EARLY NPL SUBSET

R. Morris
M. D. McIlroy
May 3, 1965

The facilities of early NPL will be a proper subset of the facilities of NPL and this document is a preliminary description of which facilities will be excluded. The order of topics follows the order of IBM Technical Report 320-0908.

Language Character Set

is the NPL set with these substitutions

↑ for |
\ for ¬
← for _

\N

and with @ deleted. Lower case letters will be mapped onto upper case. The data character set will be accepted in character constants.

Data Character Set

will be the standard 9-bit code whatever that turns out to be.

Identifiers

will be limited to 63 (31 in PL/C dialect) characters in length. The characters # and $ will not be allowed in identifiers.

Data Types

All four data types will be permitted.

Arithmetic Data

The only arithmetic data allowed will be decimal and binary integers and decimal and binary floating-point numbers.

- there will be no PICTURE attribute.
- no complex mode.

- no scaled fixed-point; integers only.

Fixed binary data will have default precision (17) bits and maximum precision 63 bits.

Fixed decimal data will be represented as binary integers and will have corresponding default and maximum precision.

Floating-point data will have maximum precision 63 bits and default precision 27 bits. This corresponds to decimal precision of about 18 digits and 8 digits, respectively.

Character String Data

fixed or variable-length as in NPL. Represented by packed 9-bit bytes. Default length 1, maximum 4096.

Bit String Data

fixed or variable-length as in NPL. Data are packed. Default length 1, maximum 1024*36.

Statement Label Data

as in NPL.

Constants

The following constants are allowed and these only
- binary and decimal integer constants
- binary and decimal floating-point constants
- character string and bit string constants without repetition fields
- statement label constants.

There will be no sterling or imaginary constants and no scaled fixed-point.

Arrays

Subscripting will be as in NPL; subscripts must be in the range ±(2**17-1). Cross sections, adjustable dimensions, HBOUND, and LBOUND are included as in NPL.

Structures

Full qualification of names is required and subscripts may not be moved right. Structures of constants and fixed

length strings are packed, others are unpacked. Cross sections may or may not be implemented for arrays of structures.

LIKE is not implemented.

STRING generic function is provided.

Type Conversion

Conversion is not allowed between character string and any other data type.

UNSPEC may not appear as an operand. Values of UNSPEC are the machine bit configurations that would result from its argument. UNSPEC in an argument position will pass bit strings that coincide with the argument of UNSPEC. These strings will be fixed length unless the argument is varying length.

Type conversion functions, FIXED, FLOAT, BIT, and CHAR on two arguments will be provided.

Expressions

Structure expressions will not be implemented and array expressions may or may not be implemented.

The generics MAX, MIN, ABS, MOD and SIGN will be provided. It may be required that all arguments have the same type. No argument may be character.

No elementary function generics will be provided.

INDEX, LENGTH, SUBSTR and STRING will be provided.

The sixteen BF logicals and REPEAT will not be provided.

Statements

These are in	These are out	These depend on the operating system.
null	DISPLAY	OPEN
assignment	RESTORE	CLOSE
CALL	SAVE	GET
END	WAIT	PUT
DO	EXIT	SPACE
GO TO	BEGIN	GROUP
IF	STOP	SKIP
ON	REPOSITION	
RETURN	TAB	

REVERT	PAGE
SIGNAL	LAYOUT
ALLOCATE	SEARCH
FREE	SORT
ENTRY	DELAY
PROCEDURE	FETCH
READ	DELETE
WRITE	
POSITION	

Program Form

Declaration must precede use.

Implicit statement must precede use of any implicitly declared variables, unless default case is assumed.

Initial values of label arrays will only be declarable via the INITIAL attribute.

Multiple closure on END's may or may not be implemented. However END labels will be checked. Only one label on PROC or DO.

Although nested procedures with internal declarations will be permitted, redeclaration of the same (qualified) name may or may not be permitted.

Storage Classes

STATIC and AUTOMATIC will be implemented. CONTROLLED will be implemented, but no pushdown will be provided. Dimensions of controlled data must be given explicitly in the ALLOCATE statement.

The asterisk notation for dimension declarations, and adjustable dimensions will be implemented.

Procedures

GENERIC attribute will not be implemented. ENTRY attribute will be implemented. Structure arguments will be unchecked despite ENTRY information.

All procedures will be assumed ABNORMAL. ABNORMAL, NORMAL, USES, SETS will be permissible attributes but will be ignored.

MAIN will not be implemented. RECURSIVE and REENTRANT will be legal but unnecessary since that's the only kind of code we'll produce.

Storage Scopes

AUTOMATIC data will go on the stack, along with subroutine temporary.

STATIC will go in the own data segment, unless it is headed external.

EXTERNAL data must be headed. The header will be the segment name. Initial value declarations for headed external (external segments) must appear all in one compilation. Segment membership of an external procedure block will be specified outside of the language. A CONTROLLED variable must be EXTERNAL and in a unique segment.

Assignment Statement

No multiple assignments.

Only the pseudovariables UNSPEC and SUBSTR will be implemented.

Structure assignment will not be implemented and array assignment will be implemented if and only if array expressions are. The only legal label assignment statements are

label variable = label variable
label variable = label constant

where the label constant is declared in the block to which the assignment is internal.

GO TO Statement

GO TO can only go to a label variable, or to a label internal to the block in which the GO TO appears.

DO Statement

This will be the IBM version current as of the date we have to decide it. Subscripted induction variables will be forbidden.

RETURN Statement

The form RETURN(expression) may require a simple variable argument. Type declarations on the entry points may or may not be ignored.

Attributes

These are in	These are out
FLOAT	COMPLEX
FIXED	PICTURE
REAL	SECONDARY
BIT	BUILTIN
CHAR	GENERIC
VAR	DEFINED
BINARY	SYMBOL
DECIMAL	NOSYMBOL
dimension	LIKE
LABEL	MAIN
SETS - ignored	PACKED
USES - ignored	ALIGNED
ABNORMAL - ignored	
NORMAL - ignored	
ENTRY	
INTERNAL	
EXTERNAL	
AUTOMATIC	
CONTROLLED	
STATIC	
INITIAL	
RECURSIVE - ignored	
REENTRANT - ignored	

The file attributes will be put in as it becomes clear which ones are appropriate.

The range option on LABEL attributes will be ignored.

INITIAL CALL will not be implemented. Initial values of multidimensional arrays and of structures will not be implemented.

Declare Statement

Factoring will be done to one level. Factored level numbers may not be implemented. Conflicting factored attributes will not be resolved.

Asynchronous facility is out

Macros are out

Modes of I-O Data Transmission

List directed is in.

Format directed is in, IOH capability to be defined.

Data directed is out.

Data List Specification

Perhaps we will leave repetitive specs out and put in the HOLD and CROSS options. A data specification would only have one level of parentheses.

READ and WRITE

These options are in	These are out
FILE	PRINT
STRING	SEGMENT
data	CALL
HOLD	TASK
CROSS	ZERO
	FROM

The KEY option depends on specifications yet to come.

File Positioning

Rewinding should be the duty of OPEN. Just what part of OPEN we want should be up to our I-O experts.

Positioning of the input-output scan pointer on a record can be done by the POSITION statement.

Record spacing and line skipping is handled by SPACE and SKIP. In NPL, IBM doesn't expect to be able to move records backwards. Until further notice, both are out.

Subroutines for I-O

The subroutines required for I-O should be specified at an early time. We should attempt to make as many of them programmable in NPL as possible. Then presumably I-O statements will simply expand into calls.

Built-in Functions

These are provided	These are not
MAX arith args	FLOOR
MIN arith args	CEIL
MOD integers	TRUNC
ABS	BINARY

SIGN	DECIMAL
FIXED 2 args	PRECISION
FLOAT 2 args	ADD
BIT 2 args	MULTIPLY
CHAR 2 args	DIVIDE
SUBSTR 3 args	COMPLEX
INDEX	REAL
LENGTH	IMAG
LBOUND	CONJG
HBOUND	special functions
UNSPEC	REPEAT
	BFnnnn
	array fns other than LBOUND, HBOUND

Built-in Functions

These are in	These are out
ONLOC	other ON functions
DATE	COMPLETE
TIME	ROUND
ALLOCATION	COUNT
POINT	
STRING	

Built-in function names may be overriden by DCL.

Abbreviations

All standard abbreviations for keywords and built-in functions will be recognized.

Conditions and the Prefix

Condition prefixes for machine traps will be observed.

FIXEDOVERFLOW means the machine condition.

These conditions are in	These are out
OVERFLOW	FINISH
UNDERFLOW	ERROR
ZERODIVIDE	
FIXEDOVERFLOW	
SIZE	
SUBSCRIPTRANGE	
CHECK	
CONDITION	

The file conditions must be decided upon as part of the I-O specifications.

The CHECK condition may or may not be turnable-off by a prefix.

Format Items

No picture specifications.

Fortran-style format specifications are up to the IOH. Probably we should compile calls for all format precedures, but expect that most will only be defined by future NPL-coded programs.