

Chapter 23. PL/1-KT DIFFERENCES FROM SUBSET "G" STANDARD

23.1. Missing features

- Basic quantitative restrictions of compiler implementation are associated with the use of commands and 80x86 processors. It results in the limit for maximum possible size for the following types of data:

- a) BINARY FIXED no more than 32 bits (from FIXED (1) to FIXED (31))

- b) BIT no more than 32-bit (from BIT (1) to BIT (32))

- a) CHARACTER no more than 254 bytes (from CHAR (1) to CHAR (254))

- d) BINARY FLOAT no more than 64 bits (from FLOAT (1) to FLOAT (53))

- Due to specifics of the LINK-KT implementation the total size of EXE-file (commands + data) should not exceed 8 MB. These restrictions do not apply to the amount of data made available by the ALLOCATE operator and resources size.

- Template PICTURE is implemented as a specification output PUT EDIT only.

- STRING built-in function can be used only within the PUT and GET, STRING argument can be a variable of type CHAR or CHAR VAR only.

- Arrays and bit strings with varying length are not allowed. It is also impossible to define lengths of arrays and strings in the form of expressions. The length of an array or string can be declared as constant only. If you need to pass arrays of different lengths as parameters of procedures, you should not pass the array itself, you should pass its address (you need to base array on parameter inside of procedure) instead
dcl test entry ((1:100) fixed); call test (m);

you can do

dcl test entry (ptr, fixed); call test (addr (m), 100);

- Attribute description DEFINED is implemented with restrictions:

- base must be defined before;

- base cannot have EXT attribute;

- base cannot be an element of the array or the upper level of the structure.

Offset from the base is not specified by POSITION attribute, but direct setting in DEFINED specifier, for example: dcl x fixed def (y +100);

- Not implemented operator A = B; where A - a variable of array and B is scalar variable, except A = 0. You can compare the arrays for equality to zero.

- Attribute LIKE structure can only refer to the structure described above. Name LIKE must be completely specified.

- Third parameter of builtin SUBSTR for bit strings should be constant.

- Some built-ins from subset G (ATANH, COPY, and others) are missing.

23.2. Additional features

- The complex numbers has been added, but with restrictions: they are used only for numbers FLOAT type only, conversions between complex and real types are not allowed and you need to write complex constants in quotes.
- Built-ins ASCII and RANK are added.
- New operators LEAVE and CONTINUE.
- In the ALLOCATE statement you can specify the number of bytes allocated explicitly.
- The preprocessor operator %REPLACE allows elements enumeration by a comma.
- Using %INCLUDE, you can specify your own library.
- The keyword MAIN can be specified without OPTIONS.
- For text files input/output operations for variables of type CHARACTER VARYING with READ and WRITE is allowed.
- Operator GET EDIT allows format A.
- Using the control characters in string constants and variables is allowed.
- Allow the type description:
DCL X (10) CHAR (10) STATIC INIT ((10), '0123456789 ');
instead of the required by subset G description:
DCL X (10) CHAR (10) STATIC INIT ((10) (1) '0123456789 ');
- An opportunity to write instead DO opening bracket {, and instead of END - closing bracket} was added, for example:
IF X = 1 THEN {; Y = 2; Z = 3;}
- In the statement GO TO the word "TO" can be omitted, for example GO M;
- In the output for the operator PUT you can specify the operator STOP, for example:
PUT DATA (X, Y, STOP);
- You can set the initial value of pointers other than NULL, for example:
DCL X1 FIXED,
X2 BIT (16) BASED (P),
P PTR STATIC INIT (X1);
You can specify the elements of structures:
DCL P PTR STATIC INIT (X1.X2.X3);
- For initial value built-in DIMENSION (necessarily abbreviated - DIM), can be used, for example:
DCL X1 (1:100) FIXED,
LEN FIXED STATIC INIT (DIM (X1, 1));
- Built-ins LBOUND, HBOUND and DIMENSION with a zero value of the second parameter return the size of the type CHARACTER or CHARACTER VARYING:
DCL X1 (1:100) CHAR (45);

`Y = DIM (X1, 0) / * assigns the Y value of 45 * /`

- For built-in functions LBOUND, HBOUND and DIMENSION you can omit second parameter, then it is set to 1.
- You can set the response code for Windows in the statement STOP.
- You can specify the keyword KEY rather than KEYFROM.
- You can omit the keyword ERROR, for example, instead of the SIGNAL ERROR (150), you can write SIGNAL (150);
- You can put the single-line comments (valid until the end of the line) as well as in the language C++. Such comments begin with `"/ /"`, for example:
`X +1; // example single-line comment`
- If the second parameter of the transformation built-in CHARACTER is negative constant, it is assumed to be the length of the response string, starting from the right (not from left).
- You can use a CALL procedure-function returning FIXED BINARY or BIT.
- The keyword CALL can be omitted.
- Built-in ROUND for bit strings performs a cyclic shift.
- You can set a prefix of disposability "1" before the DO-group.
- In the assignment statement is admissible list of variables.
- After the symbolic constants, you can specify the repeat factor in parentheses, for example: `S = '- * -' (10);`
- Cycle DO REPEAT; ... END; denotes an infinite loop.
- It is possible to list a number of conditions in a single ON-statement.
- Built-in TRIM is added, which removes any space and zero code from the beginning and end of strings.
- You can set to zero all the local variables of procedure by one assignment operator, pointing to the left side the name of it procedure.
- New assignment operators for arithmetic type `X+=1; X-=1; X*= 2; X/= 2` are added;
- DIMENSION statement in the description of arrays is allowed.
- Built-in DATE4Y, returning all four digits of the current year, is added.
- Built-in REPLACE, replacing the portions of strings is added.
- Qualifier W in text constants is added to indicate the Windows encoding.

- The ability to explicitly the number of bytes specify in the statements READ and WRITE is added.
- Attribute IMPORT for Windows procedures is added.
- It is possible to write directly in the operator STOP inside of PUT EDIT.
- After the END of the cycle operator, you can write the loop variable, or words from the title REPEAT/WHILE.
- The ability to access the CPU registers directly and to write any commands in the program using UNSPEC is added.
- Russian equivalents for keywords are added by using predefined operator %REPLACE.

You can change, the cycle parameter along the loop, such as

```
DO I = 1 TO 100, J = 1 TO 100, K = 100 TO 1 BY -1; ... END;
```

23.3. Some comments on the implementation

ON operator doesn't free the memory that contains the variable, the use of which caused this ON situation. So, you cannot close the file in the exchange with which the ON-situation if it is ON ENDFILE.

The implementation process of recursive procedures is as follows:

- At entering into the stack we copy to it static frame containing the address of any object in the class memory AUTOMATIC;
- On finish recursive procedure restores all the data from the previous instance of the stack back into the static frame.

If a recursive procedure calls routines and calls to objects in memory class AUTOMATIC, in fact, work is being done with the addresses of the static frame. If the routine is its own recursive procedure, there may be unpredicted results. In addition, the result will be unpredictable if the parameters recursive procedure to access external variables by value. These parameters must be enclosed in parentheses.

To convert DECIMAL to FLOAT DECIMAL, intermediate conversion to CHAR and CHAR in FLOAT is used. Thus, the expression $X = 1/3$, where X - FLOAT will be slower than $X = 1/3e0$;

The compiler checks whether the end of the procedure-function body can be formally reached (without the use of operator RETURN) and warns if so.

When you exit the block bounded by BEGIN; and END; settled in this unit ON-operators are not canceled.

When setting procedure parameters by constants, no temporary variables for these constants are created within the procedure. However, this can be done by enclosing the constant parameters in parentheses.

Conditional statements are not supported for variables of type BINARY FIXED, if their addition or subtraction gives intermediate results more than the absolute value of 2147483647

Memory for ON unit organized as a 32 word stack. This stack is the same for all units. Thus, at any point in the program is not possible presence of more than 32 open ON units, or you will get a message: CONDITIONAL STACK OVERFLOW.

As a compilation result we obtain a relocatable object code in Intel format OMF 8086/8087, in this format, the names of external procedures can be up to 40 bytes.

Names of external procedures and data may include Russian letters, but in this case you should only use LINK-KT, other linkers may give an error.

CHAPTER 24. DIFFERENCES FROM PL/1-KT IBM OS PL/I, and full PL/I (ANSI X3.53)

1. Several procedures, BEGIN blocks and DO groups cannot end with a single END.
2. Procedures cannot have multiple entry points.
3. Function must return a scalar value.
4. Character strings in the description RETURNS must be an integer constant.
5. BEGIN block cannot be an attribute OPTIONS.
6. Character set (small, Russian letters and pseudo graphic) differs from the full set of ANSI.
7. Comments should not contain a combination of characters /* */ (unless the key compilation H is set).
8. Prefixes enabling/disabling exception handling is not allowed.
9. Binary constants are not allowed, repetition factor can use for symbolic constants only and it is written after them (in ANSI PL/I – before symbolic constants).
10. Do not use undeclared variables.
11. Length of string variables and structures in the declaration must be an integer constant, the exception - the string length of the formal parameter (here * is allowed)
12. The lower and upper bounds must be integer constants.
13. There should not be indexed labels before the keywords PROCEDURE and FORMAT, any label before DECLARE, THEN and ELSE not allowed.
14. Array, covering not continuous piece of memory (due to the fact that it is part of the structure) cannot be an argument of the procedure.
15. Expressions in the IF and DO WHILE must be bit strings of length 1.

16. You cannot use REURN in order to exit from the ON-operator.
17. Following statements are not supported:
DISPLAY, EVENT, EXIT, FLOW, HALT, ENTER, NOFLOW, ORDER, OTHERWISE, PRIORITY, REENRANT, RELEASE, REORDER, REPLY, RETCODE, SELECT, SYSTEM, RASK, TRKOFL, UNLOCK, WAIT, WHEN.
18. Storage class CONTROLLED is not supported.
19. In the ALLOCATE statement IN option is not supported.
20. Attribute CONNECTED is not supported.
21. Assignment by operator BY NAME is not allowed.
22. COPY attribute is not supported in statements PUT and GET, DATA attribute does not output array indices, GET DATA does not allow semicolon and arbitrary data order.
23. In the statement PUT options ALL, FLOW and SNAP are not allowed.
24. Format list cannot contain variables and expressions.
25. In the statement READ options LOCATE and IGNORE are not supported.
26. Statement REWRITE cannot contain option FROM.
27. Unaligned bit strings (regardless of attributes and ALIGNED UNALIGNED row always have type ALIGNED) are not supported.
28. The following file attributes are not supported:
ADDBUFF, BUFFERED, BUFND, BUFNI, BUFOFF, BUFSP, CTL360, D, DB, FETCH, FS, GENKEY, INDEXAREA, INDEXED, KEYLENGTH, KEYLOC, LEAVE, NCP, NOLOCK, NOWRITE, PASSWORD, REREAD, REUSE, SCALARVARYING, SIS, SKIP, SYSIN, SYSPRINT, TOTAL, TP (M / R), VS, VSAM.
29. DEFAULT statement is not supported.
30. The following attributes are implicitly supported, but cannot be set explicitly:
CONSTANT, MEMBER, NONVARYING, PRECISION, REAL, STRUCTURE.
31. The following attributes are not supported:
AREA, BIT VARYING, CONDITION, CONTROLLED, GENERIC, LOCAL, OFFSET, POSITION (instead POSITION constant with the "+" is used).
32. For class variables DEFINED attribute ISUB cannot be used.
33. In the operator DECLARE option REFER cannot be used.
34. Variables initialization is only valid for a storage class STATIC.
35. Repetition factor in the operator INITIAL must be an integer constant.

36. INITIAL values can be arithmetic or string constants, as well as built-in functions NULL and DIM. For pointers (POINTER), names of variables and elements of structures can be set.

37. FIXED DECIMAL attribute can have a non-zero scale factor, but not a negative scale factor.

38. The following attributes of procedures are not supported:
ARG, ASSEMBLER, BACKWARD, CALL, COBOL, EXCLUSIVE, FORTRAN, IRREDUCIBLE, NOMAP, NOMAPIN, NOMAPOUT, REDUCIBLE, TRANSIENT, U, WHEN.

39. Templates in output cannot have formats A, E, I, K, R, T, X, Y.

40. The MIN and MAX should have only two arguments.

41. Built-in functions cannot return non-scalar result.

42. The following built-in functions and variables are not supported:
ADD, AFTER, ALL, ALLOCATION, ANY, ATANH, BEFORE, COMPILETIME, COMPLETION, COMPLEX, CONJ, COPY, COUNT, COUNTER, CURRENTSTORAGE, DATAFIELD, DECAT, DOT, EMPTY, ERF, ERFC, EVERY, HIGH, IMAG, LOW, MULTIPLY, OFFSET, ONCHAR, ONCOUNT, ONFIELD, ONLOC, ONSOURCE, PARMSET, PLIRETV, POINTER, POLY, PRECISION, PRIORITY, PROD, REAL, REPEAT, REVERSE, SAMEKEY, SOME, STATUS, STORAGE, SUBTRACT, SUM, VALID.

43. The following pseudo variables are not supported:
COMPLETION, IMAG, ONCHAR, PRIORITY, REAL, STATUS.

44. Operators must return a scalar result.

45. Expression on arrays are not allowed, but you can assign one array to another, compare them with each other ("equal" and "not equal"), as well as reset an array to zero using assignment operator and compare an array with zero.

46. The division operator "/" is not valid for variables FIXED BINARY without losing remainder.

47. Array indices in the expression must be an integer.

48. The procedure can only contain one argument list.

49. Arithmetic operators and built-in functions must have arithmetic operands.

50. The relational operators require two arithmetic operands, or operands of the same type.