

Mark Crispin
SU-AI
7 October 1977

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Bit name	Value	Meaning
%TPCBS	0,,40	this terminal is using the "intelligent terminal protocol". THIS BIT MUST BE ON.
%TPORS	0,,10	the server should process output resets instead of ignoring them. IT IS HIGHLY RECOMMENDED THAT THIS BIT BE ON; OTHERWISE THERE MAY BE LARGE DELAYS IN ABORTING OUTPUT.

Bit name	Value	Meaning
%TOCLC	100000,,0	convert lower-case input to upper case. Many terminals have a "shift lock" key which makes this option useless. NORMALLY OFF.
%TOSA1	2000,,0	characters 001-037 should be displayed using the Stanford/ITS extended ASCII graphics character set instead of uparrow followed by 100+character. NORMALLY OFF.
%TOMOR	200,,0	the system should provide "***MORE**" processing when the cursor reaches the bottom line of the screen. ***MORE** processing is described in ITS TTY. NORMALLY ON.

```
%TOROL      100,,0      the terminal should scroll when attempting
                        output below the bottom line of the screen
                        instead of wrapping around to the top.
                        NORMALLY OFF.
```

[illegible]

[illegible]

[illegible]

BUCKY BITS

Name	Value	Description
%TXTOP	4000	This character has the [TOP] key depressed.
%TXSFL	2000	Reserved, must be zero.
%TXSFT	1000	Reserved, must be zero.
%TXMTA	400	This character has the [META] key depressed.
%TXCTL	200	This character has the [CONTROL] key depressed.
%TXASC	177	The ASCII portion of the character

is sent as three bytes. The first byte is always 034 octal (that is why 034 must be quoted). The next byte contains the "bucky bits", ie, the %TXTOP through %TXCTL bits, shifted over 7 bits (ie, %TXTOP becomes 20) with the 100 bit on. The third byte contains the %TXASC part of the character. Hence the character <CONTROL><META>[LINE FEED] is sent as 034 103 012.

The intelligent terminal protocol also is involved when a network interrupt (INR/INS) is received by the user program. The user program should increment a count of received network interrupts when this happens. It should not do any output, and if possible abort any output in progress, if this count is greater than zero (NOTE: the program MUST allow for the count to go less than zero).

Since the server no longer knows where the cursor is, it suspends all output until the user informs it of the cursor position. This also gives the server an idea of how much was thrown out in case it has to have some of the aborted output displayed at a later time. The user program does this when it receives a %TDORS from the server. When this happens it should decrement the "number of received network interrupts" count described in the previous paragraph and then send 034 followed by 020, the vertical position, and the horizontal position of where the cursor currently is located on the user's screen.

[illegible]

[illegible]

[illegible]

%TD code	Value	Meaning
%TDDL	204	Clear the character position the cursor is on. The cursor does not move.
%TDCRL	207	If the cursor is not on the bottom line of the screen, move cursor to the beginning of the next line and clear that line. If the cursor is at the bottom line, scroll up.
%TDNOP	210	No-op; should be ignored.
%TDORS	214	Output reset. This code serves as a data mark for aborting output much as IAC DM does in the ordinary TELNET protocol.
%TDQOT	215	Quotes the following character. This is used when sending 8-bit codes which are not %TD codes, for instance when loading programs into an intelligent terminal. The following character should be passed through intact to the terminal.
%TDFS	216	Non-destructive forward space. The cursor moves right one position; this code will not be sent at the end of a line.
%TDMV0	217	General cursor position code. Followed by two bytes; the new vertical and horizontal positions.
%TDCLR	220	Erase the screen. Home the cursor to the top left hand corner of the screen.
%TDBEL	221	Generate an audio tone, bell, whatever.
%TDILP	223	Insert blank lines at the cursor; followed by a byte containing a count of the number of blank lines to insert. The cursor is unmoved. The line the cursor is on and all lines below it move down; lines moved off the bottom of the screen are lost.
%TDDL	224	Delete lines at the cursor; followed by a count. The cursor is unmoved. The first line deleted is the one the cursor is on. Lines below those deleted move up. Newly-created lines at the bottom of the screen are blank.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

\000\000\000

[illegible]