

IDENTIFICATION DIVISION — BASIC FORMATS

PROGRAM-ID. *program-name.*

[AUTHOR. *author-name.*]

[INSTALLATION. *any sentence or group of sentences.*]

[DATE-WRITTEN. *any sentence or group of sentences.*]

[DATE-COMPILED. *any sentence or group of sentences.*]

[SECURITY. *any sentence or group of sentences.*]

[REMARKS. *any sentence or group of sentences.*]

ENVIRONMENT DIVISION — BASIC FORMATS

See individual processor publications.

DATA DIVISION — BASIC FORMATS

File Description

FD *file-name* COPY *library-name.*

FD *file-name* [RECORDING MODE IS *mode*]

[BLOCK CONTAINS *integer-1* {RECORD|CHARACTER|IS}]

[RECORD CONTAINS [*integer-2* TO *integer-3* CHARACTER|IS]

LABEL RECORD|IS {ARE|IS} {STANDARD|OMITTED}

[VALUE OF *data-name-1* IS *literal* [*data-name-2* IS . . .]]

DATA RECORD|IS {ARE|IS} *data-name-3* [*data-name-4* . . .]

Record Description

level-number *data-name-1* [REDEFINES *data-name-2*]

COPY *data-name-3* [FROM LIBRARY]

level-number {FILLER|data-name-1} [REDEFINES *data-name-2*]

[SIZE IS *integer-1* {CHARACTER|DIGIT|IS}]

[USAGE IS {COMPUTATIONAL|DISPLAY} [OCCURS *integer-2* TIMES]

[SIGNED] [SYNCHRONIZED {LEFT|RIGHT}] [POINT LOCATION

IS {LEFT|RIGHT} *integer-3* PLACE|IS] [CLASS IS {ALPHABETIC|NUMERIC|ALPHANUMERIC|AN}

[PICTURE IS *any allowable combination of characters and symbols as described in Chapter 6*]

[JUSTIFIED {LEFT|RIGHT}] {ZERO SUPPRESS|CHECK PROTECT|FLOAT DOLLAR SIGN}

[LEAVING *integer-4* PLACE|IS] [BLANK WHEN ZERO]

[VALUE IS *literal*]

DATA-ITEM PICTORIAL CHARACTERS

(n) pictorial character repeated n times

Numeric		
9 any numeric character	S operational sign	
V assumed decimal point	P scale factor	
Alphabetic		
A any alphabetic character or space		
Alphanumeric		
X any character	. true decimal point	
Z numeric character or space	* numeric character or *	
\$ dollar sign †	0 true zero	
- minus sign or space †	B true space	
+ plus or minus sign †	CR credit symbol	
; true comma	DB debit symbol	

† These symbols may be either fixed or floating

LIST OF COBOL WORDS

ACCEPT	FLOAT	PROCEED
ADD	FOR	PROGRAM-ID
ADDRESS	FROM	PROTECT
ADDRESSES		
AFTER	GIVING	QUOTE
ALL	GO	QUOTES
ALPHABETIC	GREATER	
ALPHANUMERIC		READ
ALTER	HIGH-VALUE	RECORD
ALTERNATE	HIGH-VALUES	RECORDING
AN		RECORDS
AND	IDENTIFICATION	REDEFINES
APPLY	IF	REEL
ARE	IN	REELS
AREA	INPUT	REMARKS
AREAS	INPUT-OUTPUT	RENAMING
ASSIGN	INSTALLATION	REPLACING
AT	INTO	RERUN
AUTHOR	I-O-CONTROL	RESERVE
AUTHORS	IS	REWIND
		RIGHT
		ROUNDED
		RUN
BLANK	JUSTIFIED	
BLOCK		SECTION
BLOCKS	LABEL	SECURITY
BY	LEADING	SELECT
	LEAVING	SENTENCE
CHARACTER	LEFT	SIGN
CHARACTERS	LESS	SIGNED
CHECK	LIBRARY	SIZE
CLASS	LOCATION	SOURCE-COMPUTER
CLOSE	LOCK	SPACE
COBOL	LOW-VALUE	SPACES
COMPUTATIONAL	LOW-VALUES	SPECIAL-NAMES
COMPUTE		STANDARD
CONFIGURATION	MEMORY	STATUS
CONSTANT	MODE	STOP
CONTAIN	MOVE	SUBTRACT
CONTAINS	MULTIPLE	SUPPRESS
COPY	MULTIPLY	SYNCHRONIZED
CORRESPONDING		
	NEGATIVE	
DATA	NEXT	TALLY
DATE-COMPILED	NO	TALLYING
DATE-WRITTEN	NOT	THAN
DEPENDING	NOTE	THEN
DIGIT	NUMERIC	THROUGH
DIGITS		THRU
DISPLAY	OBJECT-COMPUTER	TIME
DIVIDE	OBJECT-PROGRAM	TIMES
DIVISION	OCCURS	TO
DOLLAR	OF	
	OFF	UNTIL
ELSE	OMITTED	UPON
END	ON	USAGE
ENTER	OPEN	
ENVIRONMENT	OPTIONAL	VALUE
EQUAL	OR	VARYING
ERROR	OTHERWISE	
EVERY	OUTPUT	WHEN
EXAMINE		WITH
EXIT	PERFORM	WORDS
	PICTURE	WORKING-STORAGE
FD	PLACE	WRITE
FILE	PLACES	
FILE-CONTROL	POINT	ZERO
FILLER	POSITIVE	ZEROES
FIRST	PROCEDURE	ZEROS

IBM Reference Card COBOL

SPECIAL CHARACTERS USED IN THE COBOL SOURCE LANGUAGE		
Name	Character	Card Code
space		(blank)
plus sign	+	12
minus sign	-	11
hyphen	-	11
multiplication sign	*	11-4-8
check protection symbol	*	11-4-8
division sign	/	0-1
left parenthesis	(0-4-8
right parenthesis)	12-4-8
comma	,	0-3-8
period	.	12-3-8
decimal point	.	12-3-8
dollar sign	\$	11-3-8
equal sign	=	3-8
quotation mark	'	4-8

ARITHMETIC EXPRESSIONS — SEQUENCE OF SYMBOLS					
First Symbol	Second Symbol				
	Variable	* or / or **	+ or -	()
Variable	-	P	P	-	P
* or / or **	P	-	P	P	-
+ or -	P	-	-	P	-
(P	-	P	P	-
)	-	P	P	-	P

CONDITIONAL EXPRESSIONS SEQUENCE OF SYMBOLS						
First Symbol	Second Symbol					
	C	OR	AND	NOT	()
C	-	P	P	-	-	P
OR	P	-	-	P	P	-
AND	P	-	-	P	P	-
NOT	P†	-	-	-	P	-
(P	-	-	P	P	-
)	-	P	P	-	-	P

† Permissible only if the condition itself does not contain a NOT.

ARITHMETIC OPERATORS	RELATIONAL OPERATORS	
OP Meaning	Long Form	Key
+ addition	IS GREATER THAN	GREATER
- subtraction or negation	IS EQUAL TO	EQUAL TO or =
* multiplication	IS LESS THAN	LESS
/ division	IS NOT GREATER THAN	NOT GREATER
** exponentiation	IS NOT EQUAL TO	NOT EQUAL TO or NOT =
	IS NOT LESS THAN	NOT LESS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
PAGE SERIAL - A B IDENT.

PROCEDURE DIVISION — BASIC FORMATS

ACCEPT *data-name* [FROM *mnemonic-name*]

ADD {*data-name-1*
literal-1} [{*data-name-2*
literal-2} . . .] [{TO
GIVING} *data-name-n*]
[ROUNDED] [ON SIZE ERROR *any imperative statement*]

ALTER *procedure-name-1* TO PROCEED TO *procedure-name-2*
[*procedure-name-3* TO PROCEED TO *procedure-name-4* . . .]

CLOSE *file-name-1* [REEL] [WITH {LOCK
NO REWIND}] [*file-name-2* . . .]

COMPUTE *data-name-1* [ROUNDED] = *arithmetic expression*
[ON SIZE ERROR *any imperative statement*]

DISPLAY {*data-name-1*
literal-1} [{*data-name-2*
literal-2} . . .] [UPON *mnemonic-name*]

DIVIDE {*data-name-1*
literal-1} INTO {*data-name-2*
literal-2} [GIVING *data-name-3*]
[ROUNDED] [ON SIZE ERROR *any imperative statement*]

ENTER *language-name*.

EXAMINE *data-name* { TALLYING { ALL
LEADING
UNTIL FIRST }
literal-1 [REPLACING BY *literal-2*] }
{ REPLACING { ALL
LEADING
UNTIL FIRST } *literal-3* BY *literal-4* }

EXIT.

GO TO [*procedure-name*]

GO TO *procedure-name-1* *procedure-name-2* [*procedure-name-3* . . .]
DEPENDING ON *data-name*

MOVE {*data-name-1*
literal} TO *data-name-2* [*data-name-3* . . .]

MOVE CORRESPONDING *data-name-1* TO *data-name-2* [*data-name-3* . . .]

MULTIPLY {*data-name-1*
literal-1} BY {*data-name-2*
literal-2} [GIVING *data-name-3*]
[ROUNDED] [ON SIZE ERROR *any imperative statement*]

NOTE *any comment.*

OPEN [INPUT *file-name-1* [*file-name-2* . . .]] [OUTPUT *file-name-3*
[*file-name-4* . . .]]

PERFORM *procedure-name-1* [THRU *procedure-name-2*]

PERFORM *procedure-name-1* [THRU *procedure-name-2*]
{ *integer-1*
data-name-1 } TIME(S)

PERFORM *procedure-name-1* [THRU *procedure-name-2*]
UNTIL *condition-1*

PERFORM *procedure-name-1* [THRU *procedure-name-2*]
VARYING *data-name-1* FROM { *numeric-literal-1*
data-name-2 }
BY { *numeric-literal-2*
data-name-3 } UNTIL *condition-1*

PERFORM *procedure-name-1* [THRU *procedure-name-2*]

VARYING *subscript-name-1* FROM { *integer-1*
data-name-1 } BY { *integer-2*
data-name-2 }

UNTIL *condition-1* [AFTER *subscript-name-2* FROM { *integer-3*
data-name-3 }]

BY { *integer-4*
data-name-4 } UNTIL *condition-2*] [AFTER *subscript-name-3*

FROM { *integer-5*
data-name-5 } BY { *integer-6*
data-name-6 } UNTIL *condition-3*]

READ *file-name* RECORD [INTO *area-name*]
[AT END *any imperative statement*]

STOP { *literal*
RUN }

SUBTRACT {*data-name-1*
literal-1} [{*data-name-2*
literal-2} . . .]

FROM {*data-name-n*
literal-n} [GIVING *data-name-m*]

[ROUNDED] [ON SIZE ERROR *any imperative statement*]

WRITE *record-name* [FROM *area-name*]

Underscored words are key words; words not underscored are optional words; brackets, [], enclose words and phrases which may be used or omitted as required; braces, { }, are used to indicate a choice of two or more variant forms.