

Published: 11/9/66

Identification

Changes to the PL/I Language
D. B. Wagner, R. M. Graham, and J. D. Harkins

Purpose

A number of language changes are necessary if PL/I is to be used for general system programming. In addition, a few PL/I restrictions are patently ridiculous and should be relaxed. These changes in the language are documented in this section.

Changes to the PL/I Language

1. FREEing a based variable allocated into an area should be allowed, in spite of the restriction on p.117 of the PL/I manual. This is necessary for the file system. (Note: This feature will be added to PL/I in the future by IBM.)
2. External procedures should take the RECURSIVE attribute by default (p.73). This is a Multics-wide standard.
3. The Shell needs some way of catching signal conditions for which no on-unit is currently set. Therefore the "standard system action" for programmer-named conditions should be a call to a library subroutine with which the Shell can communicate. A reasonable implementation might be the following call:

```
call unclaimed_signal (name,panel);
```

Where name is the name of the condition and panel is the saved machine conditions. It should be noted that the condition-name must be accessible in some manner.

4. The keywords in the source program should be written with lower case letters. Elsewhere both upper and lower-case letters should be permitted. This includes "data streams" which the object program processes as well as character-string constants, comments, and identifiers other than keywords in the source program. No upper-lower case mapping should be done, i.e., "a" and "A" are different identifiers.
5. The first procedure statement in an external procedure may have the OPTIONS attribute. The following options are defined:
 - a. executeonly
This causes the general form of the program and particularly the save and return sequences for calls to the procedure to be as specified in MSPM BD.7.03.

b. validate (check)

This option is used by sensitive procedures to cause a very careful (machine-language-coded) check of calling arguments. Before any significant amount of code is executed by the procedure (specifically, before the argument list is accessed at all) a call is to be made to the procedure check as follows:

```
call check (argp);
```

where argp is a pointer to the argument list of the procedure having this option.

c. callback (idl,...,idn)

Calls from this procedure to the specified identifiers must have information added to the argument list giving information about the arguments included, as specified in BD.7.02. If the identifier list is omitted, all calls and function references are augmented with argument information.

d. rename ((identifier 1, identifier 2),...)

This option causes identifier 1, a name built into the compiler, to be renamed to identifier 2. In EPL only the two identifiers "stat_" and "free_\$(free_)" may be renamed in this manner. However, in PL/I all identifiers should be renamable, including the names of runtime routines.

7. The data part of an external variable a whose name does not contain a "\$" is located at <stat_>|[a]. Note that if the identifier "stat_" has been renamed "mystat" by the rename option, then a will be located at <mystat>|[a]. An external entry or variable whose name does not contain a "\$" is located at <a>|[a]. An external entry or variable whose name is of the form a\$b is located at <a>|[b]. In this latter case both a and b may each be up to 31 characters long.
8. The GE 645 provides a mechanism for executing code with interrupts inhibited: When bit 28 is on in each of a sequence of instructions, no interrupts may occur while the sequence is executing (in master mode). To make it possible for PL/I-compiled programs to use this mechanism the special condition prefixes

```
nointerrupt
```

```
interrupt
```

are provided. Default is of course interrupt. In a block in which the nointerrupt prefix is in effect, all compiled

code which is executed only in that block should have bit 28 on. If the block consists only of the following, its entire execution (after the save and before the return) should be noninterruptable:

- a. assignments involving unsubscripted real scalars
 - b. IF's involving only the above
 - c. GO TO's known to be local
9. A number of built-in functions have been added. They are described in BP.0.03.
 10. The collating sequence for the data character sequence will be that of ASCII which is described in BC.2.01.
 11. The quotation mark for string constants is the double quote (") rather than the single quote (').
 12. Relative ptrs.