

Published: 11/09/67  
(Supersedes: BY.13.02, 04/18/67;  
BY.13.02, 06/05/67)

### Identification

Generate\_ptr  
D. L. Boyd

### Purpose

Given a specified external symbol and segment name, the generate\_ptr procedure returns a pointer to the offset of the symbol in the segment if it is a segment symbol, to the entry point in the segment's linkage segment if it is an entry symbol, and to the offset of the symbol in the segment's symbol segment if it is a symbol table reference. Generate\_ptr also returns the class of the symbol. Generate\_ptr returns a pointer to the first location in the segment named if a null character string is specified in place of the external symbol name.

### Usage

The call to generate\_ptr is:

```
call generate_ptr (segment, symbol, pointer, class, option);  
dcl (segment, symbol) char(*), pointer ptr,  
    class fixed bin (17), option fixed bin (17);
```

The arguments are:

- |    |         |   |
|----|---------|---|
| 1) | segment | segment name  |
| 2) | symbol  | external symbol name  |
| 3) | pointer | return argument, pointer to segment and symbol                                |
| 4) | class   | return argument, class of the symbol  |
| 5) | option  | if = 0, ignore trap before definition<br>if = 1, allow trap before definition |

Argument one, segment, is the name of the segment in which argument two, the symbol name, is to be found. Argument three, pointer, will be a pointer to the segment name

and external symbol, if found. If symbol is a null character string, pointer will be set equal to <segment>|0. Argument four, class, will contain the class code of the symbol designated by arguments one and two (see BD.7.01 for a discussion of class codes). If symbol is a null character string, class will equal zero. Argument five, option, is designed to regulate the use of a trap before definition. In this case, a trap before definition means that another procedure may be executed before the generated pointer is returned (see BD.7.01). If option equals 1, a trap before definition is allowed. If option equals 0, a trap before definition is not allowed. In the case of the null character string in place of a symbol, no traps before definition are executed. If a trap does take place, the trap pointer is turned off using the same rules as link\_fault (BD.7.01). There are no machine conditions to be passed to the called procedure and a null non-varying specifier is used instead.

Generate\_ptr makes calls to the function getsegptr in the Segment Management Module (BD.3.02). Getsegptr returns to generate\_ptr a pointer to the segment that is being referenced, if the segment has been initiated.

If generate\_ptr recognizes a fatal error, it calls seterr (BY.11.01) which writes information in <error\_out> identifying which of the following two subconditions occurred,

<u>Error Number</u>	<u>Meaning</u>
1	Segment or symbol was not a valid string specifier.
2	Symbol found contained an error.

It then signals the condition "generate\_ptr\_err" with the option forbidding return (BY.12.03). If generate\_ptr is unable to locate the segment name or the external symbol, a null pointer is returned in argument three, pointer, and no error is signalled. In this case, argument four, class, is set equal to zero if the segment was not found. Class is set equal to one if the symbol was not found in the segment. Class is set equal to two if the linkage section for the segment was not found. Class is set equal to three if the symbol section for the segment was not found.