

GENERAL  ELECTRIC
COMPANY

575 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS TELEPHONE 491-6300

CAMBRIDGE
INFORMATION
SYSTEMS
LABORATORY

just
✓
file
PL17

TO: F. CORBATO ✓
R. DALEY
V. VOYDOCK
R. GARDNER

FROM: R. FREIBURGHOUSE

DATE: NOVEMBER 13, 1969

The attached MPM section describes a set of simple list_directed I/O routines usable from a Fortran or PL/I procedure. The routines are de-bugged and installed in System_Library_3.



R. Freiburghouse

RF/jk

write_list_

Miscellaneous Call
Author_Maintained Library
11/11/69

Name: write_list_

This procedure can be used by PL/1 or Fortran programs to write out on the console in free formats the values of the internal variables.

The data to be written out must be of one of the four forms: real decimal fixed-point number, real decimal floating-point number, bit-string and character-string.

Usage:

```
call write_list_(arg1,arg2,...,argn)
```

Note:

The maximum number of arguments is 64.

Argi must be a scalar variable or constant.

The data-type of each argument is obtained from the descriptor, and conversion will take place to transform the value of each argument to its appropriate character string representation on the console.

Real decimal floating-point number is printed in E format , with 8 decimal places for single precision number and 19 decimal places for double precision number.

Character-string is printed without the enclosing quotes.

write_list_ uses tio_ as its interface with Multics.

Example:

```
x:  proc;
    dcl  a float, b char(5), c bit(3), d fixed;
    .
    .
    call write_list_(a,b,c,d);
    .
    call write_list_("x=", s, "m=", m);
    .
    end;
```

Console script:

~~0.17500000e-04~~ ~~name~~ ~~"101"~~ ~~b~~ ~~-5~~

~~x=~~ ~~-0.21200000e+03~~ ~~m=~~ ~~3192~~

Miscellaneous Call
Author-Maintained Library
11/11/69

Name: read_list_

This procedure can be used by PL/1 or Fortran programs to read in free-formatted data from the console.

The data read in must be of one of the four forms: real decimal fixed-point constant, real decimal floating-point constant, bit-string constant and character-string constant.

Usage:

```
call read_list_(arg1,arg2,...,argn)
```

Note:

The maximum number of arguments is 64.

Argi must be a scalar variable or a scalar bit- or character-string.

The syntax for the constants are those as defined in the Multics PL/1 Language Specification with two extensions:

1. Identifiers are recognized as character-string constants, and do not therefore need to be enclosed in quotes. An identifier begins with a letter and contains letters, numbers, under-bars, or dollar-signs.
2. Floating-point constants may be written with or without the exponent field.

The delimiters allowed in the input line are space (␣), tab (HT), comma (,), semi-colon (;), and new-line character (NL)....

Combination of the delimiters have the following meaning:

HT ≡ ␣

␣ ≡ ,

␣, ≡ ,

,␣ ≡ ,

␣␣ ≡ ␣

,, ≡ ,0,

␣NL ≡ NL

,NL ≡ ,ONL

The data-type for the input constant is recorded and compared with the expected data-type for the argument from the descriptor, and the constant is converted to the internal representation accordingly.

read_list_ may be declared in the user program as external entry, but the attributes of its parameters should not be, since the declaration will suppress the creation of descriptors.

read_list_ uses tio_ as its interface with Multics.

Just prior to reading from the console the routine will type the message "Ready for n values".

Example:

```
x: proc;  
  decl a float, b char(5), c bit(3), d fixed;  
  .  
  .  
  call read_list_(a,b,c,d);  
  .  
  .  
  end;
```

Console script:

```
Ready for 4 values  
17.5e-6,"name"/"101"b  
Ready for 1 value  
-5
```