TO: MSPM Distribution
FROM: N. I. Morris
SUBJECT: BE.11.00
DATE: 12/21/66

"Typer" has been corrected to accept the standard string specifiers. The format of these specifiers is described on Page 2 of this memo.

In addition, the calling sequence to "typer" has been modified to take a string size. If the size is not given, typer will get it from the string dope.

Note that "typer" expects the offset part of the string dope to be zero. Hence, "typer" will not work with strings which are structure elements.

Published: 12/21/66
(Supercedes: BE.11.00, 4/15/66)

Identification

On-line Character Stream Input/Output for Interaction with Simulated 645 Programs.
D. J. Slosberg, N. I. Morris

Purpose

The 645 segment typer, along with 635 escape code, enables a simulated 645 program to interact with the user via the on-line operators' console.

Method

The user program executes a standard call to one of two entries in typer. Typer then escapes to 635 code which uses GEINOS to read or write the typewriter on the operators' console. Escape code number 2 is used for this purpose.

Input

To read a character string from the typewriter in EPL:

```
lth = typer$rdtype  (string,-n-);
```

where,

"string" is the name of a non-varying EPL character string. The typed string (in ASCII) will be returned in "string" with no canonical conversion.

"n" is the maximum number of characters in the returned string (i.e. the size of the string). If "n" is omitted, the size will be picked up from the string specifier. "lth" is the number of characters actually read into "string".

The request for input is indicated on the console by the message, "TYPE-". The user then types his message and presses the "end of message" button. The "operator input error" button is used as a kill character. There is no erase character.

If the typed line exceeds the length of the character string, "string", or rdtypes's buffers (see restrictions) the user will be asked to retype the line.

Output

To write a character string on the typewriter in EPL:

```
call typer$wrtype (string,-n-);
```

where,

"string" is the name of a non-varying EPL character to be typed.

"n" is the number of characters to be typed. If "n" is ommitted, the number will be the size of the string as determined from the specifier.

Assembly language calls

To call typer from within an EPLBSA program for input:

```
call <typer> | [rdtype] (arglist)
```

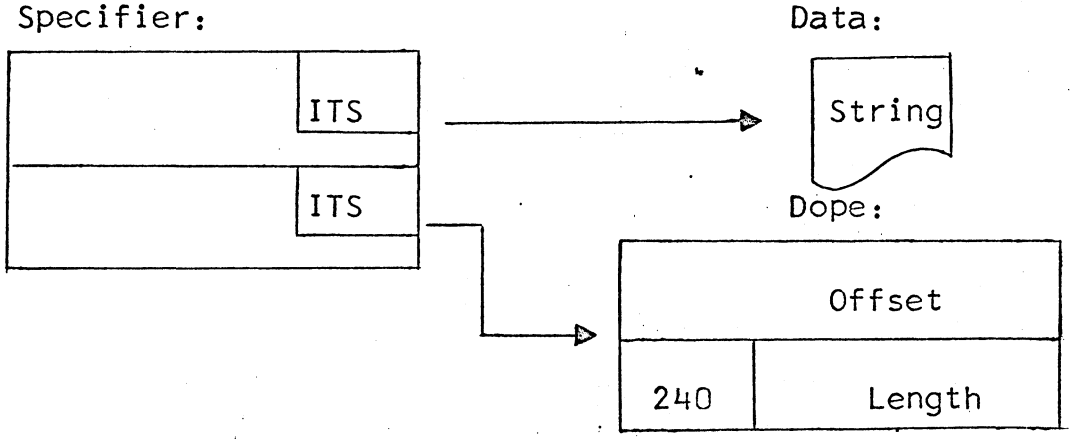
for output:

```
call <typer> | [wrtype] (arglist)
```

where arglist is a multics standard argument list.

Character String Specifiers

A character string specifier consists of 2 ITS pairs. The first points to the data origin of the actual Character string. The second points to the dope vector for the character string. The dope vector consists of two words. The first, which is usually zero, is the offset (in bits) of the origin of the string. The second word contains octal 240 in the first 9 bits and the size of the string, in bits, in the remaining 27 bits.



Note that there are "length"/9 characters in the string.

Character Set

1) The available characters are the GE-635 standard character set. The four GE characters which are not ASCII characters, are used as follows:

GE	ASCII
\	^ (software)
↑	| (vertical line)
' (apostrophe)	´ (accent acute)
←	_ (underline)

2) Octal escape - XXX is the escape for the ASCII character whose octal representation is XXX.

3) Upper case shift - U designates that all subsequent alphabetics are to be considered upper case until a L sequence is encountered.

4) Lower case shift - L designates that all subsequent alphabetics are to be considered lower case until a U sequence is encountered. This is the initial mode.

5) The ASCII characters NL and HT will be typed out on output as a carriage return and tab respectively. However, carriage returns and tabs do not appear in the character string on input.

Restrictions

In order to conserve storage, typer obtains (and returns, before returning to caller) two 64 word pages for buffers, one for the ASCII string, the other for the GE string before (or after) conversion. This places a limitation on the length of the character string. This limit is 256 ASCII characters or 384 GE characters (including escapes). For most applications, the restriction of 256 ASCII characters will apply. However, it is possible to overflow the GE character buffer, if there are many escapes.

If buffer overflow occurs on input, the user will be asked to retype the line. On output, as much of the string as possible will be typed.