

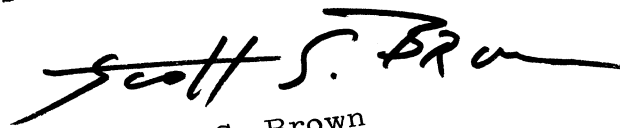
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

Here it is. Feel free to make any corrections you like.

CompSys has \$ 1,342 remaining as of whenever the accounting runs were done this weekend.

Code converter is in ~~add~~ ~~ml~~ ~~u~~ ~~ssb~~ ~~codes~~
I don't think that anyone is using it. Dave Clark may want to put it on tape for the summer unless someone wants to ~~install~~ ^{SUBMIT} it. Ken Pogran has documentation.

I will be available at KI7-3233 until tomorrow and at
30 Woodland Road
Longmeadow
Massachusetts 01106
thereafter.



Scott S. Brown



Errors and Inadequacies in the Teletypewriter Device Interface Module

Scott S. Brown

When input includes escapes:

1. It is impossible to edit an escape sequence. That is, `0139#3` results in input of 3 not `0133`.

That is the correct way - not on error!

★

2. To the canonicalizer, escapes appear to occupy only one character position. Thus,

Typist: `abcdef0133(BS)5_`
Printed line: `abcdef0133`
Input: `abcdef[`

That can be documented rather than fixed.

3. Backspaces and new line characters which are entered as escapes are canonicalized exactly if a regular backspace or new line character also appears. Thus,

Typist: `a(SP)(BS)010_`
and Typist: `a010_`

both having

Printed line: `a010_`

?

cause different strings to be input. Backspaces and new line characters entered as escapes should never be canonicalized.

why not?

When input includes horizontal tabs:

4. It is impossible to backspace before a horizontal tab. Thus,

Typist: `abcde(HT)f(BS)11#`
Printed line: `#bcde f`
Input: `abcdef`

SOMETIMES

TABS ARE A PITA IN ILLOGICAL PLACES

When the nelem argument is not large:

5. No more characters than are needed to fill the input buffer are processed. Subsequent backspacing into the range read is not seen (see also 6 and 8). Thus

`call ios_read ("user_input", readp, 0, 10, nelemt, status);`
Typist: `abcdefghij(BS)10_`
Printed line: `abcdefghij`
Input: `abcdefghij`

Next read should input (BS)10

That is the correct operation.

6. Fewer than nelem characters may be read:

`call ios_read ("user_input", readp, 0, 10, nelemt, status);`
Typist: `(SP)(BS)a#bcdefg##ijklm`
Printed line: `a#bcdefg##ijklm`
Input: `bcdefg`

Should have a mode which expands tabs to blanks.

errors in ttydim_ page two

The rule that only the first nelem characters typed are considered, however, holds only when the canonicalizer is invoked, when a backspace or a new line character appears, and if escapes are counted as one character. Thus

```
call ios_$read ("user_input", readp, 0, 10, nelem, status);
Typist:        a#bcdefg##ijklm
Printed line:  a#bcdefg##ijklm
Input:         bcdeiijklm
```

7. ~~A backspace or a new line character cannot be read alone. In:~~ *No input message can consist entirely of BACKSPACES AND NEW LINE CHARACTERS*

```
do i = 1 to 4;
call ios_$read ("user_input", readp, 0, 1, nelem, status);
end;
Typist:        a(BS)_
Printed line:  a
```

the first read call will successfully read the a. On the second read call ttydim_ will call tty_read which will read the backspace and call the canonicalizer which will canonicalize the backspace to a null string. Tty_read will then return to ttydim_ with a zero nelem. Ttydim_ now thinks that tty_read could not find any input so it causes your process to go blocked until another line is entered. When your process receives a wakeup ttydim_ will call tty_read again and the underscore will be read, the backspace having been lost.

8. After reading the required characters tty_read scans the remainder of the line for a kill character. If it locates a kill character it discards all characters typed before the kill character without regard to their position in the printed line (i.e. before canonicalization), without verifying that the kill character is not escaped, and without verifying that the dim is in error-kill mode. Thus even in iomode not erkl:

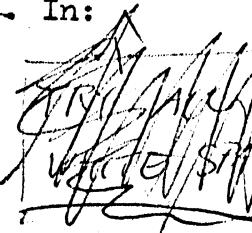
```
call ios_$read ("user_input", readp, 0, 5, nelem, status);
Typist:        abcdefg(NL)@_
Printed line:  @cdefg
Input:         -
```

When in iomode erkl:

9. The convention that an erase character which is not the only graphic in a character position erases that character position is inconvenient. An erase character should erase only characters which were typed before it was typed. Characters typed after the erase character are surely desired. Changing this convention would allow a user to backspace and change a character which he has mistyped.

But you can't tell from the printed image what was in.

NOT A PROBLEM IN MY DIM



FLX IN TTY-READ



When printing:

~~★~~
DONE

10. Tty_write will output backspace characters even if the printing element is at the first position of the line.

Done!

MAYBE 11. TRAILING WHITE SPACE IS REMOVED
BEFORE CANONICALIZATION ?

It should
be removed or part
of canonicalization,
if followed by a CR/
LF or New Line.

I ONLY PLAN TO FIX
THE [^]STARTED ITEMS

APPENDUM TO ERRORS IN TTYPIV

SCOTT BROWN

12. IF A NEW LINE CHARACTER APPEARS IN A STRING FOR WHICH THE CANONICALIZER IS INVOKED THEN THE NEW LINE IS TREATED AS A CARRIAGE RETURN AND A NEW LINE IS APPENDED TO THE MESSAGE
EXAMPLE

TYPIST: ~~ABC~~ (SP)(BS) abc 012 ---

PRINTED LINE: abc 012 ---

INPUT: ~~ABC~~
[abc]

ALTERNATIVELY

TYPIST: abc 012 ---

PRINTED LINE: abc 012 ---

INPUT: [abc]
[---]

Should
produce CONVERSION
never, but it
doesn't matter which.

It would be a valuable service if you could document the entire canonicalization, including the tradeoffs & design decisions in it.