

wIntegrate

Emulation Commands Reference

Version 6.0.0
December 2004

IBM Corporation
555 Bailey Avenue
San Jose, CA 95141

Licensed Materials – Property of IBM

© Copyright International Business Machines Corporation 2004. All rights reserved.

AIX, DB2, DB2 Universal Database, Distributed Relational Database Architecture, NUMA-Q, OS/2, OS/390, and OS/400, IBM Informix®, C-ISAM®, Foundation.2000™, IBM Informix® 4GL, IBM Informix® DataBlade® module, Client SDK™, Cloudscape™, Cloudsync™, IBM Informix® Connect, IBM Informix® Driver for JDBC, Dynamic Connect™, IBM Informix® Dynamic Scalable Architecture™ (DSA), IBM Informix® Dynamic Server™, IBM Informix® Enterprise Gateway Manager (Enterprise Gateway Manager), IBM Informix® Extended Parallel Server™, i.Financial Services™, J/Foundation™, MaxConnect™, Object Translator™, Red Brick® Decision Server™, IBM Informix® SE, IBM Informix® SQL, InformiXML™, RedBack®, SystemBuilder™, U2™, UniData®, UniVerse®, wIntegrate® are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

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

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

Documentation Team: Alan Buckley, Tim Rasmussen, David Robertshaw

US GOVERNMENT USERS RESTRICTED RIGHTS

Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

1. Introduction	8
2. Overview	10
3. Emulation file syntax	12
Backslash format strings.....	14
Host sequence parsing commands.....	15
4. Emulation Commands	16
AddTaggedAttr.....	17
AlarmMessage.....	18
AlarmOff.....	19
AlarmOn.....	20
AlphaNumericBlue.....	21
AlphaNumericCyan.....	22
AlphaNumericGreen.....	23
AlphaNumericPurple.....	24
AlphaNumericRed.....	25
AlphaNumericWhite.....	26
AlphaNumericYellow.....	27
AssignFont.....	28
AttributeEffect.....	29
AttributeMode.....	30
AttributeMode Char.....	32
AttributeMode Line.....	33
AttributeMode Page.....	34
AttributeMode Wrap.....	35
AttributeOff.....	36
AttributeOn.....	37
AutoFF.....	38
AutoScroll.....	39
AutoScrollOff.....	40
AutoScrollOn.....	41
AutoWrap.....	42
AutoWrapOff.....	43

AutoWrapOn	44
BackSpace	45
BackTab	46
Bell	47
BlackBackground	48
CR	49
ChangeColorOff	50
ChangeColorOn	51
CharacterMapTable	52
ChgTaggedAttr	53
ClearAOL	54
ClearAllUnprotected	55
ClearEOF	56
ClearEOL	57
ClearEOP	58
ClearSOL	59
ClearSOP	60
ClearScreen	61
Color Back	62
Color Fore	63
CursorCode	64
CursorDown	66
CursorHome	67
CursorLeft	68
CursorOff	69
CursorOn	70
CursorRight	71
CursorUp	72
DelayNewLine	73
DeleteChar	74
DeleteLine	75
DisableProtect	76
DisplayChar	77
DisplayMap	79
DisplayNulls	80
DisplayOff	81
DisplayOn	82
DoubleHeight	83
DoubleHeightLineBottom	84

DoubleHeightLineTop	85
DoubleSizeLineBottom	86
DoubleSizeLineTop.....	87
DoubleWidthLine.....	88
Effect.....	89
Effect ANSI	90
Effect All.....	91
Effect CharacterRange	92
Effect Mask	93
Effect Order.....	94
EffectExtent.....	95
EffectIgnoreBold	96
EffectIgnoreReverse	97
EmbeddedAttributes	98
EmulationMode	99
EnableProtect	101
EraseCharsInLine	102
EraseScreen	103
EraseUsingWriteAttr	104
FormFeed	105
ForwardSpace.....	106
GraphicMapTable	107
GraphicModeOff.....	108
GraphicModeOn.....	109
HardReset.....	110
Ignore.....	111
IgnoreErrors	112
InsertChar	113
InsertLine	114
InsertModeOff	115
InsertModeOn	116
Key_.....	117
KeyboardLock.....	119
KeyboardUnlock.....	120
Library	121
LineFeed.....	122
LockShiftGLFont	123
LockShiftGRFont.....	124
MapSequence.....	125

MapTable	126
MarginBell	127
MarginBellOff	128
MarginBellOn	129
MarginBellPsn	130
MosaicBlue	131
MosaicContiguous	132
MosaicCyan	133
MosaicGreen	134
MosaicHold	135
MosaicPurple	136
MosaicRed	137
MosaicRelease	138
MosaicSeparated	139
MosaicWhite	140
MosaicYellow	141
MoveX	142
MoveXY	143
MoveY	144
NewBackground	145
NewLine	146
NormalHeight	147
NormalVideo	148
PrinterIgnore	149
PrinterMap	150
ProtectEffect	151
ProtectFullIntensity	152
ProtectHalfIntensity	153
ProtectOff	154
ProtectOn	155
ProtectTab	156
RestoreCursor	157
Reveal	158
ReverseLF	159
ReverseNewLine	160
ReverseVideo	161
SaveCursor	162
ScrollDown	163
ScrollUp	164

SetMarginBell.....	165
SingleLine	166
SingleShift.....	167
SoftReset	168
StartCommand.....	169
Tab.....	170
TabClear	171
TabClearAll	172
TabSet	173
TaggedAttr	174
TaggedAttrOff	175
TaggedAttrOn	176
TekGraphic	177
TelnetTerminal.....	178
TermId.....	179
TerminalCommand	180
Xoff.....	182
Xon.....	183
Index	184

Introduction

Terminal emulation can be confusing. There is no need to read or understand this manual to use wIntegrate. All you need to know is which emulation to choose in Setup Terminal so that your PC matches the host.

In the normal course of events, you will never need to change the terminal emulation files as supplied with wIntegrate. However this manual is provided in case you need to create your own terminal emulation files, or modify the functionality of emulations to suit specific needs.

Much of the information sent to a terminal by a host computer is not the text you see on the screen, but the instructions as to where the text should be positioned, and text attributes such as dim and bright.

There is no single standard for these instructions. They differ between terminal manufacturers and between specific models from the same manufacturer according to the features available on each terminal. The instructions usually include unprintable characters such as "Escape" (ASCII character 27)

To ensure maximum compatibility, most terminals are supplied with several emulations, so they can work with instructions designed for other terminals. The host computer must be told which terminal emulation is required for each port, so that it can send instructions in the appropriate format.

In practice it doesn't matter which emulation you choose on your PC, provided it matches the setting on the host computer. If the PC and host do not match, text will not be displayed properly, and may even appear as a single unformatted block. At worst, the COM port will hang, and must be reset with Run Restart Port or by exiting and restarting wIntegrate.

wIntegrate stores terminal emulations as text files with the WIT extension in the WINTSYS\WIT directory. This means that capable users can create and modify their own terminal emulation files if required.

You will probably find it quicker to modify a wIntegrate WIT file to match your host,

rather than the other way round. To do so, you should first call up the Setup Character box and turn on Decimal Display for Invalid Characters. Then it's a matter of analysing the character sequences as they are displayed on the screen, and setting them to the appropriate keywords.

The ControlCodes option in Edit Record is invaluable for analysing terminal emulation files

Terminal definitions are applied to a session only when a session is first created or when changing the terminal emulation in Setup Terminal.

2

Overview

This section gives a brief overview of the most common elements of an emulation.

Terminal attributes

Terminal attributes control the way characters are displayed on the screen, for example Dim, Reverse and Underline

There are different ways of handling terminal attributes, depending on the terminal type.

The AttributeMode specifies how many characters an attribute sequence will effect and if the attribute character itself takes up a space on the screen.

The EmulationMode will set the AttributeMode for the type of terminal that is specifies.

Another technique for controlling screen display is the tagged attribute on/off method. This marks certain cells as using a non-specific terminal attribute. (This is a simple On or Off flag for each character cell.) The actual terminal attribute to be applied is set separately, and is applied to all cells with the flag set. When this method is used, it is possible to display only one terminal attribute on screen at one time.

TaggedAttrOn sets the flag for the character positions in subsequent PRINT or CRT statements. TaggedAttrOff turns the flag off. The terminal display for all character positions with the flag set is subsequently changed by use of TaggerAttr Dim, TaggedAttr Reverse, or TaggedAttr Normal, etc.

Some terminals combine the tagged attribute on/off method with the embedded attributes method.

Cursor addressing

Cursor addressing defines the way the parameters in the MoveXY, MoveX and MoveY functions are processed. This positions the cursor at a specific character "cell" on the

screen.

The `EmulationMode` command automatically sets the correct cursor addressing code for a specific type of emulation.

The `CursorCode` command can be used to change the cursor addressing for the following commands in the emulation file.

The cursor addressing code itself can be appended as an optional argument to an individual `Move` command.

Character graphics

Certain terminals allow individual characters to be displayed as either the "real" character or a graphics character such as a vertical line. To switch between these two character sets, the terminal reacts to the `GraphicModeOn` and `GraphicModeOff` control sequences. To emulate these terminals, `wIntegrate` stores the "real" and graphics characters in the keyword `GraphicMapTable`.

Other terminals use a sequence of control characters to define each graphics character. `wIntegrate` stores these as `DisplayMap` commands

Some terminals, such as the `Wyse50`, use both.

The mapping between a character on the host and the character used for display is defined in a mapping table.

A mapping table is defined using the `MapTable` command. Each emulation can have multiple numbered map tables. Map table 0 is used for mapping the standard characters and map table 1 is used to provide the mapping for the Graphics characters set.

Emulation file syntax

The files in the WIT and WIT_EXT directory are plain text formatted files.

The first line of the file must always consist of the text "[Terminal]".

Each line in these files is either a command, comment or empty line. Comment line start with an asterisk symbol. Comments can also be added to the end of command lines by using a semicolon followed by an asterisk but not all commands support end of line comments.

Tabulation stops or spaces may be used to align the commands on different lines as these characters are ignored unless they appear between quotes.

Any pair of quotes, single, back or double may be used to surround the “host sequence” but it is also possible to double up the quote character e.g. "e" is the same as `e` or "e".

The "host sequences" and other strings in the command lines are in the Backslash Format (see the backslash table below).

Many host sequences also contain string or numeric information and to allow these encoded parameters to be identified the "host sequence" may contain special instructions that consist of a percent followed by a character (see the host sequences parsing table below).

There are several commands that require a parameter such as MoveX, other commands may have an optional parameter such as CursorLeft and LineFeed while others don't require a parameter such as KeyboardLock.

These parameters once extracted from the host sequence are ignored if not required by the command. E.g. KeyboardUnlock = "e%fu" means that the host needs to send a three character sequence to unlock the keyboard and it does not matter what the second character is. The “TerminalCommand” also uses a percent sequence in the second argument, script command, but those should not

to be confused with “host sequence” percent parameters.

The order of the host sequences in the WIT file(s) is important. If a WIT contained the lines :-

```
CursorLeft = "\e[%dL";* Move the cursor left
```

```
KeyboardLock = "\e[L";* Disable keyboard input
```

The host could never lock the keyboard because every time the host sent the sequence “\e[L” to lock the keyboard the cursor would move one position to the left. If the lines were reversed then the host could lock the keyboard but would need to send either “\e[OL” or “\e1L” to move the cursor one space to the left.

A keyword can be defined more than once. This permits several control sequences to have the same effect. It also simplifies the definition of graphic character translation tables.

For example, suppose your host application has some programs which use FF (formfeed) to clear the screen, and others use the CursorHome and ClearEOP (clear to end of page). When the former method is used, data on the screen will be moved to the back pages in memory. However the second method will result in the screen being lost.

With wIntegrate you can define both of these sequences as FF, so that both result in the same action.

Normally syntax errors in a emulation file are ignored and just result in the invalid line being ignored. For stricter error checking turn off the IgnoreErrors flag by executing "Set IgnoreErrors = 0" from a script or using the "example\script\wc.wis" example script.

Backslash format strings

The following table describes the backslash format used for all the strings within the emulation file.

Sequence	Description
\z	Ascii character 0 (NULL)
\f	Ascii character 12 (Formfeed)
\t	Ascii character 9 (Tab)
\n	Ascii character 10 (Linefeed)
\r	Ascii character 13 (Carriage Return)
\e	Ascii character 27 (Escape)
\ddd	Ascii character ddd where ddd is three decimal digits in the range 000 to 255
\xhh	Ascii character hh where hh is two hexadecimal digits in the range 00 to FF
\\	A single back slash character
\cdddd	IBSFont char(dddd) where dddd is four digits in the range 0000 to 1023. This is only valid in a few commands. E.g MapTable
\m	This is only valid for key definitions and is used to indicate the key data is not to be transmitted to the host but executed as a script.

Host sequence parsing commands

The following table describes the percentage commands to parse out variable sections of the received escape sequence.

Command	Description
%f	Single character parameter
%v	Variable length string parameter. Must be less than 250 characters.
%d	Variable length decimal parameter
%a	Variable number of variable length decimal parameters separated by semi-colons

Emulation Commands

This chapter details each of the emulation commands provided with wIntegrate. You can use these commands to modify or create a new emulation.

AddTaggedAttr

Syntax

AddTaggedAttr *effect_list* = *seq*

Description

This sequence adds effects to the tagged character attribute effect changing the way it is displayed.

Parameters

The following table describes the parameters of the AddTaggedAttr command:

Parameter	Description
<i>effect_list</i>	The effects to add. Can be either "Normal" or one or more of "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret"
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Example

Tagged attribute display sequences from the wyse50 emulation

```
ChgTaggedAttr Normal = "\e`A"  
AddTaggedAttr Flash = "\e`B"  
AddTaggedAttr Secret = "\e`C"  
AddTaggedAttr Underline = "\e`E"  
AddTaggedAttr Reverse = "\e`F"  
AddTaggedAttr Dim = "\e`G"
```

Related Emulation Commands

[ChgTaggedAttr](#)

AlarmMessage

Syntax

AlarmMessage [=] *seq*

Description

This sequence sets the alarm message that will appear in the status line every second when the alarm is on.

Parameters

The following table describes the parameters of the AlarmMessage command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This must contain one string argument that is used for the alarm message.

Related Emulation Commands

[AlarmOn](#), [AlarmOff](#)

AlarmOff

Syntax

AlarmOff [=], *seq*

Description

This sequence turns of the alarm started by the AlarmOn command.

Parameters

The following table describes the parameters of the AlarmOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlarmMessage](#), [AlarmOn](#)

AlarmOn

Syntax

AlarmOn [=] *seq*

Description

This sequence turns on the alarm that will sound a bell every second and show the alarm message if one is defined.

Parameters

The following table describes the parameters of the AlarmOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlarmMessage](#), [AlarmOff](#)

AlphaNumericBlue

Syntax

AlphaNumericBlue [=] *seq*

Description

This sequence causes the following characters to be displayed as blue numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericBlue command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericCyan](#), [AlphaNumericGreen](#), [AlphaNumericPurple](#), [AlphaNumericRed](#), [AlphaNumericWhite](#), [AlphaNumericYellow](#)

AlphaNumericCyan

Syntax

AlphaNumericCyan [=] *seq*

Description

This sequence causes the following characters to be displayed as cyan numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericCyan command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericGreen](#), [AlphaNumericPurple](#), [AlphaNumericRed](#), [AlphaNumericWhite](#), [AlphaNumericYellow](#)

AlphaNumericGreen

Syntax

AlphaNumericGreen [=] *seq*

Description

This sequence causes the following characters to be displayed as green numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericGreen command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericCyan](#), [AlphaNumericPurple](#), [AlphaNumericRed](#), [AlphaNumericWhite](#), [AlphaNumericYellow](#)

AlphaNumericPurple

Syntax

AlphaNumericPurple [=] *seq*

Description

This sequence causes the following characters to be displayed as purple numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericPurple command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericCyan](#), [AlphaNumericGreen](#), [AlphaNumericRed](#), [AlphaNumericWhite](#), [AlphaNumericYellow](#)

AlphaNumericRed

Syntax

AlphaNumericRed [=] *seq*

Description

This sequence causes the following characters to be displayed as red numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericRed command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericCyan](#), [AlphaNumericGreen](#), [AlphaNumericPurple](#), [AlphaNumericWhite](#), [AlphaNumericYellow](#)

AlphaNumericWhite

Syntax

AlphaNumericWhite [=] *seq*

Description

This sequence causes the following characters to be displayed as white numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericWhite command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericCyan](#), [AlphaNumericGreen](#), [AlphaNumericPurple](#), [AlphaNumericRed](#), [AlphaNumericYellow](#)

AlphaNumericYellow

Syntax

AlphaNumericYellow [=] *seq*

Description

This sequence causes the following characters to be displayed as yellow numbers and letters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the AlphaNumericYellow command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AlphaNumericBlue](#), [AlphaNumericCyan](#), [AlphaNumericGreen](#), [AlphaNumericPurple](#), [AlphaNumericRed](#), [AlphaNumericWhite](#)

AssignFont

Syntax

AssignFont [=] *seq*, *bank*, *map_table*

Description

This sequence assigns the character map table to use for an emulation font bank.

Parameters

The following table describes the parameters of the AssignFont command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>bank</i>	The emulation character bank (0-3)
<i>map_table</i>	The character map table for the font bank (0-31).

Related Emulation Commands

[MapTable](#)

AttributeEffect

Syntax

AttributeEffect *effect_list* = *seq*

Description

This sequence changes the display of the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use ChgTaggedAttr instead.

Parameters

The following table describes the parameters of the AttributeEffect command:

Parameter	Description
<i>effect_list</i>	The effects to change. Can be either "Normal" or one or more of "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret"
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ChgTaggedAttr](#)

AttributeMode

Syntax

AttributeMode = *state*

Description

Sets how attributes are placed on the screen.

This settings is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the AttributeMode command:

Parameter	Description
=	Equals symbol
<i>state</i>	The start up attribute state. See table below.

Values for state

Value	Description
<i>Char</i>	Each character can have a different attribute
<i>Line</i>	Attributes continue until the next attribute or the end of the line
<i>Page</i>	Attributes continue until the next attribute or the end of the page
<i>Wrap</i>	Attributes continue until the next attribute, wrapping at the end of the page
<i>Embedded</i>	Add this word with Line, Page or Wrap if the attribute takes up a space on the screen. This is the default if not specified.
<i>NonEmbedded</i>	Add this word with Line, Page or Wrap if the attribute does not take up a space.

Related Emulation Commands

[EmulationMode](#)

AttributeMode Char

Syntax

AttributeMode Char [=] *seq*

Description

This command specifies the sequence to change the emulations attributing to character attributing. i.e Each character can have a different attribute.

Parameters

The following table describes the parameters of the AttributeMode Char command:

Parameter	Description
=	Equals symbol
<i>seq</i>	Escape sequence received from the host

Example

wyse60 sequence to change to character attributing

```
AttributeMode Char = "\eel"
```

Related Emulation Commands

[AttributeMode](#), [AttributeMode Line](#), [AttributeMode Page](#), [AttributeMode Wrap](#)

AttributeMode Line

Syntax

AttributeMode Line [*embed*] = *seq*

Description

This sequence changes the emulation to line attributing where an attribute will continue until the next attribute or the end of the line.

Parameters

The following table describes the parameters of the AttributeMode Line command:

Parameter	Description
<i>embed</i>	Use the word "Embedded" if an attribute uses a space on the screen. Use the word "NonEmbedded" if an attribute doesn't use a space on the screen. The default is "Embedded"
=	Equals symbol
<i>seq</i>	The control sequence that invokes this command.

Related Emulation Commands

[AttributeMode](#), [AttributeMode Char](#), [AttributeMode Page](#), [AttributeMode Wrap](#)

AttributeMode Page

Syntax

AttributeMode Page [*embed*] = *seq*

Description

This sequence changes the emulation to page attributing where an attribute will continue until the next attribute or the end of the page.

Parameters

The following table describes the parameters of the AttributeMode Page command:

Parameter	Description
<i>embed</i>	Use the word "Embedded" if an attribute uses a space on the screen. Use the word "NonEmbedded" if an attribute doesn't use a space on the screen. The default is "Embedded"
=	Equals symbol
<i>seq</i>	The control sequence that invokes this command.

Related Emulation Commands

[AttributeMode](#), [AttributeMode Char](#), [AttributeMode Line](#), [AttributeMode Wrap](#)

AttributeMode Wrap

Syntax

AttributeMode Wrap [*embed*] = *seq*

Description

This sequence changes the emulation to wrap attributing where an attribute will continue until the next attribute, wrapping around at the end of the page.

Parameters

The following table describes the parameters of the AttributeMode Wrap command:

Parameter	Description
<i>embed</i>	Use the word "Embedded" if an attribute uses a space on the screen. Use the word "NonEmbedded" if an attribute doesn't use a space on the screen. The default is "Embedded"
=	Equals symbol
<i>seq</i>	The control sequence that invokes this command.

Related Emulation Commands

[AttributeMode](#), [AttributeMode Char](#), [AttributeMode Line](#), [AttributeMode Page](#)

AttributeOff

Syntax

AttributeOff [=] *seq*

Description

This sequence turns off the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use TaggedAttrOff instead.

Parameters

The following table describes the parameters of the AttributeOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOff](#)

AttributeOn

Syntax

AttributeOn [=] *seq*

Description

This sequence turns on the tagged attribute. Characters written to the screen after this sequence will include the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use TaggedAttrOn instead.

Parameters

The following table describes the parameters of the AttributeOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOn](#)

AutoFF

Syntax

AutoFF [=] *flag*

Description

This settings causes a home command followed immediately by a clear to end of page command to act as if a formfeed had been received.

This allows the current screen to be saved in back page memory when the host doesn't use a clear screen or form feed sequence.

Parameters

The following table describes the parameters of the AutoFF command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[CursorHome](#), [ClearEOP](#), [FormFeed](#), [ClearScreen](#), [EraseScreen](#), [ClearAllUnprotected](#)

AutoScroll

Syntax

AutoScroll [=] *flag*

Description

Allows screen to scroll up when the cursor attempts to move down when at the bottom of the screen.

This setting is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the AutoScroll command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[EmulationMode](#), [AutoScrollOn](#), [AutoScrollOff](#)

AutoScrollOff

Syntax

AutoScrollOff [=] *seq*

Description

This sequence disables automatic scrolling. This stops the screen from scrolling when a line feed occurs at the bottom of the screen or a reverse line feed occurs at the top of the screen or the ScrollUp or ScrollDown sequences are used.

Parameters

The following table describes the parameters of the AutoScrollOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AutoScrollOn](#), [AutoScroll](#)

AutoScrollOn

Syntax

AutoScrollOn [=] *seq*

Description

This sequence enables automatic scrolling when a line feed occurs at the bottom of the screen or a reverse line feed occurs at the top of the screen or the ScrollUp or ScrollDown sequences are used.

Parameters

The following table describes the parameters of the AutoScrollOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AutoScrollOff](#), [AutoScroll](#)

AutoWrap

Syntax

AutoWrap [=] *flag*

Description

This setting allows the cursor to go to the start of the next row when the last character in a line is written.

This setting is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the AutoWrap command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[EmulationMode](#), [AutoWrapOn](#), [AutoWrapOff](#)

AutoWrapOff

Syntax

AutoWrapOff [=] *seq*

Description

This sequence disables auto wrap, so that a character displayed in the last column of the screen does not automatically move the cursor to the next line.

Parameters

The following table describes the parameters of the AutoWrapOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AutoWrapOn](#), [AutoWrap](#)

AutoWrapOn

Syntax

AutoWrapOn [=] *seq*

Description

This sequence enables auto wrap, so that a character displayed in the last column of the screen will automatically move the cursor to the next line.

Parameters

The following table describes the parameters of the AutoWrapOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AutoWrapOff](#), [AutoWrap](#)

BackSpace

Syntax

BackSpace [=] *seq*

Synonym

BS

Description

Moves the cursor back, positioning the cursor at the last column of the previous line if the cursor is at the beginning of the line.

Parameters

The following table describes the parameters of the BackSpace command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[CursorLeft](#), [ForwardSpace](#)

BackTab

Syntax

BackTab [=] *seq*, [*emul*]

Description

This sequence moves the cursor to a previous tab stop.

For the TVI range of emulations and emulation that specify TVI in the *emul* parameter this command will move to the beginning of the previous unprotected area of the screen.

Parameters

The following table describes the parameters of the BackTab command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If this sequence contains a numeric argument the number specifies the number of tab stops to go back.
<i>emul</i>	For non televideo emulation, this can be set to "TVI" to use the TVI version of the back tab command.

Related Emulation Commands

[Tab](#), [TabClear](#), [TabClearAll](#), [TabSet](#)

Bell

Syntax

Bell [=] *seq*

Description

This sequence sounds the bell.

Parameters

The following table describes the parameters of the Bell command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

BlackBackground

Syntax

BlackBackground [=] *seq*

Description

This sequence sets the background color to black until the end of the line or a NewBackground sequence.

This command applies to the viewdata emulation only.

Parameters

The following table describes the parameters of the BlackBackground command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[NewBackground](#)

CR

Syntax

CR [=] *seq*

Description

This sequence moves the cursor to the beginning of the line

Parameters

The following table describes the parameters of the CR command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

ChangeColorOff

Syntax

ChangeColorOff [=] *seq*

Description

This sequence modifies the behavior of the emulation so the current write attributes color component is unaffected by the monochrome attribute off sequence.

Parameters

The following table describes the parameters of the ChangeColorOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ChangeColorOn](#)

ChangeColorOn

Syntax

ChangeColorOn [=] *seq*

Description

This sequence modifies the behavior of the emulation so the current write attributes color component is overwritten with the default colors when the monochrome attribute off sequence is received.

Parameters

The following table describes the parameters of the ChangeColorOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ChangeColorOff](#)

CharacterMapTable

Syntax

CharacterMapTable *host_char*, *local_char*, [...]

Description

This sequence is kept for backwards compatability only. It is identical to using the MapTable command for table 0.

Parameters

The following table describes the parameters of the CharacterMapTable command:

Parameter	Description
<i>host_char</i>	The character received from the host (or sent to the host for characters 32 to 255).
<i>local_char</i>	The character displayed locally on the PC
...	Additional <i>host_char</i> , <i>local_char</i> pairs can be added to the same line

Example

Map the # symbol to the UK currency symbol

```
CharacterMapTable = "#", "\163"
```

Related Emulation Commands

[MapTable](#)

ChgTaggedAttr

Syntax

ChgTaggedAttr *effect_list* = *seq*

Description

This sequence changes the display of the tagged attribute.

Parameters

The following table describes the parameters of the ChgTaggedAttr command:

Parameter	Description
<i>effect_list</i>	The effects to change. Can be either "Normal" or one or more of "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret"
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[AddTaggedAttr](#)

ClearAOL

Syntax

ClearAOL [=] *seq*

Description

This sequence clears all the unprotected characters in the current line.

Parameters

The following table describes the parameters of the ClearAOL command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearSOL](#), [ClearEOL](#)

ClearAllUnprotected

Syntax

ClearAllUnprotected [=] *seq*

Description

Clears all unprotected characters on the screen with the current write attribute and moves the cursor to the first unprotected character.

Parameters

The following table describes the parameters of the ClearAllUnprotected command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearScreen](#), [FormFeed](#), [EraseScreen](#)

ClearEOF

Syntax

ClearEOF [=] *seq*

Synonym

ClearLTP

Description

This sequence clears all the characters until the end of the current field or the end of the line, whichever is sooner.

The end of the field is the next protected character.

Parameters

The following table describes the parameters of the ClearEOF command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

ClearEOL

Syntax

ClearEOL [=] *seq*

Description

This sequence clears all unprotected characters from the current cursor position until the end of the line.

Parameters

The following table describes the parameters of the ClearEOL command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearAOL](#), [ClearSOL](#)

ClearEOP

Syntax

ClearEOP [=] *seq*

Description

This sequence clears the unprotected characters from the current cursor position until the end of the screen.

Parameters

The following table describes the parameters of the ClearEOP command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearSOP](#)

ClearSOL

Syntax

`ClearSOL [=] seq`

Description

This sequence clears the screen from the current cursor position to the start of the current line.

Parameters

The following table describes the parameters of the ClearSOL command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearAOL](#), [ClearEOL](#)

ClearSOP

Syntax

ClearSOP [=] *seq*

Description

This sequence clears the screen from the current cursor position to the start of the screen.

Parameters

The following table describes the parameters of the ClearSOP command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearEOP](#)

ClearScreen

Syntax

ClearScreen [=] *seq*

Description

This sequence clears the screen with the current clear attributes.

Parameters

The following table describes the parameters of the ClearScreen command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[FormFeed](#), [EraseScreen](#), [ClearAllUnprotected](#)

Color Back

Syntax

Color Back *number* = *seq*

Synonym

Colour Back

Description

This sequence sets the write attribute background color.

Parameters

The following table describes the parameters of the Color Back command:

Parameter	Description
<i>number</i>	The color number chosen by this sequence (0 - 15). The color numbers represent the color defined in setup colors.
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Color Fore](#)

Color Fore

Syntax

Color Fore *number* = *seq*

Synonym

Colour Fore

Description

This sequence sets the write attribute foreground color.

Parameters

The following table describes the parameters of the Color Fore command:

Parameter	Description
<i>number</i>	The color number to be used by this sequence (0-15). This number represents the color as shown in setup colors.
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Color Back](#)

CursorCode

Syntax

CursorCode [=] *mode*

Description

This command sets how following lines in the emulation file decode cursor positioning parameters.

The EmulationMode command automatically sets the correct mode for its emulation so this command is only required to override this.

The cursor movement commands can also take an extra parameter that allows the cursor code to be overridden for just one command.

Parameters

The following table describes the parameters of the CursorCode command:

Parameter	Description
=	Equals symbol
<i>mode</i>	The cursor positioning mode. See table below.

Values for mode

The following table shows the calculations made to the parameters in the move sequence to calculate the row and column. p1 is the first parameter in the escape sequence, p2 is the second parameter in the sequence, x is the final column, y is the final row.

Value	Description
<i>L</i>	$y = p1 - 32$ and $x = p2 - 32$
<i>A</i>	$y = p1 - 32$ and $x = p2$. p2 is encoded as binary coded decimal
<i>H</i>	$x = p1$ and $y = p2$
<i>S</i>	$x = p1 - 32$ and $y = p2 - 32$

Value	Description
<i>V</i>	y = p1 and x = p2. If x or y is greater than zero their value is decreased by one.
<i>D</i>	x = p1 and y = p2 where p1 and p2 are encoded as ascii digits.

Example

CursorCode 'L'

MoveXY = \eY%f%f %f bytes represent Y&X

PRINT @(12,10) returns: (027)Y(042)(044)

CursorCode 'A'

MoveXY = \011%f\016%f %f bytes represent Y&X

PRINT @(12,10) returns: (011)(074)(016)(018)

CursorCode 'H'

MoveXY = \e=%f%f %f represent X & Y

PRINT @(12,10) returns: (027)=(012)(010)

CursorCode 'V'

MoveXY = \e[%v;%vH %v represent Y&X

PRINT @(12,10) returns: (027)[13;11H which is the same as:
(027)[(049)(052);(049)(049)H

Related Emulation Commands

[MoveX](#), [MoveXY](#), [MoveY](#), [EmulationMode](#)

CursorDown

Syntax

CursorDown [=] *seq*

Description

This sequence moves the cursor down. The cursor does not move below the bottom scroll row.

Parameters

The following table describes the parameters of the CursorDown command:

Parameter	Description
=	Equals Symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to move down.

Example

Cursor down from ansi emulation

```
CursorDown = "\e[%dB"
```

Cursor down from wyse50 emulation

```
CursorDown = "\022"
```

Related Emulation Commands

[CursorUp](#), [LineFeed](#)

CursorHome

Syntax

CursorHome [=] *seq*, [*emul*]

Description

This sequence moves the cursor to the home position. The home position is the top left hand corner of the screen or current scroll area.

If this sequence is used in a TVI emulation or with the *emul* parameter set to TVI the cursor is moved to the first unprotected character after the home position.

Parameters

The following table describes the parameters of the CursorHome command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>emul</i>	The literal "TVI" to use the Cursor Home functionality from the TVI emulation.

CursorLeft

Syntax

CursorLeft [=] *seq*

Description

This sequence moves the cursor left.

Parameters

The following table describes the parameters of the CursorLeft command:

Parameter	Description
=	Equals Symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of columns to move left.

Example

Cursor left from ansi emulation

```
CursorLeft = "\e[%dD"
```

Cursor left from addsvp emulation

```
CursorLeft = "\021"
```

Related Emulation Commands

[CursorRight](#), [BackSpace](#)

CursorOff

Syntax

CursorOff [=] *seq*

Description

The sequence hides the cursor

Parameters

The following table describes the parameters of the CursorOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[CursorOn](#)

CursorOn

Syntax

CursorOn [=] *seq*

Description

This sequence re-displays the screen cursor after it has been hidden by the cursor off sequence.

Parameters

The following table describes the parameters of the CursorOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[CursorOff](#)

CursorRight

Syntax

CursorRight [=] *seq*

Description

This sequence moves the cursor right.

Parameters

The following table describes the parameters of the CursorRight command:

Parameter	Description
=	Equals Symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of columns to move right.

Example

Cursor right from ansi emulation

```
CursorRight = "\e[%dC"
```

Cursor down from addsvp emulation

```
CursorRight = "\006"
```

Related Emulation Commands

[CursorLeft](#), [ForwardSpace](#)

CursorUp

Syntax

CursorUp [=] *seq*

Description

This sequence moves the cursor up. The cursor does not move above the top scroll row.

Parameters

The following table describes the parameters of the CursorUp command:

Parameter	Description
=	Equals Symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to move up.

Example

Cursor up from ansi emulation

```
CursorUp = "\e[%dA"
```

Cursor up from wyse50 emulation

```
CursorUp = "\011"
```

Related Emulation Commands

[CursorDown](#)

DelayNewLine

Syntax

DelayNewLine [=] *flag*

Description

Stop the cursor auto wrapping at the end of a line until the next character is received. This applies only when the auto wrap mode is enabled.

This setting is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the DelayNewLine command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[EmulationMode](#)

DeleteChar

Syntax

DeleteChar [=] *seq*

Description

This sequence is used to delete one or more characters from the current line.

Parameters

The following table describes the parameters of the DeleteChar command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of charactes to delete.

Related Emulation Commands

[InsertChar](#)

DeleteLine

Syntax

DeleteLine [=] *seq*

Description

This sequence deletes one or more lines from the current cursor position.

Parameters

The following table describes the parameters of the DeleteLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to delete.

Related Emulation Commands

[InsertLine](#)

DisableProtect

Syntax

DisableProtect [=] *seq*

Description

This sequence disables protected characters enabled with the EnableProtect sequence. When the protect status is disabled protected characters can be cleared or overwritten.

Parameters

The following table describes the parameters of the DisableProtect command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[EnableProtect](#)

DisplayChar

Syntax

DisplayChar [=] *seq*, *char*

Description

This sequence maps an escape sequence from the host to a character on the screen. The character to be displayed on the screen can be from any of the characters in the IBSfont character set.

This command replaces the DisplayMap command used previously.

Parameters

The following table describes the parameters of the DisplayChar command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>char</i>	The character to display on the screen from the IBSFont. Use the format "\cNNNN" where NNNN is the character number from the IBSFont. To see a list of characters run the script "example\script\ChrSet2.wis"

Example

The following is the single line character mapping from the addsvp emulation

```
DisplayChar = "\eH:", "\c0256" ;* Single horizontal
DisplayChar = "\eH6", "\c0257" ;* Single vertical
DisplayChar = "\e1L", "\c0258" ;* Single top left
DisplayChar = "\e1H", "\c0259" ;* Single top right
DisplayChar = "\e1D", "\c0260" ;* Single bottom left
DisplayChar = "\e1@", "\c0261" ;* Single bottom right
```

Related Emulation Commands

[DisplayMap](#), [MapTable](#)

DisplayMap

Syntax

DisplayMap [=] *seq*, *char*

Description

This sequence displays the specified character.

This command has been replaced by the DisplayChar command that allows translation to any character in the IBSTFont.

Parameters

The following table describes the parameters of the DisplayMap command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>char</i>	The character to display.

Related Emulation Commands

[MapTable](#), [DisplayMap](#)

DisplayNulls

Syntax

DisplayNulls [=] *flag*

Description

Allows received character zero's to be displayed.

If this command is not in the emulation character zero's will not be displayed.

Parameters

The following table describes the parameters of the DisplayNulls command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

DisplayOff

Syntax

DisplayOff [=] *seq*

Description

This sequence turns the screen display off. When the display is off the session will show an empty black screen.

The display is turned back on with the DisplayOn sequence.

Parameters

The following table describes the parameters of the DisplayOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DisplayOn](#)

DisplayOn

Syntax

DisplayOn [=] *seq*

Description

This sequence turns the display back on after it has been hidden by a DisplayOff sequence.

Parameters

The following table describes the parameters of the DisplayOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DisplayOff](#)

DoubleHeight

Syntax

DoubleHeight [=] *seq*

Description

This sequence produces double height characters until the end of the line.

This sequence needs to be repeated on the following line for the bottom half of the characters.

It is only implemented for the viewdata emulation.

Parameters

The following table describes the parameters of the DoubleHeight command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

DoubleHeightLineBottom

Syntax

DoubleHeightLineBottom [=] *seq*

Description

This sequence changes the current row into double height bottom characters.

Parameters

The following table describes the parameters of the DoubleHeightLineBottom command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleHeightLineTop](#)

DoubleHeightLineTop

Syntax

DoubleHeightLineTop [=] *seq*

Description

This sequence changes the current row into double height top characters.

Parameters

The following table describes the parameters of the DoubleHeightLineTop command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleHeightLineBottom](#)

DoubleSizeLineBottom

Syntax

DoubleSizeLineBottom [=] *seq*

Description

This sequence changes the current row into double size top characters.

Parameters

The following table describes the parameters of the DoubleSizeLineBottom command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleSizeLineTop](#)

DoubleSizeLineTop

Syntax

DoubleSizeLineTop [=] *seq*

Description

This sequence changes the current row into double size top characters.

Parameters

The following table describes the parameters of the DoubleSizeLineTop command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleSizeLineBottom](#)

DoubleWidthLine

Syntax

DoubleWidthLine [=] *seq*

Description

This sequence changes the current row into double width characters.

Parameters

The following table describes the parameters of the DoubleWidthLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Effect

Syntax

Effect *effect_list* = *seq*

Description

This sequence changes to the specific write attribute.

Parameters

The following table describes the parameters of the Effect command:

Parameter	Description
<i>effect_list</i>	The effect to change to. Can be either "Normal" or one or more of "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret"
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Effect All](#)

Effect ANSI

Syntax

Effect ANSI

Description

This setting defines the effect mode to be ANSI compatible. This is used with the Effect All command to specify how the effect parameter(s) are decoded.

Parameters

None

Related Emulation Commands

[Effect All](#), [Effect Mask](#), [Effect Order](#)

Effect All

Syntax

Effect All [=] *seq*, [*l*]

Description

This sequence changes the current write attributes.

The sequence should contain one parameter which is decoded by the current emulation mode to produce the required effect.

Parameters

The following table describes the parameters of the Effect All command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This must contain one parameter that is decoded to decide which effect to use.
<i>l</i>	This is used to specify an alternate attributing scheme and should only be used after an "Effect ANSI" command.

Related Emulation Commands

[Effect](#), [Effect ANSI](#), [EmulationMode](#)

Effect CharacterRange

Syntax

Effect CharacterRange [=] *lower_char*, *upper_char*

Description

This sequence sets the range of characters that change the write attribute. This new attribute is determined by the mask used in the Effect Mask command.

The EmulationMode command also sets this parameter.

Parameters

The following table describes the parameters of the Effect CharacterRange command:

Parameter	Description
=	Equals symbol
<i>lower_char</i>	The effect character at the beginning of the range.
<i>upper_char</i>	The effect character at the end of the range.

Related Emulation Commands

[EmulationMode](#), [Effect Mask](#), [Effect All](#)

Effect Mask

Syntax

Effect Mask *effect_list*

Description

This setting defines the effect mode to match each bit of the byte parameter from the Effect All with an effect.

Parameters

The following table describes the parameters of the Effect Mask command:

Parameter	Description
<i>effect_list</i>	This is the list of effects in the order they appear in the byte parameter (first bit 0, second bit 1 ... last bit 7). Use the effect names separated by spaces with the word unused placed for any unused bits. The effect names are "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret".

Example

Define effect sequence for wyse50

```
Effect Mask Secret Flash Reverse Underline Unused Unused Dim
Effect All = "\eG%f"
```

Related Emulation Commands

[Effect All](#), [Effect ANSI](#), [Effect Order](#), [Effect CharacterRange](#)

Effect Order

Syntax

Effect Order *effect_list*

Description

This setting is used with Effect All to define the effect decoding of the parameter so that each effect is a value.

Parameters

The following table describes the parameters of the Effect Order command:

Parameter	Description
<i>effect_list</i>	This is the list of effects in the order they appear in the byte. The order of the effect numbers (1,2,3,4...). Use the effect names separated by spaces with the word unused placed for any unused bits. The effect names are "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret".

Related Emulation Commands

[Effect All](#), [Effect Mask](#), [Effect ANSI](#)

EffectExtent

Syntax

EffectExtent *extent*

Description

This command sets how many characters are affected when the current attribute is changed.

This setting is also set by the EmulationMode command.

Note: This command is obsolete, use the EmulationMode or AttributeMode command instead.

Parameters

The following table describes the parameters of the EffectExtent command:

Parameter	Description
<i>extent</i>	The extent of the screen affected by an attribute change. See below.

Values for extent

Value	Description
<i>0</i>	The character written
<i>1</i>	The characters until the next attribute change or the end of the page
<i>2</i>	The characters until the next attribute change or the end of the line

Related Emulation Commands

[EmulationMode](#), [AttributeMode](#)

EffectIgnoreBold

Syntax

EffectIgnoreBold [=] *flag*

Description

Allows the bold attribute to be treated as an effect or color brightener.

Parameters

The following table describes the parameters of the EffectIgnoreBold command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

EffectIgnoreReverse

Syntax

EffectIgnoreReverse [=] *flag*

Description

Allow the reverse attribute to be treated as an effect or color modifier.

Parameters

The following table describes the parameters of the EffectIgnoreReverse command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

EmbeddedAttributes

Syntax

EmbeddedAttributes *flag*

Description

This command specifies if a change of write attributes uses a space on the screen.

The EmulationMode command sets this value.

Note: This command is obsolete, use the EmulationMode or AttributeMode command instead.

Parameters

The following table describes the parameters of the EmbeddedAttributes command:

Parameter	Description
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[EmulationMode](#), [AttributeMode](#)

EmulationMode

Syntax

EmulationMode *terminal_type*

Description

This command sets the base mode for the emulation. This defines the initial states and attribute masks for the emulation.

This command will also enable additional/different processing for adding characters to the screen and any command variations necessary.

It should only be used once in an emulation file and placed at, or near to, the beginning of the file.

Parameters

The following table describes the parameters of the EmulationMode command:

Parameter	Description
<i>terminal_type</i>	See the following table for the valid terminal types

Values for terminal_type

Value	Description
<i>NONE</i>	Basic settings compatible with version 3.x
<i>ANSI</i>	ANSI emulation
<i>DG</i>	DG413
<i>SM</i>	SM4407
<i>PRIME</i>	PT250
<i>PRISM</i>	Prism 9
<i>WYSE</i>	Wyse 60
<i>TVI</i>	TVI 955

Value	Description
<i>IBM</i>	IBM3151
<i>QUME</i>	QVT119
<i>VIEWDATA</i>	Viewdata
<i>AT386</i>	AT 386
<i>VT100</i>	VT 100
<i>VT220</i>	VT 220
<i>VT420</i>	VT 420
<i>SCOANSI</i>	SCO ANSI
<i>TVI910</i>	Televideo 910
<i>TVI920</i>	Televideo 920
<i>TVI925</i>	Televideo 925
<i>TVI950</i>	Televideo 950
<i>TVI955</i>	Televideo 955
<i>TVI965</i>	Televideo 965
<i>WYSE50</i>	Wyse 50
<i>WYSE60</i>	Wyse 60
<i>DG413</i>	DG 413
<i>PT200</i>	Prime PT200
<i>PT250</i>	Prime PT250
<i>PRISM5</i>	Prism 5
<i>PRISM8</i>	Prism 8
<i>PRISM9</i>	Prism 9
<i>IBM3151</i>	IBM3151
<i>SM4407</i>	SM 4407
<i>QVT119</i>	Qume QVT119

EnableProtect

Syntax

EnableProtect [=] *seq*

Description

This sequence enables protected characters. When the protected characters are enabled they can not be overwritten or cleared until a DisableProtect sequence is executed.

Parameters

The following table describes the parameters of the EnableProtect command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DisableProtect](#)

EraseCharsInLine

Syntax

EraseCharsInLine [=] *seq*

Description

This sequence clears a number of characters from the current cursor position.

Parameters

The following table describes the parameters of the EraseCharsInLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of characters to erase.

EraseScreen

Syntax

EraseScreen = *seq*

Description

This sequence clears the screen and resets all the attributes and the scroll region.

Parameters

The following table describes the parameters of the EraseScreen command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearScreen](#), [ClearAllUnprotected](#), [FormFeed](#)

EraseUsingWriteAttr

Syntax

EraseUsingWriteAttr [=] *flag*

Description

When erasing use write attributes instead of erase attributes.

This is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the EraseUsingWriteAttr command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

Related Emulation Commands

[EmulationMode](#)

FormFeed

Syntax

FormFeed [=] *seq*

Synonym

FF

Description

This sequence clears the screen always using the current write attributes.

Parameters

The following table describes the parameters of the FormFeed command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ClearScreen](#), [ClearAllUnprotected](#), [EraseScreen](#)

ForwardSpace

Syntax

ForwardSpace = *seq*

Description

This sequence moves the cursor forward. This sequence will move to the next line if auto wrap is on.

Parameters

The following table describes the parameters of the ForwardSpace command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of columns to move right.

Related Emulation Commands

[CursorRight](#)

GraphicMapTable

Syntax

GraphicMapTable *host_char, local_char, [...]*

Description

This command is obsolete and kept for backwards compatability only. It is identical to using the MapTable command for table number 1.

Parameters

The following table describes the parameters of the GraphicMapTable command:

Parameter	Description
<i>host_char</i>	The character received from the host (or sent to the host for characters 32 to 255).
<i>local_char</i>	The character displayed locally on the PC
...	Additional <i>host_char</i> , <i>local_char</i> pairs can be added to the same line

Related Emulation Commands

[MapTable](#)

GraphicModeOff

Syntax

GraphicModeOff [=] *seq*

Description

This sequence switches the emulation to use map table 0 for it's character mapping.

Parameters

The following table describes the parameters of the GraphicModeOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[GraphicModeOn](#)

GraphicModeOn

Syntax

GraphicModeOn [=] *seq*

Description

This sequence switches the emulation to use map table 1 for it's character mapping.

Parameters

The following table describes the parameters of the GraphicModeOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[GraphicModeOff](#)

HardReset

Syntax

HardReset [=] *seq*

Description

This sequence performs a hard reset.

Parameters

The following table describes the parameters of the HardReset command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[SoftReset](#)

Ignore

Syntax

Ignore [=] *seq*

Description

This sequence will be ignored by the emulation.

Parameters

The following table describes the parameters of the Ignore command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

IgnoreErrors

Syntax

IgnoreErrors [=] *flag*

Description

Ignore errors on loading wit files.

The default state is on because previous versions of this application did not perform as much syntax checking. This allows customers to keep their same WIT files between versions even if there are errors in their WIT files.

Parameters

The following table describes the parameters of the IgnoreErrors command:

Parameter	Description
=	Equals symbol
<i>flag</i>	True, Yes, On or 1 to turn on this feature, False, No, Off or 0 to turn off this feature.

InsertChar

Syntax

InsertChar [=] *seq*

Description

This sequence inserts one or more spaces at the current cursor position.

Parameters

The following table describes the parameters of the InsertChar command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of spaces to insert.

Related Emulation Commands

[DeleteChar](#)

InsertLine

Syntax

InsertLine [=] *seq*

Description

This sequence inserts one or more blank lines at the current cursor position.

Parameters

The following table describes the parameters of the InsertLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to insert.

Related Emulation Commands

[DeleteLine](#)

InsertModeOff

Syntax

InsertModeOff [=] *seq*

Description

This sequence turns off insert mode, so characters placed on the screen will be overwritten.

Parameters

The following table describes the parameters of the InsertModeOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[InsertModeOn](#)

InsertModeOn

Syntax

InsertModeOn [=] *seq*

Description

This sequence turns on insert mode, so that when characters are displayed on the screen they shift existing characters to the right.

Parameters

The following table describes the parameters of the InsertModeOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[InsertModeOff](#)

Key_

Syntax

Key_*[state]**name* [=] *value*

Description

This command programs a soft key for use with the emulation.

These commands are only executed if the Load keys check box is on in Setup Terminal or the Key LoadEmulation script command is used.

If the same key is defined twice (say in an emulation and in an emulation extension file) the last definition is used.

Note: Unlike other commands the state and name parameters are part of the Key_ command name.

Parameters

The following table describes the parameters of the Key_ command:

Parameter	Description
<i>state</i>	The Shift state for the key. Can be Shift, Control, Alt or a combination of the three. These can also be abbreviated to S, C or A respectively.
<i>name</i>	The name of the key to program. The key name can be found by looking at the name in the Setup Keyboard menu option or in the list for "SetupKeyboard" in the "Client Scripting Reference".
=	Equals symbol
<i>value</i>	The key definition. If this is prefixed with a "\m" the rest of the value is treated as script, otherwise the value is sent to the host after being converted from backslash format.

Example

The following definitions are from the key_norm emulation extension that defines keys for scrolling around the back pages.

```
Key_PageUp = "\mInvoke ScrollPrevPage"  
Key_PageDown = "\mInvoke ScrollNextpage"  
Key_UpArrow = "\mInvoke ScrollUpLine"  
Key_DownArrow = "\mInvoke ScrollDownLine"  
Key_Control_PageDown = "\mInvoke ScrollBackWidth"  
Key_Control_PageUp = "\mInvoke ScrollAcrossWidth"  
Key_End = "\mInvoke ScrollEndPage"  
Key_Home = "\mInvoke ScrollTopPage"  
Key_Shift_Insert = "\mInvoke EditPaste"  
Key_Control_Insert = "\mInvoke EditCopy"
```

KeyboardLock

Syntax

KeyboardLock [=] *seq*

Description

This sequence locks the keyboard so that is stops unprogrammed keys from being sent to the host, instead a bell is sounded.

Programmed keys such as edit or function keys can not be locked.

Parameters

The following table describes the parameters of the KeyboardLock command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[KeyboardUnlock](#)

KeyboardUnlock

Syntax

KeyboardUnlock [=] *seq*

Description

This sequence unlocks the keyboard after it has been locked with the KeyboardLock escape sequence.

Parameters

The following table describes the parameters of the KeyboardUnlock command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[KeyboardLock](#)

Library

Syntax

Library [=] *filename*

Description

This command loads a script library into memory allowing the direct use of any subroutine or functions as if they are part of script language.

This is useful as it allows more complicated scripting to be run as part of a TerminalCommand command without having to run another script file or specifically include the Library script command in the TerminalCommand statement. As the library is cached in memory it also does not need to be reloaded to be run which means it will also execute faster.

Parameters

The following table describes the parameters of the Library command:

Parameter	Description
=	Equals symbol
<i>filename</i>	The name of the script library. If the name consists only of a leaf name it is loaded from the wintsys\wit_ext folder. If the name contains a partial path it is assumed to be in a sub folder of the folder where the application was installed.

Example

The following line in the anis.wit emulation includes the library "wintsys\wit_ext\Ansiext.wis"

```
Library = "Ansiext"
```

LineFeed

Syntax

LineFeed [=] *seq*

Synonym

LF

Description

This sequence moves the cursor down and scrolls the screen when it reaches the bottom line.

Parameters

The following table describes the parameters of the LineFeed command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[CursorDown](#)

LockShiftGLFont

Syntax

LockShiftGLFont [=] *seq*, *bank*

Description

This sequence selects the emulation character bank to use for the display of 7 bit characters (characters with ASCII codes 32 to 127).

Parameters

The following table describes the parameters of the LockShiftGLFont command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>bank</i>	The emulation bank to select (0-3)

Related Emulation Commands

[LockShiftGRFont](#)

LockShiftGRFont

Syntax

LockShiftGRFont [=] *seq*

Description

This sequence select an emulation (character set) bank to use for 8 bit characters (characters with ASCII codes 160 to 255).

Parameters

The following table describes the parameters of the LockShiftGRFont command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[LockShiftGLFont](#)

MapSequence

Syntax

MapSequence [=] *seq*, *new_seq*

Description

This sequence replaces the specified received characters that are at the beginning of a sequence and then passes this on to be interpreted by the emulation.

Parameters

The following table describes the parameters of the MapSequence command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>new_seq</i>	The replacement start of the sequence

Example

The following line from the vt100 emulation converts the 8 bit version of the Command Sequence Introducer to the 7bit version. This prevents the need to define each sequence twice for 7bit and 8bit emulation.

```
MapSequence = "\155", "\e[" ;* CSI
```

MapTable

Syntax

MapTable [*table*], *host_char*, *local_char*, [...]

Description

This sequence sets up a character mapping between the characters on the host and the characters from the IBSFont that are displayed on the screen. For ASCII codes 32 to 255 the character mapping will also convert characters entered by the keyboard to the value on the host.

The command can be repeated multiple times with each command adding more character mappings to the specified character mapping table.

Parameters

The following table describes the parameters of the MapTable command:

Parameter	Description
<i>table</i>	The table number to add character mappings to. If omitted this parameter defaults to table 0.
<i>host_char</i>	The character received from the host (or sent to the host for characters 32 to 255).
<i>local_char</i>	The character displayed locally on the PC
...	Additional <i>host_char</i> , <i>local_char</i> pairs can be added to the same line

Example

Following maps the English pound currency symbol

```
MapTable = "#", "\163"
```

Related Emulation Commands

[DisplayChar](#)

MarginBell

Syntax

MarginBell [=] *column*

Description

Margin bell position. If column is zero then the margin bell is turned off. Any other value is taken as the position to sound the bell as the user types a character at this column.

If this command is not specified the margin bell is turned off.

Parameters

The following table describes the parameters of the MarginBell command:

Parameter	Description
=	Equals symbol
<i>column</i>	The column position to sound the margin bell or 0 to turn off the margin bell.

Related Emulation Commands

[MarginBellOn](#), [MarginBellOff](#), [MarginBellPsn](#), [SetMarginBell](#)

MarginBellOff

Syntax

MarginBellOff [=] *seq*

Description

This sequence turns the margin bell off.

Parameters

The following table describes the parameters of the MarginBellOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MarginBellOn](#), [MarginBellPsn](#), [MarginBell](#), [SetMarginBell](#)

MarginBellOn

Syntax

MarginBellOn [=] *seq*

Description

This sequence enables the margin bell to sound when a character is typed at the margin bell column

Parameters

The following table describes the parameters of the MarginBellOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MarginBellOff](#), [MarginBellPsn](#), [MarginBell](#), [SetMarginBell](#)

MarginBellPsn

Syntax

MarginBellPsn [=] *seq*

Description

This sequence sets the column for the margin bell

Parameters

The following table describes the parameters of the MarginBellPsn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This sequence must include one numeric parameter to specify the margin bell column.

Related Emulation Commands

[MarginBellOn](#), [MarginBellOff](#), [MarginBell](#), [SetMarginBell](#)

MosaicBlue

Syntax

MosaicBlue [=] *seq*

Description

This sequence causes the following characters to be displayed as blue mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicBlue command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicContiguous](#), [MosaicCyan](#), [MosaicGreen](#), [MosaicHold](#), [MosaicPurple](#), [MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#), [MosaicYellow](#)

MosaicContiguous

Syntax

MosaicContiguous [=] *seq*

Description

This sequence makes following mosaic characters on the line display as solid.

This command applies to the viewdata emulation only.

Parameters

The following table describes the parameters of the MosaicContiguous command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicSeparated](#)

MosaicCyan

Syntax

MosaicCyan [=] *seq*

Description

This sequence causes the following characters to be displayed as cyan mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicCyan command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicGreen](#), [MosaicHold](#), [MosaicPurple](#), [MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#), [MosaicYellow](#)

MosaicGreen

Syntax

MosaicGreen [=] *seq*

Description

This sequence causes the following characters to be displayed as green mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicGreen command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicCyan](#), [MosaicHold](#), [MosaicPurple](#),
[MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#), [MosaicYellow](#)

MosaicHold

Syntax

MosaicHold [=] *seq*

Description

This sequence displays a copy of the previous mosaic character on top of an attribute to hide the space character inserted on attribute/color changes.

This command applies to the viewdata emulation only.

Parameters

The following table describes the parameters of the MosaicHold command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicRelease](#)

MosaicPurple

Syntax

MosaicPurple [=] *seq*

Description

This sequence causes the following characters to be displayed as purple mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicPurple command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicCyan](#), [MosaicGreen](#), [MosaicHold](#),
[MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#), [MosaicYellow](#)

MosaicRed

Syntax

MosaicRed [=] *seq*

Description

This sequence causes the following characters to be displayed as red mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicRed command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicCyan](#), [MosaicGreen](#), [MosaicHold](#),
[MosaicPurple](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#), [MosaicYellow](#)

MosaicRelease

Syntax

MosaicRelease [=] *seq*

Description

This sequence releases a previous HoldMosaic sequence so that a space is once again embedded in the line for a change of color or attribute.

Parameters

The following table describes the parameters of the MosaicRelease command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicHold](#)

MosaicSeparated

Syntax

MosaicSeparated [=] *seq*

Description

This sequence causes the mosaic character following it to be displayed with separated blocks in a 2 x 3 grid.

Parameters

The following table describes the parameters of the MosaicSeparated command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicContiguous](#)

MosaicWhite

Syntax

MosaicWhite [=] *seq*

Description

This sequence causes the following characters to be displayed as white mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicWhite command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicCyan](#), [MosaicGreen](#), [MosaicHold](#), [MosaicPurple](#), [MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicYellow](#)

MosaicYellow

Syntax

MosaicYellow [=] *seq*

Description

This sequence causes the following characters to be displayed as yellow mosaic graphic characters until the next display sequence or the end of the line.

This command applies to the viewdata emulation only. i.e. EmulationMode = "VIEWDATA" must be set.

Parameters

The following table describes the parameters of the MosaicYellow command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MosaicBlue](#), [MosaicContiguous](#), [MosaicCyan](#), [MosaicGreen](#), [MosaicHold](#), [MosaicPurple](#), [MosaicRed](#), [MosaicRelease](#), [MosaicSeparated](#), [MosaicWhite](#)

MoveX

Syntax

MoveX [=] *seq*, [*cursor_code*]

Description

This sequence moves the cursor to the specified column position on the current line.

Parameters

The following table describes the parameters of the MoveX command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This must contain one parameter that is used to specify the column position
<i>cursor_code</i>	Specifies how to interpret the parameter. Entered as a CursorCode letter or an emulation mode name. If this parameter is omitted the code for the EmulationMode or the last CursorCode command is used.

Related Emulation Commands

[MoveY](#), [MoveXY](#), [CursorCode](#), [EmulationMode](#)

MoveXY

Syntax

MoveXY [=] *seq*, [*cursor_code*]

Description

This sequence moves the cursor to the specified line and column.

Parameters

The following table describes the parameters of the MoveXY command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This command must contain two parameters to specify the line and column positions.
<i>cursor_code</i>	Specifies how to interpret the parameter. Entered as a CursorCode letter or an emulation mode name. If this parameter is omitted the code for the EmulationMode or the last CursorCode command is used.

Related Emulation Commands

[MoveX](#), [MoveY](#), [CursorCode](#), [EmulationMode](#)

MoveY

Syntax

MoveY [=] *seq*, [*cursor_code*]

Description

This sequence moves the cursor to the specified line without changing the column position.

Parameters

The following table describes the parameters of the MoveY command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. This sequence must contain one parameter that is decoded to specify the line to move to.
<i>cursor_code</i>	Specifies how to interpret the parameter. Entered as a CursorCode letter or an emulation mode name. If this parameter is omitted the code for the EmulationMode or the last CursorCode command is used.

Related Emulation Commands

[MoveX](#), [MoveXY](#), [CursorCode](#), [EmulationMode](#)

NewBackground

Syntax

NewBackground [=] *seq*

Description

This sequence sets the background color until the end of the line or the next background color sequence or BlackBackground command.

This command is available for the viewdata emulation only.

Parameters

The following table describes the parameters of the NewBackground command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[BlackBackground](#)

NewLine

Syntax

NewLine [=] *seq*

Synonym

NL

Description

This sequence moves the cursor to the start of the next line.

Parameters

The following table describes the parameters of the NewLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to move down.

Related Emulation Commands

[LineFeed](#)

NormalHeight

Syntax

NormalHeight [=] *seq*

Description

This sequence sets normal height characters to the end of the line or next DoubleHeight sequence.

It applies to the viewdata emulation only.

Parameters

The following table describes the parameters of the NormalHeight command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleHeight](#)

NormalVideo

Syntax

NormalVideo [=] *seq*

Description

This sequence disables reverse video.

Parameters

The following table describes the parameters of the NormalVideo command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ReverseVideo](#)

PrinterIgnore

Syntax

PrinterIgnore [=] *seq*

Description

This sequence stops the specified characters from being sent to the printer.

Parameters

The following table describes the parameters of the PrinterIgnore command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[PrinterMap](#)

PrinterMap

Syntax

PrinterMap [=] *seq*, *printer_seq*

Description

This command changes a sequence of characters sent by the host to a different sequence during printing.

This has been implemented to allow ANSI-based emulations such as VT100 to throw a new page properly when an on-screen report is sent to the printer.

Parameters

The following table describes the parameters of the PrinterMap command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>printer_seq</i>	The sequence of characters to send to the printer

Example

From the ANSIprt emulation extension. Ignore the home sequence and map ANSI form feeds to ASCII code 12 for printing.

```
PrinterIgnore = "\e[%aH"  
PrinterMap = "\e[%dJ", "\012"
```

Related Emulation Commands

[PrinterIgnore](#)

ProtectEffect

Syntax

ProtectEffect *effect_list* = *seq*

Description

This sequence changes the display of the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use ChgTaggedAttr instead.

Parameters

The following table describes the parameters of the ProtectEffect command:

Parameter	Description
<i>effect_list</i>	The effects to change. Can be either "Normal" or one or more of "Dim", "Bold", "Reverse", "UnderLine", "Flash" or "Secret"
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ChgTaggedAttr](#)

ProtectFullIntensity

Syntax

ProtectFullIntensity [=] *seq*

Description

This sequence causes protected characters to be written with normal intensity.

Parameters

The following table describes the parameters of the ProtectFullIntensity command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ProtectHalfIntensity](#)

ProtectHalfIntensity

Syntax

ProtectHalfIntensity [=] *seq*

Description

This sequence causes protected characters to be written using dim intensity.

Parameters

The following table describes the parameters of the ProtectHalfIntensity command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ProtectFullIntensity](#)

ProtectOff

Syntax

ProtectOff [=] *seq*

Description

This sequence turns off the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use TaggedAttrOff instead.

Parameters

The following table describes the parameters of the ProtectOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOff](#)

ProtectOn

Syntax

ProtectOn [=] *seq*

Description

This sequence turns on the tagged attribute. Characters written to the screen after this sequence will include the tagged attribute.

Note: This sequence is obsolete and kept for backward compatability only. Use TaggedAttrOn instead.

Parameters

The following table describes the parameters of the ProtectOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOn](#)

ProtectTab

Syntax

ProtectTab [=] *seq*

Description

This sequence moves the cursor to the next tab stop unless protect mode is enabled in which case the cursor is moved to the next unprotected field.

Parameters

The following table describes the parameters of the ProtectTab command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Tab](#)

RestoreCursor

Syntax

RestoreCursor [=] *seq*

Description

This sequence restores the cursor to its position and the attributes to their state when the last SaveCursor sequence was executed.

Parameters

The following table describes the parameters of the RestoreCursor command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[SaveCursor](#)

Reveal

Syntax

Reveal [=] *seq*

Description

This sequence reveals all the concealed (secret) text on the screen.

This command is available for the viewdata emulation only.

Parameters

The following table describes the parameters of the Reveal command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

ReverseLF

Syntax

ReverseLF [=] *seq*

Description

This sequence moves the cursor up the screen one or more lines scrolling the screen down if it is already at the top of the scroll area.

Parameters

The following table describes the parameters of the ReverseLF command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to move up.

Related Emulation Commands

[LineFeed](#), [CursorUp](#)

ReverseNewLine

Syntax

ReverseNewLine [=] *seq*

Description

This sequence moves the cursor up one or more lines and positions the cursor at the left hand side of the line. If the cursor is already at the top of the scroll area the scroll area is scrolled down.

Parameters

The following table describes the parameters of the ReverseNewLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to move up.

Related Emulation Commands

[NewLine](#), [ReverseLF](#)

ReverseVideo

Syntax

ReverseVideo [=] *seq*

Description

This sequence reverses the appearance of the screen.

Note: This sequence has not been implemented and is currently ignored.

Parameters

The following table describes the parameters of the ReverseVideo command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[NormalVideo](#)

SaveCursor

Syntax

SaveCursor [=] *seq*

Description

This sequence saves the cursor position and current attributes. These are then later restored by the RestoreCursor sequence.

Parameters

The following table describes the parameters of the SaveCursor command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[RestoreCursor](#)

ScrollDown

Syntax

ScrollDown [=] *seq*

Description

This sequence scrolls the screen down without changing the cursor position.

Parameters

The following table describes the parameters of the ScrollDown command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to scroll down.

Related Emulation Commands

[ScrollUp](#)

ScrollUp

Syntax

ScrollUp [=] *seq*

Description

This sequence scrolls the screen up without changing the cursor position.

Parameters

The following table describes the parameters of the ScrollUp command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If the sequence contains a numeric parameter this will specify the number of lines to scroll up.

Related Emulation Commands

[ScrollDown](#)

SetMarginBell

Syntax

SetMarginBell [=] *seq*

Description

This sequence sets the column that rings the margin bell to the current column.

Parameters

The following table describes the parameters of the SetMarginBell command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[MarginBellOn](#), [MarginBellOff](#), [MarginBellPsn](#), [MarginBell](#)

SingleLine

Syntax

SingleLine [=] *seq*

Description

This sequence changes the current cursor line to normal characters.

Parameters

The following table describes the parameters of the SingleLine command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[DoubleHeightLineTop](#), [DoubleHeightLineBottom](#), [DoubleSizeLineTop](#),
[DoubleSizeLineBottom](#), [DoubleWidthLine](#)

SingleShift

Syntax

SingleShift [=] *seq*, *bank*

Description

This sequence sets the bank (character set) for the next 7 bit character (characters with ASCII code 32 to 127).

Parameters

The following table describes the parameters of the SingleShift command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>bank</i>	The bank number to use (0-3)

Related Emulation Commands

[LockShiftGLFont](#), [GraphicModeOn](#), [GraphicModeOff](#)

SoftReset

Syntax

SoftReset [=] *seq*

Description

This sequence performs a soft reset.

Parameters

The following table describes the parameters of the SoftReset command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[HardReset](#)

StartCommand

Syntax

StartCommand [=] *script*

Description

This command specifies a script line to be run when the emulation is first run or when the emulation is reset.

This command can be repeated multiple times to specify multiple script lines to run.

Parameters

The following table describes the parameters of the StartCommand command:

Parameter	Description
=	Equals symbol
<i>script</i>	The script line to execute

Tab

Syntax

Tab [=] *seq*

Description

This sequence moves the cursor to the next tab stop.

Parameters

The following table describes the parameters of the Tab command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[ProtectTab](#), [TabClear](#), [TabClearAll](#), [TabSet](#)

TabClear

Syntax

TabClear [=] *seq*

Description

This sequence clears the tab stop at the current column.

Parameters

The following table describes the parameters of the TabClear command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TabSet](#), [TabClearAll](#), [Tab](#), [ProtectTab](#)

TabClearAll

Syntax

TabClearAll [=] *seq*

Description

This sequence clears all the horizontal tab stops.

Parameters

The following table describes the parameters of the TabClearAll command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TabClear](#), [TabSet](#), [Tab](#), [ProtectTab](#)

TabSet

Syntax

TabSet [=] *seq*

Description

This sequence sets a tab stop at the current cursor position

Parameters

The following table describes the parameters of the TabSet command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TabClear](#), [TabClearAll](#), [Tab](#), [ProtectTab](#)

TaggedAttr

Syntax

TaggedAttr [=] *effect*

Description

Sets how the tagged attribute will appear on the screen.

This is also set by the EmulationMode command.

Parameters

The following table describes the parameters of the TaggedAttr command:

Parameter	Description
=	Equals Symbol
<i>effect</i>	The effect to use for the tagged attribute. See below.

Values for effect

Use "Normal" or one or more of the following effect names

Value	Description
<i>Dim</i>	Low intensity
<i>Reverse</i>	Reverse foreground and background
<i>Underline</i>	Underline
<i>Flash</i>	Blinking text
<i>Secret</i>	Hidden text
<i>Bold</i>	High intensity

Related Emulation Commands

[EmulationMode](#)

TaggedAttrOff

Syntax

TaggedAttrOff [=] *seq*

Description

This sequence turns off the tagged attribute.

Parameters

The following table describes the parameters of the TaggedAttrOff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOn](#)

TaggedAttrOn

Syntax

TaggedAttrOn [=] *seq*

Description

This sequence turns on the tagged attribute. Characters written to the screen after this sequence will include the tagged attribute.

Parameters

The following table describes the parameters of the TaggedAttrOn command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[TaggedAttrOff](#)

TekGraphic

Syntax

TekGraphic [=] *seq*

Description

This sequence puts the emulation into TEK graphics mode where following characters and sequences are interpreted as TEK graphic sequences and draw lines and shapes on the screen.

Parameters

The following table describes the parameters of the TekGraphic command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

TelnetTerminal

Syntax

TelnetTerminal [=] *terminal_type*

Description

Sets the default telnet terminal type used for telnet negotiation. This is the same value as seen in the Setup Communications, Windows Sockets, Telnet, Terminal Type menu option.

Parameters

The following table describes the parameters of the TelnetTerminal command:

Parameter	Description
=	Equals symbol
<i>terminal_type</i>	Terminal type to use

TermId

Syntax

TermId [=] *seq*, *response*

Description

This sequence sends the characters specified back to the host when it is received.

Parameters

The following table describes the parameters of the TermId command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command
<i>response</i>	The data to send back to the host

TerminalCommand

Syntax

TerminalCommand [=] *seq*, *script_command*, [*option*]

Description

This command executes a script statement when the sequence is received.

Parameters

The following table describes the parameters of the TerminalCommand command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command. If parameters are included in this sequence they can be passed on to the script command run.
<i>script_command</i>	The text of the script command to run. See the following table for special values that can be used to substitute parameters from the received sequence.
<i>option</i>	Single character option to modify the execution behavior of the command. See the table below.

Values for script_command

The following values can be placed into the script command and will be replaced when the sequence is received from the host. In the following table p is the parameter number, and n is a numeric adjustment.

Value	Description
<i>%p[+/-n]b</i>	Substitute the ascii code of the byte from the parameter optionally adding or subtracting n from it.
<i>%pd</i>	Substitute the number (ascii digits) from the parameter.

Value	Description
<i>%p[+n]s</i>	Substitute the bytes of the parameter optionally starting from the nth byte
<i>%p[+n]f</i>	Substitute the bytes of the parameter after converting it to backslash format optionally starting from the nth byte.

Values for option

Value	Description
<i>I</i>	Immediate. This command executes the script command immediately and continues processing the characters received. It should not be used with scripts that multi-task or take a long time to process
<i>P</i>	Execute this command even when terminal command processing has been turned off when printing

Example

Run a script to define the numeric keypad in the vt100 emulation.

```
TerminalCommand= "\e=", "Script 'wintsys\\wit_scr\\vtnumkey', 1"
```

A command with parameter substitution to move the cursor on the same line.

```
TerminalCommand = "\e[%db", "x=0%ls;if x<1 then x=1;Display AT  
x-1,Cursor(GET_Y)"
```

Implement printing in the vt100 emulation.

```
TerminalCommand = "\e[i", "Script 'wintsys\\wit_scr\\PrintScr'"  
TerminalCommand = "\e[0i", "Script 'wintsys\\wit_scr\\PrintScr'"  
TerminalCommand = "\e[4i", "Chain 'wintsys\\wit_scr\\printOff' " ,P  
TerminalCommand = "\e[?4i", "Chain 'wintsys\\wit_scr\\PrintOff' " ,P  
TerminalCommand = "\e[5i", "Script 'wintsys\\wit_scr\\printOn'"  
TerminalCommand = "\e[?5i", "Script 'wintsys\\wit_scr\\CopyPrt'"
```

Xoff

Syntax

Xoff [=] *seq*

Description

When Xon has been defined this sequence will stop data from being sent to the host.

Normally this is set to "\019" if used.

See the Xon entry for when this command should be used.

Parameters

The following table describes the parameters of the Xoff command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Xon](#)

Xon

Syntax

Xon [=] *seq*

Description

This sequence re-enables the sending of data to the host after it has been stopped by the Xoff sequence.

Normally this is set to "\017" if used.

Including this sequence disables the software flow control from the serial communications and instead implements it within the emulation.

This command should only be used if Xon/Xoff can not be enabled as part of the serial communications setup, for example if the emulation uses the Xon or Xoff characters.

Parameters

The following table describes the parameters of the Xon command:

Parameter	Description
=	Equals symbol
<i>seq</i>	The control sequence to invoke this command

Related Emulation Commands

[Xoff](#)

Index

A

- AddTaggedAttr, [17](#)
- AlarmMessage, [18](#)
- AlarmOff, [19](#)
- AlarmOn, [20](#)
- Alarms
 - Setting the message, [18](#)
 - Turning off, [19](#)
 - Turning on, [20](#)
- AlphaNumericBlue, [21](#)
- AlphaNumericCyan, [22](#)
- AlphaNumericGreen, [23](#)
- AlphaNumericPurple, [24](#)
- AlphaNumericRed, [25](#)
- AlphaNumericWhite, [26](#)
- AlphaNumericYellow, [27](#)
- AssignFont, [28](#)
- Attribute mode
 - Character, [32](#)
 - Line, [33](#)
 - Page, [34](#)
 - Page with wrapping, [35](#)
 - Setting default, [30](#)
- AttributeEffect, [29](#)
- AttributeMode, [30](#)
- AttributeMode Char, [32](#)
- AttributeMode Line, [33](#)
- AttributeMode Page, [34](#)
- AttributeMode Wrap, [35](#)
- AttributeOff, [36](#)
- AttributeOn, [37](#)

Attributes

- Changing the current attribute, [91](#)
- Changing the tagged attribute, [17](#), [53](#)
- Changing to a specific effect, [89](#)
- Setting the attribute mask, [93](#)
- Specify ANSI attributing, [90](#)
- Specifying single character effects, [92](#)
- Tagged attribute appearance, [174](#)
- Turning off the tagged attribute, [175](#)
- Turning on the tagged attribute, [176](#)

- AutoFF, [38](#)
- Automatic form feeds, [38](#)
- AutoScroll, [39](#)
- AutoScrollOff, [40](#)
- AutoScrollOn, [41](#)
- AutoWrap, [42](#)
- AutoWrapOff, [43](#)
- AutoWrapOn, [44](#)

B

- Backslash format strings, [14](#)
- BackSpace, [45](#)
- BackTab, [46](#)
- Base emulation mode, [99](#)
- Bell, [47](#)
- BlackBackground, [48](#)

C

- ChangeColorOff, [50](#)
 - ChangeColorOn, [51](#)
 - Character set
 - Assigning font bank, [28](#)
 - Displaying a character for a sequence, [77](#)
 - Host to PC mapping tables, [126](#)
 - Turning off graphics characters, [108](#)
 - Turning on graphics characters, [109](#)
 - CharacterMapTable, [52](#)
 - ChgTaggedAttr, [53](#)
 - ClearAllUnprotected, [55](#)
 - ClearAOL, [54](#)
 - ClearEOF, [56](#)
 - ClearEOL, [57](#)
 - ClearEOP, [58](#)
 - Clearing characters in line, [102](#)
 - Clearing the current field, [56](#)
 - Clearing the current line
 - All of the line, [54](#)
 - To the end of the line, [57](#)
 - To the start of the line, [59](#)
 - Clearing the page
 - To end of page, [58](#)
 - To start of page, [60](#)
 - Clearing the screen
 - Clear screen, [61](#)
 - Erase screen and reset attributes, [103](#)
 - Form feed, [105](#)
 - Unprotected characters, [55](#)
 - ClearScreen, [61](#)
 - ClearSOL, [59](#)
 - ClearSOP, [60](#)
 - Color Back, [62](#)
 - Color Fore, [63](#)
 - Colors
 - Setting the background color, [62](#)
 - Setting the foreground color, [63](#)
 - CR, [49](#)
 - Cursor
 - Hiding, [69](#)
 - Restoring position, [157](#)
 - Saving position, [162](#)
 - Showing, [70](#)
 - Cursor movement
 - Back one or more tabs, [46](#)
 - Back space, [45](#)
 - Beginning of following line, [146](#)
 - Beginning of previous line, [160](#)
 - Beginning of the line, [49](#)
 - Down one line, [66](#)
 - Down with scrolling, [122](#)
 - Forward one space, [106](#)
 - Left, [68](#)
 - Right, [71](#)
 - To next tab stop, [170](#)
 - Up, [72](#)
 - Up with scrolling, [159](#)
 - Cursor positioning
 - Move to top left, [67](#)
 - Cursor postioning
 - Move to column, [142](#)
 - Move to column and line, [143](#)
 - Move to line, [144](#)
 - Setting interpretation of parameters, [64](#)
 - CursorCode, [64](#)
 - CursorDown, [66](#)
 - CursorHome, [67](#)
 - CursorLeft, [68](#)
 - CursorOff, [69](#)
 - CursorOn, [70](#)
 - CursorRight, [71](#)
 - CursorUp, [72](#)
- D
- DelayNewLine, [73](#)

DeleteChar, [74](#)
DeleteLine, [75](#)
Deleting
 Characters, [74](#)
 Lines, [75](#)
DisableProtect, [76](#)
Display
 Turning off, [81](#)
 Turning on, [82](#)
Display of character zero, [80](#)
DisplayChar, [77](#)
DisplayMap, [79](#)
DisplayNulls, [80](#)
DisplayOff, [81](#)
DisplayOn, [82](#)
Double height
 Bottom line, [84](#)
 Line top, [85](#)
 Turning off, [166](#)
 Viewdata emulation, [83](#)
Double height and width
 Line bottom, [86](#)
 Line top, [87](#)
 Turning off, [166](#)
Double width line, [88](#)
DoubleHeight, [83](#)
DoubleHeightLineBottom, [84](#)
DoubleHeightLineTop, [85](#)
DoubleSizeLineBottom, [86](#)
DoubleSizeLineTop, [87](#)
DoubleWidthLine, [88](#)

E

Effect, [89](#)
Effect All, [91](#)
Effect ANSI, [90](#)

Effect CharacterRange, [92](#)
Effect Mask, [93](#)
Effect Order, [94](#)
EffectExtent, [95](#)
EffectIgnoreBold, [96](#)
EffectIgnoreReverse, [97](#)
Effects
 Changing the current effect, [91](#)
 Changing the tagged attribute, [17](#), [53](#)
 Changing to a specific effect, [89](#)
 Setting the effect mask, [93](#)
 Specify ANSI effects, [90](#)
 Specifying single character effects, [92](#)
 Tagged attribute appearance, [174](#)
 Turning off the tagged attribute, [175](#)
 Turning on the tagged attribute, [176](#)
EmbeddedAttributes, [98](#)
EmulationMode, [99](#)
EnableProtect, [101](#)
EraseCharsInLine, [102](#)
EraseScreen, [103](#)
EraseUsingWriteAttr, [104](#)

F

FF, [105](#)
Forcing form feeds, [38](#)
FormFeed, [105](#)
ForwardSpace, [106](#)

G

GraphicMapTable, [107](#)
GraphicModeOff, [108](#)
GraphicModeOn, [109](#)

H

HardReset, [110](#)

Host sequence parsing commands, [15](#)

I

Ignore, [111](#)

IgnoreErrors, [112](#)

Ignoring sequences, [111](#)

Ignoring syntax errors, [112](#)

Initial scripts, [169](#)

InsertChar, [113](#)

Inserting

 Characters, [113](#)

 Insert mode off, [115](#)

 Insert mode on, [116](#)

 Lines, [114](#)

InsertLine, [114](#)

InsertModeOff, [115](#)

InsertModeOn, [116](#)

K

Keyboard

 Locking, [119](#)

 Unlocking, [120](#)

KeyboardLock, [119](#)

KeyboardUnlock, [120](#)

Key_, [117](#)

L

Library, [121](#)

Line attributes

 Double height, [84](#), [85](#)

 Double height and width line, [86](#), [87](#)

 Double width, [88](#)

 Single height/width line, [166](#)

LineFeed, [122](#)

LockShiftGLFont, [123](#)

LockShiftGRFont, [124](#)

M

Mapping parts of a sequence, [125](#)

MapSequence, [125](#)

MapTable, [126](#)

Margin bell

 Setting at current position, [165](#)

 Setting default position, [127](#)

 Setting position, [130](#)

 Turning off, [128](#)

 Turning on, [129](#)

MarginBell, [127](#)

MarginBellOff, [128](#)

MarginBellOn, [129](#)

MarginBellPsn, [130](#)

MosaicBlue, [131](#)

MosaicContiguous, [132](#)

MosaicCyan, [133](#)

MosaicGreen, [134](#)

MosaicHold, [135](#)

MosaicPurple, [136](#)

MosaicRed, [137](#)

MosaicRelease, [138](#)

MosaicSeparated, [139](#)

MosaicWhite, [140](#)

MosaicYellow, [141](#)

MoveX, [142](#)
MoveXY, [143](#)
MoveY, [144](#)

N

NewBackground, [145](#)
NewLine, [146](#)
NL, [146](#)
NormalHeight, [147](#)
NormalVideo, [148](#)

P

PrinterIgnore, [149](#)
PrinterMap, [150](#)
Printing
 Changing a sequence, [150](#)
 Removing a sequence, [149](#)
Programming keys, [117](#)
Protect mode
 Disabling, [76](#)
 Enabling, [101](#)
ProtectEffect, [151](#)
ProtectFullIntensity, [152](#)
ProtectHalfIntensity, [153](#)
ProtectOff, [154](#), [155](#)
ProtectTab, [156](#)

R

RestoreCursor, [157](#)
Reveal, [158](#)
ReverseLF, [159](#)

ReverseNewLine, [160](#)
ReverseVideo, [161](#)

S

SaveCursor, [162](#)
Script Library, [121](#)
Scripts
 Initial scripts, [169](#)
 Invoking from a sequence, [180](#)
ScrollDown, [163](#)
Scrolling
 Enabling automatic scrolling, [41](#)
 Preventing automatic scrolling, [40](#)
 Scrolling the page down, [163](#)
 Scrolling the page up, [164](#)
 Setting default behavior, [39](#)
ScrollUp, [164](#)
Sending a response to the host, [179](#)
SetMarginBell, [165](#)
Setting base emulation mode, [99](#)
Setting up keys, [117](#)
Showing syntax errors, [112](#)
Single height/width line, [166](#)
SingleLine, [166](#)
SingleShift, [167](#)
SoftReset, [168](#)
StartCommand, [169](#)

T

Tab, [170](#)
Tab stops
 Clearing a tab stop, [171](#)
 Clearing all tab stops, [172](#)
 Moving back a tab stop, [46](#)

- Moving to the next tab, [170](#)
- Setting a tab stop, [173](#)
- TabClear, [171](#)
- TabClearAll, [172](#)
- TabSet, [173](#)
- Tagged attribute
 - Appearance, [174](#)
 - Turning off, [175](#)
 - Turning on, [176](#)
- TaggedAttr, [174](#)
- TaggedAttrOff, [175](#)
- TaggedAttrOn, [176](#)
- TekGraphic, [177](#)
- Tektonics graphics, [177](#)
- Telnet terminal type, [178](#)
- TelnetTerminal, [178](#)
- TermId, [179](#)
- Terminal type, [178](#)
- TerminalCommand, [180](#)

W

- Wrapping
 - Delaying at the end of a line, [73](#)
 - Turning off automatic wrapping, [43](#)
 - Turning on automatic wrapping, [44](#)
- Wrapping characters
 - Setting default, [42](#)

X

- Xoff, [182](#)
- Xon, [183](#)