

A TUTORIAL ON ESCAPE SEQUENCE PROGRAMMING OF T6530 TERMINAL USING TACL

INTRODUCTION:

This tutorial attempts to describe some of Escape Sequences programming of the T6530 Tandem terminal emulator enabling you to programmatically accomplish things like clearing screen, changing video attributes, writing to line 25 of terminal, etc.

ESCAPE SEQUENCES:

The 6530 terminal and devices that emulate the 6530 terminal recognize a set of escape characters that allow you to control cursor position, set the size and video attributes of characters displayed on the screen, etc. Most escape sequences have the following format: escape-character number [number]

The escape character is the 27th character in the ASCII character set. To specify an escape sequence, store binary values as their decimal equivalents in a TACL STRUCT or DELTA variable. The easiest way to send these values to your terminal is to use #OUTPUT. To specify an escape sequence in a STRUCT variable, set a BYTE value to the decimal value of the escape character and the numbers that perform the desired operation and use REDEFINES on these two variables. To define an escape sequence using #DELTA, specify a decimal number followed by the I command; for information about #DELTA, see the TACL Reference Manual in TIM.

An escape sequence consists of a string of ASCII characters, starting with the Esc (1BH) control character and followed by one or more graphic characters that comprise the escape code and any additional parameters. When the 6530 receives a recognized escape sequence, it performs the function specified. Unrecognized escape sequences cause the 6530 to issue a command error.

Your application program can send escape sequences to the 6530 by sending the ASCII code for the Esc character (1BH) followed by the appropriate graphic character(s). In conversational mode, the user can generate escape sequences from the keyboard by pressing the Esc key followed by one or more alphanumeric keys. (This does not work in Outside View terminal emulator due to the ESC key being mapped to F16, the Tandem help key).

Escape sequences can be used to alter the video attributes of the terminal, among other things. Video attributes can be used in any of the operating modes(either CONVERSATIONAL and BLOCK). They specify how characters appear on the screen. The video attributes available with the 6530 are:

- * Blinking or nonblinking
- * Normal or alternate intensity
- * Normal or reverse video
- * Normal or underscored
- * Normal or invisible (not displayed)

The following macro demonstrates the above video attributes of T6530 terminal using TACL and ESCAPE sequences.

EXAMPLE MACRO

```
=====
?SECTION display MACRO
#FRAME
== Define escape sequences
[#DEF ascii STRUCT
BEGIN
BYTE byt0 VALUE 7;
CHAR bell REDEFINES byt0;

BYTE byt1 VALUE 27; ==ascii value of ESCape character
CHAR esc REDEFINES byt1;

BYTE byt2a VALUE 36; ==ascii value of $
CHAR dollar REDEFINES byt2a;

BYTE byt2 VALUE 37; ==ascii value of %
CHAR perc REDEFINES byt2;

BYTE byt3 VALUE 38; ==ascii value of &
CHAR amp REDEFINES byt3;

BYTE byt4 VALUE 64; ==ascii value of @
CHAR at REDEFINES byt4;

BYTE byt5 (0:1) VALUE 27 73; == clear screen
CHAR clr (0:1) REDEFINES byt5; == escape sequence
END;
] == End ascii

== Clear the screen
#OUTPUT [ascii:clr(0:1)]
#OUTPUT The screen was just cleared.

== Display text with special video attributes
#OUTPUT Blinking text: [ascii:esc]6b These words are &
blinking[ascii:esc]6[ascii:at]

#OUTPUT Inverted text: [ascii:esc]6[ascii:dollar]These words&
are inverted [ascii:esc]6[ascii:at]

#OUTPUT Inverted text: [ascii:esc]6[ascii:perc]These words &
are inverted and dim [ascii:esc]6[ascii:at]

#OUTPUT [ascii:esc]o[ascii:esc]6[ascii:amp]These words are &
inverted blinking in line 25[ascii:esc]6[ascii:at]

#OUTPUT Here is the bell...[ascii:bell]
#OUTPUT And this is normal text.
#UNFRAME
=====
```

Note: To run this macro, load the file and type DISPLAY on the TACL prompt. The macro displays a series of lines with different display attributes.

The first #OUTPUT command clears the screen. This is done by setting an array with the values of the ESC character as the first character and 73 (for clearing screen) as the second character. (73 is the ASCII value of ' I ' character which is the ESCAPE code for clearing screen - see towards the bottom of this tutorial for a comprehensive list of ESCAPE codes on Tandem) and then sending the array to the terminal.

The above macro uses STRUCT variable to store escape sequence. The same can be accomplished by using a DELTA variable as follows:

```
?SECTION cls TEXT
#PUSH x
#SET/TYPE DELTA/x 0J 27i
#OUTPUT [#DELTA/COMMANDS x/I]
#POP x
```

The remaining #OUTPUT commands too work in the similar way. For example, to display inverted text, the fourth #OUTPUT command first outputs the escape sequence for displaying inverted text, then the text to be displayed in inverted mode, and then the escape sequence corresponding to normal video (ESC6@).

Writing to the status line (i.e. LINE 25) of the terminal screen also requires the use of ESCAPE sequences.

STRUCTURE OF THE STATUS LINE (LINE 25 OF THE SCREEN)

The bottom row of the screen contains the message/status line, which has the following format:

```
Column
1-- 2----- 66----- 80
b   Message Area           b   Status Area
```

Columns 1 and 66 always contain a blank space.

The message area occupies columns 2 through 65. It can contain any character string sent through an escape sequence from either your application program or the keyboard in conversational mode. Text in the message area remains visible until it is cleared or reset by another escape sequence.

The status area, occupying columns 67 through 80, contains terminal status information such as the current mode of operation (CONV or BLOCK). Only the 6530 can write into this area; your application cannot address this area. The user can enable and disable the display of the status area by pressing the Ctrl-PageUp and Ctrl-PageDown keys, respectively.

SENDING ESCAPE SEQUENCES TO THE TERMINAL NON-PROGRAMMATICALLY:

To send ESCAPE sequence to the terminal, first send the ESCAPE character (HEX 1B) by pressing the ESC key on the keyboard and then send the ESCAPE code (and optionally other required parameters). For a comprehensive list of the ESCAPE codes see towards the end of this tutorial.

If you are using Outside View:

In Outside View, by default, the ESC key is mapped to F16, the Tandem help key. So it is not possible to send the ESCAPE character to the terminal using the keyboard's ESC key.

Instead, the escape sequence can be sent to the terminal by CTRL+[key combination (equivalent to ESC keypress on other terminal emulators) followed by the escape code (and other parameters).

E.g. (1) You can dim the video intensity by the following key combination CTRL+[followed by 6 and !

You can revert to normal video by the following key combination CTRL+[followed by 6 and @

(2) You can lock the keyboard using the following key combination CTRL+[followed by c

(3) You can write to the status line (line 25 of the terminal) using the following key combination CTRL+[followed by o followed by "the stuff you want to put on line 25"

Command looks like this \$N18GB1 ANAND 4> ^[oHave a nice time!

Remember that the string ^[is generated by pressing CTRL+[keys together (not by pressing ^ and then [keys)

LIST OF THE ESCAPE SEQUENCES:

```
esc 6 @ Normal video
esc 6 ! Dim intensity
esc 6 $ Reverse video
esc 6 " Blinking video
esc 6 ( Blank (invisible)
esc 6 0 Underscored
esc 6 % Reverse dim
esc 6 & Reverse blink
esc 6 , Reverse blank
esc 6 1 Underscored dim
esc 6 2 Underscored italics
esc 6 4 Underscored reverse
esc 6 5 Underscored reverse dim
esc 6 6 Underscored reverse italics
esc 6 8 Underscored blank
esc 6 ? Underscored reverse blank
esc - C Set buffer address extended *
esc - D Set cursor address extended *
esc - I Clear memory to spaces extended *
esc - J Read with address extended *
```

esc - K Read with address all extended *
esc - O Write to auxiliary port
esc - P Execute self tests
esc - r Define data type table extended *
esc 0 Print page
esc 1 Set tab
esc 2 Clear tab
esc 3 Clear all tabs
esc 6 Set video attributes
esc 7 Set video prior condition
esc 8 Set line width to 40 columns
esc 9 Set line width to 80 columns
esc : Select page *
esc ; Display page
esc < Read buffer *
esc = Read with address *
esc > Reset modified data tags *
esc ? Read terminal configuration
esc @ Delay one second
esc A Cursor up
esc C Cursor right
esc F Cursor home down
esc H Cursor home
esc I Clear memory to spaces
esc J Erase to end of page/memory
esc K Erase to end of line/field
esc L Insert line *
esc M Delete line *
esc N Disable local line editing *
esc O Insert character *
esc P Delete character *
esc S Roll up **
esc T Roll down **
esc U Next page **
esc V Previous page **
esc W Enter protect submode *
esc X Exit protect submode *
esc [Start field extended *
esc] Read with address all *
esc ^ Read terminal status
esc _ Read firmware revision level
esc a Read cursor address
esc b Unlock keyboard
esc c Lock keyboard
esc d Simulate function key
esc f Disconnect modem
esc i Back tab *

esc o Write to message area
esc p Set max page number *
esc q Reinitialize *
esc r Define data type table *
esc t Set screen width to 40 columns
esc u Define enter key function **
esc v Set terminal configuration
esc x Set I/O device configuration
esc y Read I/O device configuration
esc z Execute power-on self test

*- BLOCK mode only

** - CONV mode only

For detailed information about these ESCAPE sequences and the programming of T6530 terminal emulator, please get hold of the 6530 programmer's guide.

This tutorial is compiled by Anand Sharma. For suggestions/comments please contact me using GoAnand.com.