PR1ME Technical Update

Subject: Rev. 18 PL/I Subset G

Number: 75

Revision: 0

Date: December 1980

Applicable Hardware: All CPUs

Applicable Software: PRIMOS

Documentation Impact: The PL/I Subset G Reference Guide (IDR4031)

Abstract: PL/I G now supports the filename suffix conventions. Two new built-in functions -- TRIM and ONCODE -- and the %PAGE preprocessor command have been added. There are new compiler options controlling single-precision real numbers (-FRN) and error listing (-ERRTTY, _NOERRTTY, and -ERRLIST).

Five conditions -- FIXEDOVERFLOW, OVERFLOW, UNDERFLOW, ZERODIVIDE, and UNDEFINEDFILE -- are now supported.

Changes have been made to the OPEN TITLE options.

PL/I G now supports keyed sequential files by using MIDAS.

PTU 75

REV. 18 PL/I Subset G

FILENAME SUFFIX CONVENTIONS

The PL1G compiler now supports the filename suffix conventions. This means that a PL1G source file named "program.PL1G" will be compiled and produce a binary file named "program.BIN" (not "B_program.PL1G"), and a listing file named "program.LIST" (not "L_program.PL1G"). There is a 32-character limit on the final filenames.

All source program files not named "program.PL1G" are handled as before. That is, a PL1G source file named "program" will be compiled to yield binary and listing files named "B_program" and "L_program" respectively. Therefore, unless the user changes the names of program source files, this change will be transparent.

PROGRAM-RELATED CHANGES

● Lines in %INCLUDE files are now numbered separately. Therefore, the line numbers in listings and error messages refer to the line number in the source file. This change allows the original source file to be specified as the DBG source file, and therefore the "filename.DBG" file is no longer needed or produced.

● The %INCLUDE 'filename' filename may be up to 128 characters and may include passwords.

● The maximum size of a source line is 255 characters. (An error message will be issued if exceeded.)

● A statement may have up to 6143 tokens.

● Extralingual characters appearing in the source outside of strings and comments will now be flagged as an error.

● The STRING and SET options for I/O are now supported.

● Simple scalar promotion is now allowed; 'array = scalar' is valid to initialize the entire array. However, the array must be contiguous in memory.

TRIM Built-in Function

A new nonstandard built-in function is available.

    TRIM (character_string, bit2 [,trim_char])

TRIM removes, from the first argument, any occurrences of the character specified in the third argument. The characters may be leading and/or trailing depending upon the second argument. The first bit in the second argument implies leading and the second bit implies trailing. If the third argument is omitted, blank is the character to be removed.

Examples:

    TRIM (' ABC ', '11'B) = 'ABC'
    TRIM ('AAQRMA', '10'B, 'A') = 'QRMA'

The TRIM built-in function should be added to the list of PL1G extensions to PL/I Subset G in the Introduction and the description should be added to the Built-in Functions section.

ONCODE Built-in Function

Values returned by the ONCODE built-in function are divided into two classes according to whether or not they represent an input/output error. Values which are less than the value of the symbol "oncode_base" are input/output errors and values greater than or equal to "oncode_base" represent all other runtime errors. This symbol is defined in the file SYSCOM>ONCODES.PL1. Since the values returned by this function are subject to change, it is recommended that this file be included in the source file (%INCLUDE 'SYSCOM>ONCODES.PL1') and the symbolic keys in the file referenced instead of the numeric values themselves.

The symbols defined in this file represent all errors which are not related to input/output. Thus, these symbols should have the value of the symbol "oncode_base" added to them before they are used in calling "signl$", so that they are not confused with the input/output related errors.

Two one-dimensional arrays of character strings are also defined in this file. They contain the text of the error messages output by the default onunit handler. The array "io_oncode_message", which contains strings declared as "char(68) varying", contains the text of the input/output related error messages, and the array "oncode_message", which contains strings declared as "char(56) varying", contains the text of all the other possible error messages. To access the message corresponding to a given ONCODE value, the following constructs should be used:

```
    oncode_val = ONCODE();
    IF oncode_val > 0    oncode_val <= max_io_oncode
       THEN msg = io_oncode_message(oncode_val);
    ELSE IF oncode_val >= oncode_base
                    oncode_val < oncode_base + next_available_code
       THEN msg = oncode_message(oncode_value - oncode_base + 1);
    ELSE /* no message available */;
```

%PAGE Preprocessor Command

Listings of source programs can be arbitrarily formatted into pages by
use of the %PAGE command.   %PAGE is used to denote the final source
line to be listed on a page during printout.   Its use is   shown   below:

```
    TEST:  PROC;
       DCL (A,B,C) FIXED BIN;
                .
                .
                .
       %PAGE;
    NESTED: PROC;
```

A source listing printout of the above program will be formatted with
NESTED procedure beginning at the top of a new page.

COMPILER

The compiler now contains all the features found in Subset G as
contained in BSR X3.74, SUBSET/G, Revision 7.

FRN Option

The PL1G compiler has the -FRN (Floating point Round) option.   Use of
this option   improves   the   accuracy   of   calculations   involving
single-precision real numbers.   Such numbers are termed FLOAT BIN(23)
in PL1G.

When the -FRN option has been given, all single-precision numbers are
rounded each time they are moved from a register to main storage.   The
method of rounding is: if the last bit of the mantissa is 1, add a 1
to the second-to-last bit, then set the last bit to 0. This rounding
reduces loss of accuracy in the low-order bits when many calculations
are performed on the same number.

The -FRN option does not affect double-precision real numbers (FLOAT
BIN(47)). It causes a slight increase in execution time, and should
therefore be used only when maximum accuracy is a major consideration.

Add this information to the Compiler section.

Compiler Error - Severity Code

Four levels of errors are reported: level 1 is a warning, level 2 is an error that has been fixed, level 3 is an error that has not been fixed, and level 4 is an error that prevents continued compilation. Any error of level 3 prevents optimization and code generation if detected prior to those phases.

The error prompt ("ER!" or "Error") appears after a compilation which results in diagnostics. As a result the compiler may not be used under Rev. 17 or earlier PRIMOS.

Any binary file being created will now be deleted if errors of severity 3 or greater are detected. Therefore, a bad binary file cannot be used unintentionally.

This information should be added to COMPILER ERROR MESSAGES in the Compiler section.

Listings

The symbol table portion of the listing is now sorted alphabetically by major (level 1) variable names.

Member names are indented from their level 1 names.

New Options

-ERRTTY   List errors on the terminal (default).

-NOERRTTY   Do not list errors on terminal.

-ERRLIST   Produce an errors-only listing file.

Cross-reference Option

The cross-reference option may cause the compiler's virtual symbol space to overflow for very large programs.

DBG Interface

Use of DBG on programs in which multiple external procedures exist in a single source file is not supported. Also, the compiler does not provide %INCLUDE line numbers to DBG; therefore, all statements in %INCLUDE files are referenced as if they appeared directly on the source file line containing the %INCLUDE.

Since compilation in DEBUG mode produces extra information in the binary and SEG files, use of the -DEBUG option for large programs may require a significant amount of disk space.

Segment Usage

Compilation of PL1G programs uses segments 4004 to 4007 and 4027. In user programs, segments 4027 to 4010 are used in descending order, as the system free storage pool. This is the area in which ALLOCATE and FREE requests operate and in which some compiler-generated temporaries are allocated.

LOADING

The PL1G compiler produces V-mode and I-mode code. (I-mode code executes on Prime 450 and higher computers.) Thus, the segmented loader (SEG) must be used to load the object modules produced by the compiler. Also, the PL1G library must be loaded prior to loading the standard library. This library is named PL1GLB and is located in UFD LIB. Thus, the following commands issued to SEG's virtual loader should be used for PL1G programs (after all user modules have been loaded):

```
$LI PL1GLB
$LI
```

CONDITIONS

The following conditions are now supported in PL1G (abbreviations are underlined):

```
FIXEDOVERFLOW
OVERFLOW
UNDERFLOW
ZERODIVIDE
UNDEFINEDFILE(file)
```

These conditions should be removed from the PL1G exclusions in the Introduction. These conditions are described in PTU2600-078, Rev. 18 Subroutines.

INPUT/OUTPUT CHANGES

Revised TITLE Option Specification

The following changes are reflected in the new OPEN TITLE options:

- No implied association between -SAM/-DAM and SEQUENTIAL/DIRECT.

- An explicit -RECL keyword is available.

- The format of DIRECT files is now compatible with SEQUENTIAL files. Files may be accessed and stored in old format if -NOSIZE is specified.

* Device names must begin with @, unless opened with -DEV.

* Record sizes are now given in bytes.

* FORTRAN carriage control codes are available for PRINT files.

OPEN FILE(f) TITLE name [{-SAM|-DAM|-DEVICE}] [-APPEND]
            [[-RECL] n] [-FUNIT m] [-NOSIZE] [-CTLASA]

name	pathname or device name and must be supplied if TITLE is present. If the name begins with @, and no options are supplied, the @ is removed and the name is compared with a list of device names (TTY, MTØ, etc.). If a match is found, the file is associated with the specified device. If no match is found, default values are assumed for the other options.
-SAM|-DAM|-DEVICE	-SAM and -DAM specify the type of file to create if a non-existent disk file is being opened. If the file already exists, -SAM or -DAM are ignored. -DEVICE (-DEV) specifies that name is a device (in which case the @ is optional). If the device is not found in the device table, ERROR is signalled. Default: -SAM is assumed for SEQUENTIAL files and -DAM assumed for DIRECT files.
-APPEND	Specifies that a file opened for output will not be truncated before writing. Default: delete file before writing unless the file was already open, in which case, the file is truncated at its current position.
-RECL n	Record length (in bytes) for DIRECT files or buffer size (that is, maximum record length for other file types). Maximum is 131,062 bytes. If -RECL is omitted, n must immediately follow -SAM, -DAM, -DEVICE or -APPEND. Default: -RECL 2048.
-FUNIT m	File unit on which disk file is to be opened, or is already open. If file is already open, name is ignored. If file is already open and -APPEND is not specified, the file is truncated at its current position. Default: any available file unit will be used.
-NOSIZE	Specifies that records of a DIRECT file are to be stored in the old format (no record-length word). Default: records are stored in RDBIN/WRBIN format (record length word before data).

-CTLASA   Specifies that FORTRAN control   codes   will
                           become the first character in each line of a
                           file. The characters and their spacing effect
                           are:

                               Character   Effect

                                    +   overprinting

                                    1   advance to next page

                               blank   one line

                           This option is useful with the -FTN command
                           line option of SPOOL. This option   allows
                           SKIP(Ø) to be used for overstriking. SKIP(Ø)
                           directed to the TTY device will   cause   a
                           carriage return without a line feed.

The above options may be specified in any order but name must be the
first entry in the TITLE option. All names and options are mapped to
upper-case before processing. Therefore @tty -dev and @TTY -DEV are
equivalent. The maximum length of the TITLE option is 128 characters.

If no TITLE option is given, the following is the default:

    OPEN FILE(f) TITLE ('fname')

    fname   declared name of the file constant referenced by f, where
              f is a file variable, file-valued function or file
              constant.

The above information should be added to the Implementation-defined
Features section.

Keyed Sequential File Support (MIDAS Notes)

PL1G now supports KEYED SEQUENTIAL files by using MIDAS (Multiple Index
Data Access System).   The implementation allows single-keyed variable
or fixed-length records with character valued keys of 32 characters or
less. The reference in the KEYTO option must be a varying character
string, while the KEY and KEYFROM references may be either varying or
non-varying.

The MIDAS CREATK utility may be used to create MIDAS files for use with
PL1G, or a MIDAS file with variable length records and 32 character
keys will be automatically created at runtime if a non-existing MIDAS
file is opened for output.

The MIDAS library must NOT be included in the load sequence, only the
shared MIDAS library may be used.