P *k % p *n ESC \
Extended keypad mode reset	ESC [< 15 l
Extended keypad mode set	ESC [< 15 h

PAGE MANIPULATION

Page layout definition — ESC [*p ; *l ; *c ; *s ; *e % w

where *p = page number in range 1 - 8, default 1
*l = number of lines, default 24
*c = number of columns, default 80 or 132
*s = start of scroll area, default 1
*e = end of scroll area, default as *l

Display buffer & video format definition — ESC [*p ; *c ; *f % x

where *p = number of pages, default 8 (80 col) or 4 (132 col)
*c = number of columns, default 80
*f = format options: 0 =single active page, 1 = multiple active page

Set scrolling area — ESC [*s ; *e % v

where *s = starting line number, default 1
*e = ending line number, default is page length

80 column display mode	ESC [14 l
132 column display mode	ESC [14 h
Move active position to absolute page number	ESC [* SP P
Move forward by * number of pages	ESC [* SP Q
Move backward by * number of pages	ESC [* SP R
Save page state mode reset	ESC [< 11 l
Save page state mode set	ESC [< 11 h
Save current active page state	ESC [% y
Restore last saved page state	ESC [% z
Page display absolute	ESC [*p 5 p
Display page number *	ESC [* % p
Move forward by * number of pages	ESC [* U
Move backward by * number of pages	ESC [* V
Scroll up * lines (forward)	ESC [* S
Scroll down * lines (backward)	ESC [* T
Active page display mode reset	ESC [< 7 l
Active page display mode set	ESC [< 7 h
Screen/scroll keys action mode reset	ESC [< 13 l
Screen/scroll keys action mode set	ESC [< 13 h

PRINTING

Printer output control	ESC [< *o ; *s ; *e i

where *o = output type as listed below, default 0
0 = print page, 1 = print partial page, 2 = hardcopy off
3 = hardcopy on, 4 = direct print off, 5 = direct print on
*s = starting line number, default start of scrolled region
*e = ending line number, default end of scrolled region

Media copy ESC [*o i

where *o = 0 = print page, 4 = direct print off, 5 = direct print on

REPORTS

Report on device configuration (* = 0 by default) ESC [< * c

Give terminal configuration report	0
Give screen configuration report	1
Give printer configuration report	2
Give keyboard configuration report	3

Device attributes (* = device type, 0) ESC [* c

Terminal configuration report ESC [< 10 ; *m ; *f ; *h ; *v c

where *m = model number (9 for 12090)
*f = features (1 = down-loadable charcter set,
2 = programmable function keys,
4 = software down-load capability
16 = field-read capability)
*h = hardware options
*v = version number

Screen configuration report ESC [< 11 ; *p ; *l ; *c ; *f c

where *p = number of (80 x 25) pages available
*l = maximum lines displayable on screen
*c = maximum columns displayable on screen
*f = features (1 = per field attribute capability,
2 = per character attribute capability,
4 = 80/132 screen width selectable)

Printer configuration report (* = printer type) ESC [< 12 ; * c

Keyboard configuration report ESC [< 13 ; *l ; *n c

where *l = keyboard layout (1 = standard, 2 = data entry)
*n = nationality (1 = UK, 2 = USA, 3 = German, 4 = French,
5 = Swiss (Ger), 6 = Swiss (Fre), 7 = French Canadian)

STATUS REPORTING

Report on device status (* = device identifier) ESC [< * n

Device status report (5 = device status, 6 = cursor position) ESC [* n

Terminal status report ESC [< 10 ; *e ; *g n

where *e = emulation mode (1 = ANSI 7-bit,
2 = ANSI 8-bit, 3 = 12080 mode)
*g = general status (1 = 7-bit comms link,
2 = flow control enabled, 4 = foreground executive active)

Screen status report ESC [< 11 ; *p ; *l ; *c ; *g n

where *p = number of pages currently defined

*l = number of lines displayed (24 or 25)
*c = number of columns displayed (80 or 132)
*g = general status (1 = multiple active page format,
2 = per-character attributes)

Printer status report (*g = general status, *b = buffer left)	ESC [< 12 ; *g ; *b n
Keyboard status report (*m = mode, *g = general status)	ESC [< 13 ; *m ; *g n
Cursor position report (*l = line, *c = column)	ESC [< *l ; *c R

SYSTEM MESSAGES

Write data into system message area starting at column *	ESC [* % }
System messages not visible	ESC [< 8 l
System messages displayed on last screen line	ESC [< 8 h

TERMINAL CONTROL

Soft reset	ESC [& p
Reset to initial state	ESC c
ANSI terminal emulation mode	ESC [< 12 l
Model 12080 terminal emulation mode	ESC [< 12 h
Sound alarm	BEL

MDIS Prism-12 Emulation

When TeemTalk is running the Prism-12 emulation, the following commands will be executed in addition those listed previously for Prism-8 and Prism-9.

Data stream filler character	NUL
Cursor home	SOH
Function key control	STX *
* = Generate user defined codes	90
Do not generate any codes	91
Generate standard codes	92
Generate short codes (hex 80 - B3)	93
Set video attributes for next display field	ETX *
Cursor forward 1 character position	ACK
Sound bell	BEL
Backspace	BS
Line feed	LF
Cursor vertical position	VT *
Form feed	FF
Carriage return	CR
Select Multinational character set (in Multinational mode)	SO
Select ASCII character set (in Multinational mode)	SI
Horizontal position	DLE *
Cursor back	NAK
Cursor up	SUB
Escape sequence introducer	ESC
Special single shift	GS *
Space compression codes	Hex A0 to Hex BF

Reserved	Hex E0
Set S-mode	Hex E1
Cursor on	Hex E2
Start of system message	Hex E3 *
Cursor off	Hex E4
End of system message	Hex E5
Give terminal status	Hex E6
Set screen format	Hex E7 *
Display system message	Hex E8
Display user line 25	Hex E9
Select active screen	Hex EA *
Set R-mode	Hex EB
Set video prior conditions	Hex EC *
Screen move	Hex ED *
Filler	Hex EE
Character fill	Hex EF *
Page back	Hex F5
Page forward	Hex F6
Scroll back	Hex F7
Scroll forward	Hex F8
Run diagnostics	ESC D
Erase to end of page	ESC J
Erase to end of line	ESC K
DCS sequence	ESC P
Printer on	ESC R
Printer off	ESC T
Print screen	ESC U
CSI sequence introducer	ESC [

PT250 Emulation

AREA & LOGICAL ATTRIBUTES

Assert defined logical attributes	ESC [> 3 h
Assert defined logical attributes disabled	ESC [> 3 l
Clear (reset) selected areas	ESC $ K
Define area qualification	ESC [* o
* = All printing characters	2
Numeric characters	3
Alphabetic characters	4
Right-justify in area	5
Protected; no input accepted (default)	> 0
Must enter the area	> 1
Must fill the whole area	> 2
Set modified data tag	> 3
Define logical attributes	ESC [* v
* = All printing characters	2
Numeric characters	3
Alphabetic characters	4
Right-justify in area	5
Protected; no input accepted (default)	> 0
Must enter the area	> 1
Must fill the whole area	> 2
Set modified data tag	> 3
Field entry check	ESC $ C
Logical attributes start	ESC $ L
Logical attributes end	ESC $ M
Protected area start	ESC V
Protected area end	ESC W
Reset modified tags	ESC $ J
Selected area start	ESC F
Selected area end	ESC G

CHARACTER SET SELECTION

Set G0 alternate character set	ESC $ 2
Set G0 ASCII character set	ESC $ 0
Set G1 alternate character set	ESC $ 3
Set G1 ASCII character set	ESC $ 1

COMPRESSED COMMANDS

Compressed cursor position (*l = line, *c = column)	ESC 0 *l *c
Compressed logical area (* = logical attribute)	ESC 3 *
Compressed logical attributes (*l = line, *c = column *a = logical attributes *n = number of characters)	ESC 1 *l *c *a *n
Compressed visual area (* = visual attributes)	ESC 4 *
Compressed visual attributes (* = visual attributes)	ESC 2 *

CURSOR

Autowrap disabled	ESC [> 9 h
Autowrap enabled	ESC [> 9 l
Carriage return = carriage return only	ESC [> 1 l
Carriage return = carriage return & line feed	ESC [> 1 h
Cursor invisible	ESC $ S
Cursor movement keys immediate effect	ESC [> 13 l
Cursor movement keys require host permission	ESC [> 13 h
Cursor position report (*l = line, *c = column)	ESC [*l ; *c R
Cursor tabulation control	ESC [* W
* = Set tab at cursor position	0
Clear tab at cursor postion	2
Clear all tab stops	5
Cursor visible	ESC $ R
Home cursor absolute	ESC $ B
Home cursor relative	ESC $ A
Index (cursor down to next unlocked line)	ESC D
Line feed = line feed only	ESC [20 h
Line feed = line feed & carriage return	ESC [20 l
Move cursor down to next unlocked line 1st column	ESC E
Move cursor down * lines	ESC [* B
Move cursor down * unlocked lines to 1st column	ESC [* E
Move cursor left * columns	ESC [* D
Move cursor right * columns	ESC [* C
Move cursor to column * (absolute)	ESC [* G
Move cursor to line * (absolute)	ESC [* d
Move cursor to absolute line (*l) & column (*c)	ESC [*l ; *c H
Move cursor to relative line (*l) & column (*c)	ESC [*l ; *c f
Move cursor up * unlocked lines	ESC [* A
Move cursor up * unlocked lines to 1st column	ESC [* F
Restore cursor & attributes	ESC $ Q
Reverse index (cursor up to next unlocked line)	ESC M
Save cursor & attributes	ESC $ O
Set tab stop at current cursor position	ESC H
Tab cursor backward * tab stops	ESC [* Z
Tab cursor forward * tab stops	ESC [* I

DISPLAY

Clear display memory (except locked lines)	ESC ?
Display size	ESC [* N
* = 80 columns by 24 lines	1
80 columns by 48 lines (2 pages)	2
132 columns by 27 lines	3
80 columns by 25 lines	4
Display error message (* = string)	ESC : * ESC \
Display invisible	ESC $ E
Display memory 24 lines (1 page)	ESC [> 11 l
Display memory 48 lines (2 pages)	ESC [> 11 h
Display status line	ESC $ T

Display system line (data on line preserved)	ESC $ U
Display system line (blank)	ESC $ V
Display visible	ESC $ P
Jump scroll enabled	ESC [> 5 l
Lock lines (*l = screen line number to begin, *n = # of lines)	ESC [*l ; *n u
Next page (* = 1 page one, 2 page two)	ESC [* U
Page down	ESC $ b
Page up	ESC $ a
Previous page (* = 1 page one, 2 page two)	ESC [* V
Repeat previous character * times	ESC [* b
Screen wrap mode enabled	ESC [> 8 h
Screen wrap mode disabled	ESC [> 8 l
Scroll automatically	ESC $ W
Scroll manually only	ESC $ X
Scroll unlocked lines down * lines	ESC [* T
Scroll unlocked lines up * lines	ESC [* S
Smooth scroll enabled	ESC [> 5 h
Unlock lines (*l = screen line # to begin, *n = # of lines)	ESC [*l ; *n y

EDITING

Delete * characters from cursor position right	ESC [* P
Delete * lines from cursor position down	ESC [* M
Editing extent mode	ESC [* Q
* = Entire display	0
Line only	1
Area only	2
Enable all characters to be erased	ESC [6 h
Enable erasure of unprotected characters only	ESC [6 l
Enter character mode	ESC [> 2 l
Enter edit (block) mode	ESC [> 2 h
Erase * characters & substitute with pad characters	ESC [* X
Erase in area & substitute with pad characters	ESC [* O
* = Erase to end of area, line, or display	0
Erase from start of area, line, or display	1
Erase entire area, line, or display	2
Erase in display & substitute with pad characters	ESC [* J
* = Erase to end of display	0
Erase from start of display	1
Erase entire display	2
Erase in line & substitute with pad characters	ESC [* K
* = Erase to end of line	0
Erase from start of line	1
Erase entire line	2
Insert * blank lines	ESC [* L
Insert * space or null characters	ESC [* @
Insert mode disabled (replace mode selected)	ESC [4 l
Insert mode enabled	ESC [4 h
Pad character is null	ESC [> 7 l
Pad character is space	ESC [> 7 h

Read cursor character	ESC ;
Read cursor character normal mode	ESC [> 23 l
Read cursor character erase mode	ESC [> 23 h

GENERAL OPERATION

Action all control characters except CR & HT	ESC [> 14 h
Action all control characters including CR & HT	ESC [> 14 l
DOS merge mode disabled	ESC [> 25 l
DOS merge mode enabled	ESC [> 25 h
Host notification mode disabled	ESC [> 16 l
Host notification mode enabled	ESC [> 16 h
Local echo mode on	ESC [12 l
Local echo mode off	ESC [12 h
Reset to initial state	ESC c
Tab characters separate fields enabled	ESC [> 17 h
Tab characters separate fields disabled	ESC [> 17 l

KEYBOARD

Application program command	ESC _ * ESC \

Menu	0	Ctrl + Shift + Help	4
Help	1	Shift + Menu	5
Shift + Help	2	Ctrl + Menu	6
Ctrl + Help	3	Ctrl + Shift + Menu	7

Cursor movement keys immediate effect	ESC [> 13 l
Cursor movement keys require host permission	ESC [> 13 h
E2 mode enabled (Enter = 4 ESC sequences)	ESC [> 21 h
E2 mode disabled (Enter functions as normal)	ESC [> 21 l
Escape key disabled	ESC $ H
Escape key enabled	ESC $ I
Flow control codes from kbd not sent immediately	ESC [> 15 h
Flow control codes from keyboard sent immediately	ESC [> 15 l
Function keys (F#, PF#, PA#) terminated with CR	ESC [> 18 h
Function keys not terminated with CR	ESC [> 18 l
Hard keyboard lock disabled	ESC '
Hard keyboard lock enabled	ESC b
Load keyboard table (*d = data, *c = checksum)	ESC < *d *c ESC \
Numeric keypad application mode	ESC [> 10 h
Numeric keypad numeric mode	ESC [> 10 l
Print screen key local function	ESC [> 24 h
Print screen key normal function	ESC [> 24 l
Single shift three (* = key identifier)	ESC 0 *
Single shift two (* = key identifier)	ESC N *
Soft keyboard lock enabled	ESC $ F
Soft keyboard lock disabled	ESC $ G
Soft lock keyboard on illegal command enabled	ESC [> 19 h
Soft lock keyboard on illegal command disabled	ESC [> 19 l

LOCAL EDITING

Data block is cursor line	ESC [> 4 h
Data block is page(s) (depends on display memory)	ESC [> 4 l
Dump block data	ESC 6
Enter character mode	ESC [> 2 l
Enter edit (block) mode	ESC [> 2 h
Modified unprotected areas only transmitted	ESC [> 6 h
Modified & unmodified unprotected areas transmitted	ESC [> 6 l
Page dump waits for host request	ESC =
Transmission waits for host request	ESC S
Transmit block data (excluding locked lines)	ESC 5
Transmit selected areas	ESC [17 l
Transmit unprotected areas (depends on modified mode)	ESC [17 h

PRINTING

Auto print off	ESC [4 i
Auto print on (data not displayed)	ESC [5 i
Initiate page dump	ESC [> 2 i
Initiate screen transfer	ESC [> 1 i
Print & display received host data	ESC [> 3 i
Print screen	ESC [0 i
Print screen qualified areas	ESC [> 0 i

REPORTS

Device control string (* = string)	ESC P * ESC \
Display revision	ESC $ Z
Report potentially destructive action to host	ESC] * ESC \

* =	Reset to original state notification	0
	Clear screen notification	2
	Load keyboard table command failed	3
	Load keyboard table command successful	4

Report terminal identity	ESC [c
Report terminal identity	ESC [0 c
Report terminal status	ESC [* n

* =	Ready	0
	Report status	5
	Report cursor position	6

VISUAL ATTRIBUTES

Change visual attributes of area	ESC [* p

* =	Active position to end of area	0
	Start of area to active position	1
	Entire area	2

Change visual attributes of character	ESC [* q
Change visual attributes of display	ESC [* r

* =	Active position to end of display	0
	Start of display to active position	1
	Entire display	2

Change visual attributes of line			ESC [* t
* =	Active position to end of line		0
	Start of line to active position		1
	Entire line		2
Select graphic rendition			ESC [* m
Normal video	0	Strike-through	> 1
Dimmed	2	Invisible	> 2
Underline	4	Line drawing graphics	> 3
Flashing	5	Block drawing graphics	> 4
Inverse video	7		
Visual attributes locked			ESC [> 12 h
Visual attributes unlocked			ESC [> 12 l

ReGIS Emulation

MODE SELECTION

Exit to VT100/VT200 mode	ESC \

MOUSE BUTTON PROGRAMMING

Program a mouse button (*c = clear, *b = button, *p = press code, *r = release code)	ESC P *c $ w *b/*p/*r ESC \

SIXEL GRAPHICS

Direct sixels to host	ESC [? 2 i
Direct sixels to printer	ESC [? 0 i or ESC [? 1 i
Transmit sixel data (*p = pixel shape, *b = backgrnd colour *h = horiz. spacing, *..* = sixel data)	ESC P *p ; *b ; *h ; q *..* ESC \

COMMAND LEVEL INPUT SELECTION

Select command level input	;

COMMAND SET STORING/INVOKING

Clear all stored command sets	@.
Clear stored command set	@: character @;
Invoke command set	@ character
Store command set	@: character set definition @;

CURVE DRAWING

Draw arc from specified start position	C(A deg of drawing) start posn
Draw arc with specified centre position	C(A deg of drawing C) centre posn
Draw circle with specified centre	C(C) centre position
Draw circle specified circum. point	C circumference point
Draw closed interpolated curve intersecting current & specified positions	C(B) positions (E)
Draw closed interpolated curve intersecting current & specified positions, closed with a straight line	C(B)[] positions [](E)
Draw open interpolated curve	C(S) positions (E)
Draw open interpolated curve intersecting current & all specified positions	C(S)[] positions [](E)
Set temporary writing controls effective only for the command, using 'W' Writing Control cmd & options	C options (W(sub-opts)) opts

FILLING SHAPES

Define curved boundary shape for filling using 'C' Curve Drawing com. & options	F(C sub-options)
Define straight boundary shape for filling using 'V' Draw Vector command & options	F(V sub-options)
Set cursor position using 'P' command & options	F(P(sub-options))
Set temporary writing controls effective only for the command using 'W' Writing Control cmd & options	F(W(sub-options) options)

LOADING CHARACTER SETS

Assign name to set currently selected for loading	L(A ' set name ')
Select set for loading & assign name	L(A set number ' set name ')
Select specified character set for loading	L(A set number)
Specify character form in hex codes & ASCII reference character	L' ascii char ' hex codes

POSITION SELECTION

Select active position & move cursor to it	P[coordinates]
Specify position to move to via a PCV using PCV active only for current command	P(W(M pcm)) pcv
Specify series of positions to move to. Post-command active position = last specified position	P(S) positions & options (E)
Specify series of positions to move to. Post-command active position = pre-command active position	P(B) positions & options (E)
Specify writing with first bit of pattern	P[]

REQUESTING REPORTS

Request report of active position	R(P)
Request report of command set stored for specified character	R(M(character))
Request report of last detected error	R(E)
Request report of name assigned to char. set currently selected for loading	R(L)
Request report of total & remaining space available for storage of command sets	R(M(=))
Select GIN mode	R(P(I)

SCREEN SETTING

Select display addresing range	S(A[coords][coords]
Scroll from specified coordinates	S[coordinates]
Scroll from specified position	S(W(M pcm)) pcv
Define hardcopy region	S(H positions)
Specify printed image offset	S(H(positions)
Specify output map location lightness value (monochrome monitor)	S(M location (lightness))
Specify output map location RGB character	S(M location (rgb character))
Specify output map location HLS colour val.	S(M location (hls values))
Select specifed output map location colour for background	S(I location)
Select output map location colour approx. specified RGB character for background	S(I(rgb character))
Select output map location colour approx. specified HLS values for background	S(I(hls values))
Erase screen to current background	S(E)
Erase screen to specified background	S(I location ,E)
Specify time delay before command	S(T time)
Control cursor display	S(C control value)
Set temporary writing controls	S(W(sub-options))

TEXT DISPLAY

Select 1 of 17 standard sets of char. cell, character matrix & char. cell spacing	T(S standard set)
Select character cell size	T(S[cell width , cell height])
Select character cell & matrix height multiplier	T(H cell & matrix height mult)
Select character cell spacing	T[relative position]
Select character matrix size	T(U[matrix width , height])
Select character slant	T(I tilt)
Select character string & orientation	T(D string or S set D char or)
Select character string orientation	T(D orientation S set)
Select font	T(A font)
Select subscript/superscript/overstrike	T pcv
Select width & height multipliers for standard set 1 character matrix	T(M[width mult , height mult])
Set temporary writing controls	T(W sub-options))
Set temporary text controls	T(B) options (E)
Specify text	T' text'

VECTOR DRAWING

Draw point	V[]
Draw vector to specified coordinates	V [coordinates]
Draw vector to specified PCV	V pcv
Draw closed vector sequence	V(B) positions & options (E)
Draw open vector sequence	V(S) positions & options (E)
Set temporary writing controls	V(W(sub-options))

WRITING CONTROL

Character shading	W(S' chading character ')
Define writing pattern	W(P binary pattern)
Define writing pattern & set multiplier	W(P pattern def (M multiplier))
Pattern reversal control	W(N pattern control value)
Pixel memory bit plane selection	W(F plane selection value)
Select comp. writing with plane 0 enabled	W(F1,C)
Select comp. writing with plane 1 enabled	W(F2,C)
Select comp. writing both planes enabled	W(F3,C)
Select comp. writing for current write plane selection	W(C)
Select erase writing mode	W(E)
Select output map location colour approx. specified HLS values for foreground	W(I(hls values))
Select output map location colour approx. specified RGB character for foreground	W(I(rgb character))
Select overlay writing mode	W(V)
Select replace writing mode	W(R)
Select specified output map location foreground colour	W(I location)
Select standard writing pattern	W(P standard pattern number)
Select standard writing pattern & multiplier	W(P std pattern (M multiplier))
Set pixel change multiplier (PCM)	W(M multiplier)
Shading control	W(S(shading control value))
Shading horizontal limit selection	W(S(X)[x coordinate])
Shading vertical limit selection	W(S([, y coordinate]))

SCO Console Emulation

CONTROLLING TERMINAL PROCESSING

Sound audible tone	BEL
Select iBCSe2 compliance	ESC [= 2 L
Deselect iBCSe2 compliance	ESC [= 3 L
Save cursor position	ESC 7
Restore cursor position	ESC 8

KEYBOARD FUNCTIONS

Disable keyboard input	ESC [2 h
Enable keyboard input	ESC [2 l
Program function keys	ESC Q *key ' *data '

ASCII	105-Key ANSI	Enhanced PC-Style	*key Unshifted	*key Shifted
F1	F6	F1	0	<
F2	F7	F2	1	=
F3	F8	F3	2	>
F4	F9	F4	3	?
F5	F10	F5	4	@
F6	F11	F6	5	A
F7	F12	F7	6	B
F8	F13	F8	7	C
F9	F14	F9	8	D
F10	Help	F10	9	E
F11	Do	F11	:	F
F12	F17	F12	;	G
F13	F18		`	d
F14	F19		a	e
F15	F20		b	f
F16			c	g

*data = data string of up to 29 characters
' = a string delimiter (any character except contained in string)

COLOUR

Set foreground & background colour — ESC [2 ; *fg ; *bg m

*fg and *bg are integers as listed below:

Dull	Colour	Bold
0	Black	8
1	Blue	9
2	Green	10
3	Cyan	11
4	Red	12
5	Magenta	13
6	Yellow	14
7	White	15

Set inverse & foreground & background colour (* as above)	ESC [7 ; *fg ; *bg ; m
Set normal foreground colour (* = integer as above)	ESC [= * F
Set normal background colour (* = integer as above)	ESC [= * G
Set reverse foreground colour (* = integer as above)	ESC [= * H
Set reverse background colour (* = integer as above)	ESC [= * I
Set graphic foreground colour (* = integer as above)	ESC [= * J
Set graphic background colour (* = integer as above)	ESC [= * K
Request current colour attribute (* = integer as above)	ESC [= * M

CONTROLLING THE SCREEN DISPLAY

Assign * attribute(s) to following characters — ESC [* m

Default attributes	0
Bold on	1
Underline on	4
Flashing on	5
Inverse video on	7
Blank	8

Fill regions with attribute (* = integer as below) — ESC [= * L

* =	0	Fill with current attribute	2	Select iBCSe2 compliance
	1	Fill with normal attribute	3	Deselect iBCSe2 compliance

Select page (* = 1 - 6)	ESC [* z
Clear page	FF
Scroll display up * lines, insert blank lines	ESC [* S
Scroll display down * lines, insert blank lines	ESC [* T
Access bank 1 & 2 characters using decimal values	ESC [= * g (* = 0 - 255)
Clear tab stops (0 = cursor position, 3 = all)	ESC [* g

CURSOR

Cursor on	ESC [? 25 h
Cursor off	ESC [? 25 l
Move cursor to specified column (* = integer)	ESC [* ` *or* ESC [* G
Move cursor up * lines	ESC [* A
Move cursor down * lines	ESC [* B *or* ESC [* e
Move cursor right * columns	ESC [* C *or* ESC [* a
Move cursor left * columns	ESC [* D
Move cursor to specified line (* = integer)	ESC [* d
Move cursor to line (*l) column (*c)	ESC [*l ; *c H
Move cursor to line (*l) column (*c)	ESC [*l ; *c f
Move cursor down 1 line & to column 1	NAK
Move cursor to start of line	CR
Move cursor down 1 line in current column	LF
Move cursor up * lines & to first column	ESC [* F
Move cursor down * lines & to first column	ESC [* E
Move cursor one column left	BS
Move cursor to next tab stop	HT
Tab cursor backward * tabs	ESC [* Z
Enable cursor autowrap	ESC [? 7 h
Disable cursor autowrap	ESC [? 7 l

EDITING

Erase from cursor to end of display	ESC [0 J
Erase from start of display to cursor	ESC [1 J
Erase entire display	ESC [2 J
Erase from cursor to end of line	ESC [0 K
Erase from start of line to cursor	ESC [1 K
Erase entire line	ESC [2 K
Erase * characters from cursor right	ESC [* X
Insert * blank characters	ESC [* @
Insert * blank lines	ESC [* L
Delete * lines from cursor position down	ESC [* M
Delete * characters	ESC [* P
Set tab at cursor position	ESC H

PRINTING

Send page	ESC [2 i
Send line	ESC [? 3 i
Print page	ESC [0 i
Print all pages	ESC [? 11 i
Print cursor line	ESC [? 1 i
Transparent print mode off	ESC [? 4 i
Transparent print mode on	ESC [? 5 i
Auxiliary print mode on	ESC [5 i
Auxiliary print mode off	ESC [4 i

Siemens 97801 Emulation

CHARACTER SET SELECTION

Assign International character set to G0	ESC (@
Assign International character set to G1	ESC) @
Assign International A character set to G0	ESC (B
Assign International A character set to G1	ESC) B
Assign German character set to G0	ESC (K
Assign German character set to G1	ESC) K
Assign Brackets character set to G0	ESC (w
Assign Brackets character set to G1	ESC) w
Assign FACET character set to G0	ESC (c
Assign FACET character set to G1	ESC) c
Assign IBM character set to G0	ESC (v
Assign IBM character set to G1	ESC) v
Assign Euro character set to G0	ESC (u
Assign Euro character set to G1	ESC) u
Assign Mathematics Symbols character set to G0	ESC (t
Assign Mathematics Symbols character set to G1	ESC) t
Assign Blanks character set to G0	ESC (y
Assign Blanks character set to G1	ESC) y
Assign G2 to G0	ESC (x
Assign G2 to G1	ESC) x
Load G2 with character set (*s = @ or B, K, w ...)	ESC * *s
Load character generator address(es) with new symbol	ESC R B ... ESC \
(Three bytes address and 28 byte description of pattern in hex format)	
Switch to G0 (same as keyboard Ctrl + O)	SI
Switch to G1 (same as keyboard Ctrl + N)	SO
Switch within G0 (national/international)	ESC [5 v
Lock Change Code key	ESC [10 v
Unlock Change Code key	ESC [11 v
Report current code set	ESC [13 v

CURSOR

Cursor on	ESC [7 p
Cursor off	ESC [6 p
Move cursor up * lines	ESC [* A
Move cursor down * lines	ESC [* B
Move cursor right * columns	ESC [* C
Move cursor left * columns	ESC [* D
Move cursor to line *l and column *c	ESC [*l ; *c H
Report cursor position	ESC [6 n
Short poll of cursor position	ESC [5 p
Move cursor left one column	BS
Move cursor one tab to the right	HT
Move cursor * tabs to the left	ESC [* Z
Line Feed	LF
Carriage Return	CR
Move cursor to begining of next line	ESC E

Save cursor position	ESC [s
Move cursor to last saved position	ESC [u

DISPLAY AREA MOVEMENT

Roll mode (do not destroy data)	ESC [10 u
Scroll mode (destroy data)	ESC [11 u
Move display up * lines	ESC [* S
Move display down * lines	ESC [* T

EDITING

Insert * (1-80) blanks to the right starting at cursor position	ESC [* @
Insert * (1-24 or 25) lines, starting at cursor position	ESC [* L
Delete * character(s) ,including attributes, at cursor position	ESC [* P
Delete * line(s), including attributes, at cursor position	ESC [* M
Erase line - * determines extent	ESC [* K
Erase in screen area - * determines extent	ESC [* J

INITIALIZING SCREEN & ATTRIBUTES

Define screen moving area at line *t (top) to line *b (bottom)	ESC [*t ; *b r
Set 24 line mode	ESC [1 u
Set 25 line mode	ESC [0 u
Switch on/off blinking in 24/25th line	ESC [* p
* = Flashing on, Line 1-25 / 1-24 (25/24 line mode)	0
Flashing off, Line 1-25 / 1-24 (25/24 line mode)	1
Flashing on, line 25	2
Flashing off, line 25	3
Auto roll mode enabled	ESC [9 u
Page mode enabled	ESC [8 u
Set attribute(s) for the following character	ESC [*a ; ...*a m

*a = Normal	0	Inverse	7
Screen	2	Hidden	8
Underline	4	Store attributes	50
Blink	5		

Set delete character to blank	ESC [3 u
Set delete character to nil	ESC [2 u
Dim screen	ESC [8 p
Undim screen	ESC [9 p
Set background dark (white characters on black screen)	ESC [20 u
Set background highlight (black characters on white screen)	ESC [21 u
Set off cursor position	ESC [10 p
Video timeout enabled	ESC [5 u
Video timeout disabled	ESC [4 u
Reset screen to initial state	ESC c

KEYBOARD

Lock keyboard	ESC `
Unlock keyboard	ESC b
Keyboard repeat rate off	ESC [0 s

Keyboard repeat rate on	ESC [1 s
Sound bell	BEL
Key click disabled	ESC [2 s
Key click enabled	ESC [3 s
Report keyswitch status	ESC [0 w
Map scan codes corresponding to German keyboard values	ESC [7 u
Map scan codes corresponding to all other keyboard values	ESC [6 u
Load more national keyboards (Multiuser systems: 1024 Byte, PC-X/PC-X10: 10280 Byte)	ESC R A ... ESC \
Read actual keyboard layout	ESC E " y

SERVICE

Display all ESCape sequences (except ESC) on screen	ESC [3 v
Switch to standard mode	ESC [2 v
Display all ESCape sequences - system must be switched off to return to standard mode	ESC [4 v
Start system test and poll result	ESC [3 y
Poll firmware	ESC [4 y
Poll keyboard firmware	ESC [5 y
Fill the screen moving area with the following character	ESC [8 v
Display complete character generator	ESC [9 v
Reset the previous two commands	ESC [7 v

Stratus V102 Emulation

GENERAL OPERATION

Block mode on	ESC B
Local mode on	ESC c
Half duplex mode on	ESC D H
Full duplex mode on	ESC D F
Return to previous conversational mode	ESC C
Set terminal operating mode(s) (0 = V102, 1 = 950)	ESC [= * h
Reset terminal operating mode(s) (0 = V102, 1 = 950)	ESC [= * l
Select a terminal operating value (0 = V102, 1 = 950)	ESC [*1 ; *2 v
Select a programming compatibility mode (0 = V102, 1 = 950)	ESC [10 ; * v
Reset terminal to factory default values	ESC ~ 0
Reset terminal to saved settings values	ESC ~ 1
Reset function keys to factory default values	ESC ~ 2
Reset editing keys to factory default values	ESC ~ 3
Monitor mode on	ESC U
Monitor mode off	ESC X *or* ESC u

CHARACTER SETS & BLOCK GRAPHICS

Select character set (0 = US ASCII, 1 = UK ASCII)	ESC [9 ; * v
Special graphics mode on	ESC $
Special graphics mode off	ESC %
Select a character from the multinational character set	CTRL U *
Read 7-bit data words	ESC [= 1 l
Read 8-bit data words	ESC [= 1 h
Define block graphics area	ESC H w h

CURSOR

Cursor home	ESC [H *or* RS
Line feed	LF
Reverse line feed	ESC j
New line (line feed/carriage return)	US
Carriage return	CR
Move cursor up * lines	ESC [* A
Move cursor up one line	VT
Move cursor down one line	SYN
Move cursor down * lines	ESC [* B
Move cursor right * columns	ESC [* C
Move cursor right one column	FF
Move cursor left * columns	ESC [* D
Move cursor left one column	BS
Move cursor to line (*l) and column (*c)	ESC [*l ; *c H
Move cursor to line (*l) and column (*c)	ESC [*l ; *c f
Send cursor to line (*l) & columns 1 - 80 (*c)	ESC = *l *c
Send cursor to line (*l) & columns 81 - 132 (*c)	ESC = *l ~ *c
Send cursor to page (*p), line (*l) & columns 1 - 80 (*c)	ESC - *p *l *c
Send cursor to page (*p), line (*l) & columns 81 - 132 (*c)	ESC - *p *l ~ *c

Read cursor's line & column position	ESC ?
Read cursor's page, line & column position	ESC /
Read cursor's line & column position in decimal units	ESC [6 n
Read cursor's page, line & column position in decimal units	ESC [? 6 n

DISPLAY

Turn screen on	ESC n
Turn screen off	ESC o
Light background with dark characters	ESC b
Dark background with light characters	ESC d
Define visual attribute(s) *	ESC G *

0	Normal (default) video	8	Underline
1	Invisible normal video	9	Invisible underline
2	Flash	:	Underline & flash
3	Invisible flash	;	Invisible underline & flash
4	Reverse current background	<	Reverse & underline
5	Invisible reverse	=	Invisible reverse & underline
6	Reverse and flash	>	Reverse, underline & flash
7	Invisible reverse & flash	?	Invisible reverse, underline & flash

Attributes occupy a character space	ESC F 0
Attributes do not occupy a character space	ESC F 1
Half intensity mode	ESC [= 5 h
Full intensity mode	ESC [= 5 l
Page base attribute mode	ESC [= 2 h
Line base attribute mode	ESC [= 2 l
Select status line visual attribute(s)	ESC [3 ; * v

0	Normal	2	Normal underline
1	Reverse	3	Reverse underline

Set cursor style to * ESC . *

0	Cursor not displayed	3	Flashing underline cursor
1	Flashing block cursor	4	Steady underline cursor
2	Steady block cursor		

Enable 132 columns per line	ESC [= 3 h
Enable 80 columns per line	ESC [= 3 l

EDITING MODES

Autowrap mode on	ESC [= 7 h
Autowrap mode off	ESC [= 7 l
New line mode on	ESC [= 6 h
New line mode off	ESC [= 6 l
DOWN key sends CTRL J	ESC [= 9 h
DOWN key sends CTRL V	ESC [= 9 l
Turn on write protect mode	ESC)
Turn off write protect mode	ESC (
Turn on protect mode	ESC &
Turn off protect mode	ESC '

EDITING DATA

Enable page edit mode	ESC N
Enable line edit mode	ESC O
Enable insert mode	ESC q
Enable replace mode	ESC r
Load a replacement character	ESC e *
Insert a replacement character at cursor position	ESC Q
Insert * replacement characters at cursor position	ESC [* @
Insert line of replacement characters on current line	ESC E
Insert * lines of replacement characters starting at cursor line	ESC [* L
Delete character at cursor position	ESC W
Delete * characters starting at cursor position	ESC [* P
Delete current line & replace with replacement characters	ESC R
Delete * lines at cursor line & replace with replacement characters	ESC [* M
Erase from cursor to end of line & replace with replacement characters	ESC T
Erase line portion & replace with replacement characters (0 = from cursor, 1 = to cursor, 2 = all unprotected)	ESC [* K
Erase from cursor to end of line & replace with null characters	ESC t
Erase from cursor to end of page & replace with replacement characters	ESC Y
Erase page portion & replace with replacement characters (0 = from cursor, 1 = to cursor, 2 = all unprotected)	ESC [* J
Erase from cursor to end of page & replace with null characters	ESC y
Clear current unprotected field, replace with replacement characters	CAN
Clear all characters & replace with null characters	ESC *
V102 mode: Clear all characters & replace with replacement characters (reset protect and write protect modes)	ESC +
950 mode: Clear unprotected characters & replace with replacement characters (do not reset protect and write protect modes)	ESC +
V102 mode: Clear unprotected characters & replace with write protected space characters (reset protect mode)	ESC ,
950 mode: Clear unprotected characters & replace with write-protected space characters (do not reset protect mode)	ESC ,
Clear unprotected characters & replace with replacement characters	ESC ; *or* SUB
Clear unprotected characters & replace with null characters	ESC :

FUNCTION KEYS

Select function key set (0 = set one, 1 = set two) ESC [7 ; * v

Reprogram a function key ESC ¦ *1 *2 <message> EM

	V102 compatible *1 value			V102 compatible *1 value	
Key	Unshifted	Shifted	Key	Unshifted	Shifted
F1	1	A	F9	9	I
F2	2	B	F10	:	J
F3	3	C	F11	;	K
F4	4	D	F12	<	L
F5	5	E	F13	=	M
F6	6	F	F14	>	N
F7	7	G	F15	?	O
F8	8	H	F16	@	P

Additional *1 values:	<space>	Clear entire current function key set from memory
	0	Load function keys in sequence

*2 message destination values:	1	Send message to host
	2	Send message to terminal
	3	Send message to both host and terminal

Save function key reprogramming in non-volatile memory	ESC [= 10 h
Do not save function key reprogramming in non-volatile memory	ESC [= 10 l
Invoke a function key	ESC [* ¦

REPROGRAMMING EDITING KEYS

Reprogram the SEND key (950 mode)	ESC 0 *k *c

*k =	1	Unshifted	2	Shifted
*c =	4	ESC 4 command	7	ESC 7 command
	5	ESC 5 command	S	ESC S command
	6	ESC 6 command	s	ESC s command

Reprogram any individual editing key (V102 mode)	ESC 0 *Ps *p1 *p2 *p3
Reprogram all editing keys	ESC] *Ps *p1 ... *p60

KEYBOARD & BELL

Keyboard locked	ESC #
Keyboard unlocked	ESC "
Lock keys	ESC [= * l
Unlock keys	ESC [= * h

* =	11	Set Up	13	Clear Space
	12	Esc	14	Break

Local editing key mode enabled	ESC k
Duplex editing key mode enabled	ESC l
Load the margin bell column	ETB
Margin bell on	ESC [= 4 h
Margin bell off	ESC [= 4 l
Sound bell	BEL

LOADING & SENDING MESSAGES

Send terminal's identification	ESC M
Program the answerback message	ESC ^ <text> EM
Send the answerback message	ENQ
Display user message 1 on bottom screen line	ESC g
Display status line on bottom screen line	ESC h
Select contents of top information line	ESC [4 ; * v
Select contents of bottom information line	ESC [5 ; * v

* =	0	Blank	2	User message 1
	1	Status line	3	User message 2

Load text into user message 1	ESC f <text> CR
Load text into a user message	ESC _ *m *e <text> CR

User Message			Effect		
*m =	0	Status line message field	*e =	0	Clears message before loading
	1	User message 1		1	Writes over existing message
	2	User message 2			

Send contents of information line — ESC Z *

* = 0 User message 1
1 Status line
2 User message 2

PRINTING

Buffered copy print mode on	ESC @
Buffered copy print mode off	ESC A
Buffered transparent print mode on	ESC `
Buffered transparent print mode off	ESC a
Buffered bidirectional print mode on	DC2
Buffered bidirectional print mode off	DC4
Print unprotected formatted page	ESC P
Print all unformatted page	ESC L
Page print	ESC [0 ; * i

* =	0	Formatted all	4	Unformatted all	
	1	Formatted unprotected	5	Unformatted unprotected	

Define page print terminator — ESC p *

SCREEN MEMORY

Define number of lines on each page — ESC \ *

	Lines per Page	Number of Pages 4-Page Memory	Number of Pages 2-Page Memory
* = 1	24	4	2
2	48	2	1
3	96	1	-

Autopage mode on	ESC v
Autopage mode off	ESC w
Display previous page	ESC J
Display next page	ESC K
Display page *	ESC [1 ; * }

SCROLLING

Define a scrolling region	ESC [*t ; *b r
Enable line lock	ESC ! 1
Disable line lock	ESC ! 2

SENDING SCREEN DATA

Reprogram delimiter (*d) characters *1 & *2 — ESC x *d *1 *2

0	Field Separator	2	Start of protected field	4	Message terminator
1	End of line	3	End of protected field		

Send unprotected characters in cursor line up to & including cursor	ESC 4
Send unprotected page up to & including cursor	ESC 5
Send entire cursor line characters up to & including cursor	ESC 6
Send entire page up to & including cursor	ESC 7
Send unprotected message between start of text and end of text	ESC S
Send whole message between start of text (STX) and end of text (ETX)	ESC s

TAB STOPS

Clear typewriter tab stop at cursor position	ESC 2
Clear all typewriter tab stops	ESC 3
Create column of tab stops at cursor position	ESC 1
Move cursor forward to next typewriter or field tab stop	HT
Move cursor forward to next field tab stop	ESC i
Move cursor backward to previous typewriter or field tab stop	ESC I

TA6530 Emulation

Conversational & Block Mode

ATTRIBUTES

Set display attribute *			ESC 6 *
Normal video	SP	Inverse invisible	,
Dimmed	!	Underline dim	1
Inverse video	$	Underline italics	2
Italics	"	Underline inverse	4
Invisible	(	Underline inverse dim	5
Underline	0	Underline inverse italics	6
Inverse dim	%	Underline invisible	8
Inverse italics	&	Underline inverse invisible	?

Set video prior condition register (* = attribute as above)	ESC 7 *
Set/reset colour map table	ESC - q
Read colour configuration (Block mode)	ESC - u
Set colour configuration (Block mode)	ESC - t
Read colour mapping table (Block mode)	ESC - v
Set colour mapping table (Block mode)	ESC - x

CURSOR

Back tab (Block mode)	ESC i
Clear all tabs	ESC 3
Clear tab at current position	ESC 2
Move cursor down one line	LF
Move cursor home	ESC H
Move cursor home down	ESC F
Move cursor left one column	BS
Move cursor right one column	ESC C
Move cursor to beginning of current line	CR
Move cursor to next tab stop	HT
Move cursor up one line	ESC A
Report cursor address	ESC a
Set cursor address	DC3
Set cursor address extended (Block mode)	ESC - D
Set tab at current position	ESC 1

DISPLAY

Clear memory to spaces	ESC I
Clear memory to spaces extended	ESC - I
Display message on 25th line	ESC o
Display next page (Conversational mode)	ESC U
Display previous page (Conversational mode)	ESC V
Display page *	ESC ; *
Scroll down one line (Conversational mode)	ESC T
Scroll up one line (Conversational mode)	ESC S

Select page * (Block mode)	ESC : *
Set current line to 40 (double width) columns	ESC 8
Set current line to 80 (single width) columns	ESC 9
Set max. number of pages for display (Block mode)	ESC p *
Set page size to 24 lines by 40 characters	ESC t
Start field (Block mode)	GS
Start field extended (Block mode)	ESC [*display *data *case
Start enhanced colour field (Block mode)	ESC `
Define field using pre-defined attributes (Block mode)	FS
Write message (*m) on 25th line	ESC o *m

EDITING

Delete character (Block mode)	ESC P
Delete line (Block mode)	ESC M
Disable local line editing (Block mode)	ESC N
Erase to end of line/field	ESC K
Erase to end of page/memory	ESC J
Insert character (Block mode)	ESC O
Insert line (Block mode)	ESC L
Reset modified data tags (Block mode)	ESC >

GENERAL OPERATION

Data compression (Enhanced) (Block mode)	DC2
Data compression (Limited) (Block mode)	DC4
Define data-type table (Block mode)	ESC r
Define data-type table extended (Block mode)	ESC - r
Define/update variable table (Block mode)	ESC - s
Delay processing for one second	ESC @
End of transmission (Block mode)	EOT
Enter protect submode (Block mode)	ESC W
Exit protect submode (Block mode)	ESC X
Execute self tests	ESC - P
Execute self test while loading	ESC z
Print current screen (Conv) or selected page (Block)	ESC 0 (zero)
Read buffer (Block mode)	ESC <
Read emulation configuration	ESC ?
Read with address (Block mode)	ESC = *
Read with address extended (Block mode)	ESC - J *
Read with address all (Block mode)	ESC] *
Read with address all extended (Block mode)	ESC - K *
Read with all attributes (Block mode)	ESC Q
Reinitialize (Block mode)	ESC q
Set buffer address (Block mode)	DC1
Set buffer address extended	ESC - C
Set emulation configuration	ESC v *
Soft reset	ESC [! p
Text start (Block mode)	STX
Text end (Block mode)	ETX
Sound bell	BEL

KEYBOARD

Define Return key (Conversational mode)	ESC u *
Lock keyboard	ESC c
Unlock keyboard	ESC b
Simulate function key	ESC d *

REPORTS

Report cursor address	ESC a
Report emulation status	ESC ^
Report firmware revision level	ESC _

ANSI Mode

CHARACTER SETS

Select G0 character set *	ESC (*
Select G1 character set *	ESC) *
Shift out to G0 character set	SI
Shift out to G1 character set	SO

CURSOR

Autowrap enabled	ESC [? 7 h
Autowrap disabled	ESC [? 7 l
Clear tab stops (0 = cursor position, 2 or 3 = all)	ESC [* g
Cursor enabled	ESC [? 25 h
Cursor disabled	ESC [? 25 l
Horizontal tab	HT
LF is LFCR (new line mode)	ESC [20 h
LF is LF only	ESC [20 l
Move cursor left one column	BS
Move cursor left * columns	ESC [* D
Move cursor right * columns	ESC [* C
Move cursor down one line	LF *or* VT *or* FF
Move cursor down one line	ESC D *or* IND
Move cursor down * lines	ESC [* B
Move cursor up one line	ESC M *or* RI
Move cursor up * lines	ESC [* A
Move cursor to beginning of line	CR
Move cursor to column one of next line	NEL
Move cursor to column one of next line	ESC E
Move cursor to column one & down * lines	ESC [* E
Move cursor to column one & up * lines	ESC [* F
Move cursor to column * on current line	ESC [* G
Move cursor to line *l and column *c	ESC [*l ; *c H
Move cursor * tab stops	ESC [* I
Move cursor * previous tab stops	ESC [* Z
Set/clear * tab stops	ESC [* W
Set horizontal tab	ESC H *or* HTS

DISPLAY

Insert * spaces at cursor location	ESC [* @
Restore cursor position & video attribute	ESC 8
Save cursor position & video attribute	ESC 7
Set current line to single width (80 columns)	ESC # 5
Set current line to double width (40 columns)	ESC # 6
Scroll display down * lines	ESC [* T
Scroll display up * lines	ESC [* S
Set current video attributes	ESC [* m
Write to message field	ESC P * = * ESC \

EDITING

Delete * blank lines at cursor position	ESC [* M
Delete * characters at cursor position	ESC [* P
Erase * characters & attributes at cursor	ESC [* X
Erase field *	ESC [* N
Erase line portion (0 = from, 1 = to, 2 = all)	ESC [* K
Erase screen portion (0 = from, 1 = to, 2 = all)	ESC [* J
Insert * blank lines at cursor position	ESC [* L
Repeat last character * times	ESC [* b

GENERAL OPERATION

Cancel current escape sequence	CAN
Execute self tests	ESC [* y
Reset emulation	ESC c
Set emulation configuration	ESC P * ? * ESC \
Set mode	ESC [* h
Reset mode	ESC [* l
Sound bell	BEL
Read screen contents	ESC [* ; * ; * ; * v
Terminate current escape sequence	SUB
Transmission enabled (XON)	DC1
Transmission disabled (XOFF)	DC3

KEYBOARD

Lock keyboard	ESC `
Lock keyboard	ESC [2 h
Unlock keyboard	ESC b
Unlock keyboard	ESC [2 l

REPORTS

Report configuration values	ESC [* x
Report cursor position	ESC [6 *
Report terminal ID & version	ESC [* c
Report status of last self test	ESC [5 *

Tek 4010/4014 Emulation

ALPHANUMARIC MODE

Select G0 character set for alpha text	ESC SI
Select G1 character set for alpha text	ESC SO

CURSOR MOVEMENT

Move cursor down 1 pixel	ESC \ b
Move cursor down 16 pixels	ESC \ B
Move cursor left 1 pixel	ESC \ d
Move cursor left 16 pixels	ESC \ D
Move cursor right 1 pixel	ESC \ c
Move cursor right 16 pixels	ESC \ C
Move cursor up 1 pixel	ESC \ a
Move cursor up 16 pixels	ESC \ A
Move cursor to beginning of line	ESC \ CR
Move text cursor down	LF
Move text cursor left	BS
Move text cursor right	HT
Move text cursor up	VT
Move text cursor to start of line	CR

GENERAL OPERATION

Clear screen & enter Graphics Text mode	ESC FF
Copy screen data to parallel port	ESC ETB
Request status report	ESC ENQ
Select bypass mode	ESC CAN
Sound bell	BEL

GRAPHICS TEXT MODE

Select character size 0 (80x34)	ESC 0
Select character size 1 (40x17)	ESC 1
Select character size 2 (26x11)	ESC 2
Select character size 3 (20x8)	ESC 3
Select Graphics Text font 1 (74x35)	ESC 8
Select Graphics Text font 2 (81x38)	ESC 9
Select Graphics Text font 3 (121x58)	ESC :
Select Graphics Text font 4 (133x64)	ESC ;
Select text zoom factor 1	ESC \ e
Select text zoom factor 2	ESC \ f
Select text zoom factor 3	ESC \ g
Select text zoom factor 4	ESC \ h

MODE SELECTION

Select Alphanumeric mode	ACK
Select Alphanumeric mode	CAN
Select Alphanumeric mode	ESC \ CAN
Select GIN mode	ESC SUB

Select Graphics Text mode	CR
Select Graphics Text mode	US
Select Graphics Text mode	ESC FF
Select Incremental Point mode	RS
Select Point mode	FS
Select Vector mode	GS
Select Vector mode	ESC \ GS

VECTOR MODE

Clear Vector mode screen	ESC \ FF
Deselect write-through & selective erase	ESC SOH
Disable block fill/erase	ESC ETX
Disable dark vector	BEL
Disable dark vector	ESC BEL
Disable rectangle drawing	ESC \ r
Enable block fill/erase	ESC STX
Enable circle plotting (* = ASCII equiv. of coords)	ESC \ O *
Enable dark vector	GS
Enable dark vector	ESC GS
Enable rectangle drawing (*c = corner coords)	ESC \ R *c *c
Enable selective erase	ESC DLE
Enable write-through mode	ESC NAK
Select dot-dash line style	ESC b
Select dot-dash line style	ESC j
Select dotted line style	ESC a
Select dotted line style	ESC i
Select long dash line style	ESC d
Select long dash line style	ESC I
Select short dash line style	ESC c
Select short dash line style	ESC k
Select solid line style	ESC `
Select solid line style	ESC e
Select solid line style	ESC f
Select solid line style	ESC g
Select solid line style	ESC h
Select solid line style	ESC m
Select solid line style	ESC n
Select solid line style	ESC o

TVI 955 Emulation

GENERAL OPERATION

Block mode on	ESC B
Local mode on	ESC c
Half duplex mode on	ESC D H
Full duplex mode on	ESC D F
Return to previous conversational mode	ESC C
Set terminal operating mode(s) (0 = 955, 1 = 950)	ESC [= * h
Reset terminal operating mode(s) (0 = 955, 1 = 950)	ESC [= * l
Select a terminal operating value (0 = 955, 1 = 950)	ESC [*1 ; *2 v
Select a programming compatibility mode (0 = 955, 1 = 950)	ESC [10 ; * v
Reset terminal to factory default values	ESC ~ 0
Reset terminal to saved settings values	ESC ~ 1
Reset function keys to factory default values	ESC ~ 2
Reset editing keys to factory default values	ESC ~ 3
Monitor mode on	ESC U
Monitor mode off	ESC X *or* ESC u

CHARACTER SETS & BLOCK GRAPHICS

Select character set (0 = US ASCII, 1 = UK ASCII)	ESC [9 ; * v
Special graphics mode on	ESC $
Special graphics mode off	ESC %
Select a character from the multinational character set	CTRL U *
Read 7-bit data words	ESC [= 1 l
Read 8-bit data words	ESC [= 1 h
Define block graphics area	ESC H w h

CURSOR

Cursor home	ESC [H *or* RS
Line feed	LF
Reverse line feed	ESC j
New line (line feed/carriage return)	US
Carriage return	CR
Move cursor up * lines	ESC [* A
Move cursor up one line	VT
Move cursor down one line	SYN
Move cursor down * lines	ESC [* B
Move cursor right * columns	ESC [* C
Move cursor right one column	FF
Move cursor left * columns	ESC [* D
Move cursor left one column	BS
Move cursor to line (*l) and column (*c)	ESC [*l ; *c H
Move cursor to line (*l) and column (*c)	ESC [*l ; *c f
Send cursor to line (*l) & columns 1 - 80 (*c)	ESC = *l *c
Send cursor to line (*l) & columns 81 - 132 (*c)	ESC = *l ~ *c
Send cursor to page (*p), line (*l) & columns 1 - 80 (*c)	ESC - *p *l *c
Send cursor to page (*p), line (*l) & columns 81 - 132 (*c)	ESC - *p *l ~ *c

Read cursor's line & column position	ESC ?
Read cursor's page, line & column position	ESC /
Read cursor's line & column position in decimal units	ESC [6 n
Read cursor's page, line & column position in decimal units	ESC [? 6 n

DISPLAY

Turn screen on	ESC n
Turn screen off	ESC o
Light background with dark characters	ESC b
Dark background with light characters	ESC d
Define visual attribute(s) *	ESC G *

0	Normal (default) video	8	Underline
1	Invisible normal video	9	Invisible underline
2	Flash	:	Underline & flash
3	Invisible flash	;	Invisible underline & flash
4	Reverse current background	<	Reverse & underline
5	Invisible reverse	=	Invisible reverse & underline
6	Reverse and flash	>	Reverse, underline & flash
7	Invisible reverse & flash	?	Invisible reverse, underline & flash

Attributes occupy a character space	ESC F 0
Attributes do not occupy a character space	ESC F 1
Half intensity mode	ESC [= 5 h
Full intensity mode	ESC [= 5 l
Page base attribute mode	ESC [= 2 h
Line base attribute mode	ESC [= 2 l
Select status line visual attribute(s)	ESC [3 ; * v

0	Normal	2	Normal underline
1	Reverse	3	Reverse underline

Set cursor style to * ESC . *

0	Cursor not displayed	3	Flashing underline cursor
1	Flashing block cursor	4	Steady underline cursor
2	Steady block cursor		

Enable 132 columns per line	ESC [= 3 h
Enable 80 columns per line	ESC [= 3 l

EDITING MODES

Autowrap mode on	ESC [= 7 h
Autowrap mode off	ESC [= 7 l
New line mode on	ESC [= 6 h
New line mode off	ESC [= 6 l
DOWN key sends CTRL J	ESC [= 9 h
DOWN key sends CTRL V	ESC [= 9 l
Turn on write protect mode	ESC)
Turn off write protect mode	ESC (
Turn on protect mode	ESC &
Turn off protect mode	ESC '

EDITING DATA

Enable page edit mode	ESC N
Enable line edit mode	ESC O
Enable insert mode	ESC q
Enable replace mode	ESC r
Load a replacement character	ESC e *
Insert a replacement character at cursor position	ESC Q
Insert * replacement characters at cursor position	ESC [* @
Insert line of replacement characters on current line	ESC E
Insert * lines of replacement characters starting at cursor line	ESC [* L
Delete character at cursor position	ESC W
Delete * characters starting at cursor position	ESC [* P
Delete current line & replace with replacement characters	ESC R
Delete * lines at cursor line & replace with replacement characters	ESC [* M
Erase from cursor to end of line & replace with replacement characters	ESC T
Erase line portion & replace with replacement characters (0 = from cursor, 1 = to cursor, 2 = all unprotected)	ESC [* K
Erase from cursor to end of line & replace with null characters	ESC t
Erase from cursor to end of page & replace with replacement characters	ESC Y
Erase page portion & replace with replacement characters (0 = from cursor, 1 = to cursor, 2 = all unprotected)	ESC [* J
Erase from cursor to end of page & replace with null characters	ESC y
Clear current unprotected field, replace with replacement characters	CAN
Clear all characters & replace with null characters	ESC *
955 mode: Clear all characters & replace with replacement characters (reset protect and write protect modes)	ESC +
950 mode: Clear unprotected characters & replace with replacement characters (do not reset protect and write protect modes)	ESC +
955 mode: Clear unprotected characters & replace with write protected space characters (reset protect mode)	ESC ,
950 mode: Clear unprotected characters & replace with write-protected space characters (do not reset protect mode)	ESC ,
Clear unprotected characters & replace with replacement characters	ESC ; *or* SUB
Clear unprotected characters & replace with null characters	ESC :

FUNCTION KEYS

Select function key set (0 = set one, 1 = set two)	ESC [7 ; * v
Reprogram a function key	ESC ¦ *1 *2 <message> EM

Key	955 compatible *1 value Unshifted	Shifted	Key	955 compatible *1 value Unshifted	Shifted
F1	1	A	F9	9	I
F2	2	B	F10	:	J
F3	3	C	F11	;	K
F4	4	D	F12	<	L
F5	5	E	F13	=	M
F6	6	F	F14	>	N
F7	7	G	F15	?	O
F8	8	H	F16	@	P

Additional *1 values:	<space>	Clear entire current function key set from memory
	0	Load function keys in sequence

*2 message destination values:	1	Send message to host
	2	Send message to terminal
	3	Send message to both host and terminal

Save function key reprogramming in non-volatile memory	ESC [= 10 h
Do not save function key reprogramming in non-volatile memory	ESC [= 10 l
Invoke a function key	ESC [* ¦

REPROGRAMMING EDITING KEYS

Reprogram the SEND key (950 mode)	ESC 0 *k *c

*k =	1	Unshifted	2	Shifted
*c =	4	ESC 4 command	7	ESC 7 command
	5	ESC 5 command	S	ESC S command
	6	ESC 6 command	s	ESC s command

Reprogram any individual editing key (955 mode)	ESC 0 *Ps *p1 *p2 *p3
Reprogram all editing keys	ESC] *Ps *p1 ... *p60

KEYBOARD & BELL

Keyboard locked	ESC #
Keyboard unlocked	ESC "
Lock keys	ESC [= * l
Unlock keys	ESC [= * h

* =	11	Set Up	13	Clear Space
	12	Esc	14	Break

Local editing key mode enabled	ESC k
Duplex editing key mode enabled	ESC l
Load the margin bell column	ETB
Margin bell on	ESC [= 4 h
Margin bell off	ESC [= 4 l
Sound bell	BEL

LOADING & SENDING MESSAGES

Send terminal's identification	ESC M
Program the answerback message	ESC ^ <text> EM
Send the answerback message	ENQ
Display user message 1 on bottom screen line	ESC g
Display status line on bottom screen line	ESC h
Select contents of top information line	ESC [4 ; * v
Select contents of bottom information line	ESC [5 ; * v

* =	0	Blank	2	User message 1
	1	Status line	3	User message 2

Load text into user message 1	ESC f <text> CR
Load text into a user message	ESC _ *m *e <text> CR

User Message			Effect		
*m =	0	Status line message field	*e =	0	Clears message before loading
	1	User message 1		1	Writes over existing message
	2	User message 2			

Send contents of information line — ESC Z *

* =	0	User message 1
	1	Status line
	2	User message 2

PRINTING

Buffered copy print mode on	ESC @
Buffered copy print mode off	ESC A
Buffered transparent print mode on	ESC `
Buffered transparent print mode off	ESC a
Buffered bidirectional print mode on	DC2
Buffered bidirectional print mode off	DC4
Print unprotected formatted page	ESC P
Print all unformatted page	ESC L
Page print	ESC [0 ; * i

* =	0	Formatted all	4	Unformatted all	
	1	Formatted unprotected	5	Unformatted unprotected	

Define page print terminator — ESC p *

SCREEN MEMORY

Define number of lines on each page — ESC \ *

	Lines per Page	Number of Pages 4-Page Memory	Number of Pages 2-Page Memory
* = 1	24	4	2
2	48	2	1
3	96	1	-

Autopage mode on	ESC v
Autopage mode off	ESC w
Display previous page	ESC J
Display next page	ESC K
Display page *	ESC [1 ; * }

SCROLLING

Define a scrolling region	ESC [*t ; *b r
Enable line lock	ESC ! 1
Disable line lock	ESC ! 2

SENDING SCREEN DATA

Reprogram delimiter (*d) characters *1 & *2 — ESC x *d *1 *2

0	Field Separator	2	Start of protected field	4	Message terminator
1	End of line	3	End of protected field		

Send unprotected characters in cursor line up to & including cursor	ESC 4
Send unprotected page up to & including cursor	ESC 5
Send entire cursor line characters up to & including cursor	ESC 6
Send entire page up to & including cursor	ESC 7
Send unprotected message between start of text and end of text	ESC S
Send whole message between start of text (STX) and end of text (ETX)	ESC s

TAB STOPS

Clear typewriter tab stop at cursor position	ESC 2
Clear all typewriter tab stops	ESC 3
Create column of tab stops at cursor position	ESC 1
Move cursor forward to next typewriter or field tab stop	HT
Move cursor forward to next field tab stop	ESC i
Move cursor backward to previous typewriter or field tab stop	ESC I

Unisys T27 Emulation

FORM OPERATIONS

Set forms mode	ESC W
Reset forms mode	ESC X
Toggle forms mode	DC2
Exit forms mode	SOH

SEARCH OPERATIONS

Set search mode	ESC E
Set search character (if 80 hex or less)	ESC - *
Set search character (if greater than 80 hex)	ESC - ESC SO * ESC SI
Reset search mode	ESC F

POINTER MOVEMENT

Move pointer up	DC3
Move pointer down	LF
Move pointer right	ESC C *or* DC2
Move pointer left	BS
Align KBC to DCP	ESC &
Tab right	HT
Clear variable tabs	ESC #
Vertical tab down	VT
Home	DC4
Jump to page *	ESC $ *
Position pointer at *c column & *r row (using hex values)	ESC ^ *c *r
Position pointer at *c column & *r row (using ASCII chars.)	ESC " *c *r
Carriage return	CR *or* LF
Prevent align in cursor page	ESC 6

EDITING SEQUENCES

Insert character by line	ESC !
Insert character by page	ESC @
Delete character by line	ESC %
Delete character by page	ESC P
Line insert	ESC L
Line delete	EC M
Move line up	ESC >
Move line down	ESC <
Clear to end of line	ESC K *or* DC1
Clear to end of page	ESC J
Roll page up	ESC S
Roll page down	ESC T
Clear page	SOH *or* FF *or* ESC FF

VIDEO ATTRIBUTE SEQUENCES

Underline video highlight	ESC 3
Reverse video highlight	ESC 4
End highlight	ETB
Start video blink	CAN
Start secure video	EM
Start bright video	SUB
Set reverse video	ESC N
Set normal video	ESC O

TERMINAL RECONFIGURATION SEQUENCES

Store ASCII codes as data	ESC R A *start *length *data ESC R C
Configure Data Comm, Screen & KPT buffer sizes	ESC R B *dc *s *kpt
Reconfigure terminal	ESC R C
Selective key programming	ESC R D *hex *prog 00 ... *hex *prog 00 00
Store data in hexadecimal codes	ESC R H *start *length *data ESC R C
Program function keys F1 - F10	ESC R K *length *key *prog A 9 ... *length *key *prog A 9 A 9
Transmit error log	ESC R L
Copy temporary storage into nonvolatile RAM	ESC R P
Display message in environmental user status line	ESC R S *number of bytes *data
Transmit memory (temporary storage) contents to host	ESC R T *start *number of bytes

MISCELLANEOUS SEQUENCES

Start escape sequence	ESC
Set shift out (to extended character set)	ESC SO *or* SO
Reset shift out (shift in)	ESC SI *or* SI
Character translation	ESC ' *

* =	:	bright	7	end highlight
	/	underscore	8	blink
	.	reverse	9	secure

Reset keystroke lockout	ESC =
Fill with character *	ESC _ *
Transmit terminal screen to host	ESC (
Toggle audible alarm	ESC ? *or* BEL
Display character set	ESC space C
Load contents of permanent storage & reconfigure	ESC space D
Display version number	ESC space V
Toggle variable tabs	ESC . *or* VT
Set mobile home	ESC D
Lowercase disable	ESC Y
Lowercase enable	ESC Z
End of text processing	ETX
End of transmission	EOT

Viewdata Mode

COLOUR SELECTION

Background cell colour = previous character colour	ESC]
Cell background colour black (default)	ESC \
Select blue alphanumeric character set	ESC D
Select blue graphics character set	ESC T
Select cyan alphanumeric character set	ESC F
Select cyan graphics character set	ESC V
Select green alphanumeric character set	ESC B
Select green graphics character set	ESC R
Select magenta alphanumeric character set	ESC E
Select magenta graphics character set	ESC U
Select red alphanumeric character set	ESC A
Select red graphics character set	ESC Q
Select white alphanumeric character set	ESC G
Select white graphics character set	ESC W
Select yellow alphanumeric character set	ESC C
Select yellow graphics character set	ESC S

CHARACTER ATTRIBUTES

Block graphics	ESC Y
Border graphics	ESC Z
Flashing on	ESC H
Flashing off	ESC I
Single line height	ESC L
Double line height	ESC M

CURSOR

Cursor display on	DC1
Cursor display off	DC2
Home cursor	RS
Move cursor down one line	LF
Move cursor left one column	BS
Move cursor right one column	HT
Move cursor up one line	VT
Move cursor to beginning of line	CR

DISPLAY

Clear screen & home cursor	FF
Conceal display	ESC X
Hold graphics character	ESC ^
Release graphics character	ESC _

GENERAL OPERATION

Enter 40 column viewdata mode	ESC # 6
Enter 80 column viewdata mode	ESC # 5
Send Answerback string	ENQ

VT640 Emulation

ALPHANUMARIC MODE

Select G0 character set for alpha text	ESC SI
Select G1 character set for alpha text	ESC SO

CURSOR MOVEMENT

Move cursor down 1 pixel	ESC \ b
Move cursor down 16 pixels	ESC \ B
Move cursor left 1 pixel	ESC \ d
Move cursor left 16 pixels	ESC \ D
Move cursor right 1 pixel	ESC \ c
Move cursor right 16 pixels	ESC \ C
Move cursor up 1 pixel	ESC \ a
Move cursor up 16 pixels	ESC \ A
Move cursor to beginning of line	ESC \ CR
Move text cursor down	LF
Move text cursor left	BS
Move text cursor right	HT
Move text cursor up	VT
Move text cursor to start of line	CR

GENERAL OPERATION

Clear screen & enter Graphics Text mode	ESC FF
Copy screen data to parallel port	ESC ETB
Request status report	ESC ENQ
Select bypass mode	ESC CAN
Sound bell	BEL
Specify data (*) for graphics memory loading	ESC + * #
Specify screen address for g. memory load	ESC " *h ; *v a
Specify writing mode (0 = pix. off, 1 = on, 2 = comp)	ESC / * d
Transmit graphics memory (*h *v coords, *s bit sets)	ESC " *h ; *v *s c

GRAPHICS TEXT MODE

Select character size 0 (80x34)	ESC 0
Select character size 1 (40x17)	ESC 1
Select character size 2 (26x11)	ESC 2
Select character size 3 (20x8)	ESC 3
Select Graphics Text font 1 (74x35)	ESC 8
Select Graphics Text font 2 (81x38)	ESC 9
Select Graphics Text font 3 (121x58)	ESC :
Select Graphics Text font 4 (133x64)	ESC ;
Select text zoom factor 1	ESC \ e
Select text zoom factor 2	ESC \ f
Select text zoom factor 3	ESC \ g
Select text zoom factor 4	ESC \ h

MODE SELECTION

Select Alphanumeric mode	ACK
Select Alphanumeric mode	CAN
Select Alphanumeric mode	ESC \ CAN
Select GIN mode	ESC SUB
Select GIN mode	ESC " 4 g
Select GIN mode	ESC " 5 g
Select Graphics Text mode	CR
Select Graphics Text mode	US
Select Graphics Text mode	ESC FF
Select Incremental Point mode	RS
Select Point mode	FS
Select Vector mode	GS
Select Vector mode	ESC \ GS

VECTOR MODE

Clear Vector mode screen	ESC \ FF
Define user line style x (* = # pixels on ; # pixels off)	ESC / * a
Define user line style y (* = # pixels on ; # pixels off)	ESC / * b
Define user line style z (* = # pixels on ; # pixels off)	ESC / * c
Disable block fill/erase	ESC ETX
Disable dark vector	BEL
Disable dark vector	ESC BEL
Disable rectangle drawing	ESC \ r
Enable block fill/erase	ESC STX
Enable circle plotting (* = ASCII equiv. of coordinates)	ESC \ O *
Enable dark vector	GS
Enable dark vector	ESC GS
Enable rectangle drawing (*c = corner coordinates)	ESC \ R *c *c
Select 3 dot-dash line style	ESC e
Select 3 dot-dash line style	ESC m
Select dot-dash line style	ESC b
Select dot-dash line style	ESC j
Select dotted line style	ESC a
Select dotted line style	ESC i
Select long dot-dash line style	ESC f
Select long dot-dash line style	ESC n
Select long dash line style	ESC d
Select long dash line style	ESC I
Select medium dash line style	ESC g
Select medium dash line style	ESC o
Select short dash line style	ESC c
Select short dash line style	ESC k
Select solid line style	ESC `
Select solid line style	ESC h
Select user defined line style a	ESC x
Select user defined line style b	ESC y
Select user defined line style c	ESC z

W2119 Emulation

ALPHANUMARIC MODE

Select G0 character set for alpha text	ESC SI
Select G1 character set for alpha text	ESC SO

CURSOR MOVEMENT

Move cursor down 1 pixel	ESC \ b
Move cursor down 16 pixels	ESC \ B
Move cursor left 1 pixel	ESC \ d
Move cursor left 16 pixels	ESC \ D
Move cursor right 1 pixel	ESC \ c
Move cursor right 16 pixels	ESC \ C
Move cursor up 1 pixel	ESC \ a
Move cursor up 16 pixels	ESC \ A
Move cursor to beginning of line	ESC \ CR
Move text cursor down	LF
Move text cursor left	BS
Move text cursor right	HT
Move text cursor up	VT
Move text cursor to start of line	CR

GENERAL OPERATION

Clear screen & enter Graphics Text mode	ESC FF
Copy screen data to parallel port	ESC ETB
Deselect write-through & selective erase	ESC SOH
Enable selective erase	ESC DLE
Enable write-through	ESC NAK
Request cursor position report	ESC ENQ
Select bypass mode	ESC CAN
Sound bell	BEL

GRAPHICS TEXT MODE

Select graphics text font (120x58)	ESC :
Select graphics text font (132x38)	ESC <
Select graphics text font (132x64)	ESC ;
Select graphics text font (146x70)	ESC >
Select graphics text font (73x35)	ESC 8
Select graphics text font (80x38)	ESC 9
Select graphics text font (80x64)	ESC =
Select text zoom factor 1	ESC \ e
Select text zoom factor 2	ESC \ f
Select text zoom factor 3	ESC \ g
Select text zoom factor 4	ESC \ h

MODE SELECTION

Select Alpha mode & erase alpha memory	ESC EOT
Select Alphanumeric mode	ACK
Select Alphanumeric mode	CAN
Select Alphanumeric mode	ESC 2
Select Alphanumeric mode	ESC \ CAN
Select GIN mode	ESC SUB
Select Graphics Text mode	CR
Select Graphics Text mode	US
Select Graphics Text mode	ESC 1
Select Graphics Text mode	ESC FF
Select Incremental Point mode	RS
Select Point mode	FS
Select Vector mode	GS

VECTOR MODE

Clear Vector mode screen	ESC \ FF
Disable block fill/erase	ESC ETX
Disable dark vector	BEL
Disable dark vector	ESC BEL
Disable rectangle drawing	ESC \ r
Enable block fill/erase	ESC STX
Enable circle plotting (* = ASCII equiv. of coords)	ESC \ O *
Enable dark vector	GS
Enable dark vector	ESC GS
Enable rectangle drawing (*c = corner coords)	ESC \ R *c *c
Select dot-dash line style	ESC b
Select dot-dash line style	ESC j
Select dot-dash line style	ESC r
Select dotted line style	ESC a
Select dotted line style	ESC i
Select dotted line style	ESC q
Select long dash line style	ESC d
Select long dash line style	ESC I
Select long dash line style	ESC t
Select short dash line style	ESC c
Select short dash line style	ESC k
Select short dash line style	ESC s
Select solid line style	ESC `
Select solid line style	ESC e
Select solid line style	ESC f
Select solid line style	ESC g
Select solid line style	ESC h
Select solid line style	ESC m
Select solid line style	ESC n
Select solid line style	ESC o
Select solid line style	ESC p
Select solid line style	ESC u
Select solid line style	ESC v
Select solid line style	ESC w

Wyse Emulations

ATTRIBUTES

Assign character display attribute	ESC G *attribute
Assign line attribute	ESC G *line attribute
Assign write-protected character attribute	ESC ' *attribute
Line attribute mode on	ESC e 3
Page attribute mode on	ESC e 2
Wyse 60 character attribute mode on	ESC e 1
Wyse 60 character attribute mode off	ESC e 0

CHARACTER SETS (WY-60 only)

Define & load soft character — ESC c A *b *p *...*c CTRL Y

*b = bank 0-3, *p = position in character set (2-byte hex)
*...*c = character bit pattern (32-byte character string)

Automatic font loading enabled	ESC e O
Automatic font loading disabled	ESC e N
Load font bank with predefined character set	ESC c @ *bank *set

Bank: 0 = Bank 0, 1 = Bank 1, 2 = Bank 2, 3 = Bank 3

Set:				
	Native mode	@	Standard ANSI	G
	Multinational	A	44-line Native mode	'
	Standard ASCII	B	44-line Multinational	a
	Graphics 1	C	44-line PC Equivalent	b
	PC Equivalent	D	44-line Standard ASCII	c
	Graphics 2	E	44-line Standard ANSI	d
	Graphics 3	F		

Clear font bank (* = 0 - 3)	ESC c ? *
Specify font bank for primary character set (* = 0 - 3)	ESC c B *
Specify font bank for secondary character set (* = 0 - 3)	ESC c C *
Select primary character set for display	ESC c D
Select secondary character set for display	ESC c E

CLEARING DATA

Clear cursor column	ESC V
Clear entire rectangle in 80 column page	ESC c H *line *col *character
Clear entire rectangle in 132 column page	ESC c H *line ~ *col *character
Clear page to nulls	ESC *
Clear page to spaces	ESC +
Clear page to write-protected spaces	ESC ,
Clear unprotected column to nulls	ESC c K
Clear unprotected column to specified character	ESC c I *character
Clear unprotected line to nulls from cursor	ESC t
Clear unprotected line foreground to nulls from cursor	ESC c S
Clear unprotected line to spaces from cursor	ESC T
Clear unprotected line foreground to spaces from cursor	ESC c R
Clear unprotected page to attribute character	ESC !

Clear unprotected page to nulls	ESC :
Clear unprotected page to nulls from cursor	ESC y
Clear unprotected page foreground to nulls from cursor	ESC c Q
Clear unprotected page to spaces	ESC ; *or* SUB
Clear unprotected page to spaces from cursor	ESC Y
Clear unprotected page foreground to spaces from cursor	ESC c P
Clear unprotected page to specified character	ESC . *character
Clear unprotected rectangle in 80 column page	ESC c F *line *col *character
Clear unprotected rectangle in 132 column page	ESC c F *line ~ *col *character
Clear unprotected to end of line with nulls	ESC c L
Clear unprotected to end of line with spaces	ESC c O
Box rectangle in 80 column page	ESC c G *line *column
Box rectangle in 132 column page	ESC c G *line ~ *column
Box rectangle to right of cursor	ESC c N *width *height

COLOUR SELECTION

Wyse colour (except Wyse 60):

Select colour palette (* = palette ID in range 0 - ?)	ESC % *
Redefine attribute association	ESC m *r *c *m

*r = display attribute group or write protect to be redefined (range 0 - 8)
*c = the colour assigned (range ! - 6)
*m = the monochrome attributes assigned:
0 = Normal, - = Reverse, . = Underline, / = Underline & reverse

Wyse 60 ASCII colour:

Foreground colour palette	ESC A a *
Select border colour	ESC A b *
Background colour	ESC A c *
Assign foreground colour	ESC A d *
Assign background colour	ESC A e *
Assign display attribute to unprotected characters	ESC A f *
Redefine character attribute association	ESC A g *attr *f *b *assoc
Assign foreground/background colours	ESC A h *
Assign fore/background colours to write-prot. chars.	ESC A i *
Assign foreground colours to write-protected chars.	ESC A j *
Assign background colours to write-protected chars.	ESC A k *
Assign display attribute to write-protected characters	ESC A l *
Assign replacement character colours & attributes	ESC A y *f *b *a

CURSOR

Address cursor in current 80/132 column page	ESC a *line R *column C
Address cursor in current 80 column page	ESC = *line *column
Address cursor in specific 80 column page	ESC w @ *page *line *column
Address cursor in specific 80 column window/page	ESC - *win/pg *line *column
Address cursor column	ESC _
Address cursor row	ESC [
Autowrap mode off	ESC d .
Autowrap mode on	ESC d /
Clear all tab stops	ESC 0
Clear tab stop	ESC 2 or ESC 3
Cursor down (scroll)	LF
Cursor left	BS

Cursor right	FF
Cursor up (no scroll)	VT
Cursor to start of current line	CR
Cursor to start of next line	US
Home cursor	ESC { or RS
Line lock mode on	ESC ` H
Line lock mode off	ESC ` I
Move cursor up; scroll	ESC j
Read 80 column window/page & cursor address	ESC /
Read 80 column page number & cursor address	ESC w '
Read cursor address in current 80 column page	ESC ?
Read cursor address in current 80/132 column page	ESC b
Redefine screen as one window & clear pages	ESC x 0
Set cursor display features	ESC ' *cursor
Set tab stop	ESC 1
Tabulate cursor	ESC i
Tab cursor	HT
Tab cursor backward	ESC I
Tabs are not initialized	ESC e :
Tabs are initialized	ESC e ;

DISPLAY

80 column display	ESC ' :
132 column display	ESC ' ;
80/132 change clears screen on	ESC e /
80/132 change clears screen off	ESC e .
Activate lower window	ESC }
Activate other window/display previous page	ESC J
Activate other window/display next page	ESC K
Activate upper window	ESC]
Autopage mode off	ESC d *
Autopage on	ESC d +
Autoscrolling mode on	ESC 0
Autoscrolling mode off	ESC N
Display 24 data lines	ESC e (
Display 25 data lines	ESC e)
Display 42 data lines	ESC e *
Display 43 data lines	ESC e +
Display graphics character	ESC H *key
Display next page	ESC w C
Display previous page	ESC w B
Display specific page	ESC w *page
Divide memory into pages	ESC w *length
Economy 80 column mode on	ESC e G
Economy 80 column mode off	ESC e F
Horizontal split higher	ESC x R
Horizontal split lower	ESC x P
Program label line	ESC f
Protect mode on	ESC &
Protect mode off	ESC '
Redefine screen as one window	ESC x @
Redefine screen as one window & clear pages	ESC x 0

Restore normal display	ESC ^ 0
Reverse display	ESC ^ 1
Roll window up in page	ESC w E
Roll window down in page	ESC w F
Split screen	ESC x C *line
Split screen (two pages only)	ESC x A *line
Split screen & clear pages	ESC x 3 *line
Split screen & clear pages (2 pages only)	ESC x 1 *line
Status line format (a = extended, b = standard)	ESC ' *
Status line message	ESC F *message CR
Status line not displayed	ESC ' c
Turn display off	ESC ' 8
Turn display on	ESC ' 9

EDITING

Begin send at top of page	ESC d '
Begin send at top of screen	ESC d &
Delete cursor character	ESC W
Delete cursor column	ESC c j
Delete cursor line	ESC R
Insert column of nulls	ESC c M
Insert line of spaces	ESC E
Insert mode on, replace mode off	ESC q
Insert mode off, replace mode on	ESC r
Insert space character	ESC Q
Page edit mode on (Wyse 60 mode)	ESC e #
Page edit mode off (Wyse 60 mode)	ESC e "
Write-protect mode off	ESC (
Write-protect mode on	ESC)

GENERAL OPERATION

Answerback mode off	ESC e SP
Answerback mode on	ESC e !
Graphics mode on	ESC H CTRL B
Graphics mode off	ESC H CTRL C
Received CR mode off	ESC e 4
Received CR mode on	ESC e 5
Run script in Wyse and TVI/ADDS/HZ enhanced modes	ESC c s *filename CR
Set MODEM port receive handshake	ESC c 2 *handshake
Set MODEM port transmit handshake	ESC c 4 *handshake

HOST COMMUNICATIONS

Select terminal emulation — ESC ~ *emulation

*emulation:	Wyse 50	@	VT52	6
	Wyse 50+	"	VT100	;
	Wyse 60	4	VT220-7	<
	TVI 910+	#	VT220-8	=
	TVI 912/920	'	VT320-7	B
	TVI 925	$	VT320-8	C
	ADDS VP A2	%	DG200	)
	HZ 1500	&	DG210	:

Enhance mode on	ESC ~ !
Enhance mode off	ESC ~ SP
Monitor mode on (display control codes)	ESC U
Monitor mode off	ESC u *or* ESC X
Block mode on	ESC B
Half-duplex mode on	ESC C ESC D H
Half-duplex block mode on	ESC D H ESC B
Full-duplex mode on	ESC C ESC D F
ACK mode (6 = off, 7 = on)	ESC e *
Set MODEM port operating parameters	ESC c 0 *b *s *p *w
Set maximum data transmission speed	ESC c 6 *max
Send terminal ID	ESC SP
Program answerback message	ESC c ; *answer CTRL Y
Answerback message (= = conceal, < = send)	ESC c *

KEYBOARD FUNCTIONS

Application key mode on	ESC ~ 3
Application key mode off	ESC ~ 2
Keyboard locked	ESC # *or* SI
Keyboard unlocked	ESC " *or* SO
Caps lock on	ESC e &
Caps lock off	ESC e ‘
Clear all programmable keys	ESC c U
Clear key definition	ESC z *key DEL
Key repeat on	ESC e -
Key repeat off	ESC e ,
Margin bell on	ESC e M
Margin bell off	ESC e L
Set margin bell at cursor position	ESC ` J
Program function key definition	ESC z *fkey seq DEL
Program key direction & definition	ESC Z *dir key seq DEL
Read key direction & definition	ESC Z ~ *key
Sound Bell	BEL
Turn local edit mode on, duplex edit mode off	ESC k
Turn local edit mode off, duplex edit mode on	ESC l

LABEL LINE

Clear function key label	ESC z *field CR
Clear shifted label line message	ESC z) CR
Clear unshifted label line message	ESC z (CR
Display shifted label line	ESC z P CR
Program/display function key label	ESC z *field *label CR
Program/display unshifted label line	ESC z (*text CR
Program shifted label line	ESC z) *text CR
Shifted label line off	ESC z DEL

SENDING DATA

Begin print/send at top of page	ESC d '
Begin print/send at top of screen	ESC d &
Bidirectional mode off	ESC d $

Bidirectional mode on	ESC d %
Enable transmission (XON/XOFF handshaking)	DC1 (XON)
Stop transmission (XON/XOFF handshaking)	DC3 (XOFF)
Mark block beginning	ESC 8
Mark block end	ESC 9
Print entire formatted page	ESC P
Print formatted unprotected page	ESC @
Print unformatted page	ESC p *or* ESC L
Secondary receive mode off	ESC d SP
Secondary receive mode on	ESC d !
Send ACK	ENQ
Send cursor character	ESC M
Send entire block	ESC s
Send entire cursor line	ESC 6
Send entire page	ESC 7
Send unprotected characters in block	ESC S
Send unprotected cursor line	ESC 4
Send unprotected page	ESC 5
Send next incoming character to printer port	DLE
Transparent print mode on	ESC d # *or* CAN
Turn auxiliary print mode on	DC2
Turn print modes off	DC4

Wyse PC Term Emulation

MONITOR MODE

Monitor mode on	ESC U
Monitor mode off	ESC u *or* ESC X

SELECTING PERSONALITIES

Enhance mode off	ESC v SPACE
Enhance mode on	ESC v !
Select WYSE 50+ personality	ESC v "
Select WYSE 50 personality	ESC v @
Select WYSE ASCII personality	ESC v 4
Select TeleVideo 905 personality	ESC v b
Select TeleVideo 910+ personality	ESC v #
Select TeleVideo 925 personality	ESC v $
Select TeleVideo 950 personality	ESC v (
Select ADDS A2 personality	ESC v %
Select PC Term personality	ESC v 5
Select HZ 1500 personality	ESC v &
Select VT52 personality	ESC v 6
Select VT100 personality	ESC v ;
Select VT220 7-bit personality	ESC v <
Select VT220 8-bit personality	ESC v =
Select UNIX Console personality	ESC v]

HOST COMMUNICATIONS

Enable transmission	CTRL Q
Stop transmission	CTRL S
Send ACK	CTRL E
Full-duplex mode on	ESC C ESC }
Half-duplex mode on	ESC C ESC {
Block mode on	ESC B ESC }
Half-duplex block mode on	ESC B ESC {
Send terminal ID	ESC M
Assign COMM 1 port as host port (enhanced)	ESC e 8
Assign COMM 2 port as host port (enhanced)	ESC e 9
Send time of day	ESC SPACE 2
Enable DTR COMM 1 port handshaking	CTRL N
Enable XON-XOFF/XPC COMM 1 port handshaking	CTRL O

TERMINAL & KEYBOARD CONTROL

Local edit mode on	ESC k
Duplex edit mode on	ESC l
Sound bell	CTRL G
Unlock keyboard	ESC "
Lock keyboard	ESC #
Keyclick off	ESC <
Keyclick on	ESC >
CAPS LOCK on (enhanced)	ESC e &

CAPS LOCK off (enhanced)	ESC e '
Margin bell off	ESC n
Margin bell on	ESC o
Select standard ASCII key code mode	ESC e H
Select PC scan code mode	ESC e I
Key repeat off (enhanced)	ESC e ,
Key repeat on (enhanced)	ESC e -
Application key mode off	ESC v 2
Application key mode on	ESC v 3
Read keyboard status	ESC [
Default unit	ESC m

PROGRAMMING KEYS

Program function key definition	ESC z *key *seq DEL
Clear function key definition	ESC z *key DEL
Program key direction & definition	ESC ¦ *p1 *p2 *seq CTRL Y
Read key direction & definition	ESC Z ~ *key
Clear key direction & definition	ESC z *dir *key DEL
Clear all programmable keys	ESC c U

DISPLAY

Screen display off	ESC O
Screen display on	ESC N
Reverse screen (light background)	ESC b
Restore normal screen (dark background)	ESC d
Set cursor display features	ESC . *cursor*
Display 25 data lines	ESC ^
Display 43 data lines	ESC _
Display next page	ESC K
Display previous page	ESC J
Load user line	ESC f
Display user line	ESC g
User line display off	ESC e *or* ESC h
Clear unshifted label line	ESC z (CR
Program & display function key label	ESC z *field *label CR
Clear function key label	ESC z *field CR
Assign display attribute to a message field	ESC \ *mf *attr
Clear unprotected page to display attribute	ESC ! *attr
Assign line attribute	ESC G *attr

PROTECTING DATA

Write-protect mode off	ESC (
Write-protect mode on	ESC)
Clear cursor column to write-protected spaces	ESC V
Protect mode off	ESC '
Protect mode on	ESC &

GRAPHICS CHARACTERS

Line-drawing graphics mode on	ESC $
Line-drawing graphics mode off	ESC %

CURSOR CONTROL

Cursor left (backspace)	CTRL H
Cursor right	CTRL L
Cursor up - no scroll	CTRL K
Cursor up - scroll	ESC j
Cursor down - no scroll	CTRL V
Cursor down - scroll	CTRL J
Cursor to start of line	CTRL M
Cursor to start of next line	CTRL _
Move cursor to home position on current page	ESC { *or* CTRL ^
Move cursor to specific line	ESC [*line
End-of-line wrap mode off	ESC 0
End-of-line wrap mode on	ESC ~
Received CR mode = CR	ESC 9
Received CR mode = CRLF	ESC 8
Autopage mode off	ESC w
Autopage mode on	ESC v
Address cursor in 80-column current page	ESC = *line *col
Address cursor in specific 80-column page	ESC - *page *line *col
Address cursor in specific 80-column window	ESC - *wind/page *line *col
Read cursor address in 80-column current page	ESC ?
Read 80-column window/page number & cursor address	ESC /

EDITING

Clear all tab stops	ESC 3
Set tab stop	ESC 1
Clear tab stop	ESC 2
Tabulate cursor	ESC i *or* CTRL I
Backtab	ESC I
Insert mode on, replace mode off	ESC Z
Insert mode off, replace mode on	ESC r
Insert space character	ESC Q
Insert line of spaces	ESC E
Delete cursor character	ESC W
Delete cursor line	ESC R

CLEARING DATA

Clear page to nulls	ESC *
Clear page to spaces	ESC +
Clear page to write-protected spaces	ESC ,
Clear unprotected page to spaces	ESC ; *or* ESC +
Clear unprotected page to nulls	ESC :
Clear unprotected page to display attribute	ESC ! *attr
Clear unprotected page to spaces from cursor	ESC Y
Clear unprotected page to nulls from cursor	ESC y
Clear unprotected line to spaces from cursor	ESC T
Clear unprotected line to nulls from cursor	ESC t
Fill page with Hs	ESC F

SENDING DATA

Send line through cursor	ESC 6
Send unprotected line through cursor	ESC 4
Send page through cursor	ESC 7
Send unprotected page through cursor	ESC 5
Mark block beginning	ESC CTRL B
Mark block end	ESC CTRL C
Send entire block	ESC s
Send unprotected characters in block	ESC S
Report terminal status	ESC [
Report attribute under cursor	ESC D

PRINT FUNCTIONS

Print formatted page through cursor	ESC P
Print formatted unprotected page through cursor	ESC P
Print unformatted page through cursor	ESC p *or* ESC L
Auxiliary print mode off	ESC A
Auxiliary print mode on	ESC @
Transparent print mode off	ESC a
Transparent print mode on	ESC `
Bidirectional mode off	CTRL T
Bidirectional mode on	CTRL R
Pass next incoming character to printer (enhanced)	CTRL P *schar
Set print terminator	ESC P *t1 *t2
Define delimiters	ESC x *d1 *d2

CHARACTER SETS

Select PC character set	ESC SPACE m *charset
Set national mode	ESC SPACE U
Set multinational mode	ESC SPACE T

325 COLOUR PALETTE MODE

Select colour map values	ESC SPACE $ *fc325 *bc *map
Select a predefined colour palette	ESC SPACE # *pal325
Map blank attribute	ESC SPACE %
Map reverse attribute	ESC SPACE &
Select border colour	ESC SPACE ` *col325
Disable intensity attribute	ESC SPACE "
Enable intensity attribute	ESC SPACE !

370/350 COLOUR MAP MODE

Select foreground colour palette	ESC SPACE a *pal370
Select background colour	ESC SPACE c *col370
Redefine character attribute association	ESC SPACE g *a *fc *bc *assoc

370/350 COLOUR DIRECT MODE

Assign foreground colour	ESC SPACE d *fc370
Assign background colour	ESC SPACE e *bc370
Assign display attribute to unprotected characters	ESC SPACE f *assattr

Assign foreground/background colours	ESC SPACE h *CGAcol
Assign fore/background colours to write-protected chars	ESC SPACE i *CGAcol
Assign foreground colours to write-protected chars	ESC SPACE j *fc370
Assign background colours to write-protected chars	ESC SPACE k *fc370
Assign display attribute to write-protected characters	ESC SPACE 1 *assattr

370/350 COLOUR MISCELLANEOUS

Select border colour	ESC SPACE b *c370
Assign replacement character colours & attributes	ESC SPACE y *fc370 *bc *attr

TeemTalk Additional Commands

DISPLAY COLOUR

DEC VT modes:	Foreground colour	CSI = * F
	Background colour	CSI = * G

where * is one of the following numbers:

0	Black	4	Red	8	Grey	12	Light Red
1	Blue	5	Magenta	9	Light Blue	13	Light Magenta
2	Green	6	Brown	10	Light Green	14	Yellow
3	Cyan	7	White	11	Light Cyan	15	Light White

Reflection 4 compatibility:	Foreground colour	ESC [3 x m
	Foreground+bold colour	ESC [4 x m
	Background colour	ESC [5 x m
	Background+bold colour	ESC [6 x m

DISPLAY SIZE

Number of screen lines — ESC [= 1 ; * p

MOUSE CURSOR

Mouse cursor style (see table below for * value) — ESC [= 2 ; * p

0	Default Cursor (I-Beam)
1	
2	
3	

4	
5	
6	
7	

8	
9	
10	No Cursor

MOUSE REPORTING IN ALPHANUMERIC APPLICATIONS

Arm the emulator for mouse operation — ESC [= *arg g

where *arg is of the format:	bit 7 - 3	0 0 1 1 0
	bit 2	when set enables motion events
	bit 1	when set enables button release events
	bit 0	when set enables button press events

If all bits are cleared then any outstanding arming request is cancelled. The mouse remains armed until cancelled. When any of the selected events occur, the following report format is sent to the host:

ESC [= <Event Type> ; <Button Status> ; <Column> ; <Row> r

Where: <Event Type> is the event(s) that caused the report in the same format as the arming sequence.

<Button Status> is of the format: 00110LMR
where LMR indicates which button caused the event.

<Column> and <Row> are the alphanumeric position of the mouse.

REPORTS

Report application name & version (in DEC VT modes)	ESC [0 ; 1234 c

Notes

Index

A

B

C

D

F

G

H

I

K

L

M

O

P

Q

R

S

T

U

V

W

Notes