MULTICS SYSTEM-PROGRAMMERS' MANUAL  SECTION BZ.8.06  PAGE 1
Published: 08/29/68

Identification
Debugging Aids
B. Wolman
(Note that the following are Abstracts, which should be replaced by a full description at a later time.)

DISPLAY_ARRAY

Function of Entry:
Display the information contained in an array node.
This entry of use only to the PL/I compiler.

Calling Sequence for Entry:
call display_array (p);

Declaration of Arguments:
dcl  p  ptr;  /* points at node to be displayed */

Description of Arguments:
none
DISPLAY_AT

Function of Entry:

Display the information contained in a temporary attribute block. This entry is only to the PL/I compiler.

Calling Sequence for Entry:

call display_at (p);

Declarations of Arguments:

dcl  p  ptr; /* points at node to be displayed */

Description of Arguments:

none
DISPLAY_ATTRIBUTE

Function of Entry:
Displays the information contained in a data attribute node. This segment of use only to the PL/I compiler.

Calling Sequence for Entry:
call display_attribute (p);

Declaration of Arguments:
dcl p ptr;  /* points at node to be displayed */
DISPLAY_DESC

Function of Entry:
Prints the information contained in a data descriptor node. This segment of use only to the PL/I compiler.

Calling Sequence for Entry:
call display_desc (p);

Declaration of Arguments:
dcl p ptr; /* points at node to be displayed */

Description of Arguments:
one
DISPLAY_EXP

Function of Entry:
Displays the information contained in an expression tree. This entry is of use only to the PL/I compiler.

Calling Sequence for Entry:
call display_exp (p);

Declaration of Arguments:
    dcl p ptr;    /* points at tree to be displayed */

Description of Arguments:
none
DISPLAY_SYMB

Function of Entry:

Displays the information contained in a symbol table node. This entry of use only to the PL/I compiler.

Calling Sequence for Entry:

call display_symb (p);

Declaration of Arguments:

dcl p ptr; /* points at node to be displayed */

Description of Arguments:

none
PRT

Function of Entry:

Perform conversions and message printing for debugging modules of the PL/I compiler.

Calling Sequence for Entry:

```plaintext
call prt$rel (mess, p)
call prt$bin_dec (mess, num)
call prt$bin_oct (mess, num)
call prt$bit_oct (mess, b)
call prt$statement (k)
call prt$token (mess, p)
```

Declaration of Arguments:

```plaintext
dcl mess char(*), 
p ptr.,
num fixed,
b bit(*), /* length <=36 */
k fixed;
```

Description of Arguments:

- `prt$rel` prints `mess || bin_oct (rel(p))`
- `prt$bin_dec` prints `mess || bin_dec (num)`
- `prt$bin_oct` prints `mess || bin_oct (bit(fixed(num,36),36))`
- `prt$bit_oct` prints `mess || bin_oct (b36)` where `b36` is `b` extended on left to length 36 bits.
- `prt$statement` prints the input statement currently being processed by the PL/I compiler starting at Kth token.
- `prt$token` prints `mess || string` where string is the character string stored in the token table entry pointed to by `p`. 