IBM Library Server Print Preview

DOCNUM = SX26-3834-00
DATETIME = 02/17/95 13:20:26
BLDVERS = 1.2
TITLE = COBOL/VSE Reference Summary
AUTHOR =

PATH = /home/webapps/epubs/htdocs/book

COVER Book Cover

IBM COBOL for VSE/ESA

Reference Summary

Release 1

Document Number SX26-3834-00

Program Number
5686-068

File Number S370-40

NOTICES Notices

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" in topic FRONT_1.

EDITION Edition Notice

First Edition (April 1995)

This edition applies to Version 1 Release 1 of IBM COBOL for VSE/ESA, Program Number 5686-068, and to all subsequent releases and modifications until otherwise indicated in new editions. Changes are made periodically to this publication; consult the latest *IBM System/390, 370, 30xx, 4300, and 9370 Processors Bibliography* for current information on this product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department J58
P.O. Box 49023
San Jose, California, 95161-9023
U.S.A.

CONTENTS Table of Contents

TABLES Tables

FIGURES Figures

FRONT_1 Notices

References in this publication to IBM* products or services do not imply that they will be available everywhere IBM operates, nor that only IBM products or services can be used. Functionally equivalent products or services that do not infringe legal rights held by IBM can be used instead. Operation with products or services other than those expressly designated by IBM is your responsibility.

IBM may have patents or pending patent applications covering subject matter described herein. This document neither grants nor implies any license or immunity under any IBM or third-party patents, patent applications, trademarks, copyrights, or other similar rights, or any right to refer to IBM in any marketing activities. Other than responsibilities assumed via the Agreement for Purchase of IBM Machines and the Agreement for IBM Licensed Programs, IBM assumes no responsibility

for any infringement of third-party rights that may result from use of the subject matter disclosed in this publication or from the manufacture, use, lease, or sale of machines or programs described herein.

Licenses under utility patents held by IBM are available on reasonable and nondiscriminatory terms. IBM does not grant licenses under its appearance design patents. Direct licensing inquiries in writing to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A..

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. **This disclaimer does not apply in the United Kingdom or elsewhere if inconsistent with local law.**

Subtopics:

FRONT_1.1 Programming Interface

This *COBOL/VSE Reference Summary* is intended to present to the experienced COBOL programmer the syntax of COBOL/VSE statements and the rules for writing source programs to be compiled by the COBOL/VSE compiler. This *COBOL/VSE Reference Summary* documents General-Use Programming Interface and Associated Guidance Information provided by IBM COBOL for VSE/ESA.

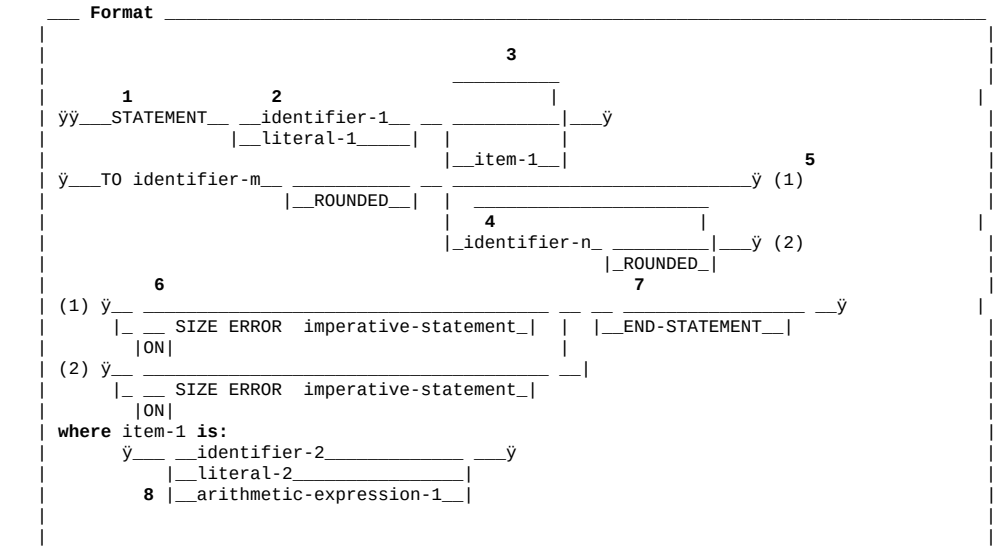
General-Use programming interfaces allow the customer to write programs that obtain the services of IBM COBOL for VSE/ESA.

FRONT_1.2 Trademarks

The following terms, denoted by an asterisk (*) on their first occurrences in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

COBOL/370	IBM	System/370
System/390	VSE/ESA	

1.0 Format Notation



1 The STATEMENT keyword must be specified and coded as shown .

2 This operand is required. Either identifier-1 or literal-1 can be coded.

3 The operand item-1 is optional. It can be coded or not, as required by the application. If coded, it may be repeated, with each entry separated by one or more COBOL separators. Entry selections allowed for this operand are described at the bottom of the diagram.

4 The operand identifier-n is optional. If specified it can be repeated with one or more COBOL separators separating each entry. Each entry can be assigned the keyword ROUNDED.

5 In cases where multiple lines must be continued, a number is given at the end of one line and the beginning of the next, in order to show syntax flow.

6 The ON SIZE ERROR phrase, with associated imperative statement, can be coded after identifier-m or the last identifier-n operand. If coded, SIZE ERROR is required; ON is optional.

7 The END-STATEMENT keyword can be coded to end the stateme nt. It is not a required delimiter.

8 An X in the left margin indicates an IBM extension on that line, in this case arithmetic-expression-1. This operand is optional.

2.0 COBOL Language Structure

Subtopics:

2.1 COBOL/VSE Characters

Table 1. COBOL/VSE Characters--Their Meanings and Uses

Character	Meaning	Use
°	Space	Punctuation character
.	Decimal point or Period	Editing character Punctuation character
<	Less than	Relation character
(	Left parenthesis	Punctuation character
+	Plus sign	Arithmetic operator Editing character
$	Dollar sign	Editing character
*	Asterisk	Arithmetic operator Editing character Comment character
)	Right parenthesis	Punctuation character

;	Semicolon	Punctuation character
-	Minus sign or Hyphen	Arithmetic operator Editing character Continuation character
/	Stroke or Slash	Arithmetic operator Editing character Line control character
,	Comma	Punctuation character Editing character
>	Greater than	Relation character
:	Colon	Punctuation character
'	Apostrophe	Non-numeric literal delimiter
=	Equal sign	Relation character Punctuation character
"	Quotation mark	Non-numeric literal delimiter
A-Z	Alphabet (uppercase)	Alphabetic characters
a-z	Alphabet (lowercase)	Alphabetic characters
0-9	Arabic numerals	Numeric characters

2.2 Separator Characters

Table 2. Separator Characters

Separator	**Meaning**
º	Space
,º	Comma
.º	Period
;º	Semicolon
(	Left parenthesis
)	Right parenthesis
:	Colon
"	Quotation marks
'	Apostrophe
==	Pseudo-text delimiter

2.3 User-Defined Words

Table 3. User-Defined Words

Types of User-Defined Words	**General Rules**
Alphabet-name Class-name Condition-name Data-name File-name Index-name Mnemonic-name Program-name (for contained program) Record-name Routine-name Symbolic-character	Each word must contain at least one alphabetic character.

Library-name Program-name Text-name	See "PROGRAM-ID Paragraph" in *COBOL/VSE Application Programming: Language Reference*.
Paragraph-name Section-name	The word need **not** contain an alphabetic character.
Level-numbers: 01-49,66,77,88 Priority-numbers: 00-99	Each word must be a 1-digit or 2-digit integer.

2.4 Figurative Constants

ALL literal

HIGH-VALUE, HIGH-VALUES

LOW-VALUE, LOW-VALUES

NULL, NULLS

QUOTE, QUOTES

SPACE, SPACES

symbolic-character

ZERO, ZEROS, ZEROES

2.5 Special Registers

ADDRESS OF

DEBUG-ITEM

LENGTH OF

LINAGE-COUNTER

RETURN-CODE

SHIFT-OUT and SHIFT-IN

SORT-CONTROL

SORT-RETURN

SORT-CORE-SIZE

SORT-FILE-SIZE

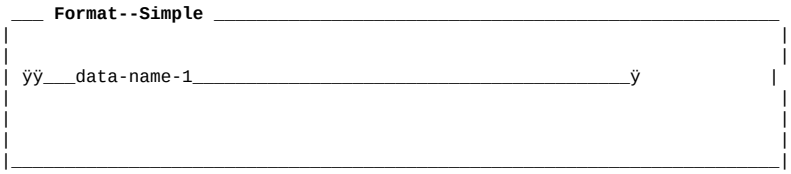
SORT-MESSAGE

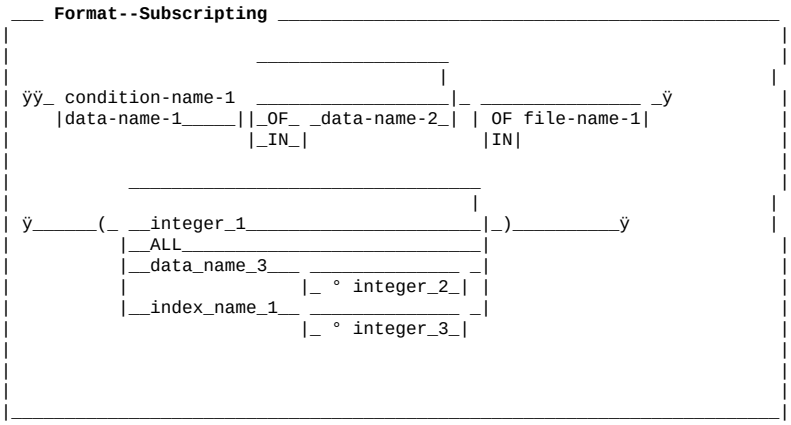
SORT-MODE-SIZE

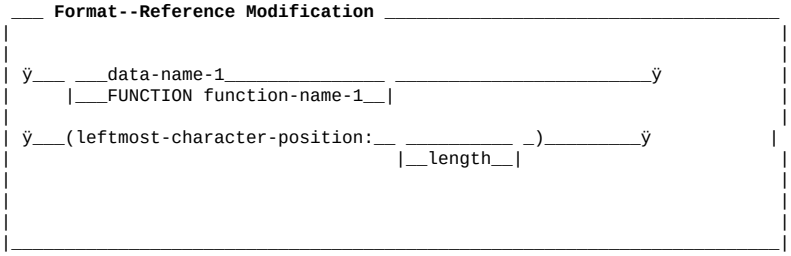
TALLY

WHEN-COMPILED

2.6 Methods of Data Reference







3.0 COBOL Program Structure

Subtopics:

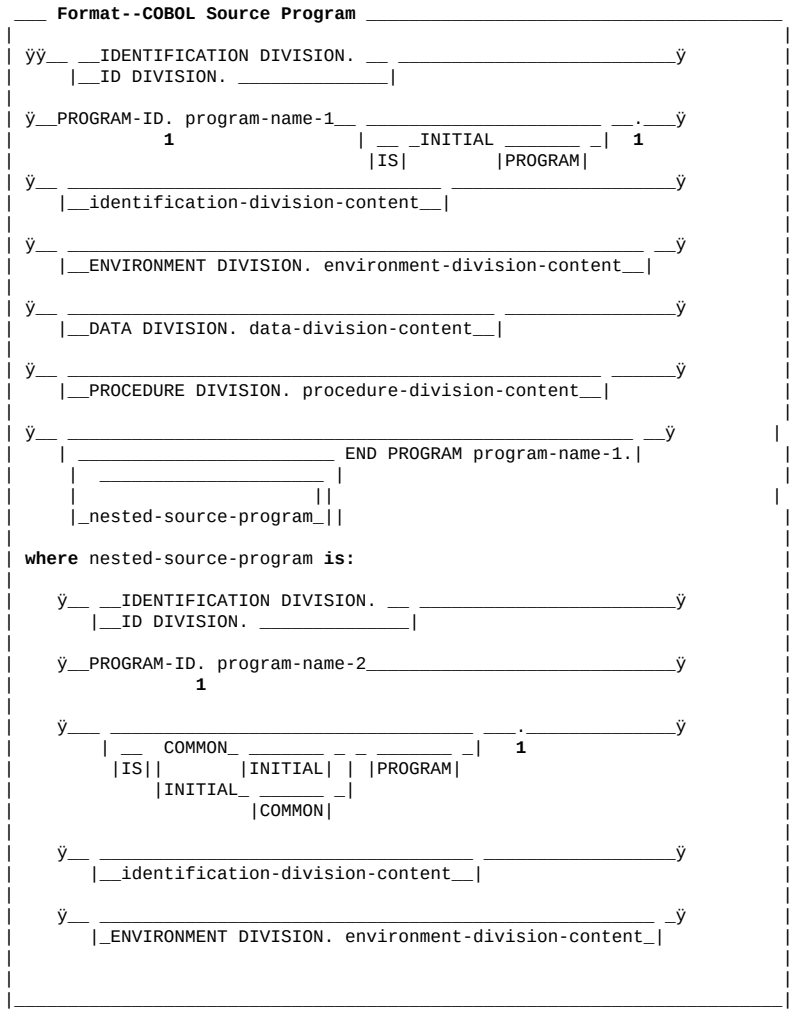
- 3.1 COBOL Program Structure
- 3.2 Intrinsic Functions

3.1 COBOL Program Structure

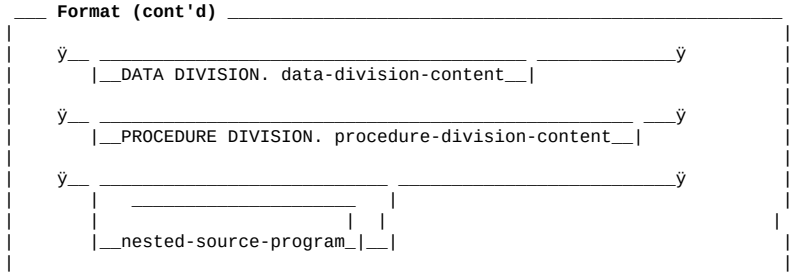
Subtopics:

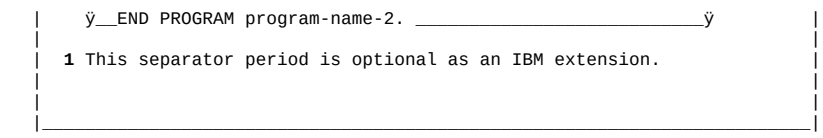
- 3.1.1 COBOL Source Program

3.1.1 COBOL Source Program

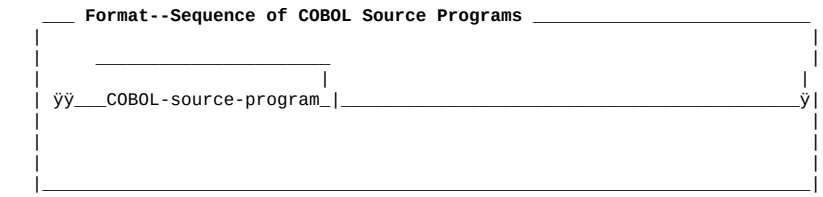


Format continued on next page.

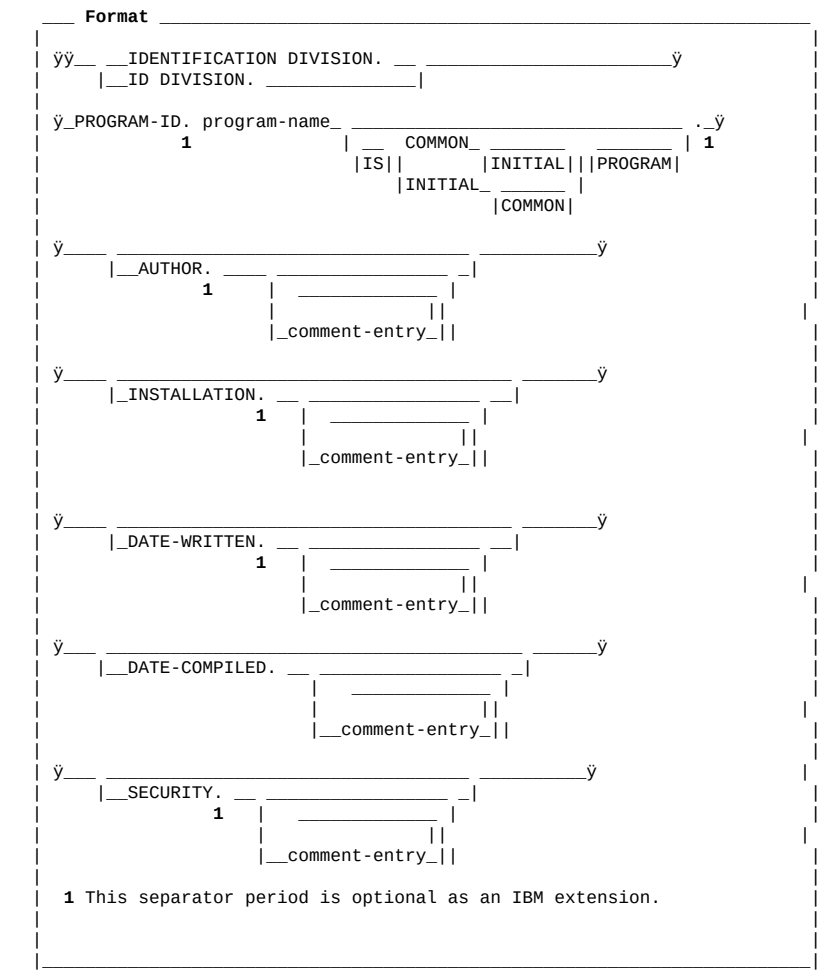




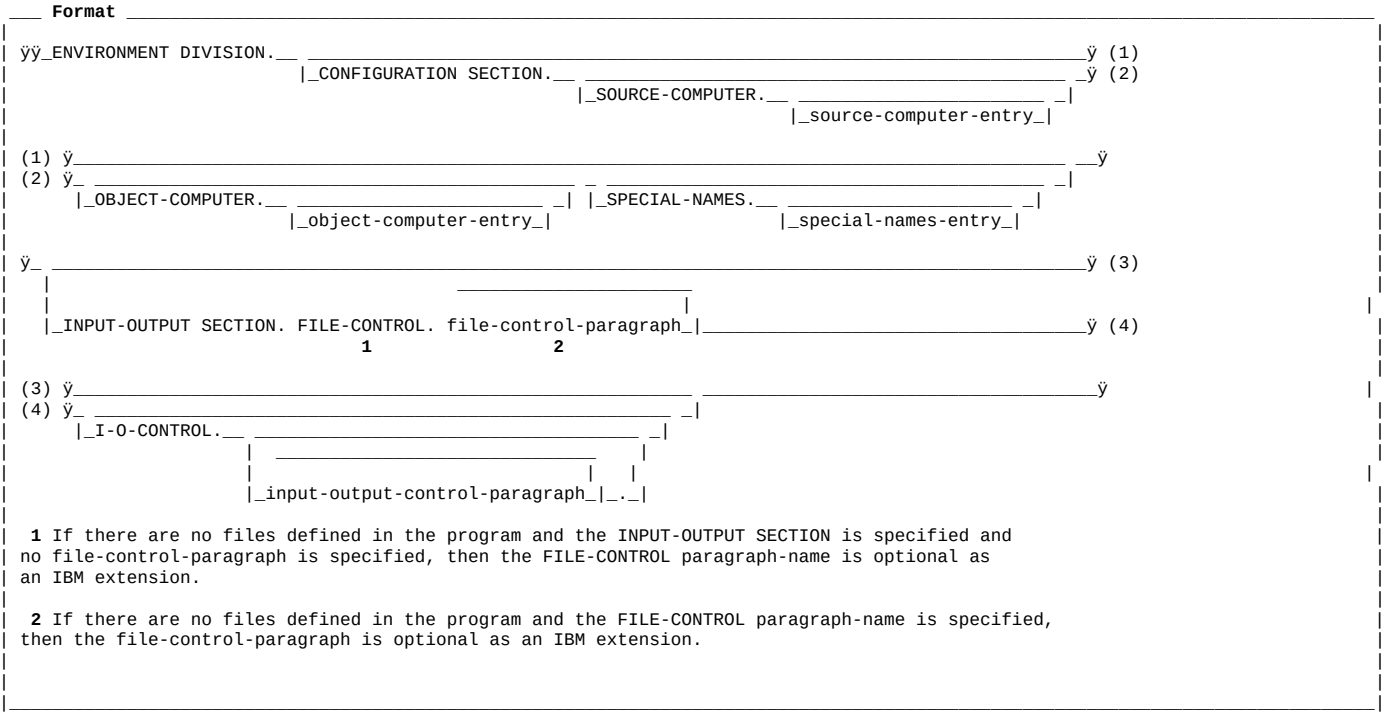
3.1.2 Sequence of COBOL Source Programs



3.1.3 Identification Division

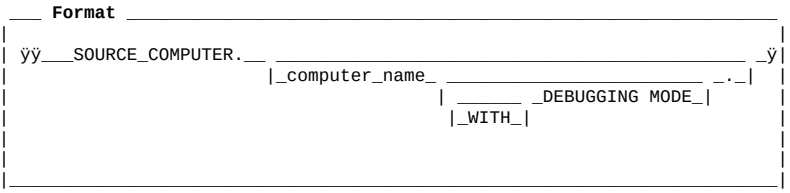


3.1.4 Environment Division--Configuration Section

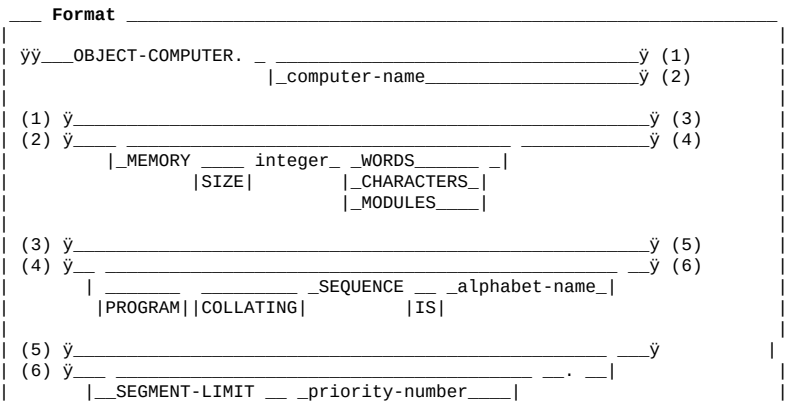


Subtopics:

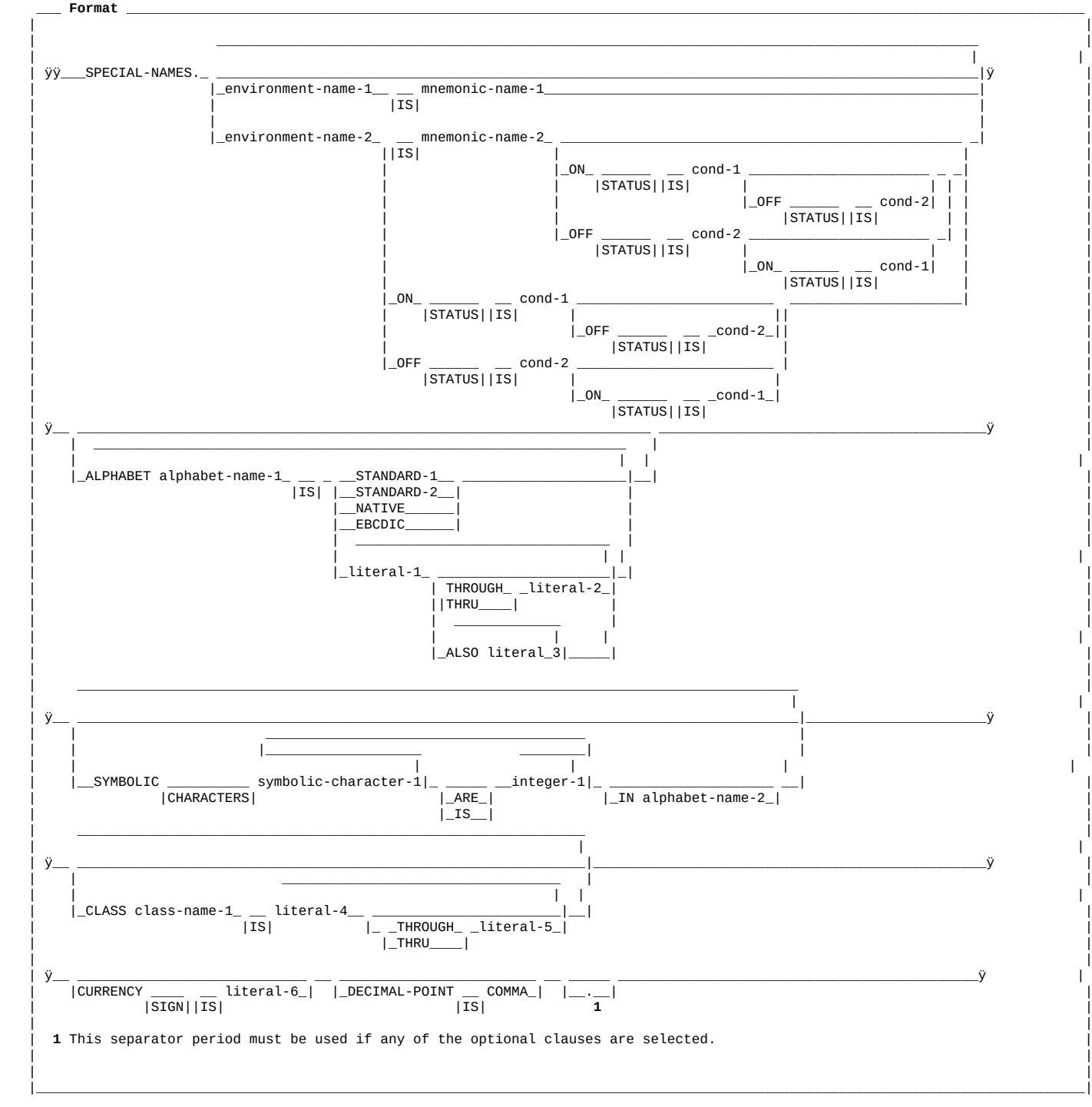
3.1.4.1 SOURCE-COMPUTER Paragraph



3.1.4.2 OBJECT-COMPUTER Paragraph



3.1.4.3 SPECIAL-NAMES Paragraph



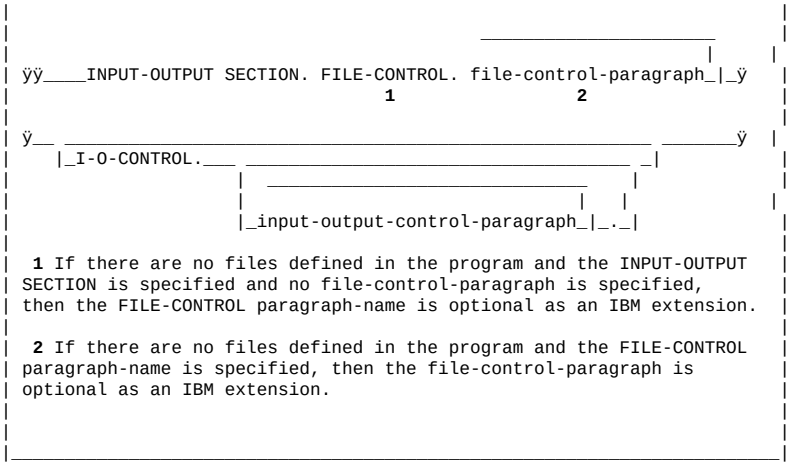
3.1.4.4 Meanings of Environment Names

Table 4. Meanings of Environment Names

Environment-Name-1	Meaning	Allowed In
SYSIN SYSIPT	System logical input unit	ACCEPT
SYSOUT SYSLIST SYSLST	System logical output unit	DISPLAY
SYSPUNCH SYSPCH	System punch device	DISPLAY
CONSOLE	Console typewriter	ACCEPT and DISPLAY
C01 - C12	Skip to channel 1 through 12, respectively	WRITE ADVANCING
CSP	Suppress spacing	WRITE ADVANCING
S01-S05	Pocket select 1-5 on punch devices	WRITE ADVANCING

3.1.5 Environment Division--Input-Output Section

Format



1 If there are no files defined in the program and the INPUT-OUTPUT SECTION is specified and no file-control-paragraph is specified, then the FILE-CONTROL paragraph-name is optional as an IBM extension.

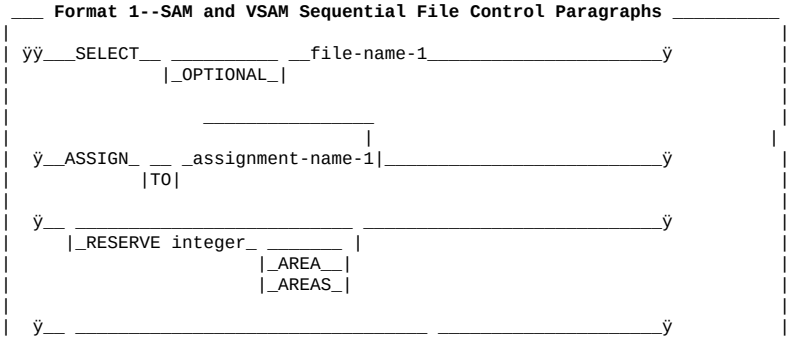
2 If there are no files defined in the program and the FILE-CONTROL paragraph-name is specified, then the file-control-paragraph is optional as an IBM extension.

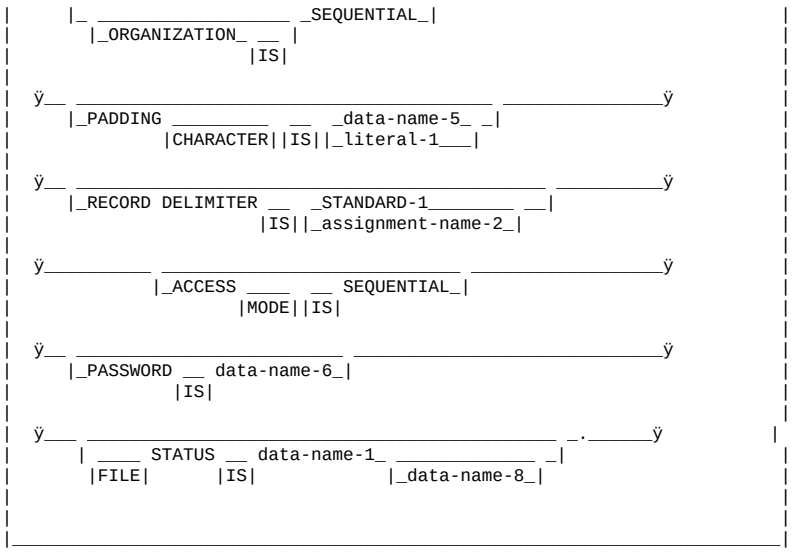
Subtopics:

- 3.1.5.1 File-Control Paragraph
- 3.1.5.2 I-O-Control Paragraph

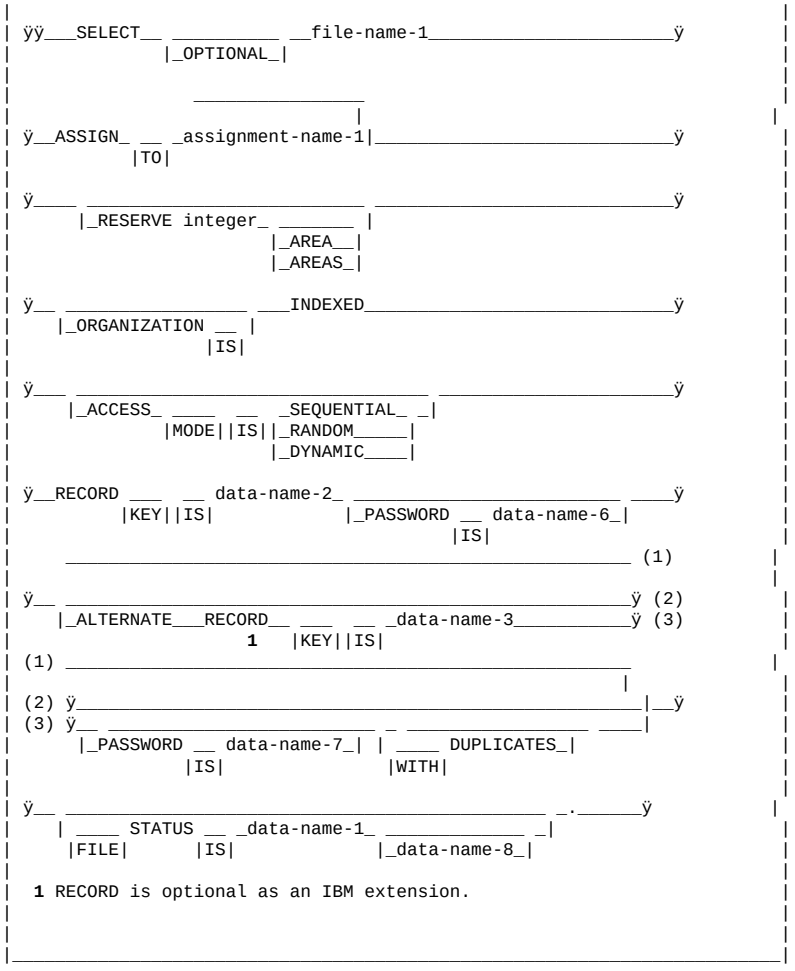
3.1.5.1 File-Control Paragraph

Format 1--SAM and VSAM Sequential File Control Paragraphs



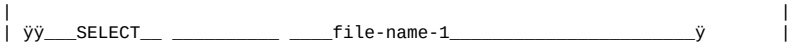


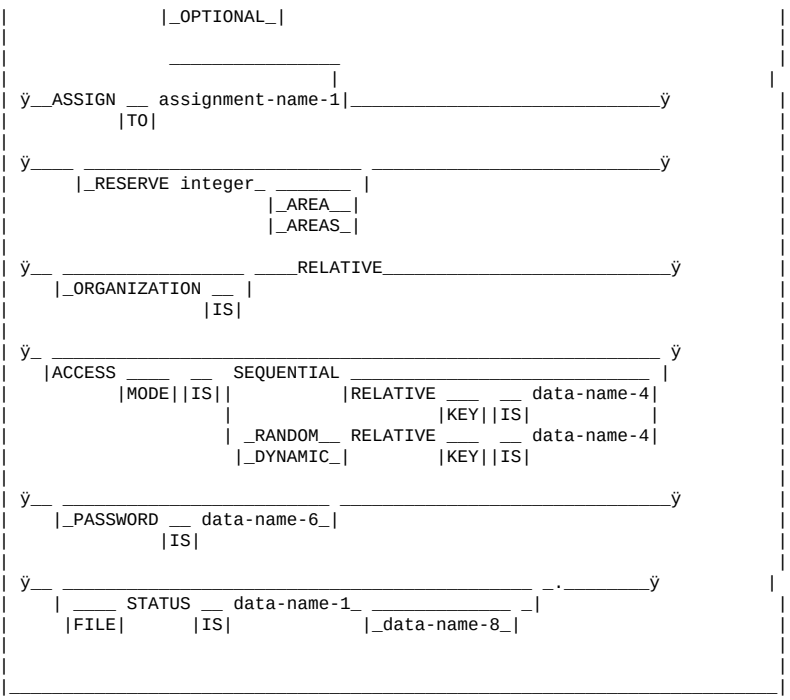
Format 2--VSAM Indexed File Control Paragraphs



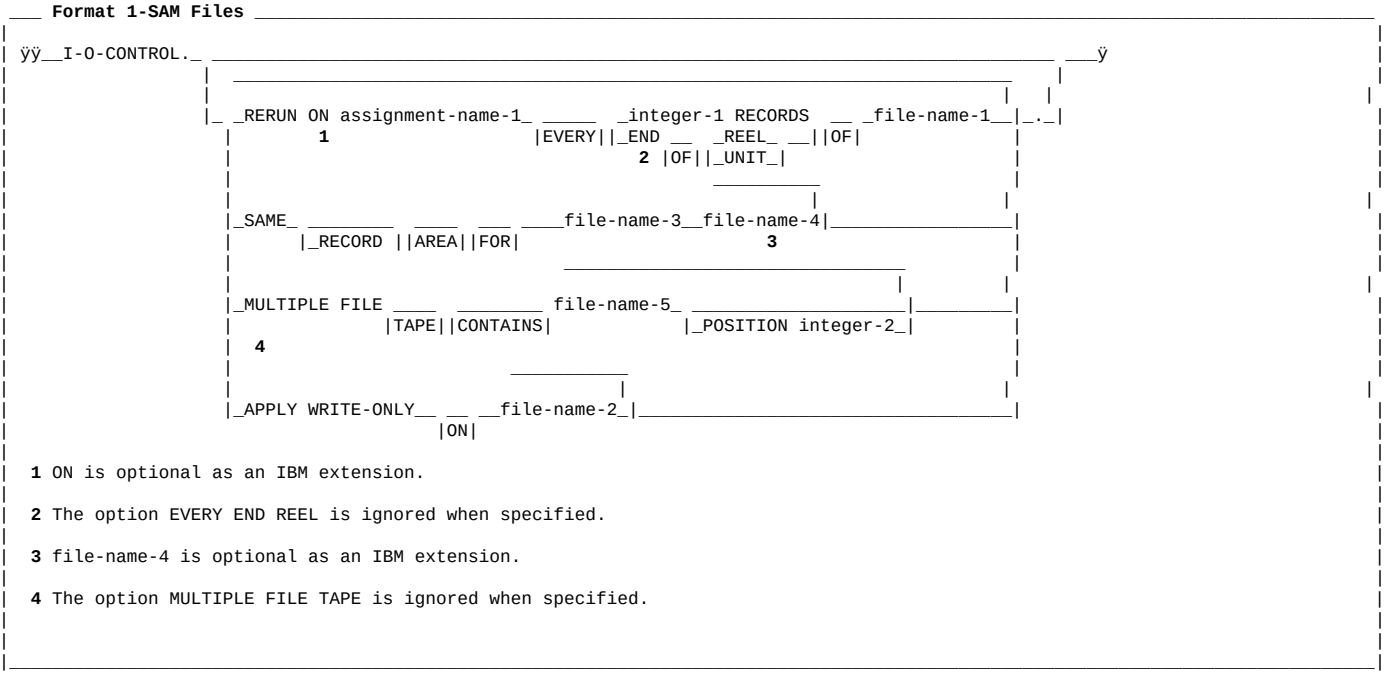
1 RECORD is optional as an IBM extension.

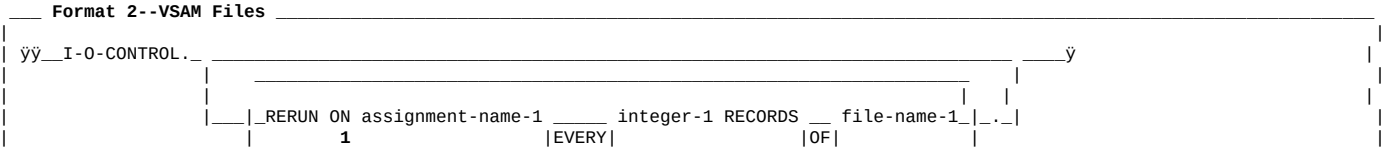
Format 3--VSAM Relative File Control Paragraphs

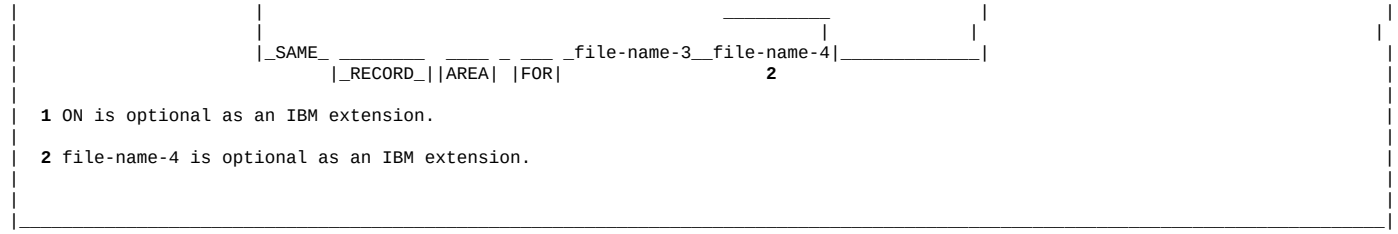


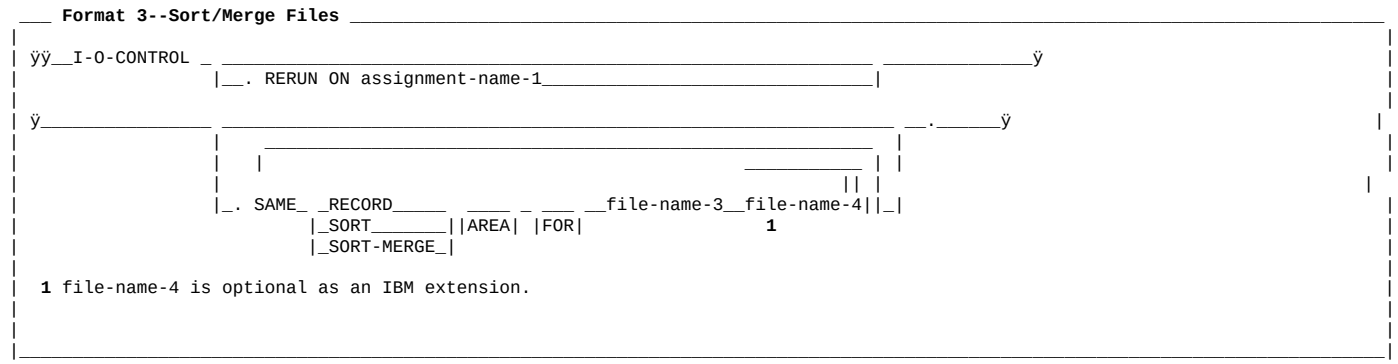


3.1.5.2 I-O-Control Paragraph

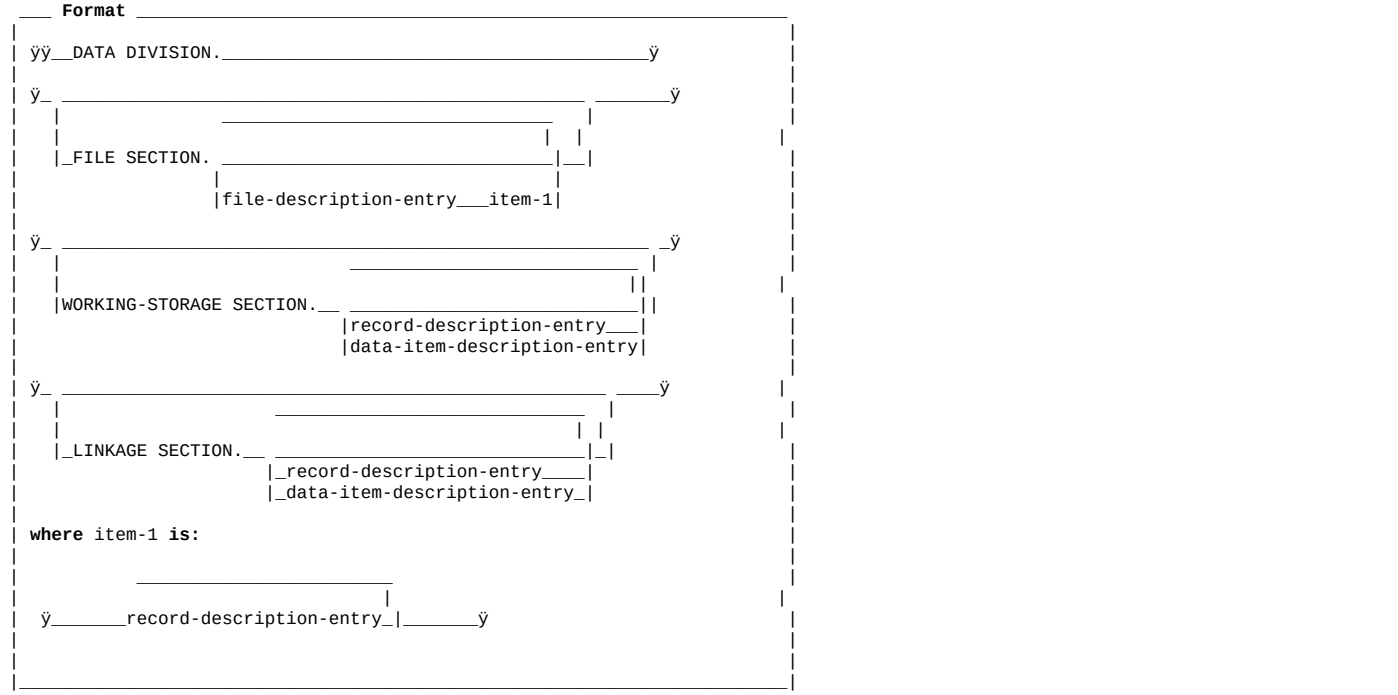








3.1.6 Data Division Structure



Subtopics:

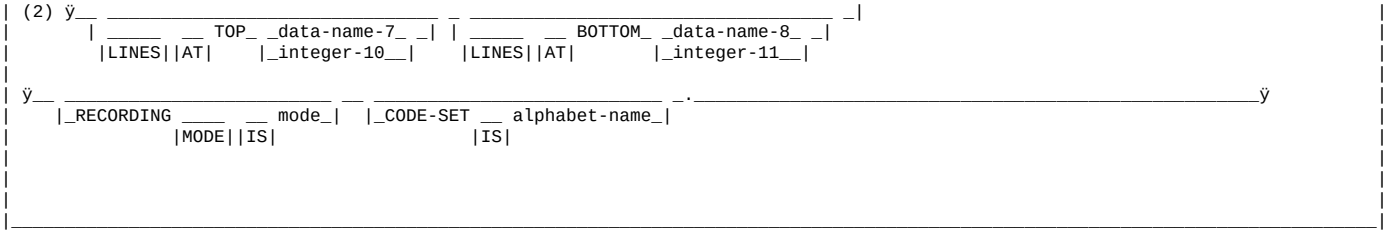
- 3.1.6.1 Classes and Categories of Data
- 3.1.6.2 File and Sort Description Entries
- 3.1.6.3 Data Description Entry

3.1.6.1 Classes and Categories of Data

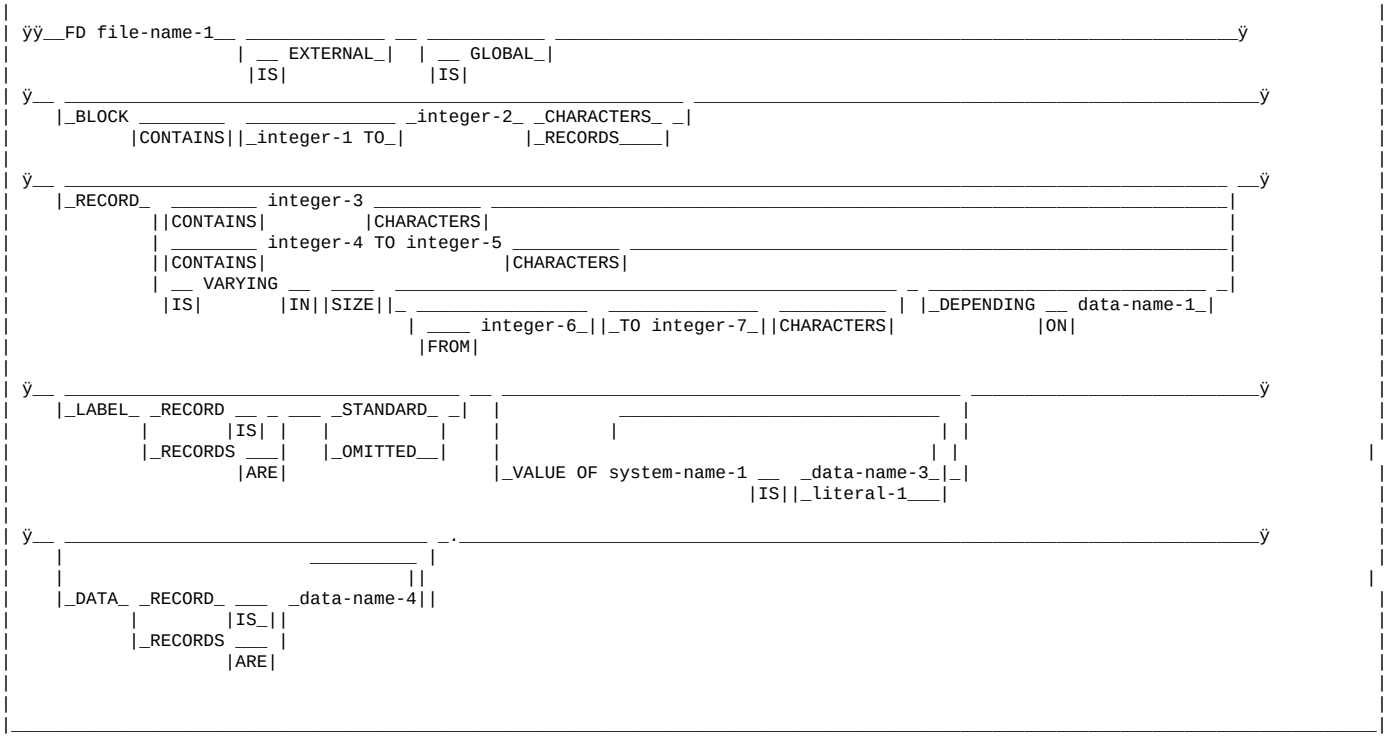
Table 5. Classes and Categories of Data

Level of Item	Class	Category
Elementary	Alphabetic	Alphabetic
	Numeric	Numeric
		Internal Floating-point
		External Floating-point
	Alphanumeric	Numeric-Edited
		Alphanumeric-Edited
		Alphanumeric
		DBCS
Group	Alphanumeric	Alphabetic
		Numeric
		Internal Floating-point
		External Floating-point
		Numeric-Edited
		Alphanumeric-Edited
		Alphanumeric
		DBCS

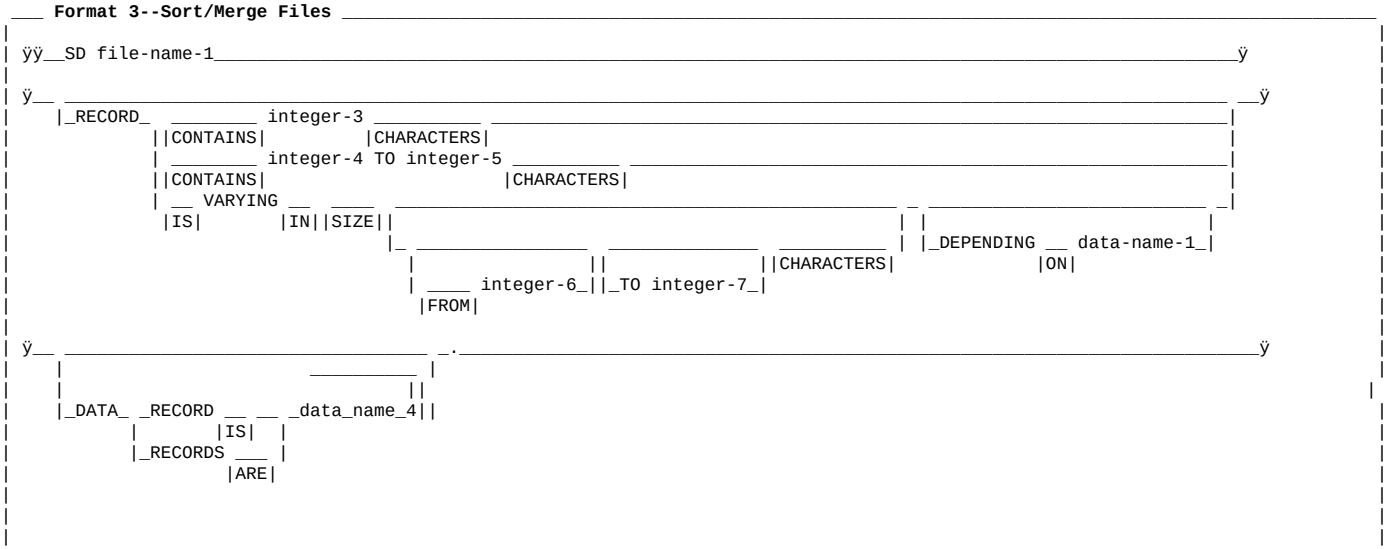
3.1.6.2 File and Sort Description Entries



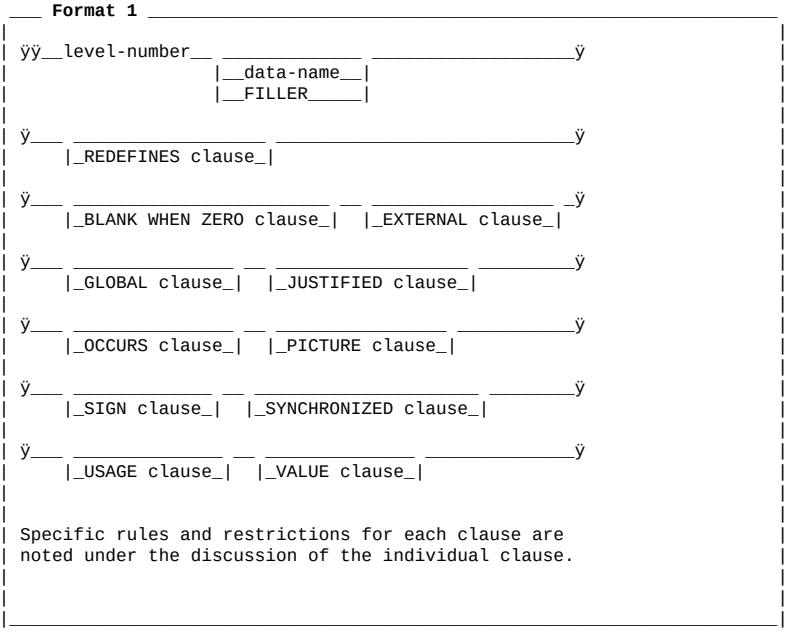
Format 2--Relative/Indexed Files

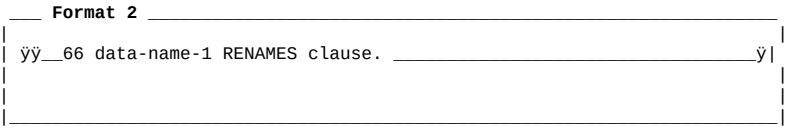


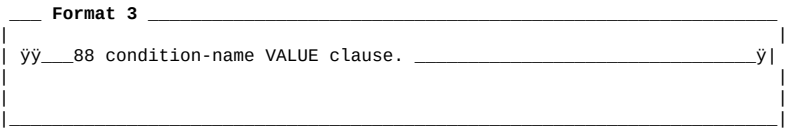
Format 3--Sort/Merge Files



3.1.6.3 Data Description Entry



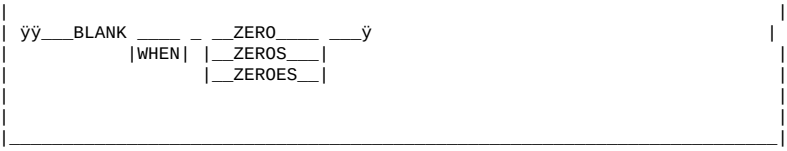




Subtopics:

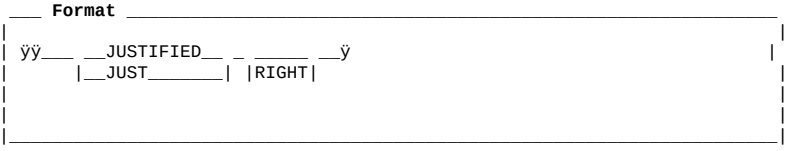
3.1.6.3.1 BLANK WHEN ZERO Clause

Format



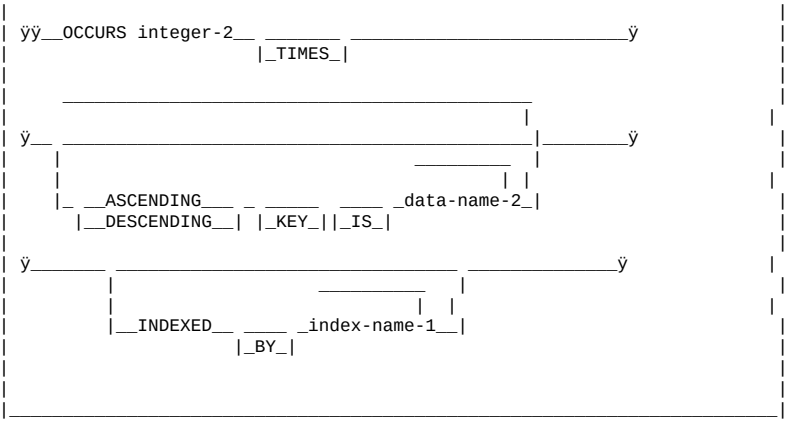
3.1.6.3.2 JUSTIFIED Clause

Format

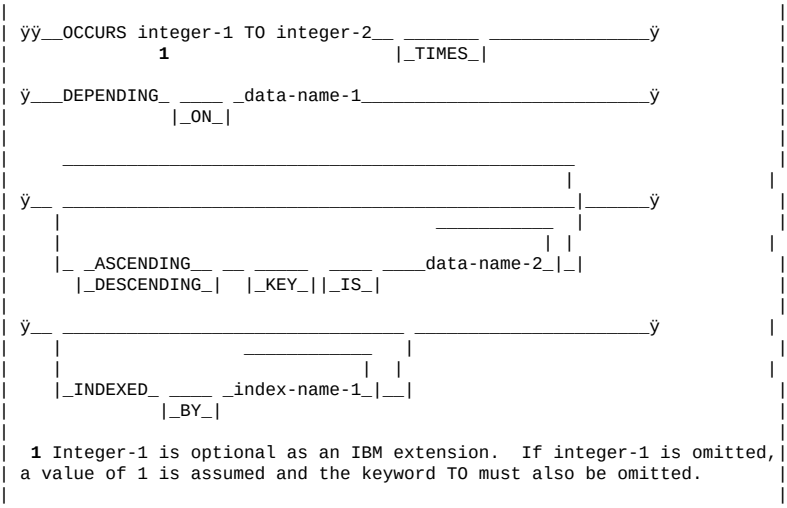


3.1.6.3.3 OCCURS Clause

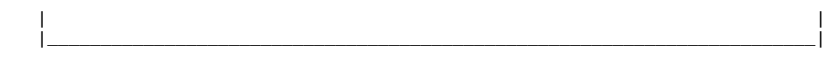
Format 1--Fixed-Length Tables



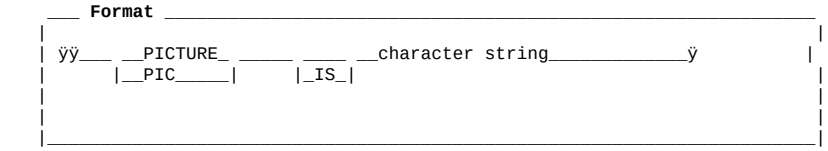
Format 2--Variable-Length Tables



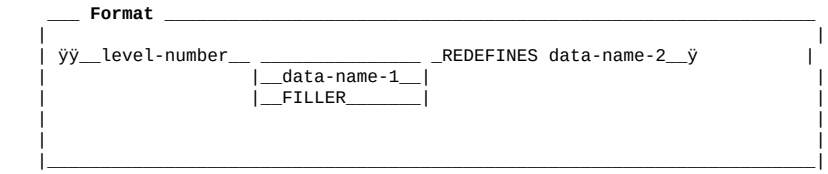
1 Integer-1 is optional as an IBM extension. If integer-1 is omitted, a value of 1 is assumed and the keyword TO must also be omitted.



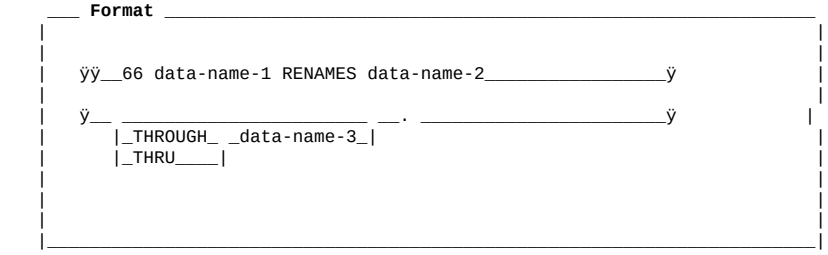
3.1.6.3.4 PICTURE Clause



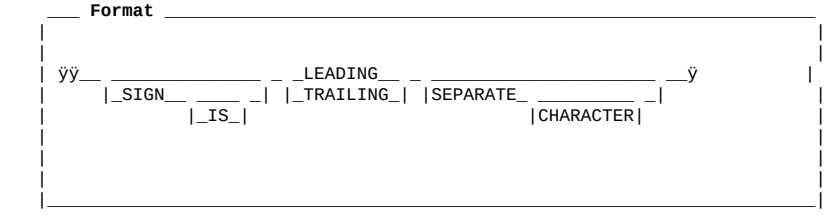
3.1.6.3.5 REDEFINES Clause



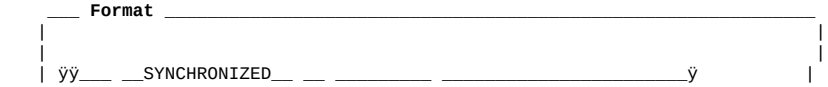
3.1.6.3.6 RENAMES Clause

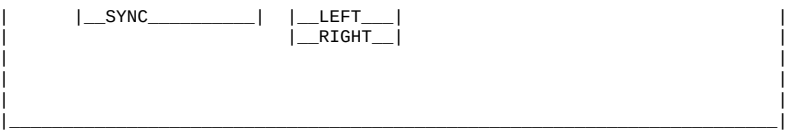


3.1.6.3.7 SIGN Clause

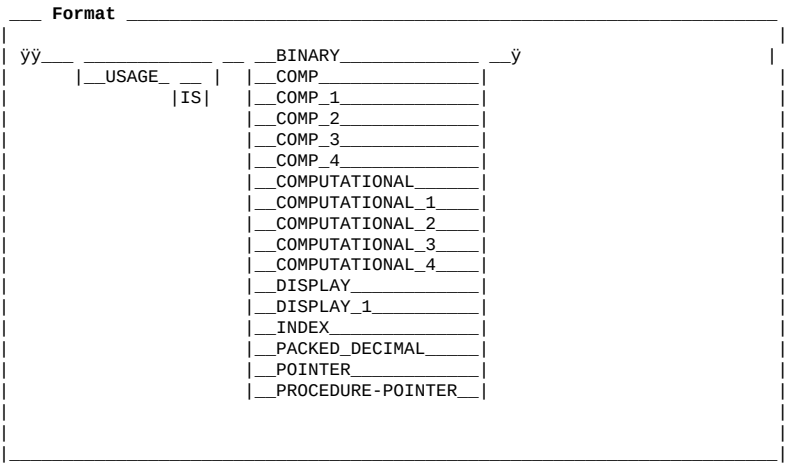


3.1.6.3.8 SYNCHRONIZED Clause

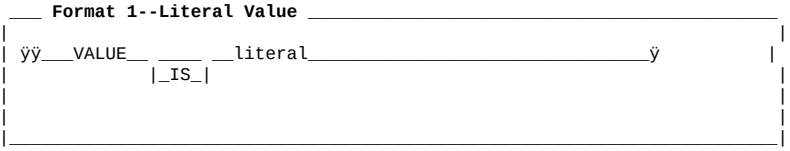


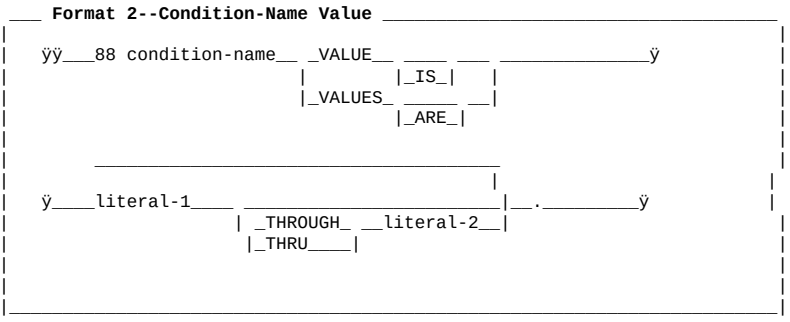


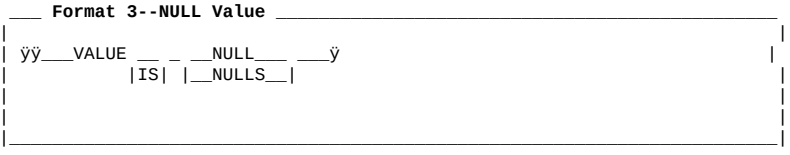
3.1.6.3.9 USAGE Clause



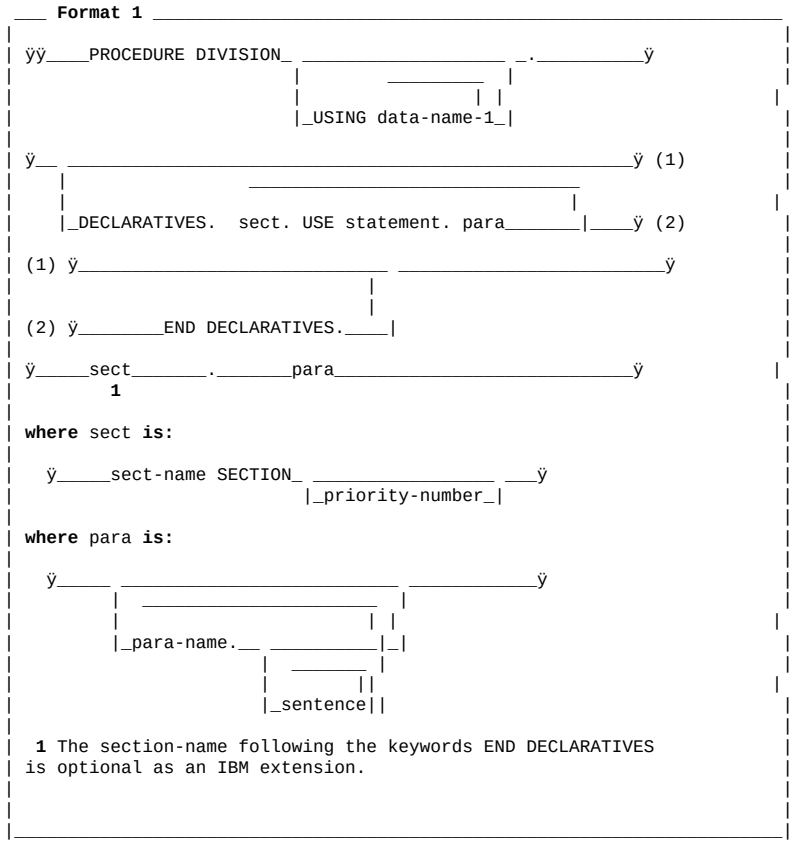
3.1.6.3.10 VALUE Clause

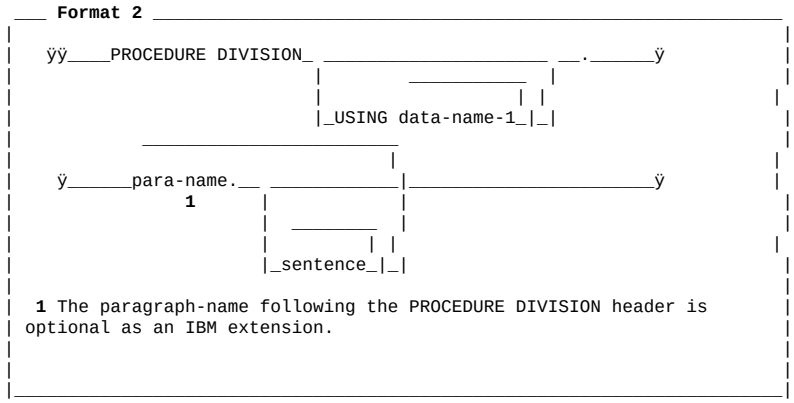






3.1.7 Procedure Division





Subtopics:

- 3.1.7.1 Arithmetic Operators
- 3.1.7.2 Class Condition
- 3.1.7.3 Class Test
- 3.1.7.4 Condition-Name Condition
- 3.1.7.5 Relation Condition
- 3.1.7.6 Relational Operators
- 3.1.7.7 Permissible Operand Comparison
- 3.1.7.8 Sign Condition
- 3.1.7.9 Switch-Status Condition
- 3.1.7.10 Complex Conditions

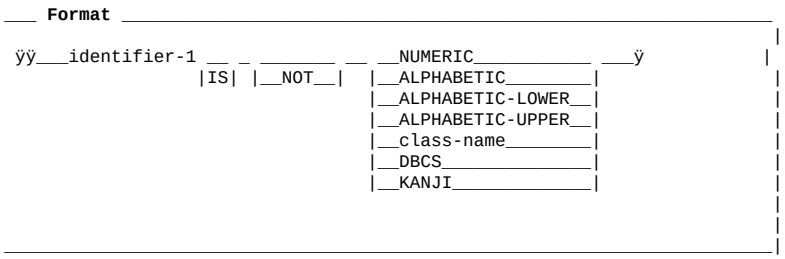
3.1.7.1 Arithmetic Operators

Table 6. Valid Arithmetic Symbol Pairs

First Symbol	Second Symbol: Identifier or Literal	* / ** + -	Unary + or Unary -	(	)
Identifier or Literal	No	Yes	No	No	Yes
*** / ** + -**	Yes	No	Yes	Yes	No
Unary + or Unary -	Yes	No	No	Yes	No
(	Yes	No	Yes	Yes	No
)	No	Yes	No	No	Yes

3.1.7.2 Class Condition

Format



3.1.7.3 Class Test

Table 7. Valid Forms of the Class Test

Type of Identifier	Valid Forms of the Class Test	
Alphabetic	ALPHABETIC	NOT ALPHABETIC
	ALPHABETIC-LOWER	NOT ALPHABETIC-LOWER
	ALPHABETIC-UPPER	NOT ALPHABETIC-UPPER
	class-name	NOT class-name
Alphanumeric, Alphanumeric-edited, or Numeric-edited	ALPHABETIC	NOT ALPHABETIC
	ALPHABETIC-LOWER	NOT ALPHABETIC-LOWER
	ALPHABETIC-UPPER	NOT ALPHABETIC-UPPER
	NUMERIC	NOT NUMERIC
	class-name	NOT class-name
External-Decimal Internal-Decimal	NUMERIC	NOT NUMERIC
DBCS	DBCS	NOT DBCS
	KANJI	NOT KANJI

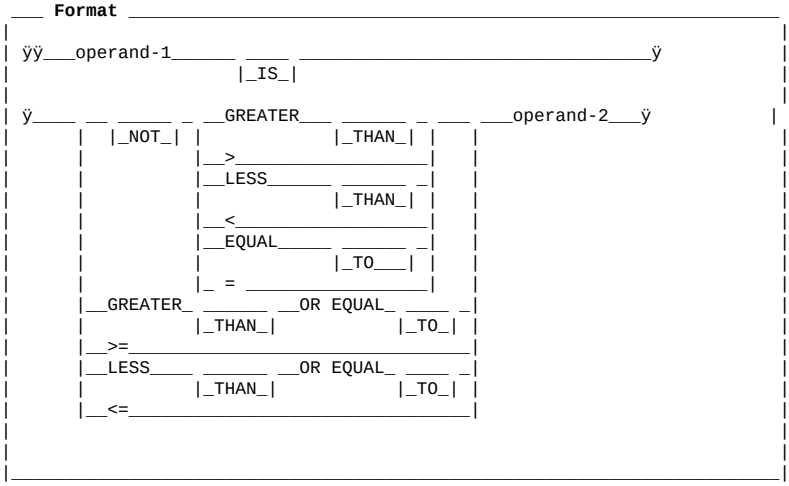
3.1.7.4 Condition-Name Condition

Format

```
ÿÿ___condition_name_____________________________________ÿ
```

3.1.7.5 Relation Condition

Format



3.1.7.6 Relational Operators

Table 8. Relational Operators and Their Meanings

Relational Operator	Can Be Written	Meaning
IS GREATER THAN	IS >	Greater than
IS NOT GREATER THAN	IS NOT >	Not greater than
IS LESS THAN	IS <	Less than
IS NOT LESS THAN	IS NOT <	Not less than
IS EQUAL TO	IS =	Equal to
IS NOT EQUAL TO	IS NOT =	Not equal to
IS GREATER THAN OR EQUAL TO	IS >=	Is greater than or equal to
IS LESS THAN OR EQUAL TO	IS <=	Is less than or equal to

3.1.7.7 Permissible Operand Comparison

Table 9. Permissible Comparisons for USAGE IS POINTER, NULL and ADDRESS OF

First Operand	Second Operand		
	PTR	ADDRESS OF	NULL
Usage is Pointer (PTR)	Yes	Yes	Yes
Address of (PTR)	Yes	Yes	Yes

Figurative Constant (NULL)	Yes	Yes	No

Note:

Yes = Comparison allowed only for EQUAL, NOT EQUAL
No = No comparison allowed

Table 10. Permissible Comparisons for USAGE IS PROCEDURE-POINTER and Figurative Constant

First Operand	Second Operand	
	Usage Procedure-Pointer	NULL
Usage is Procedure-Pointer	Yes	Yes
Figurative Constant (NULL)	Yes	No

Note:

Yes = Comparison allowed only for EQUAL, NOT EQUAL
No = No comparison allowed

Table 11. Permissible Comparisons with Non-numeric Second Operands

First Operand	Second Operand						
	GR	AL	AN	ANE	NE	FC(2)	NNL
Non-numeric Operand							
Group (GR)	NN	NN	NN	NN	NN	NN	NN
Alphabetic (AL)	NN	NN	NN	NN	NN	NN	NN
Alphanumeric (AN)	NN	NN	NN	NN	NN	NN	NN
Alphanumeric-Edited (ANE)	NN	NN	NN	NN	NN	NN	NN
Numeric-Edited (NE)	NN	NN	NN	NN	NN	NN	NN
Figurative Constant (FC(2))	NN	NN	NN	NN	NN		
Non-numeric Literal (NNL)	NN	NN	NN	NN	NN		
Numeric Operand							
Figurative Constant ZERO (ZR)	NN	NN	NN	NN	NN		
Numeric Literal (NL)	NN(1)	NN(1)	NN(1)	NN(1)	NN(1)		
External Decimal (ED)	NN(1)	NN(1)	NN(1)	NN(1)	NN(1)	NN(1)	NN(1)
Binary (BI)							
Arithmetic Expression (AE)							
Internal Decimal (ID)							
Internal Floating-Point (IFP)							
External Floating-Point (EFP)	NN	NN	NN	NN	NN	NN	NN
Floating-Point Literal (FPL)							

Note:

NN = Comparison for nonnumeric operands
NU = Comparison for numeric operands
Blank = Comparison is not allowed
(1)Integer item only
(2)Includes all figurative constants except ZERO

Table 12. Permissible Comparisons with Numeric Second Operands

First Operand	Second Operand								
	ZR	**NL**	**ED**	**BI**	**AE**	**ID**	**IFP**	**EFP**	**FPL**
Non-numeric Operand									
Group (GR)	NN	NN(1)	NN(1)					NN	
Alphabetic (AL)	NN	NN(1)	NN(1)					NN	
Alphanumeric (AN)	NN	NN(1)	NN(1)					NN	
Alphanumeric-Edited (ANE)	NN	NN(1)	NN(1)					NN	
Numeric Edited (NE)	NN	NN(1)	NN(1)					NN	
Figurative Constant (FC(2))			NN(1)					NN	
Non-numeric Literal (NNL)			NN(1)					NN	
Numeric Operand									
Figurative Constant ZERO (ZR)			NU	NU	NU	NU	NU	NU	
Numeric Literal (NL)			NU	NU	NU	NU	NU	NU	
External Decimal (ED)	NU	NU	NU	NU	NU	NU	NU	NU	NU
Binary (BI)	NU	NU	NU	NU	NU	NU	NU	NU	NU
Arithmetic Expression (AE)	NU	NU	NU	NU	NU	NU	NU	NU	NU
Internal Decimal (ID)	NU	NU	NU	NU	NU	NU	NU	NU	NU
Internal Floating-Point (IFP)	NU	NU	NU	NU	NU	NU	NU	NU	NU
External Floating-Point (EFP)	NU	NU	NU	NU	NU	NU	NU	NU	NU
Floating-Point Literals (FPL)			NU	NU	NU	NU	NU	NU	

Note:

NN = Comparison for nonnumeric operands
NU = Comparison for numeric operands
Blank = Comparison is not allowed
(1)Integer item only
(2)Includes all figurative constants except ZERO

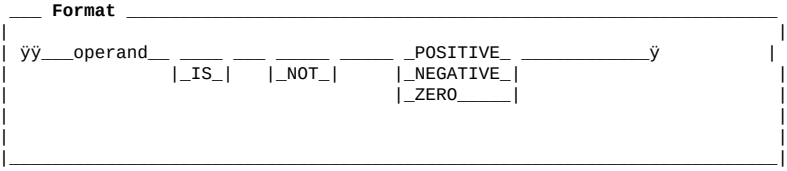
Table 13. Comparisons for Index Names and Index Data Items

Operands Compared	**Index-Name**	**Index Data Item**	**Data-Name (Numeric Integer Only)**	**Literal (Numeric Integer Only)**	**Arithmetic Expression**

Index-Name	Compare occurrence number	Compare without conversion	Compare occurrence number with data-name	Compare occurrence number with literal	Compare occurrence number with arithmetic expression
Index Data Item	Compare without conversion	Compare without conversion	Illegal	Illegal	Illegal

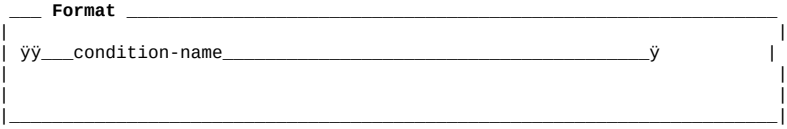
3.1.7.8 Sign Condition

Format



3.1.7.9 Switch-Status Condition

Format



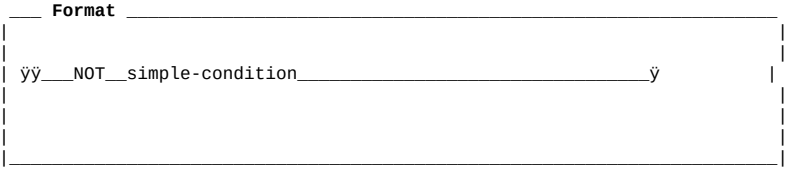
3.1.7.10 Complex Conditions

Table 14. Logical Operators and Their Meanings

Logical Operator	**Name**	**Meaning**
AND	Logical conjunction	The truth value is **true** when both conditions are true.
OR	Logical inclusive OR	The truth value is **true** when either or both conditions are true.
NOT	Logical negation	Reversal of truth value (the truth value is **true** if the condition is false).

3.1.7.11 Negated Simple Conditions

Format



3.1.7.12 Combined Conditions

Format

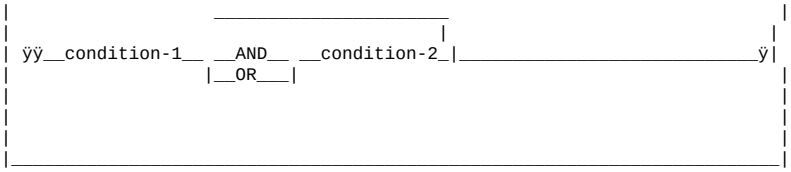


Table 15. Logical Operators and Evaluation Results of Combined Conditions

Value For C1	Value For C2	C1 AND C2	C1 OR C2	NOT (C1 AND C2)	NOT C1 AND C2	NOT (C1 OR C2)	NOT C1 OR C2
True	True	True	True	False	False	False	True
False	True	False	True	True	True	False	True
True	False	False	True	True	False	False	False
False	False	False	False	True	False	True	True

3.1.7.13 Abbreviated Combined Relation Conditions

Format

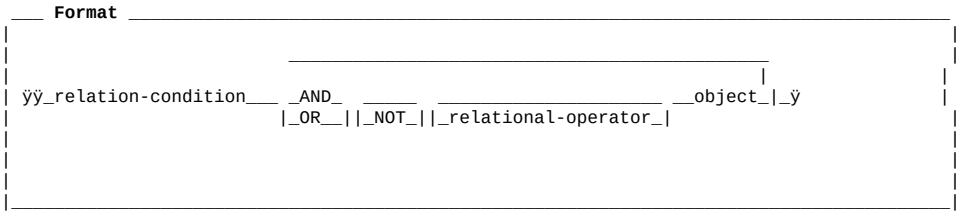


Table 16. Examples of Abbreviated Combined Relation Conditions

Abbreviated Combined Relation Condition	Equivalent
A = B AND NOT < C OR D	((A = B) AND (A NOT < C)) OR (A NOT < D)
A NOT > B OR C	(A NOT > B) OR (A NOT > C)
NOT A = B OR C	(NOT (A = B)) OR (A = C)
NOT (A = B OR < C)	NOT ((A = B) OR (A < C))
NOT (A NOT = B AND C AND NOT D)	NOT (((A NOT = B) AND (A NOT = C)) AND (NOT (A NOT = D)))

3.1.7.14 Status Key Values and Meanings

Table 17. Status Key Values and Meanings

High-Order Digit	Meaning	Low-Order Digit	Meaning
0	Successful Completion	0	No further information
		2	The input-output statement was successfully executed, but a duplicate key was detected. For a READ statement the key value for the current key of reference was equal to the value of the same key in the next record within the current key of reference. For a REWRITE or WRITE statement, the record just written created a duplicate key value for at least one alternate record key for which duplicates are allowed.
		4	A READ statement was successfully executed, but the record length being processed did not conform to the fixed file attributes for that file.
		5	An OPEN statement is successfully executed but the referenced optional file is not present at the time the OPEN statement is executed. If the open mode is I-O , the file has been created. This does not apply to VSAM sequential files. File status 0 is returned.
		7	For a CLOSE statement with the NO REWIND, REEL/UNIT, or FOR REMOVAL phrase or for an OPEN statement with the NO REWIND phrase, the referenced file was on a non-reel/unit medium. **Note:** The EVERY END/REEL option is ignored.
1	At end condition	0	A sequential READ statement was attempted and no next logical record existed in the file because the end of the file had been reached, or the first READ was attempted on an optional input file that was not present.
		4	A sequential READ statement was attempted for a relative file and the number of significant digits in the relative record number was larger than the size of the relative key data item described for the file.
2	Invalid key condition	1	A sequence error exists for a sequentially accessed indexed file. The prime record key value has been changed by the program between the successful execution of a READ statement and the execution of the next REWRITE statement for that file, or the ascending requirements for successive record key values were violated.
		2	An attempt was made to write a record that would create a duplicate key in a relative file; or an attempt was made to write or rewrite a record that would create a duplicate prime record key or a duplicate alternate record key without the DUPLICATES phrase in an indexed file. This key value applies to an indexed file in which the alternate key has been declared 'UNIQUE'.
		3	An attempt was made to randomly access a record that does not exist in the file, or a START or random READ statement was attempted on an optional input file that was not present.
		4	An attempt was made to write beyond the externally defined boundaries of a relative or indexed file. Or, a sequential WRITE statement was attempted for a relative file and the number of significant digits in the relative record number was larger than the size of the

			relative key data item described for the file.
3	Permanent error condition	0	No further information
		4	A permanent error exists because of a boundary violation; an attempt was made to write beyond the externally-defined boundaries of a sequential file.
		5	An OPEN statement with the INPUT, I-O, or EXTEND phrase was attempted on a nonoptional file that was not present.
		7	An OPEN statement was attempted on a file that would not support the open mode specified in the OPEN statement. Possible violations are: 1. The EXTEND or OUTPUT phrase was specified but the file would not support write operations. 2. The I-O phrase was specified but the file would not support the input and output operations permitted. 3. The INPUT phrase was specified but the file would not support read operations.
		8	An OPEN statement was attempted on a file previously closed with lock.
3	Permanent error condition	9	The OPEN statement was unsuccessful because a conflict was detected between the fixed file attributes and the attributes specified for that file in the program. These attributes include the organization of the file (sequential, relative, or indexed), the prime record key, the code set, the maximum record size, the record type (fixed or variable), and the blocking factor.
4	Logic error condition	1	An OPEN statement was attempted for a file in the open mode.
		2	A CLOSE statement was attempted for a file not in the open mode.
		3	For a mass storage file in the sequential access mode, the last input-output statement executed for the associated file prior to the execution of a REWRITE statement was not a successfully executed READ statement. For relative and indexed files in the sequential access mode, the last input-output statement executed for the file prior to the execution of a DELETE or REWRITE statement was not a successfully executed READ statement.
		4	A boundary violation exists because an attempt was made to rewrite a record to a file and the record was not the same size as the record being replaced, or an attempt was made to write or rewrite a record that was larger than the largest or smaller than the smallest record allowed by the RECORD IS VARYING clause of the associated file-name.
4	Logic error condition	6	A sequential READ statement was attempted on a file open in the input or I-O mode and no valid next record had been established because: 1. The preceding READ statement was unsuccessful but did not cause an at end condition. 2. The preceding READ statement caused an at end condition.
		7	The execution of a READ statement was attempted on a file not open in the input or I-O mode.
		8	The execution of a WRITE statement was attempted on a file not open in the I-O,

			output, or extend mode.
		9	The execution of a DELETE or REWRITE statement was attempted on a file not open in the I-O mode.
9	Implementor-defined condition	0	No further information.
		1	For VSAM only: Password failure.
		2	Logic error.
		3	For VSAM only: Resource not available.
		4	For VSAM with CMPR2 compiler-option only: No file position indicator for sequential request.
		5	For VSAM only: Invalid or incomplete file information.
		6	For VSAM and SAM: No DLBL statement specified for this file.
		7	For VSAM only: OPEN statement execution successful: File integrity verified.

3.2 Intrinsic Functions

Argument types and function types are abbreviated as follows:

A = alphabetic
I = integer
N = numeric
X = alphanumeric

Table 18. Table of Functions

Function-name	***Arguments***	***Type***	***Value returned***
ACOS	N1	N	Arccosine of N1
ANNUITY	N1, I2	N	Ratio of annuity paid for I2 periods at interest of N1 to initial investment of one
ASIN	N1	N	Arcsine of N1
ATAN	N1	N	Arctangent of N1
CHAR	I1	X	Character in position I1 of program collating sequence
COS	N1	N	Cosine of N1
CURRENT-DATE	None	X	Current date and time and difference from Greenwich Mean Time
DATE-OF-INTEGER	I1	I	Standard date equivalent (YYYYMMDD) of integer date
DAY-OF-INTEGER	I1	I	Julian date equivalent (YYYYDDD) of integer date
FACTORIAL	I1	I	Factorial of I1
INTEGER	N1	I	The greatest integer not greater than N1
INTEGER-OF-DATE	I1	I	Integer date equivalent of standard date (YYYYMMDD)
INTEGER-OF-DAY	I1	I	Integer date equivalent of Julian date (YYYYDDD)
INTEGER-PART	N1	I	Integer part of N1
LENGTH	A1 or N1 or	I	Length of argument

	X1		
LOG	N1	N	Natural logarithm of N1
LOG10	N1	N	Logarithm to base 10 of N1
LOWER-CASE	A1 or X1	X	All letters in the argument are set to lowercase
MAX	A1... or I1...or N1...or X1...	X I N X	Value of maximum argument; note that the type of function depends on the arguments
MEAN	N1...	N	Arithmetic mean of arguments
MEDIAN	N1...	N	Median of arguments
MIDRANGE	N1...	N	Mean of minimum and maximum arguments
MIN	A1... or I1...or N1...or X1...	X I N X	Value of minimum argument; note that the type of function depends on the arguments
MOD	I1,I2	I	I1 modulo I2
NUMVAL	X1	N	Numeric value of simple numeric string
NUMVAL-C	X1 X1,X2	N	Numeric value of numeric string with optional commas and currency sign
ORD	A1 or X1	I	Ordinal position of the argument in collating sequence
ORD-MAX	A1... or N1...or X1...	I	Ordinal position of maximum argument
ORD-MIN	A1...or N1...or X1...	I	Ordinal position of minimum argument
PRESENT-VALUE	N1 N2...	N	Present value of a series of future period-end amounts, N2, at a discount rate of N1
RANDOM	I1 or none	N	Random number
RANGE	I1... or N1...	I N	Value of maximum argument minus value of minimum argument; note that the type of function depends on the arguments.
REM	N1,N2	N	Remainder of N1/N2
REVERSE	A1 or X1	X	Reverse order of the characters of the argument
SIN	N1	N	Sine of N1
SQRT	N1	N	Square root of N1
STANDARD-DEVIATION	N1...	N	Standard deviation of arguments
SUM	I1... or N1...	I N	Sum of arguments; note that the type of function depends on the arguments.
TAN	N1	N	Tangent of N1
UPPER-CASE	A1 or X1	X	All letters in the argument are set to uppercase

VARIANCE	N1...	N	Variance of arguments
WHEN-COMPILED	None	X	Date and time when program was compiled

Subtopics:

3.2.1 Language Elements that Allow Function Identifiers

A number of COBOL language elements allow the use of function identifiers. The following underlined elements in the COBOL language syntax allow a numeric function identifier:

1. **Reference Modification**

   (leftmost-character-position : length)

2. **Conditional Expressions, Relation Condition**

   arithmetic-expression-1 IS (condition) arithmetic-expression-2

3. **Conditional Expressions, Sign Condition**

   arithmetic-expression-1 IS (NOT) POSITIVE|NEGATIVE|ZERO

4. **COMPUTE Statement**

   COMPUTE ... = arithmetic-expression-1

5. **COPY Statement**

   COPY ... REPLACING identifier-1 BY identifier-2
   (only text replacement to be done)

   Numeric and integer values are also allowed.

6. **EVALUATE Statement**

   EVALUATE expression-1 ALSO expression-2
     WHEN arithmetic-expression-1 THRU arithmetic-expression-2
     ALSO arithmetic-expression-3 THRU arithmetic-expression-4

7. **PERFORM Statement**

   PERFORM...UNTIL condition-1

8. **SEARCH Statement**

   SEARCH ALL ... WHEN ... IS = arithmetic-expression-1 AND
                       ... IS = arithmetic-expression-2

9. A numeric function identifier can be used as an argument to another function where numeric arguments are allowed.

   FUNCTION function-name-1 [({argument-1}...)]

The following underlined elements in the COBOL language syntax allow an alphanumeric function identifier:

1. **Conditional Expressions, Class Condition**

   IF identifier-1 IS NUMERIC|ALPHABETIC|etc.

2. **Conditional Expressions, Relation Condition**

   identifier-1 IS (NOT) (condition) identifier-2

3. **CALL Statement**

   CALL identifier-1

4. **CANCEL Statement**

   CANCEL identifier-1

5. **COPY Statement**

   COPY ... REPLACING identifier-1 BY identifier-2
   (only text replacement to be done)

6. **DISPLAY Statement**

   DISPLAY identifier-1

7. **EVALUATE Statement**

   EVALUATE identifier-1 ALSO identifier-2
     WHEN identifier-3 THRU identifier-4
     ALSO identifier-5 THRU identifier-6

8. **INITIALIZE Statement**

   INITIALIZE ... REPLACING ... BY identifier-2

9. **INSPECT Statement**

   INSPECT identifier-1
   (without the REPLACING or CONVERTING phrases)

   INSPECT ... TALLYING ... FOR ALL|LEADING identifier-3
   INSPECT ... BEFORE|AFTER INITIAL identifier-4
   INSPECT ... REPLACING CHARACTERS BY identifier-5
   INSPECT ... REPLACING ALL|LEADING|FIRST identifier-3 BY
                identifier-5
   INSPECT ... CONVERTING identifier-6 TO identifier-7

10. **MOVE Statement**

    MOVE identifier-1 TO ...

11. **PERFORM Statement**

    PERFORM...UNTIL condition-1

12. **RELEASE Statement**

    RELEASE record-name FROM identifier

13. **REWRITE Statement**

    REWRITE record-name FROM identifier-1

14. **SEARCH Statement**

    SEARCH ALL ... WHEN data-name-1 IS EQUAL TO identifier-3 AND

```
                        data-name-2 IS EQUAL TO identifier-4
```

15. **STRING Statement**

```
    STRING identifier-1 ... DELIMITED BY identifier-2 ...
```

16. **UNSTRING Statement**

```
    UNSTRING identifier-1
      DELIMITED BY identifier-2 OR (ALL) identifier-3 ...
```

17. **WRITE Statement**

```
    WRITE record-name FROM identifier-1 ...
```

18. An alphanumeric function identifier can be used as an argument to another function where alphanumeric arguments are allowed.

```
    FUNCTION function-name-1  [({argument-1}... )]  [reference modifier]
```

3.2.2 List of Functions

Subtopics:

3.2.2.1 ACOS

___ **Format** _____

```
|                                                                          |
| ÿÿ___FUNCTION ACOS (argument-1)_____________________ÿ                    |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

3.2.2.2 ANNUITY

```
 ___ Format _____________________________________________________________
|                                                                          |
| ÿÿ___FUNCTION ANNUITY (argument-1 argument-2)_______ÿ                    |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

3.2.2.3 ASIN

```
 ___ Format _____________________________________________________________
|                                                                          |
| ÿÿ___FUNCTION ASIN (argument-1)_____________________ÿ                    |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

3.2.2.4 ATAN

```
 ___ Format _____________________________________________________________
|                                                                          |
| ÿÿ___FUNCTION ATAN (argument-1)_____________________ÿ                    |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

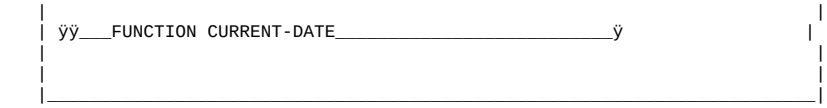
3.2.2.5 CHAR

```
 ___ Format _____________________________________________________________
|                                                                          |
| ÿÿ___FUNCTION CHAR (argument-1)_____________________ÿ                    |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

3.2.2.6 COS

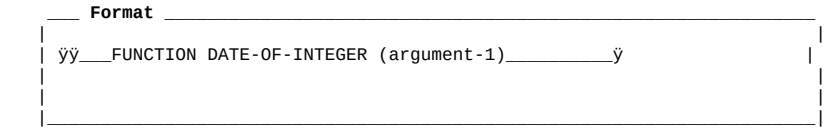
```
 ___ Format _____________________________________________________________
|                                                                          |
| ÿÿ___FUNCTION COS (argument-1)_____________________ÿ                     |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

```
 ___ Format _____________________________________________________________
```



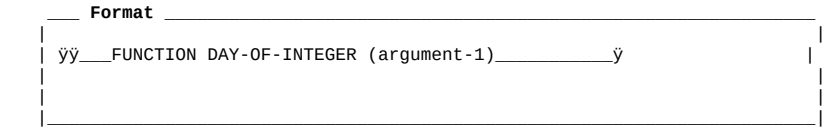
3.2.2.7 DATE-OF-INTEGER

Format



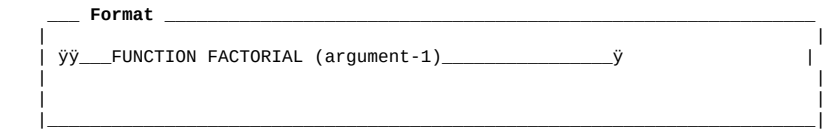
3.2.2.8 DAY-OF-INTEGER

Format



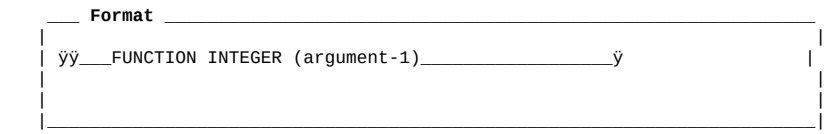
3.2.2.9 FACTORIAL

Format



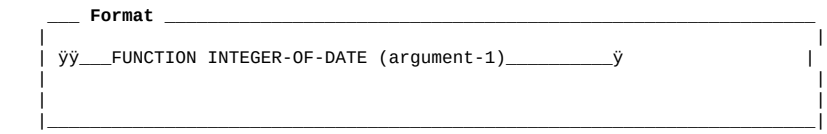
3.2.2.10 INTEGER

Format



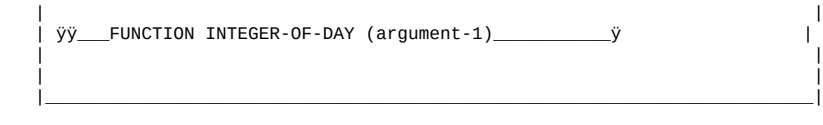
3.2.2.11 INTEGER-OF-DATE

Format



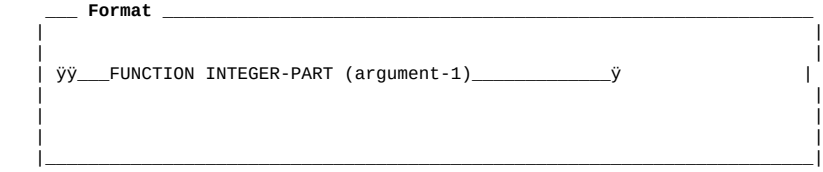
3.2.2.12 INTEGER-OF-DAY

Format



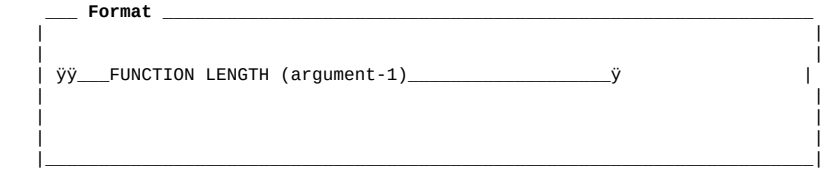
3.2.2.13 INTEGER-PART

Format



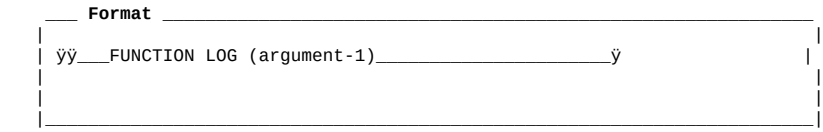
3.2.2.14 LENGTH

Format



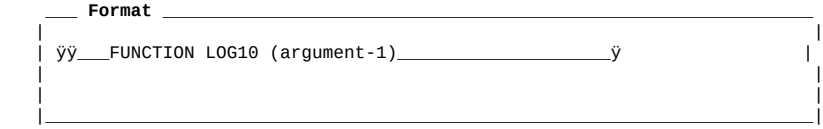
3.2.2.15 LOG

Format



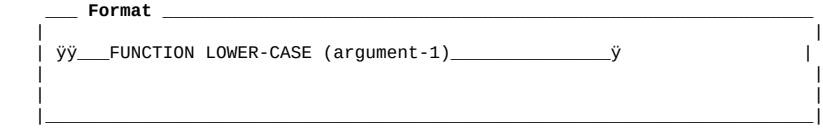
3.2.2.16 LOG10

Format

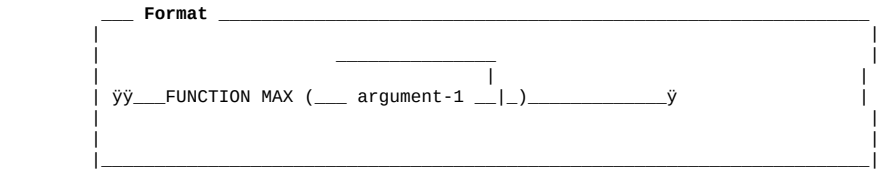


3.2.2.17 LOWER-CASE

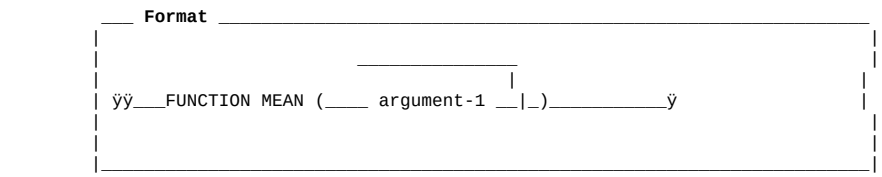
Format



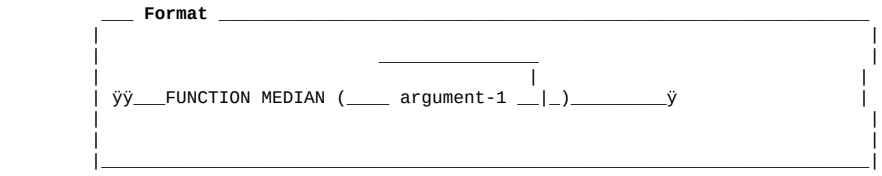
3.2.2.18 MAX



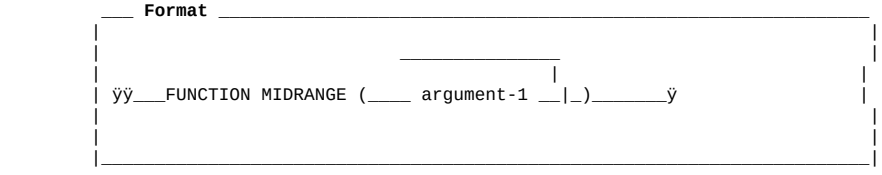
3.2.2.19 MEAN



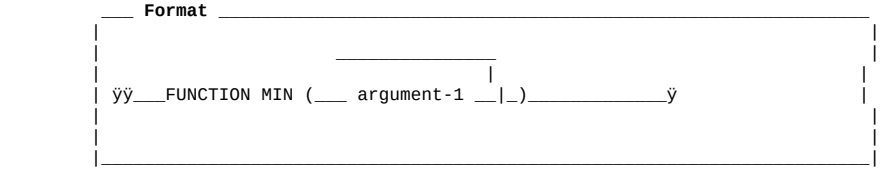
3.2.2.20 MEDIAN



3.2.2.21 MIDRANGE

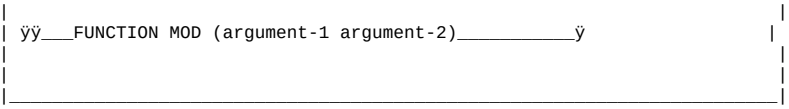


3.2.2.22 MIN



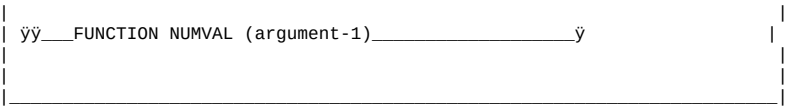
3.2.2.23 MOD

Format

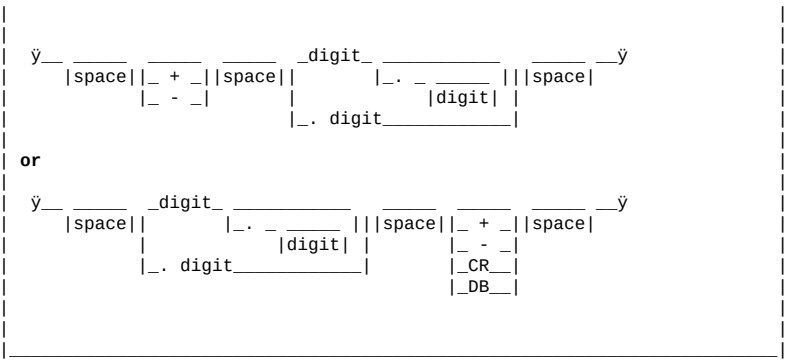


3.2.2.24 NUMVAL

Format

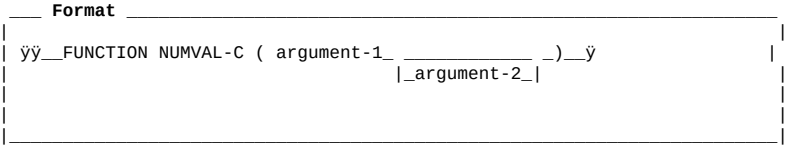


Argument-1
must be a nonnumeric literal or an alphanumeric data item whose content has the following formats:

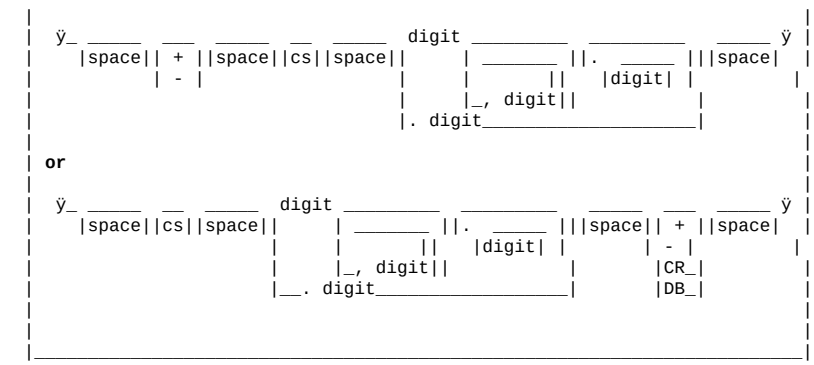


3.2.2.25 NUMVAL-C

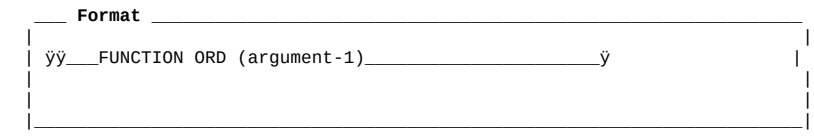
Format



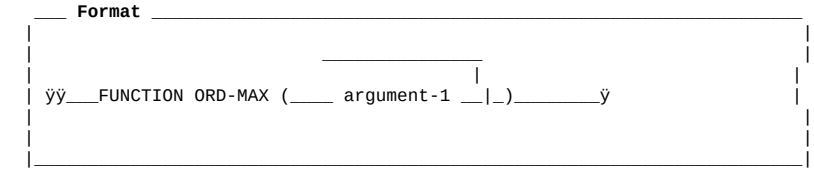
Argument-1
must be a nonnumeric literal or an alphanumeric data item whose content has the following formats:



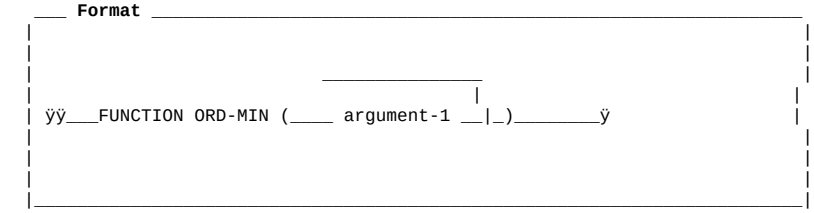
3.2.2.26 ORD



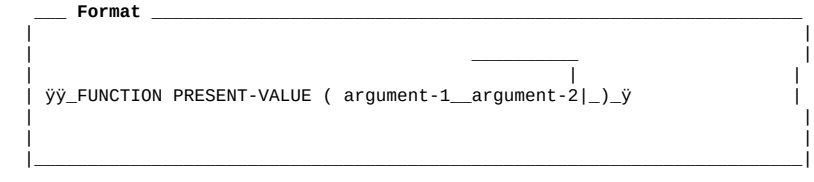
3.2.2.27 ORD-MAX



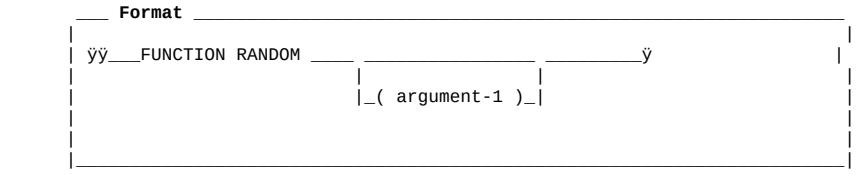
3.2.2.28 ORD-MIN



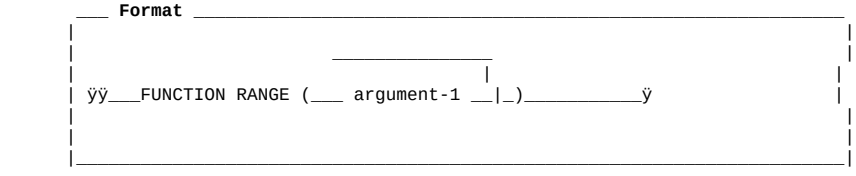
3.2.2.29 PRESENT-VALUE



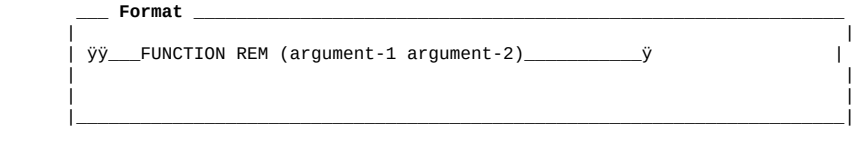
3.2.2.30 RANDOM



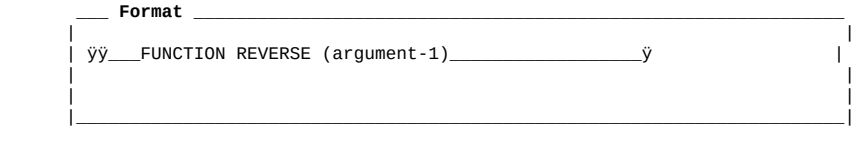
3.2.2.31 RANGE



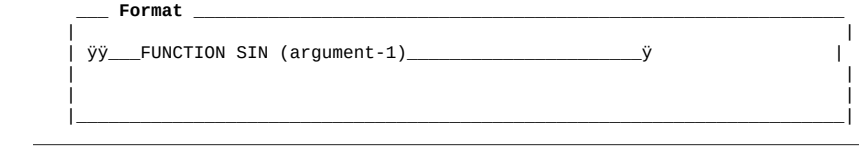
3.2.2.32 REM



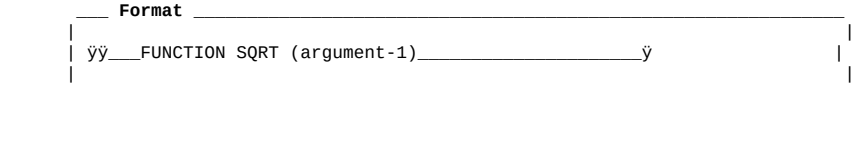
3.2.2.33 REVERSE



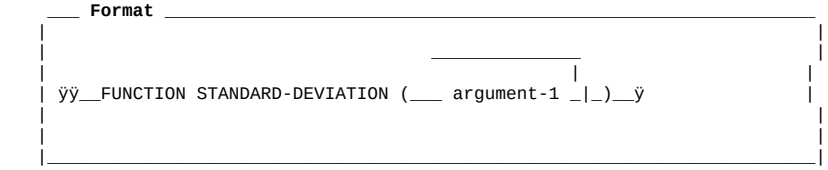
3.2.2.34 SIN



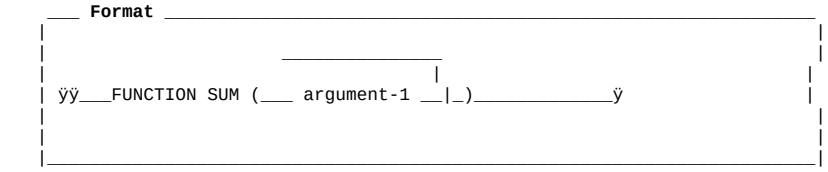
3.2.2.35 SQRT



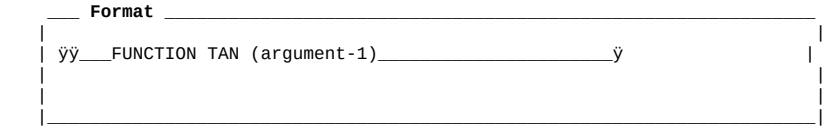
3.2.2.36 STANDARD-DEVIATION



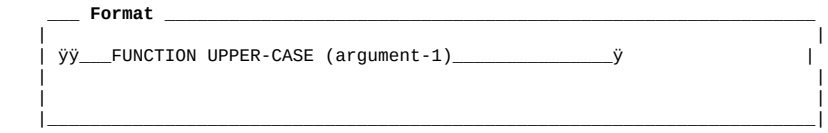
3.2.2.37 SUM



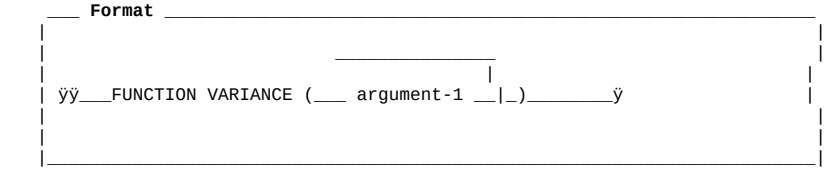
3.2.2.38 TAN



3.2.2.39 UPPER-CASE



3.2.2.40 VARIANCE



3.2.2.41 WHEN-COMPILED

```
 ___ Format ___________________________________________________________
|                                                                      |
| ÿÿ___FUNCTION WHEN-COMPILED________________________ÿ                 |
|                                                                      |
|                                                                      |
|______________________________________________________________________|
```

4.0 Procedure Division Statements

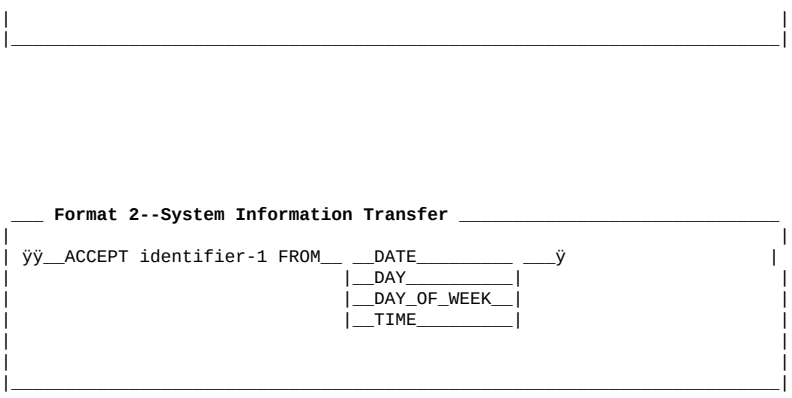
Subtopics:

4.1 Procedure Division Statements

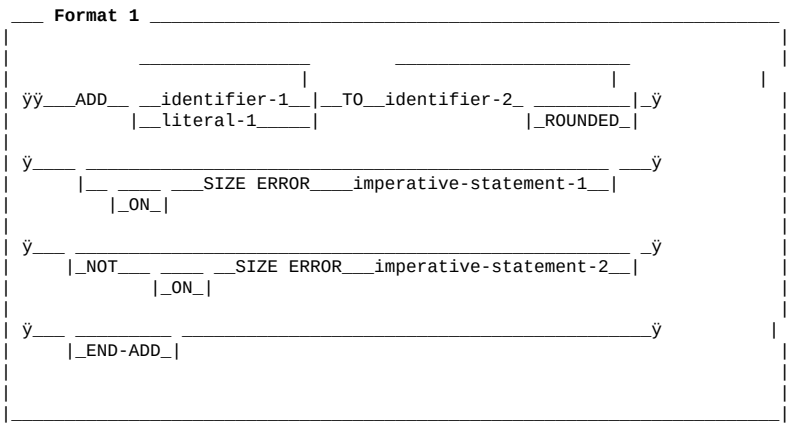
Subtopics:

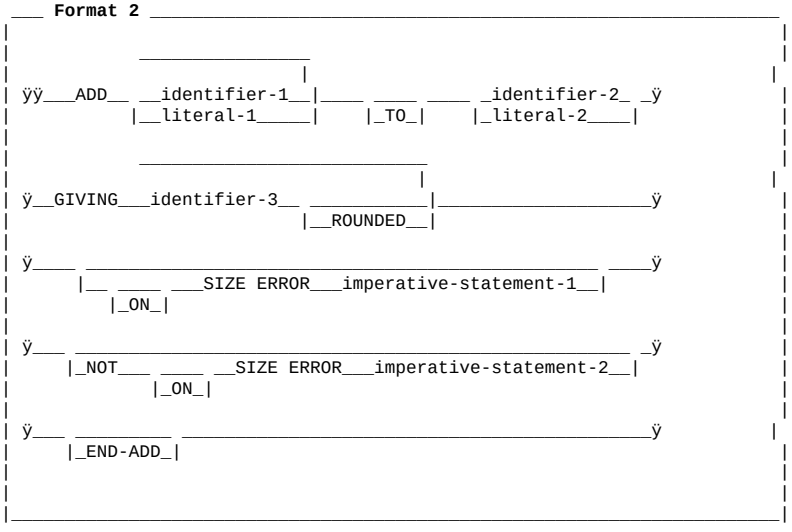
4.1.1 ACCEPT Statement

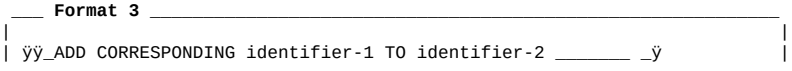
```
 ___ Format 1-Data Transfer ___________________________________________
|                                                                      |
| ÿÿ__ACCEPT identifier__ ______________________________ ___ÿ          |
|                        |_FROM_ __mnemonic-name_____ __|              |
|                               |__environment-name__|                 |
|                                                                      |
```

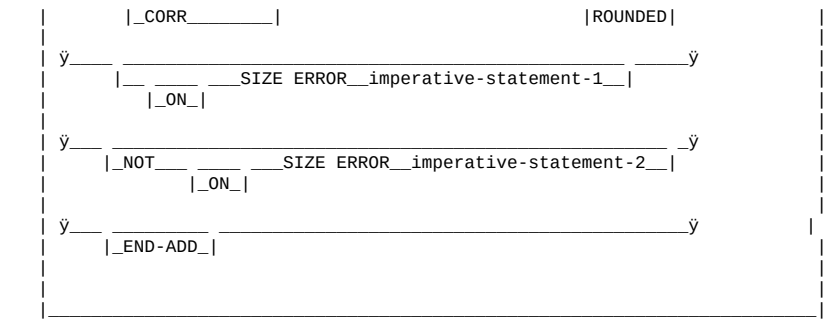


4.1.2 ADD Statement

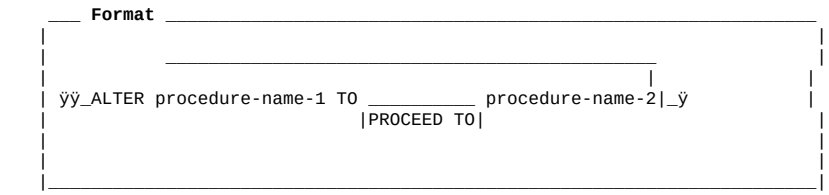




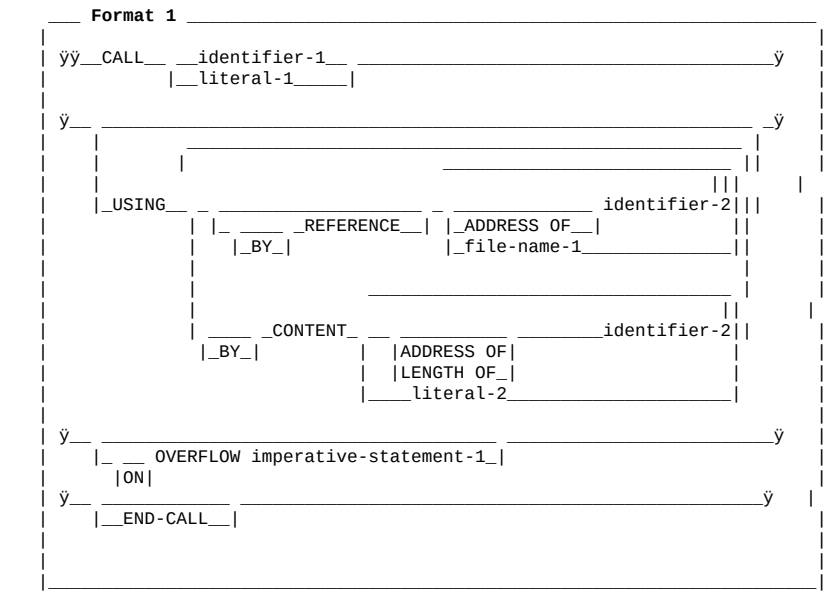


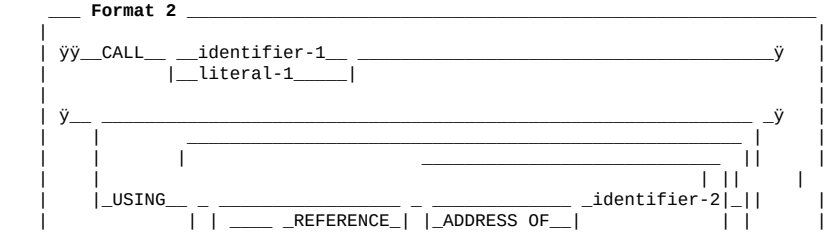


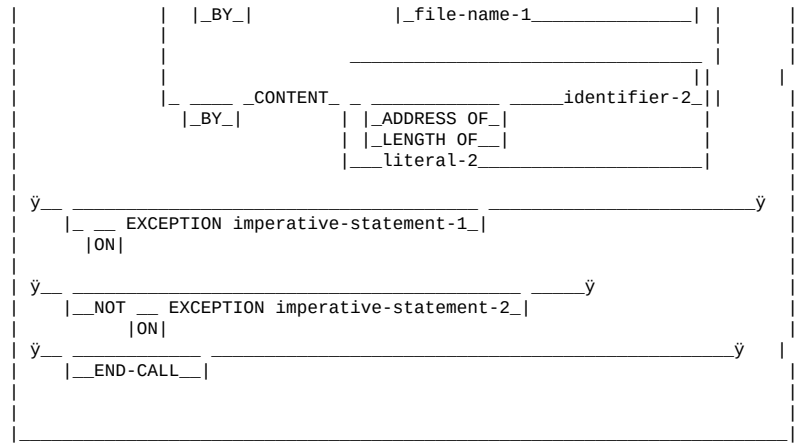
4.1.3 ALTER Statement



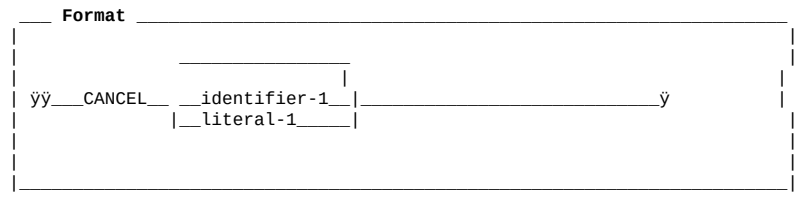
4.1.4 CALL Statement



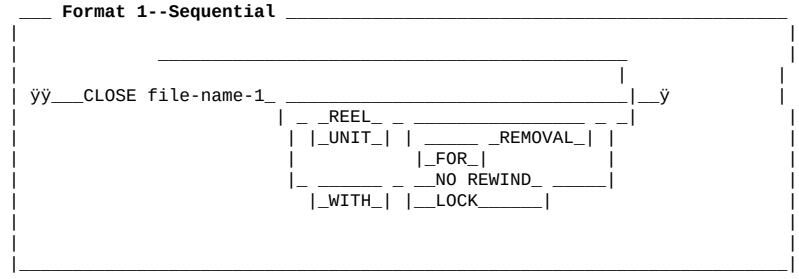




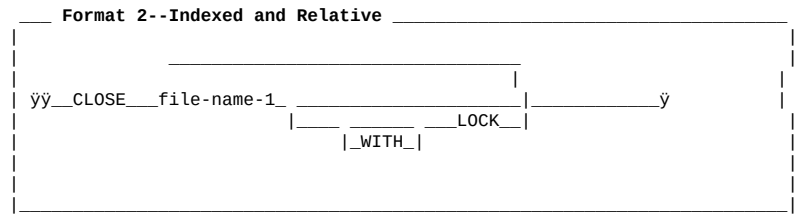
4.1.5 CANCEL Statement



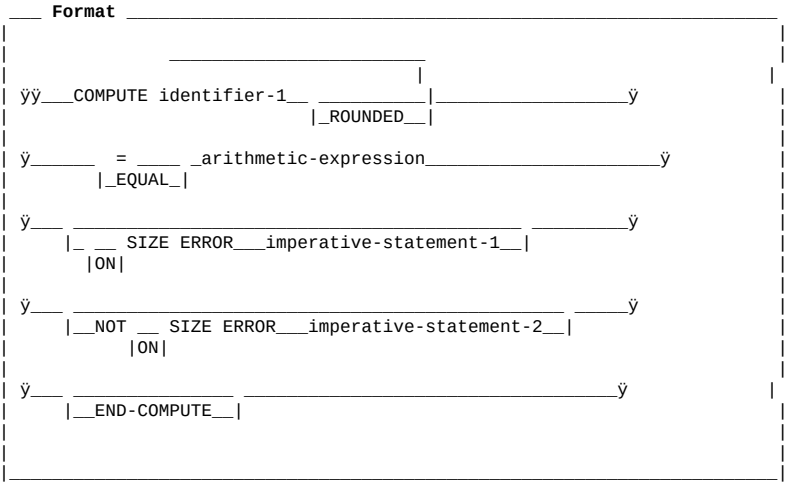
4.1.6 CLOSE Statement



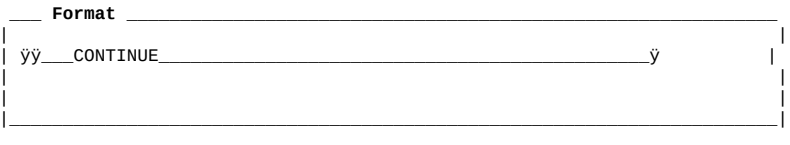
Note: The option EVERY END/REEL is ignored.



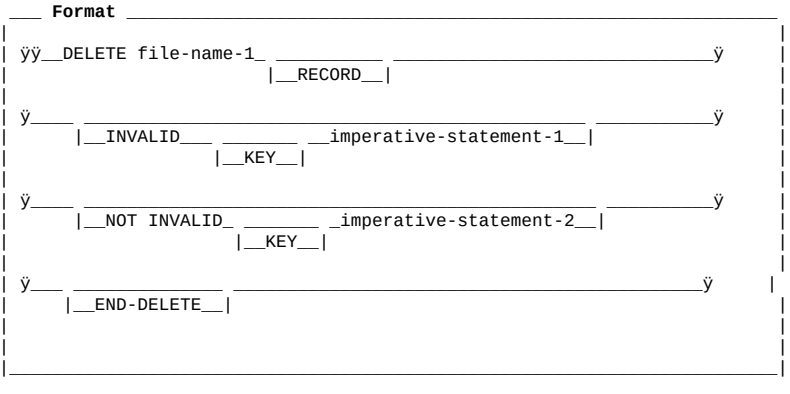
4.1.7 COMPUTE Statement



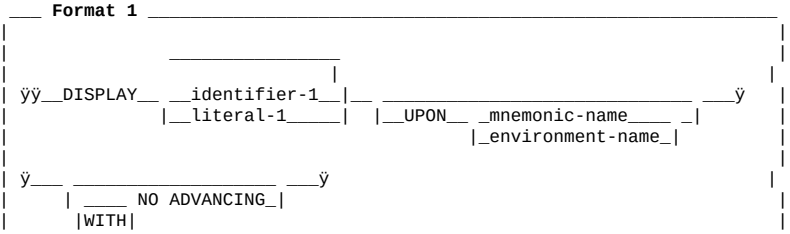
4.1.8 CONTINUE Statement



4.1.9 DELETE Statement

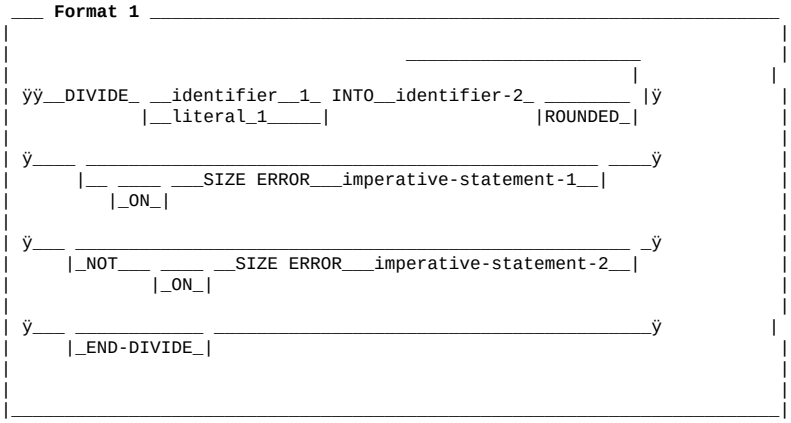


4.1.10 DISPLAY Statement

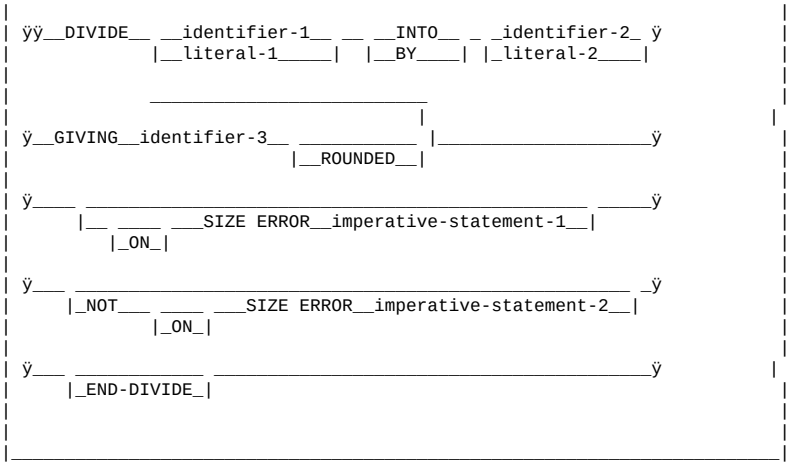


4.1.11 DIVIDE Statement

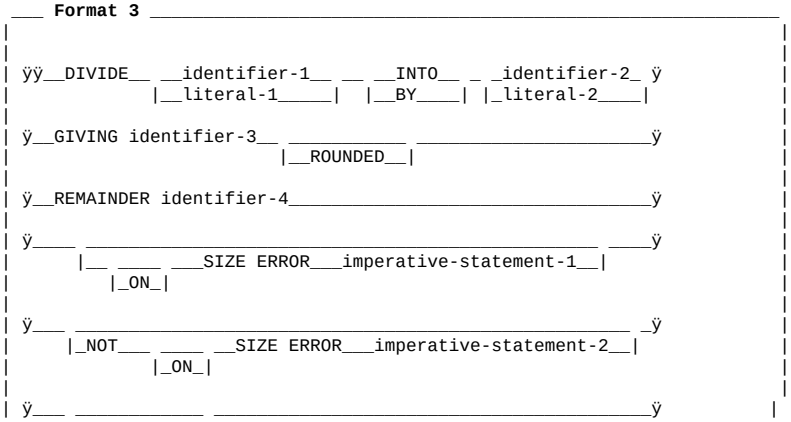
Format 1



Format 2



Format 3



```
|     |_END-DIVIDE_|                                                       |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

4.1.12 ENTRY Statement

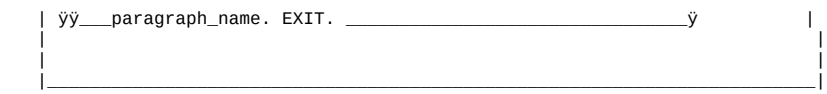
```
 ___ Format _______________________________________________________________
|                                                                          |
| ÿÿ___ENTRY literal__ ______________________ ___ÿ                         |
|                     |          ___________  |                            |
|                     |                     | |                            |
|                     |__USING__data-name-1|_|                             |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

4.1.13 EVALUATE Statement

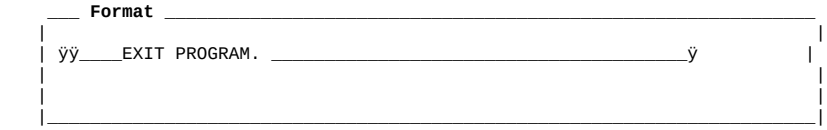
```
 ___ Format _______________________________________________________________
|                                                                          |
| ÿÿ__EVALUATE__ _identifier-1_   __________________________ _ÿ            |
|               |_literal-1____||   ______________________ |               |
|               |_expression-1_||                         ||               |
|               |_TRUE_________||__ALSO__ __identifier-2__||               |
|               |_FALSE________|         |__literal-2_____|                |
|                                        |__expression-2__|                |
|                                        |__TRUE__________|                |
|                                        |__FALSE_________|                |
|                                                                          |
|    ________________________________________________________              |
|   | ________________________________                       |             |
|                                      |                     |             |
| ÿ___WHEN phrase-1 _________________ |imperative-statement-1|ÿ           |
|                  | _______________ |                                     |
|                  |                ||                                     |
|                  |__ALSO phrase-2_||                                     |
|                                                                          |
| ÿ__ ____________________________________ _ ______________ _ÿ             |
|    |__WHEN OTHER imperative-statement-2_| |_END-EVALUATE_|               |
|                                                                          |
|    where  phrase_1  is:                                                  |
|                                                                          |
|   ÿ_ _ANY________________________________________________ _ÿ             |
|     |_condition-1________________________________________|               |
|     |_TRUE_______________________________________________|               |
|     |_FALSE______________________________________________|               |
|     |_ _____ _ identifier-3 __ _________________________ |               |
|       |_NOT_| |literal-3___|  | THROUGH _ _identifier-4 |                |
|               |arith-exp-1_|   |THRU___| |_literal-4___|                 |
|                                          |_arith-exp-2_|                 |
|                                                                          |
|     where phrase-2 is:                                                   |
|                                                                          |
|   ÿ_ _ANY________________________________________________ _ÿ             |
|     |_condition-2________________________________________|               |
|     |_TRUE_______________________________________________|               |
|     |_FALSE______________________________________________|               |
|     |_ _____ _ identifier-5 _ __________________________ |               |
|       |_NOT_| |literal-5___| | THROUGH _ _identifier-6_ |                |
|               |arith-exp-3_|  |THRU___| |_literal-6____|                 |
|                                         |_arith-exp-4__|                 |
|                                                                          |
|                                                                          |
|__________________________________________________________________________|
```

4.1.14 EXIT Statement

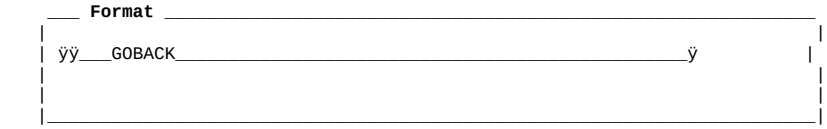
```
 ___ Format _______________________________________________________________
|                                                                          |
```



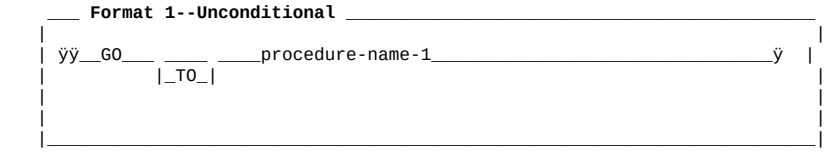
4.1.15 EXIT PROGRAM Statement

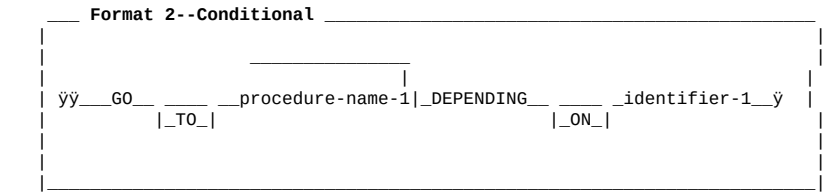


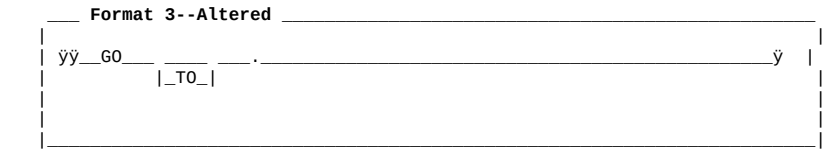
4.1.16 GOBACK Statement

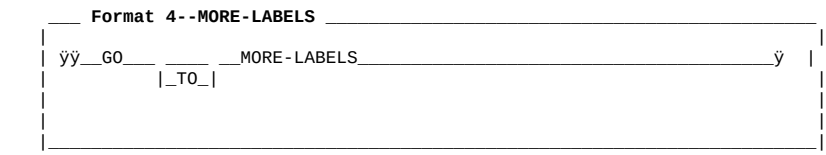


4.1.17 GO TO Statement

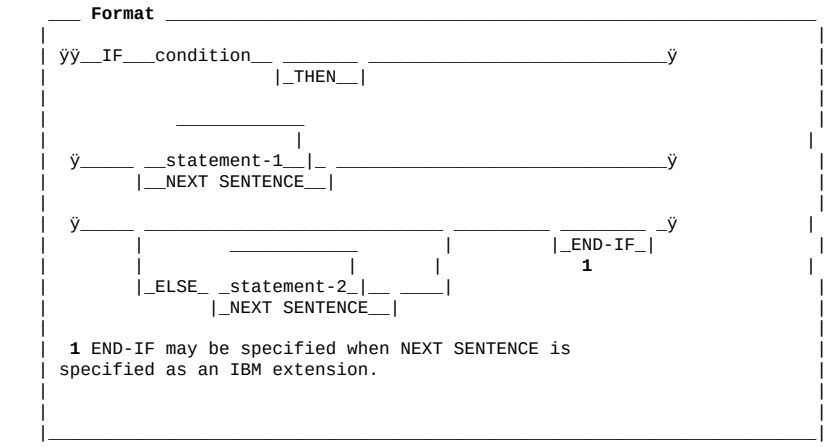




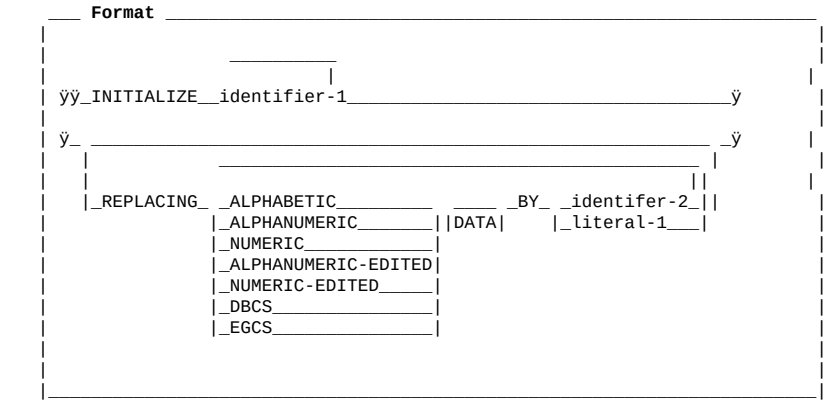




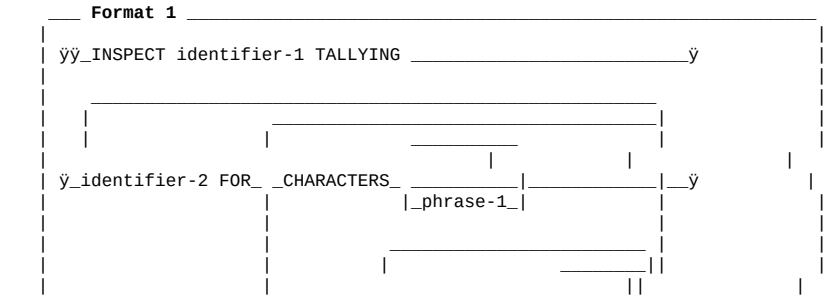
4.1.18 IF Statement

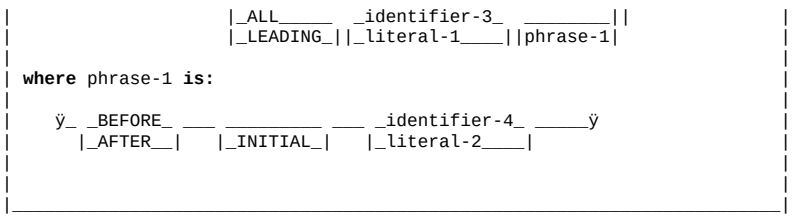


4.1.19 INITIALIZE Statement

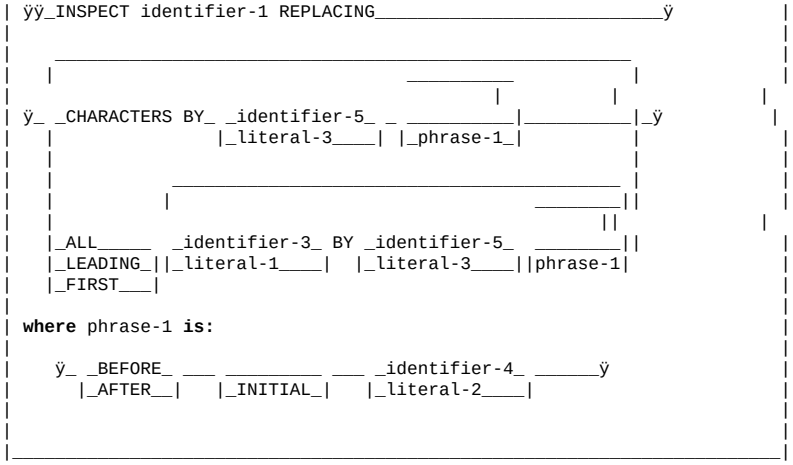


4.1.20 INSPECT Statement

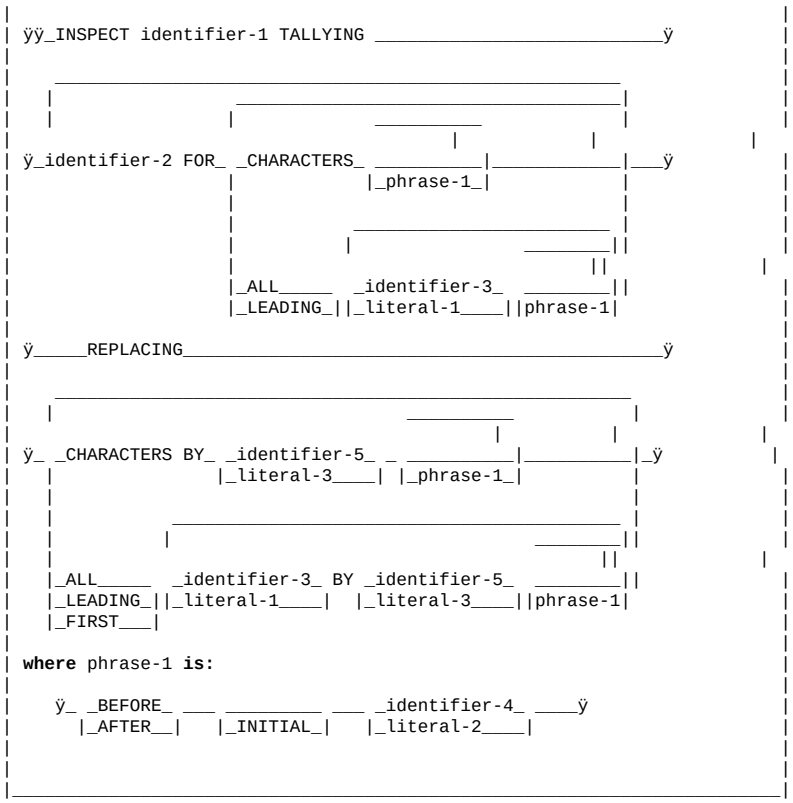




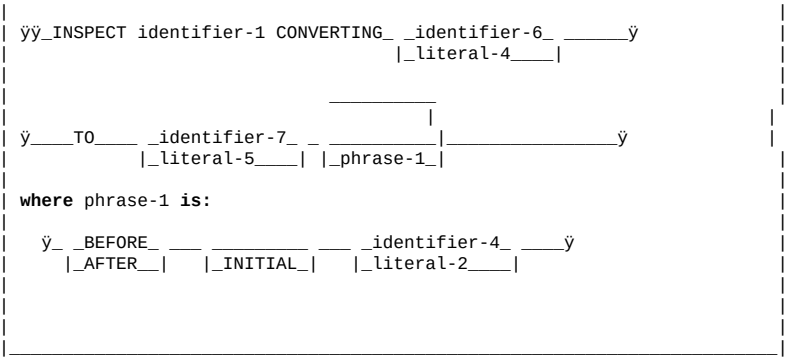
Format 2



Format 3



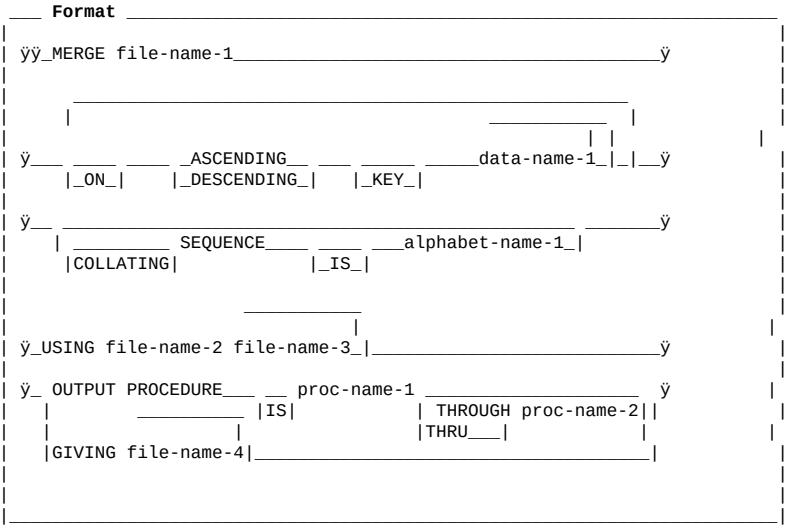
Format 4



where phrase-1 **is:**

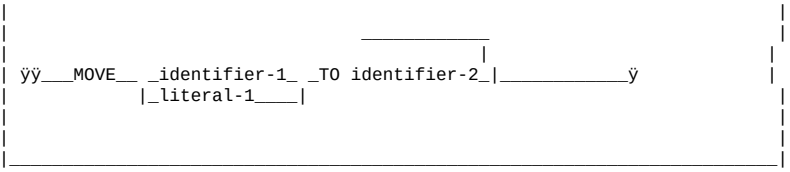
4.1.21 MERGE Statement

Format

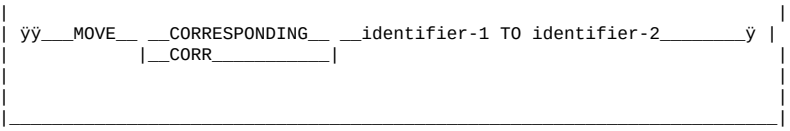


4.1.22 MOVE Statement

Format 1



Format 2



Subtopics:

4.1.22.1 Valid and Invalid Elementary Moves

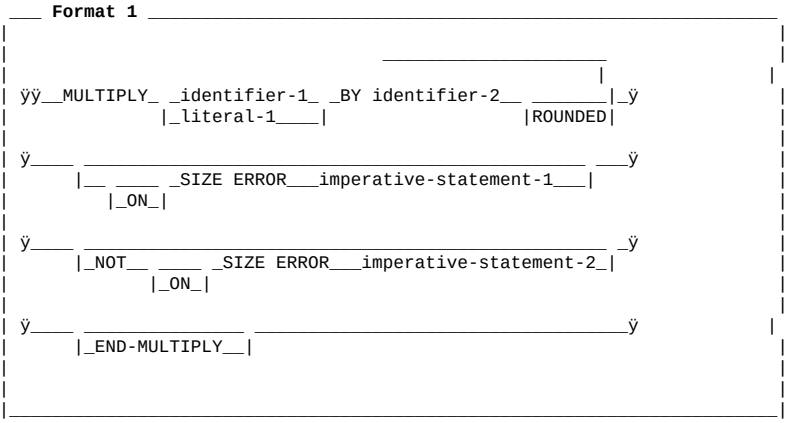
Table 19. Valid and Invalid Elementary Moves

	Receiving Item Category							
Sending Item Category	**Alpha-betic**	**Alpha-numeric**	**Alpha-numeric Edited**	**Numeric**	**Numeric Edited**	**External Floating Point**	**Internal Floating Point**	**DBCS(1)**
Alphabetic and SPACE	Yes	Yes	Yes	No	No	No	No	No
Alphanumeric(2)	Yes	Yes	Yes	Yes(3)	Yes(3)	Yes(8)	Yes(8)	No
Alphanumeric-Edited	Yes	Yes	Yes	No	No	No	No	No
Numeric Integer and ZERO(4)	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Numeric Noninteger(5)	No	No	No	Yes	Yes	Yes	Yes	No
Numeric-Edited	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Floating-Point(6)	No	No	No	Yes	Yes	Yes	Yes	No
DBCS(7)	No	No	No	No	No	No	No	Yes

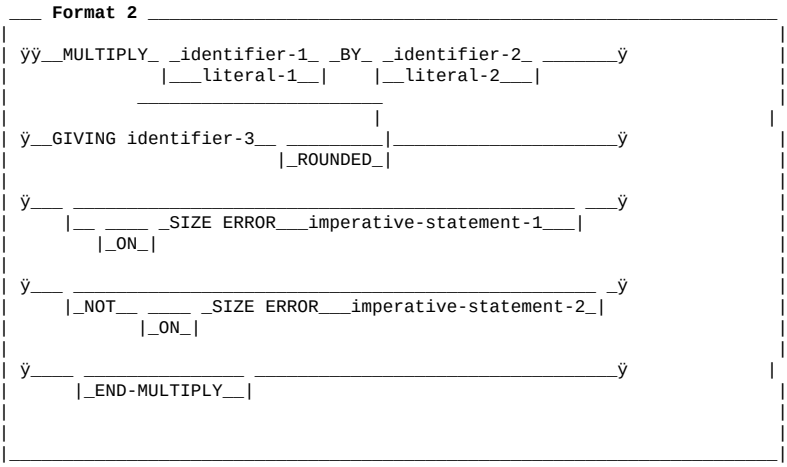
Note:

(1) Includes DBCS data items.
(2) Includes nonnumeric literals.
(3) Figurative constants and nonnumeric literals must consist only of numeric characters and will be treated as numeric integer fields.
(4) Includes integer numeric literals.
(5) Includes noninteger numeric literals.
(6) Includes floating-point literals, external floating-point data items (USAGE DISPLAY), and internal floating-point data items (USAGE COMP-1 or USAGE COMP-2).
(7) Includes DBCS data items, DBCS literals, and SPACE.
(8) Figurative constants and nonnumeric literals must consist only of numeric characters and will be treated as numeric integer fields. The ALL literal may not be used as a sending item.

4.1.23 MULTIPLY Statement

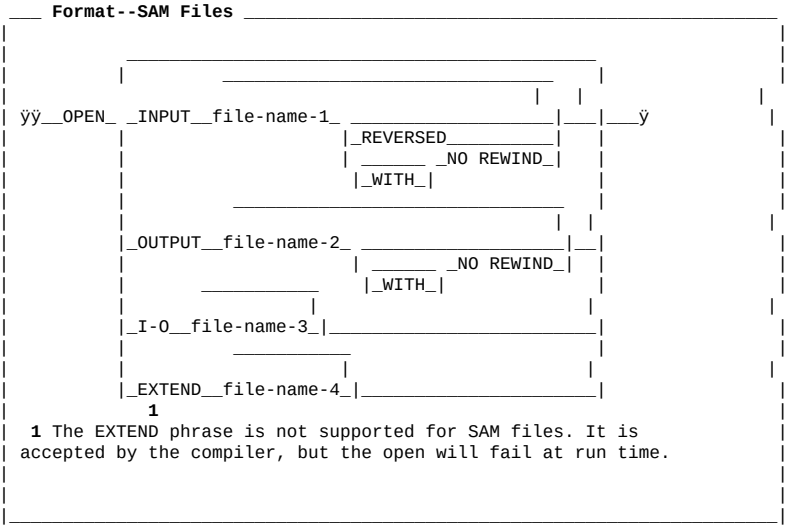


Format 2

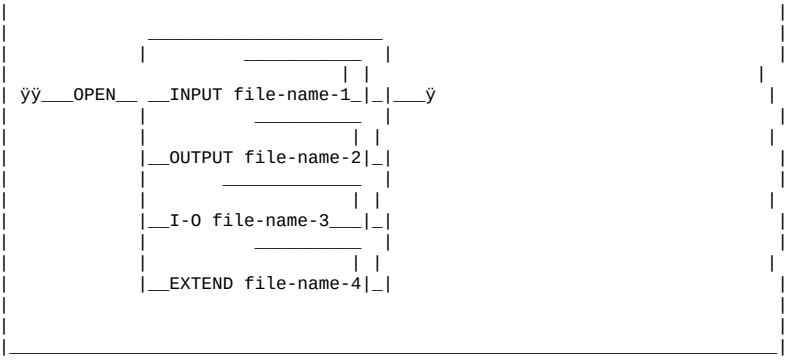


4.1.24 OPEN Statement

Format--SAM Files

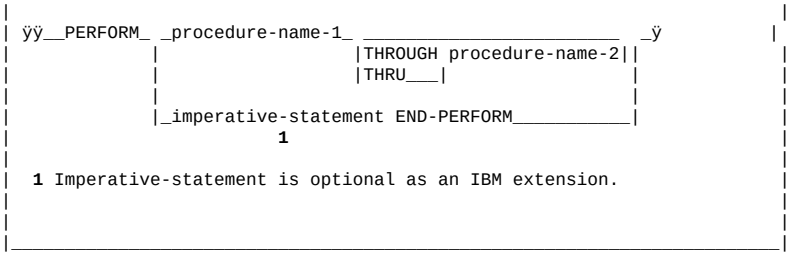


Format--VSAM Files

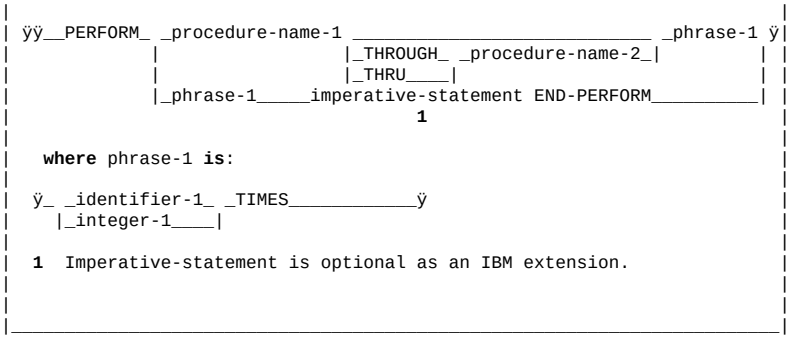


4.1.25 PERFORM Statement

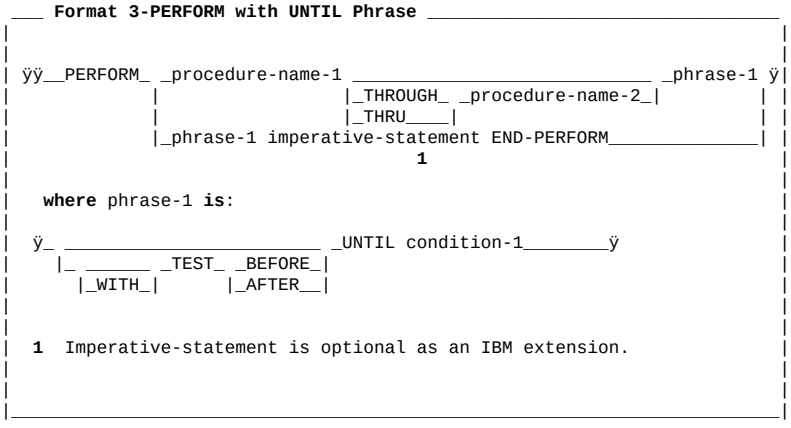
Format 1--Basic PERFORM Statement



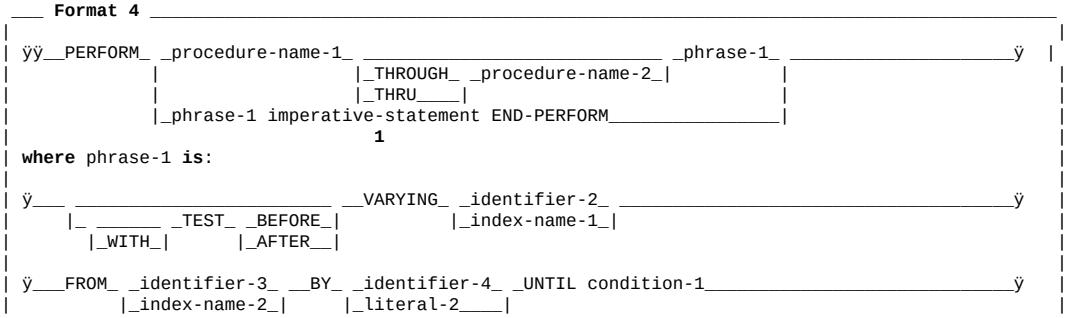
Format 2-PERFORM with TIMES Phrase

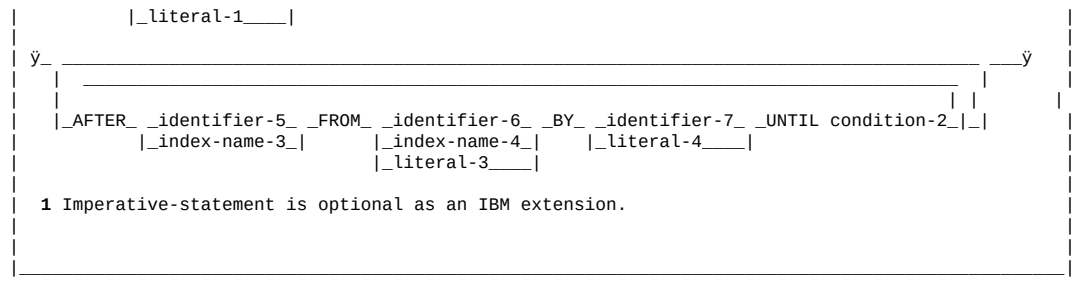


Format 3-PERFORM with UNTIL Phrase

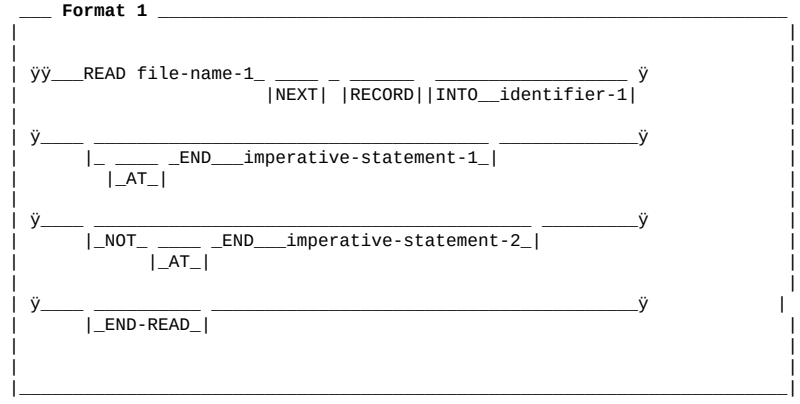


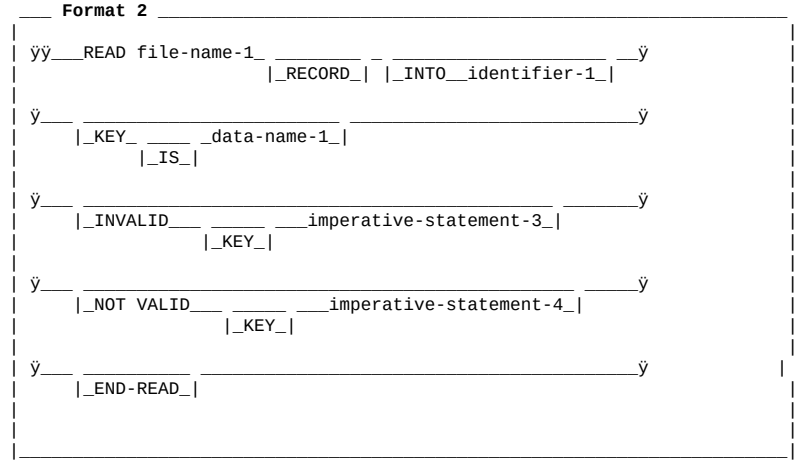
Format 4



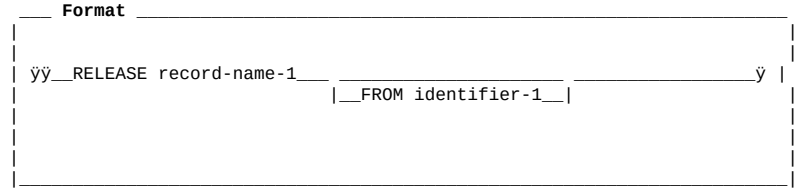


4.1.26 READ Statement

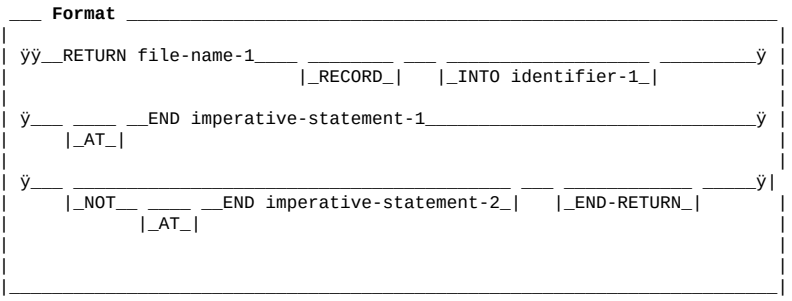




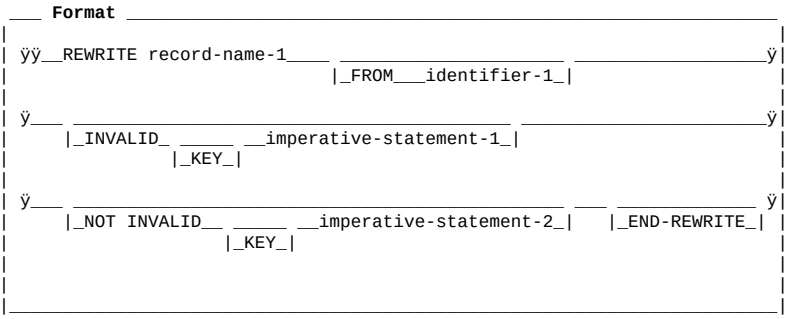
4.1.27 RELEASE Statement



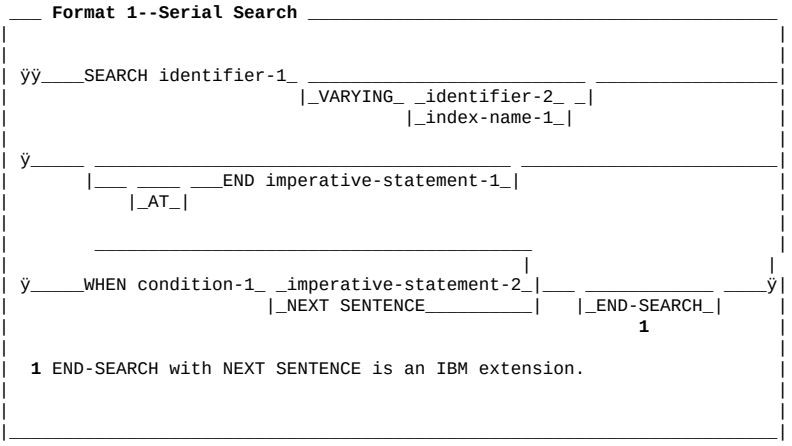
4.1.28 RETURN Statement

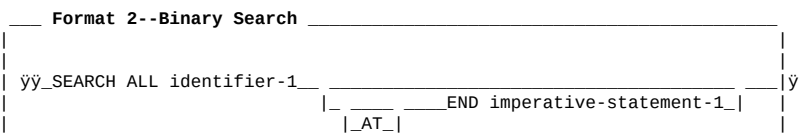


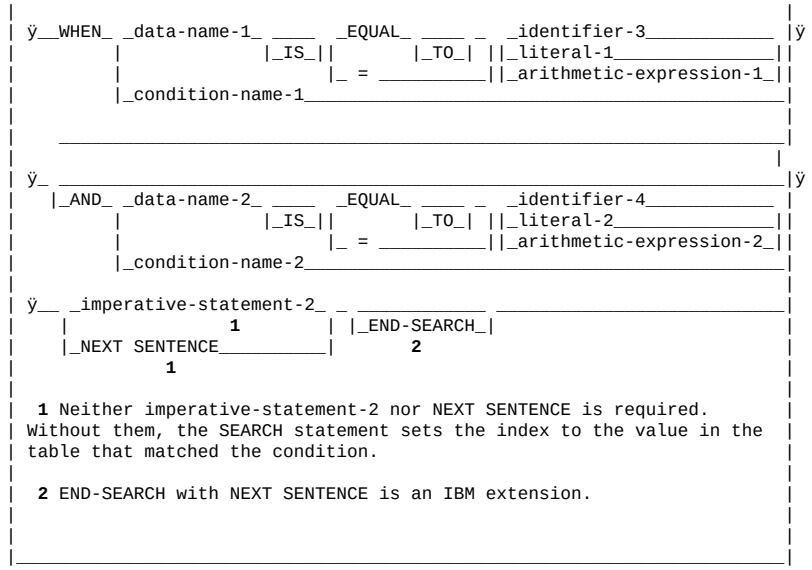
4.1.29 REWRITE Statement



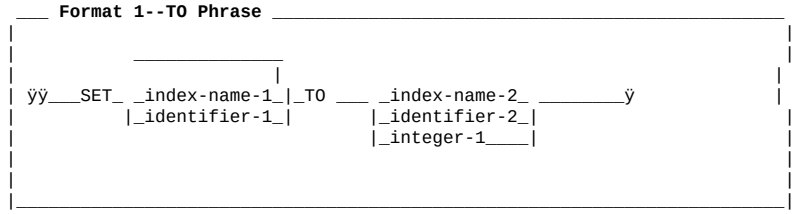
4.1.30 SEARCH Statement

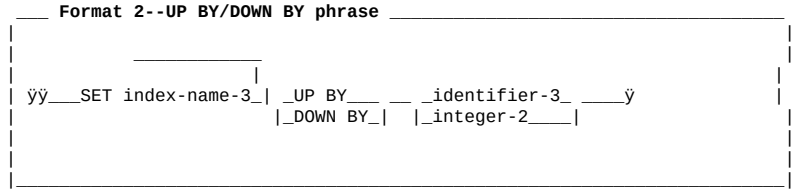


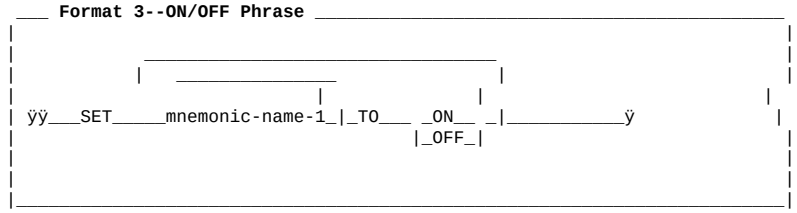


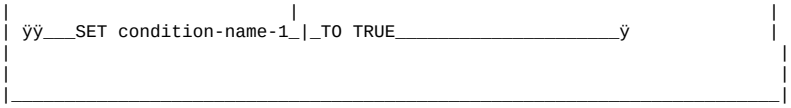


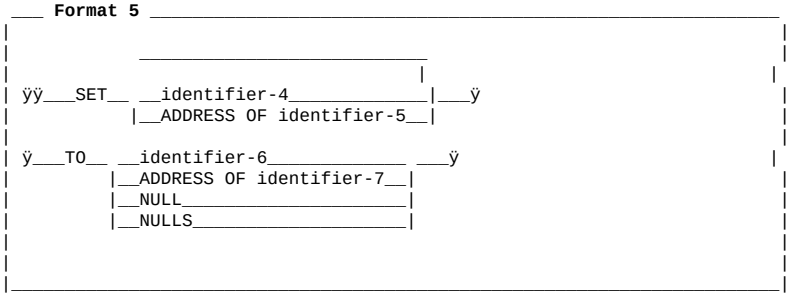
4.1.31 SET Statement

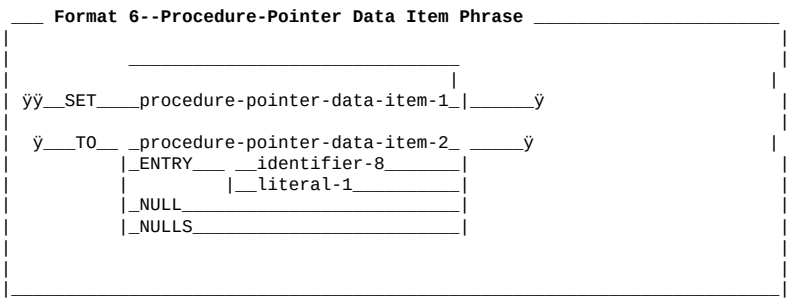




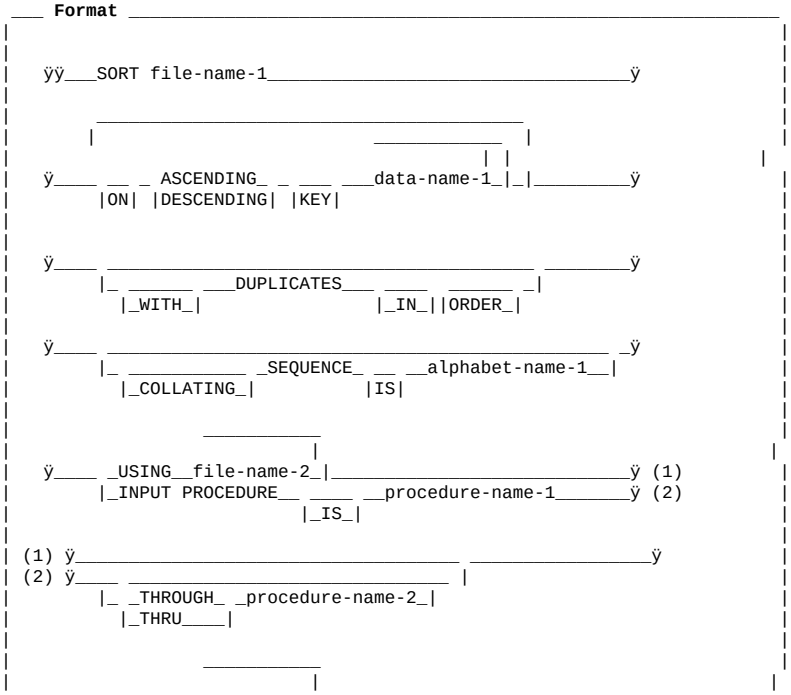


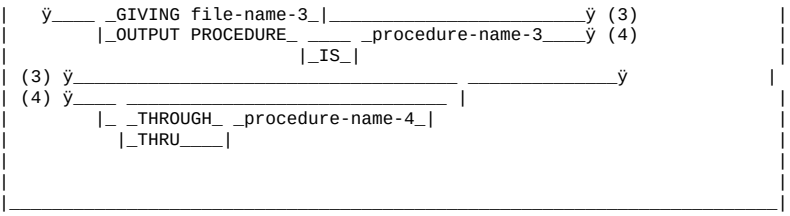




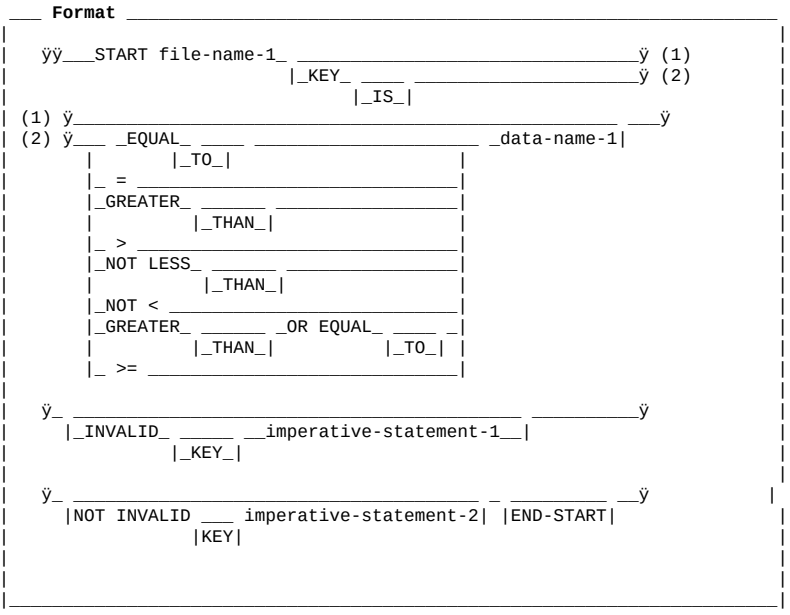


4.1.32 SORT Statement

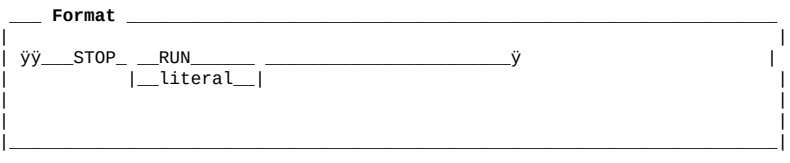




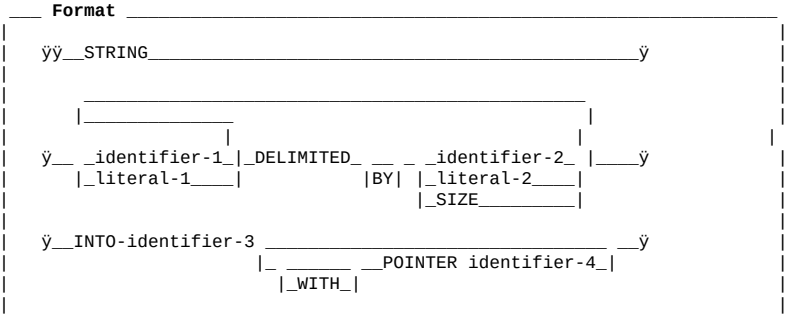
4.1.33 START Statement

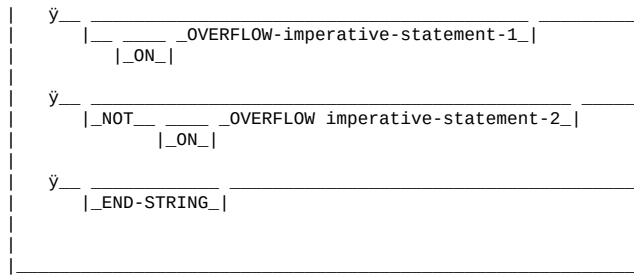


4.1.34 STOP Statement

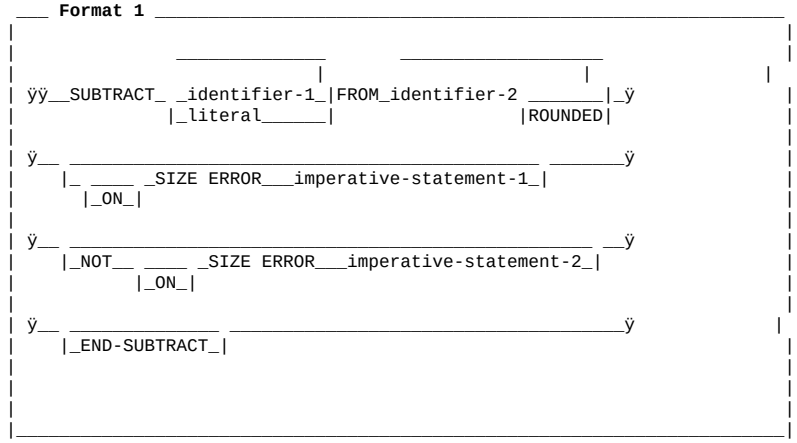


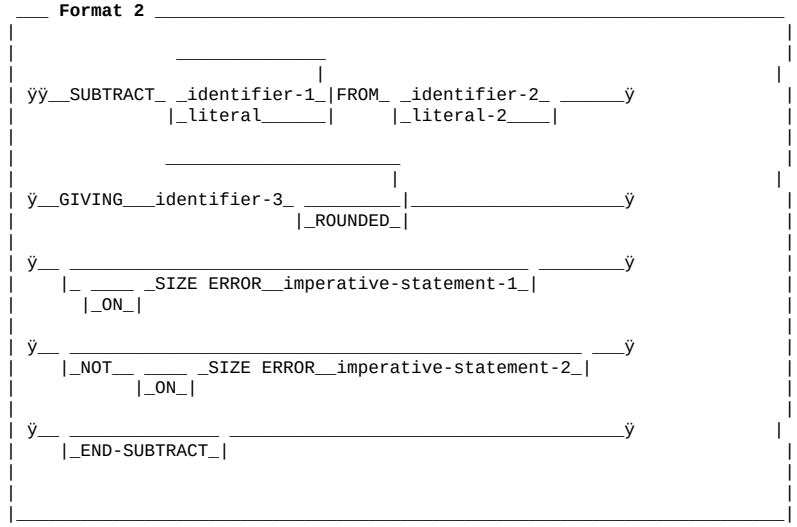
4.1.35 STRING Statement

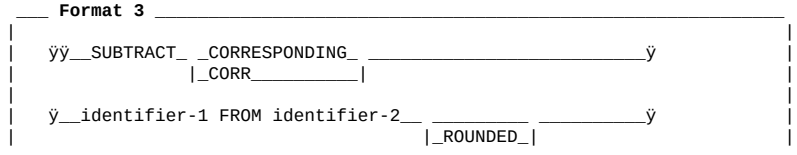


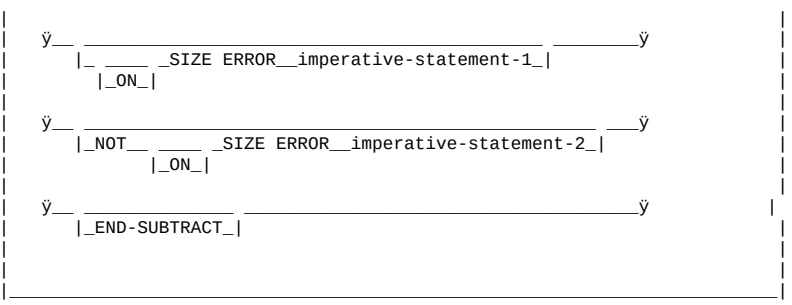


4.1.36 SUBTRACT Statement

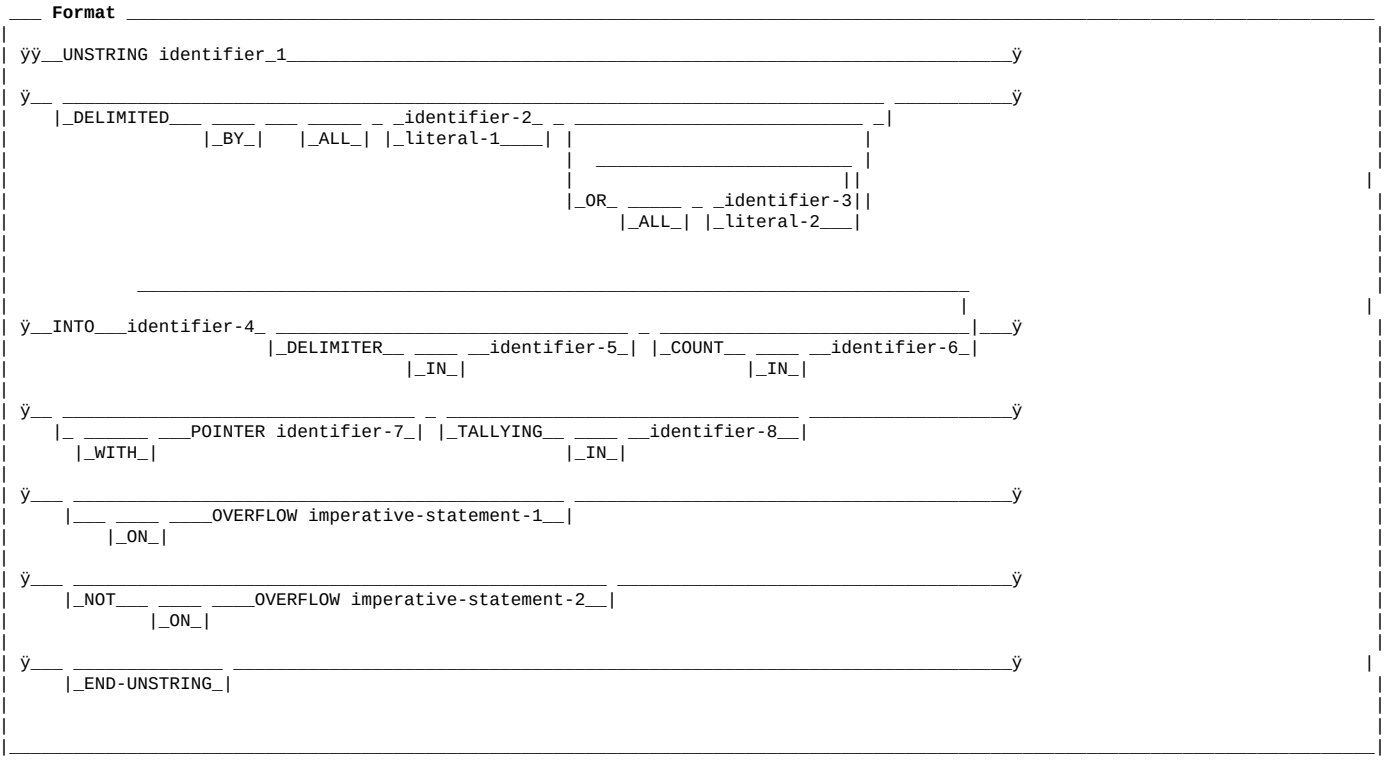




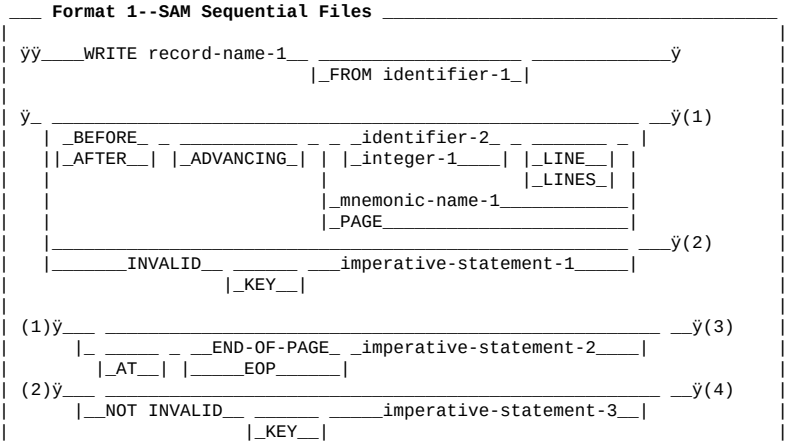


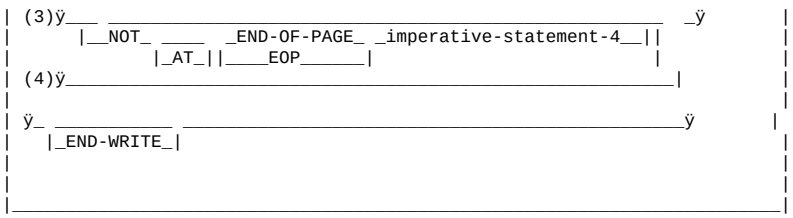


4.1.37 UNSTRING Statement

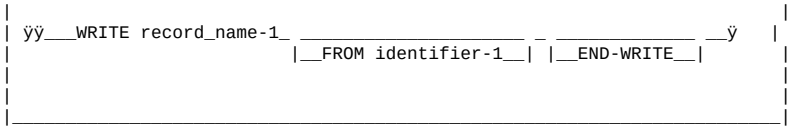


4.1.38 WRITE Statement

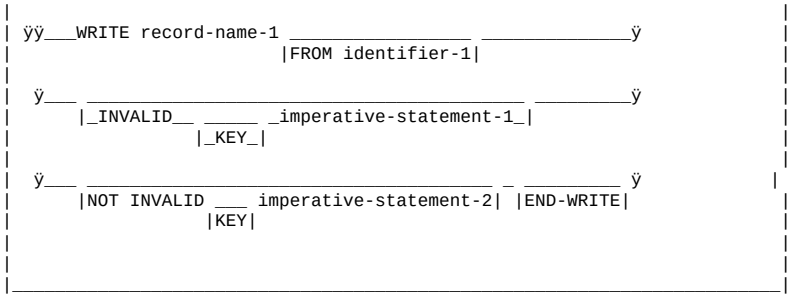




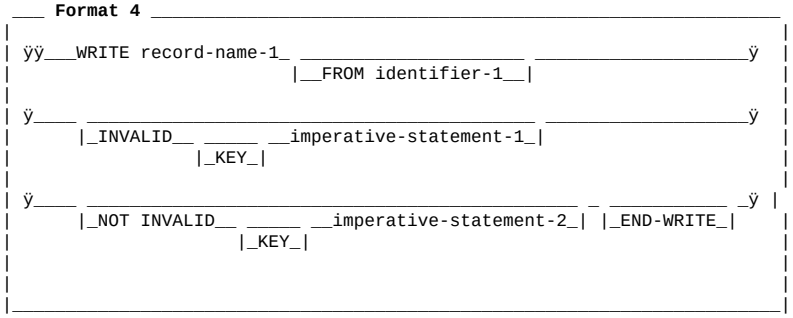
Format 2--VSAM Sequential Files



Format 3--VSAM Indexed Files



Format 4



5.0 Compiler Directing Statements

Subtopics:

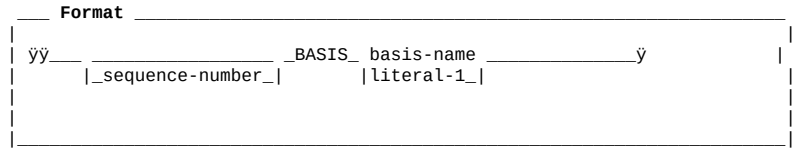
- 5.1 Compiler Directing Statements

5.1 Compiler Directing Statements

Subtopics:

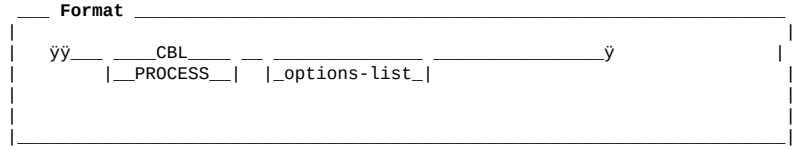
5.1.1 COBOL BASIS Statement

Format



5.1.2 CBL (PROCESS) Statement

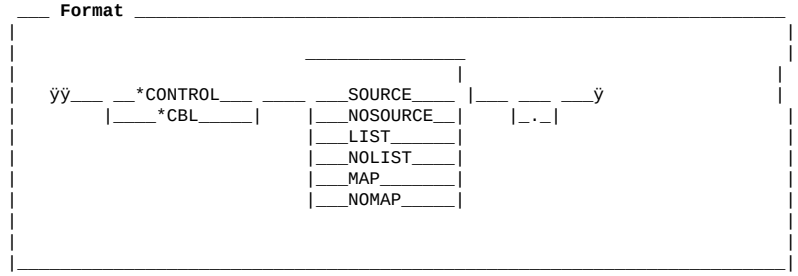
Format



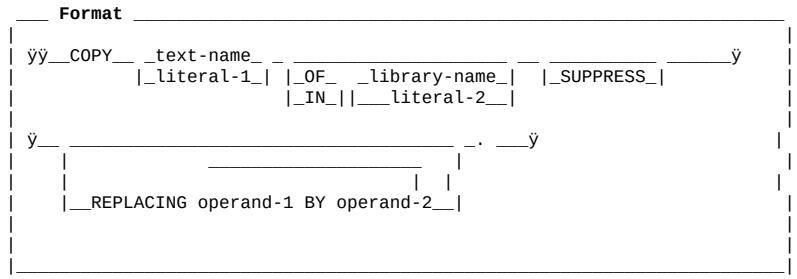
For a list of options you can use with the CBL statement, see Appendix E, "Compiler Options" in topic APPENDIX1.5.

5.1.3 *CONTROL (*CBL) Statement

Format

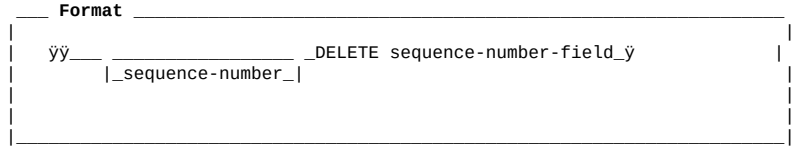


5.1.4 COPY Statement

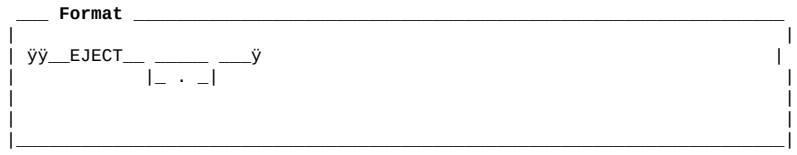


Note: If OF or IN phrase is not used under VSE and will be treated as a comment if coded in your program. It is retained for COBOL/370* compatibility.

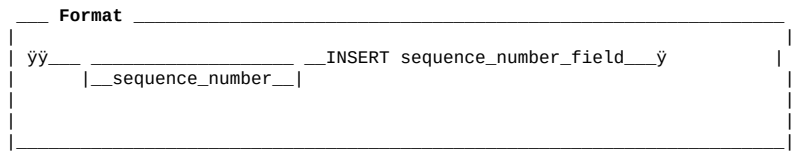
5.1.5 DELETE Statement



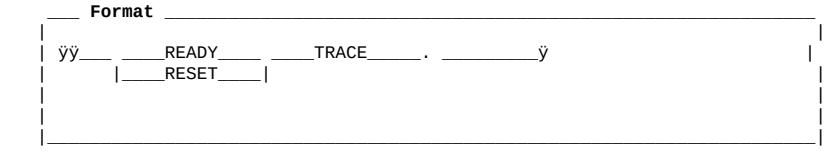
5.1.6 EJECT Statement



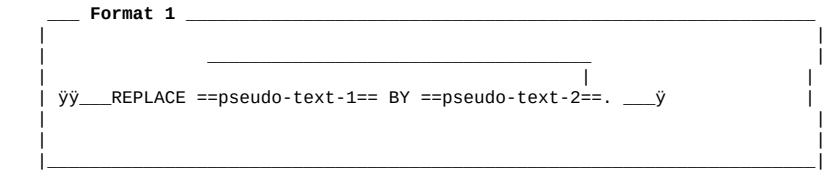
5.1.7 INSERT Statement

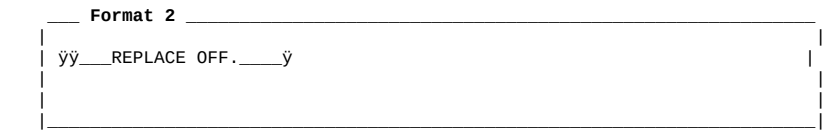


5.1.8 READY or RESET TRACE Statement

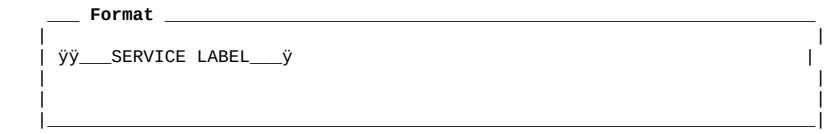


5.1.9 REPLACE Statement

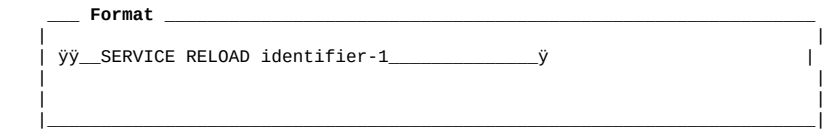




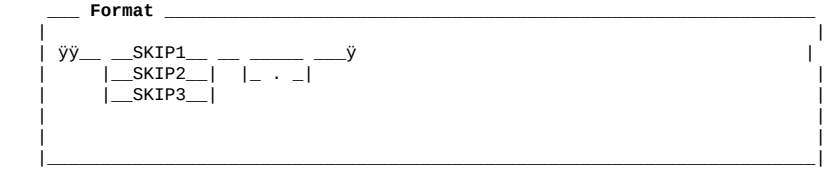
5.1.10 SERVICE LABEL Statement



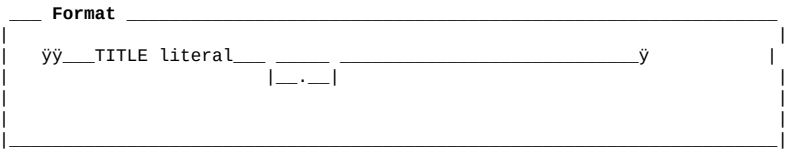
5.1.11 SERVICE RELOAD Statement



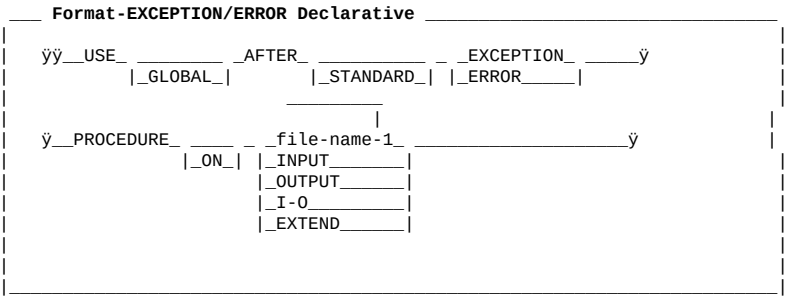
5.1.12 SKIP1/2/3 Statement

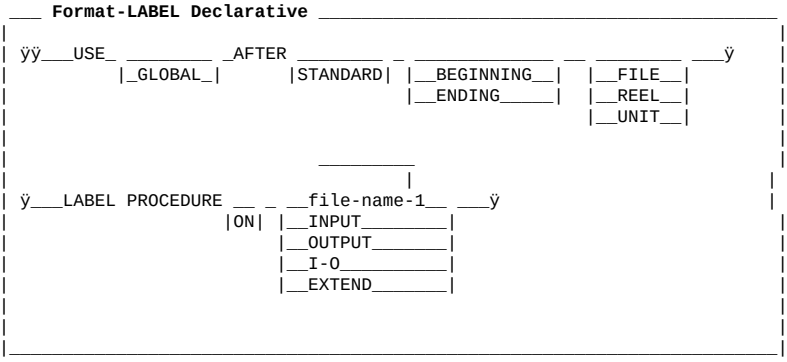


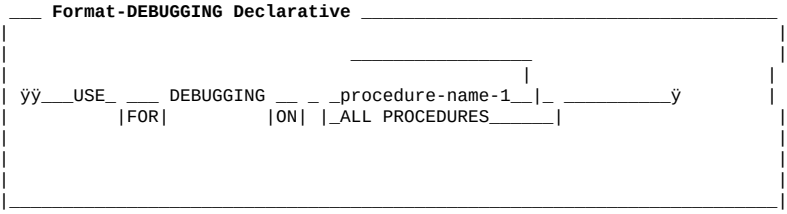
5.1.13 TITLE Statement



5.1.14 USE Statement







APPENDIX1 Appendixes

Subtopics:

APPENDIX1.1 Appendix A. COBOL/VSE Compiler Limits

The following table lists the compiler limits for COBOL/VSE programs. Other operating systems may impose further limits. The numbers are **guidelines** to the limits.

Table 20. COBOL/VSE Compiler Limits

Language Element	**Limit(s)**
Size of program	999,999 lines
Number of literals	4,194,303(1)
Total length of literals	4,194,303 bytes(1)
Reserved Word Table entries	1536
COPY REPLACING ... BY ... (items per COPY statement)	N/A
Number of COPY libraries	N/A
Block size of COPY library	32,767 bytes
Identification Division	
Environment Division	
Configuration Section	
SPECIAL-NAMES paragraph	
function-name IS	18
UPSI-n ... (switches)	0-7
alphabet-name IS ...	N/A
literal THRU/ALSO ...	256
Input-Output Section	
FILE-CONTROL paragraph	
SELECT file-name ...	65,535
ASSIGN system-name ...	N/A(2)
ALTERNATE RECORD KEY data-name ...	253
RECORD KEY length	N/A(3)
RESERVE integer (buffers)	255(4)
I-O-CONTROL paragraph	
RERUN ON system-name ... integer RECORDS	32,767 16,777,215
SAME RECORD AREA	255
FOR file-name ...	255
SAME SORT/MERGE AREA	N/A(2)
MULTIPLE FILE ... file-name	N/A(2) **Note:** The MULTIPLE FILE TAPE phrase is ignored.
Data Division	
File Section	
FD file-name ...	65,535
LABEL data-name ... (if no optional clauses)	255
Label record length	80 bytes
DATA RECORD dnm ...	N/A(2)
BLOCK CONTAINS integer	1,048,575(5)
RECORD CONTAINS integer	1,048,575(5)
Item length	1,048,575 bytes(5)
SD file-name ...	65,535
DATA RECORD dnm ...	N/A(2)
Sort record length	32,751 bytes

Working-Storage Section	
items without the EXTERNAL attribute	134,217,727 bytes
items with the EXTERNAL attribute	134,217,727 bytes
77 data-names	16,777,215 bytes
01-49 data-names	16,777,215 bytes
88 condition-name ...	N/A
VALUE literal ...	N/A
66 RENAMES ...	N/A
PICTURE character string	30
Numeric item digit positions	18
Num-edit character positions	249
PICTURE replication ( )	16,777,215
PIC repl (editing)	32,767
DBCS Picture replication ( )	8,388,607
Group item size:	
File section	1,048,575 bytes
Elementary item size	16,777,215 bytes
VALUE initialization	16,777,215 bytes
(Total length of all value literals)	
OCCURS integer	16,777,215
Total number of ODOs	4,194,303(1)
Table size	16,777,215 bytes
Table element size	8,388,607 bytes
ASC/DES KEY ...	
(per OCCURS clause)	12 KEYS
Total length	256 bytes
INDEXED BY ... (index names)	
(per OCCURS clause)	12
Total num of indexes (index names)	65,535
Size of relative index	32,765
Linkage Section	134,217,727 bytes
Total 01 + 77 (data items)	N/A
Procedure Division	
Procedure + constant area	4,194,303 bytes(1)
USING identifier ...	32,767
Procedure-names	1,048,575(1)
Subscripted data-names per verb	32,767
Verbs per line (TEST)	7
ADD identifier ...	N/A
ALTER pn1 TO pn2 ...	4,194,303(1)
CALL ... BY CONTENT id	2,147,483,647 bytes
CALL id/lit	
USING id/lit...	16380
CALL literal ...	4,194,303(1)
Active programs in run unit	32,767
number of names called (DYN)	N/A
CANCEL id/lit ...	N/A
CLOSE file-name ...	N/A
COMPUTE identifier ...	N/A
DISPLAY id/lit ...	N/A
DIVIDE identifier ...	N/A
ENTRY USING id/lit ...	N/A
EVALUATE ... subjects	64
EVALUATE ... WHEN clauses	256
GO pn ... DEPENDING	255
INSPECT TALLY/REPL clauses	N/A
MERGE file-name ASC/DES KEY ...	N/A
Total key length	3,072 bytes(6)
USING file-name ...	9(7)
MOVE id/lit TO id ...	N/A
MULTIPLY identifier ...	N/A
OPEN file-name	N/A
PERFORM	4,194,303
SEARCH ... WHEN ...	N/A
SET index/id ... TO	N/A
SET index ... UP/DOWN	N/A
SORT file-name ASC/DES KEY	N/A
Total key length	3,072 bytes(6)
USING file-name ...	9(7)
STRING identifier ...	N/A
DELIMITED id/lit ...	N/A
UNSTRING	
DELIMITED id/lit OR id/lit ...	255
INTO id/lit ...	N/A
USE ... ON file-name ...	N/A

Note:

(1) Items included in 4,194,303 byte limit for procedure plus constant area.
(2) Treated as comment; there is no limit.
(3) No compiler limit, but VSAM limits it to 255 bytes.
(4) The SAM limit is 2.
(5) Compiler limit shown, but SAM limits it to 32,767 bytes.
(6) For DFSORT/VSE, the limit is 3,072 bytes.

| (7) For DFSORT/VSE, the limit is 9 files. |

APPENDIX1.2 Appendix B. EBCDIC and ASCII Collating Sequences

The ascending collating sequences for both the EBCDIC (Extended Binary Coded Decimal Interchange Code) and ASCII (American National Standard Code for Information Interchange) character sets are shown in this appendix. In addition to the symbol and meaning for each character, the ordinal number (beginning with 1), decimal representation, and hexadecimal representation are given.

Subtopics:

APPENDIX1.2.1 EBCDIC Collating Sequence

Table 21. EBCDIC Collating Sequence

Ordinal Number	Symbol	Meaning	Decimal Representation	Hex Representation
65	°	Space	64	40
.				
.				
.				
.				
.				
.				
.				
.				
.				
75	¢	Cent sign	74	4A
76	.	Period, decimal point	75	4B
77	<	Less than sign	76	4C
78	(	Left parenthesis	77	4D
79	+	Plus sign	78	4E
80	|	Vertical bar, Logical OR	79	4F
81	&	Ampersand	80	50
.				
.				
.				
.				
.				
.				
.				
.				
.				
91	!	Exclamation point	90	5A
92	$	Dollar sign	91	5B
93	*	Asterisk	92	5C
94	)	Right parenthesis	93	5D
95	;	Semicolon	94	5E
96	¬	Logical NOT	95	5F

97	-	Minus, hyphen	96	60
98	/	Slash	97	61
.				
.				
.				
.				
.				
.				
.				
.				
.				
108	,	Comma	107	6B
109	%	Percent sign	108	6C
110	_	Underscore	109	6D
111	>	Greater than sign	110	6E
112	?	Question mark	111	6F
.				
.				
.				
.				
.				
.				
.				
.				
.				
123	:	Colon	122	7A
124	#	Number sign, pound sign	123	7B
125	@	At sign	124	7C
126	'	Apostrophe, prime sign	125	7D
127	=	Equal sign	126	7E
128	"	Quotation marks	127	7F
.				
.				
.				
.				
.				
.				
.				
.				
.				
130	a		129	81
131	b		130	82
132	c		131	83
133	d		132	84
134	e		133	85
135	f		134	86
136	g		135	87
137	h		136	88
138	i		137	89
.				
.				
.				
.				
.				

.				
.				
.				
.				
146	j		145	91
147	k		146	92
148	l		147	93
149	m		148	94
150	n		149	95
151	o		150	96
152	p		151	97
153	q		152	98
154	r		153	99
.				
.				
.				
.				
.				
.				
.				
.				
163	s		162	A2
164	t		163	A3
165	u		164	A4
166	v		165	A5
167	w		166	A6
168	x		167	A7
169	y		168	A8
170	z		169	A9
.				
.				
.				
.				
.				
.				
.				
.				
194	A		193	C1
195	B		194	C2
196	C		195	C3
197	D		196	C4
198	E		197	C5
199	F		198	C6
200	G		199	C7
201	H		200	C8
202	I		201	C9
.				
.				
.				
.				
.				

.				
.				
.				
.				
210	J		209	D1
211	K		210	D2
212	L		211	D3
213	M		212	D4
214	N		213	D5
215	O		214	D6
216	P		215	D7
217	Q		216	D8
218	R		217	D9
.				
.				
.				
.				
.				
.				
.				
.				
227	S		226	E2
228	T		227	E3
229	U		228	E4
230	V		229	E5
231	W		230	E6
232	X		231	E7
233	Y		232	E8
234	Z		233	E9
.				
.				
.				
.				
.				
.				
.				
.				
241	0		240	F0
242	1		241	F1
243	2		242	F2
244	3		243	F3
245	4		244	F4
246	5		245	F5
247	6		246	F6
248	7		247	F7
249	8		248	F8
250	9		249	F9

APPENDIX1.2.2 ASCII Collating Sequence

Table 22. ASCII Collating Sequence

Ordinal Number	Symbol	Meaning	Decimal Representation	Hex Representation
1		Null	0	0
.				
.				
.				
.				
.				
.				
.				
.				
.				
33	º	Space	32	20
34	!	Exclamation point	33	21
35	"	Quotation mark	34	22
36	#	Number sign	35	23
37	$	Dollar sign	36	24
38	%	Percent sign	37	25
39	&	Ampersand	38	26
40	'	Apostrophe, prime sign	39	27
41	(	Opening parenthesis	40	28
42	)	Closing parenthesis	41	29
43	*	Asterisk	42	2A
44	+	Plus sign	43	2B
45	,	Comma	44	2C
46	-	Hyphen, minus	45	2D
47	.	Period, decimal point	46	2E
48	/	Slant	47	2F
49	0		48	30
50	1		49	31
51	2		50	32
52	3		51	33
53	4		52	34
54	5		53	35
55	6		54	36
56	7		55	37
57	8		56	38
58	9		57	39
59	:	Colon	58	3A
60	;	Semicolon	59	3B
61	<	Less than sign	60	3C
62	=	Equal sign	61	3D
63	>	Greater than sign	62	3E
64	?	Question mark	63	3F
65	@	Commercial At sign	64	40

66	A		65	41
67	B		66	42
68	C		67	43
69	D		68	44
70	E		69	45
71	F		70	46
72	G		71	47
73	H		72	48
74	I		73	49
75	J		74	4A
76	K		75	4B
77	L		76	4C
78	M		77	4D
79	N		78	4E
80	O		79	4F
81	P		80	50
82	Q		81	51
83	R		82	52
84	S		83	53
85	T		84	54
86	U		85	55
87	V		86	56
88	W		87	57
89	X		88	58
90	Y		89	59
91	Z		90	5A
92	[	Opening bracket	91	5B
93	\	Reverse slant	92	5C
94	]	Closing bracket	93	5D
95	^	Caret	94	5E
96	_	Underscore	95	5F
97	`	Grave Accent	96	60
98	a		97	61
99	b		98	62
100	c		99	63
101	d		100	64
102	e		101	65
103	f		102	66
104	g		103	67
105	h		104	68
106	i		105	69
107	j		106	6A
108	k		107	6B
109	l		108	6C
110	m		109	6D
111	n		110	6E

112	o		111	6F
113	p		112	70
114	q		113	71
115	r		114	72
116	s		115	73
117	t		116	74
118	u		117	75
119	v		118	76
120	w		119	77
121	x		120	78
122	y		121	79
123	z		122	7A
124	{	Opening brace	123	7B
125	¦	Split vertical bar	124	7C
126	}	Closing brace	125	7D
127	~	Tilde	126	7E

APPENDIX1.3 Appendix C. Source Language Debugging

Table 23. DEBUG-ITEM Subfield Contents

Cause of Debugging Section Execution	**Statement Referred to in DEBUG-LINE**	**Contents of DEBUG-NAME**	**Contents of DEBUG-CONTENTS**
procedure-name-1 ALTER reference	ALTER statement	procedure-name-1	procedure-name-n in TO PROCEED TO phrase
GO TO procedure-name-n	GO TO statement	procedure-name-n	spaces
procedure-name-n in SORT/MERGE input/output procedure	SORT/MERGE statement	procedure-name-n	"SORT INPUT" "SORT OUTPUT" "MERGE OUTPUT" (as applicable)
PERFORM statement transfer of control	This PERFORM statement	procedure-name-n	"PERFORM LOOP"
procedure-name-n in a USE procedure	Statement causing USE procedure execution	procedure-name-n	"USE PROCEDURE"
Implicit transfer from previous sequential procedure	Previous statement executed in previous sequential procedure(*)	procedure-name-n	"FALL THROUGH"
First execution of first nondeclarative procedure	Line number of first nondeclarative procedure-name	first nondeclarative	"START PROGRAM"

Note: If the paragraph containing a statement executed in a previous sequential procedure is preceded by a section header, and control is passed through the section header, the statement number refers to the section header.

APPENDIX1.4 Appendix D. COBOL/VSE Reserved Words

This list identifies all reserved words in the COBOL/VSE product.

- Words marked under **Compiler-Directing** are compiler-directing words and are flagged with an S-LEVEL message if used as user-defined names.
- Words marked under **Non-COBOL/VSE** are COBOL 85 Standard Reserved Words that are NOT implemented in COBOL/VSE Release 1 and are flagged with an S-LEVEL message if used as user-defined names.
- Words marked under **CODASYL** are reserved for future development and are flagged with an I-LEVEL message.

Note: The contents of the reserved word table can be changed by using the WORD compiler option. See *COBOL/VSE Programming Guide* for details on how to specify an alternate reserved word table.

Table 24. Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
ACCEPT			
ACCESS			
ACQUIRE			
ADD			
ADDRESS			
ADVANCING			
AFTER			
ALL			
ALLOWING			X
ALPHABET			
ALPHABETIC			
ALPHABETIC-LOWER			
ALPHABETIC-UPPER			
ALPHANUMERIC			
ALPHANUMERIC-EDITED			
ALSO			
ALTER			
ALTERNATE			
AND			
ANY			
APPLY			
ARE			
AREA			
AREA-VALUE			
AREAS			
ARITHMETIC			X
ASCENDING			

ASSIGN			
AT			
AUTHOR			
AUTO			
AUTO-SKIP			
AUTOMATIC			
B-AND			X
B-EXOR			X
B-LESS			X
B-NOT			X
B-OR			X
BACKGROUND-COLOR			
BACKGROUND-COLOUR			
BACKWARD			
BASIS	X		
BEEP			
BEFORE			
BEGINNING			
BELL			
BINARY			
BIT			X
BITS			X
BLANK			
BLINK			
BLOCK			
BOOLEAN			X
BOTTOM			
BY			
CALL			
CANCEL			
CBL	X		
CD		X	
CF		X	
CH		X	
CHAIN			
CHAINING			
CHANGED			
CHARACTER			
CHARACTERS			
CLASS			
CLOCK-UNITS		X	
CLOSE			
COBOL			
CODE			
CODE-SET			

COL			
COLLATING			
COLOR			
COLUMN		X	
COM-REG			
COMMA			
COMMAND-LINE			
COMMIT			X
COMMITMENT			
COMMON			
COMMUNICATION		X	
COMP			
COMP-X			
COMP-0			
COMP-1			
COMP-2			
COMP-3			
COMP-4			
COMP-5			X
COMP-6			X
COMP-7			X
COMP-8			X
COMP-9			X
COMPUTATIONAL			
COMPUTATIONAL-X			
COMPUTATIONAL-0			
COMPUTATIONAL-1			
COMPUTATIONAL-2			
COMPUTATIONAL-3			
COMPUTATIONAL-4			
COMPUTATIONAL-5			X
COMPUTATIONAL-6			X
COMPUTATIONAL-7			X
COMPUTATIONAL-8			X
COMPUTATIONAL-9			X
COMPUTE			
CONFIGURATION			
CONNECT			X
CONSOLE			
CONTAINED			X
CONTAINS			
CONTENT			
CONTINUE			
CONTROL	X	X	
CONTROL-AREA			
CONTROLS		X	

CONVERTING			
COPY	X		
CORR			
CORRESPONDING			
COUNT			
CRT			
CRT-UNDER			
CURRENCY			
CURRENT			X
CURSOR			
CYCLE			X
DATA			
DATE			
DATE-COMPILED			
DATE-WRITTEN			
DAY			
DAY-OF-WEEK			
DB			X
DB-ACCESS-CONTROL-KEY			X
DB-DATA-NAME			X
DB-EXCEPTION			X
DB-FORMAT-NAME			
DB-RECORD-NAME			X
DB-SET-NAME			X
DB-STATUS			X
DBCS			
DE		X	
DEBUG-CONTENTS			
DEBUG-ITEM			
DEBUG-LINE			
DEBUG-NAME			
DEBUG-SUB-1			
DEBUG-SUB-2			
DEBUG-SUB-3			
DEBUGGING			
DECIMAL-POINT			
DECLARATIVES			
DEFAULT			X
DELETE	X		
DELIMITED			
DELIMITER			
DEPENDING			
DESCENDING			
DESTINATION		X	
DETAIL		X	

DISABLE		X	
DISCONNECT			X
DISK			
DISPLAY			
DISPLAY-1			
DISPLAY-2			X
DISPLAY-3			X
DISPLAY-4			X
DISPLAY-5			X
DISPLAY-6			X
DISPLAY-7			X
DISPLAY-8			X
DISPLAY-9			X
DIVIDE			
DIVISION			
DOWN			
DROP			
DUPLICATE			X
DUPLICATES			
DYNAMIC			
EGCS			
EGI		X	
EJECT	X		
ELSE			
EMI		X	
EMPTY			X
EMPTY-CHECK			
ENABLE		X	
END			
END-ACCEPT			
END-ADD			
END-CALL			
END-COMPUTE			
END-DELETE			
END-DISABLE			X
END-DIVIDE			
END-ENABLE			X
END-EVALUATE			
END-IF			
END-MULTIPLY			
END-OF-PAGE			
END-PERFORM			
END-READ			
END-RECEIVE		X	
END-RETURN			
END-REWRITE			

END-SEARCH			
END-SEND			X
END-START			
END-STRING			
END-SUBTRACT			
END-TRANSCEIVE			X
END-UNSTRING			
END-WRITE			
ENDING			
ENTER			
ENTRY			
ENVIRONMENT			
EOP			
EQUAL			
EQUALS			X
ERASE			X
ERROR			
ESCAPE			
ESI		X	
EVALUATE			
EVERY			
EXACT			X
EXCEEDS			X
EXCEPTION			
EXCESS-3			
EXCLUSIVE			X
EXEC			
EXECUTE			
EXHIBIT			
EXIT			
EXTEND			
EXTERNAL			
EXTERNALLY-DESCRIBED-KEY			
FALSE			
FD			
FETCH			X
FILE			
FILE-CONTROL			
FILE-ID			
FILLER			
FINAL		X	
FIND			X
FINISH			X
FIRST			
FIXED			

FOOTING			
FOR			
FOREGROUND-COLOR			
FOREGROUND-COLOUR			
FORM			X
FORMAT			X
FREE			X
FROM			
FULL			
FUNCTION			
GENERATE		X	
GET			X
GIVING			
GLOBAL			
GO			
GOBACK			
GREATER			
GROUP		X	
HEADING		X	
HIGH-VALUE			
HIGH-VALUES			
HIGHLIGHT			
I-O			
I-O-CONTROL			
ID			
IDENTIFICATION			
IF			
IN			
INDEX			
INDEX-1			X
INDEX-2			X
INDEX-3			X
INDEX-4			X
INDEX-5			X
INDEX-6			X
INDEX-7			X
INDEX-8			X
INDEX-9			X
INDEXED			
INDIC			
INDICATE		X	
INDICATOR			
INDICATORS			
INITIAL			
INITIALIZE			
INITIATE		X	

INPUT			
INPUT-OUTPUT			
INSERT	X		
INSPECT			
INSTALLATION			
INTO			
INVALID			
IS			
JAPANESE			
JUST			
JUSTIFIED			
KANJI			
KEEP			X
KEPT			
KEY			
KEYBOARD			
LABEL	X		
LAST		X	
LD			X
LEADING			
LEFT			
LEFT-JUSTIFY			
LENGTH			
LENGTH-CHECK			
LESS			
LIKE			
LIMIT		X	
LIMITS		X	
LINAGE			
LINAGE-COUNTER			
LINE			
LINE-COUNTER		X	
LINES			
LINKAGE			
LOCALLY			X
LOCK			
LOW-VALUE			
LOW-VALUES			
MANUAL			
MEMBER			X
MEMORY			
MERGE			
MESSAGE		X	
MODE			
MODIFIED			

MODIFY			X
MODULES			
MORE-LABELS			
MOVE			
MULTIPLE			
MULTIPLY			
NAME			
NATIVE			
NEGATIVE			
NEXT			
NO-ECHO			
NO			
NORMAL			X
NOT			
NULL			
NULLS			
NUMBER		X	
NUMERIC			
NUMERIC-EDITED			
OBJECT-COMPUTER			
OCCURS			
OF			
OFF			
OMITTED			
ON			
ONLY			X
OPEN			
OPTIONAL			
OR			
ORDER			
ORGANIZATION			
OTHER			
OUTPUT			
OVERFLOW			
OWNER			X
PACKED-DECIMAL			
PADDING			
PAGE			
PAGE-COUNTER		X	
PALETTE			
PARAGRAPH			X
PASSWORD			
PERFORM			
PF		X	
PH		X	
PIC			

PICTURE			
PLUS		X	
POINTER			
POSITION			
POSITIVE			
PRESENT			X
PREVIOUS			
PRINT-SWITCH			
PRINTER			
PRINTER-1			
PRINTING		X	
PRIOR			X
PROCEDURE			
PROCEDURE-POINTER			
PROCEDURES			
PROCEED			
PROCESS			
PROCESSING			
PROGRAM			
PROGRAM-ID			
PROMPT			
PROTECTED			X
PURGE		X	
QUEUE		X	
QUOTE			
QUOTES			
RANDOM			
RANGE			
RD		X	
READ			
READY	X		
REALM			X
RECEIVE		X	
RECONNECT			X
RECORD			
RECORD-NAME			X
RECORDING			
RECORDS			
REDEFINES			
REEL			
REFERENCE			
REFERENCES			
RELATION			X
RELATIVE			
RELEASE			

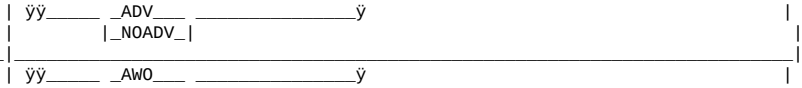
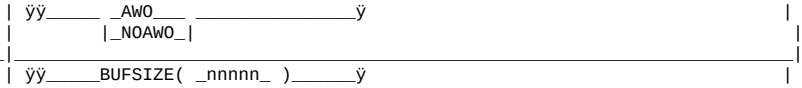
RELOAD	X		
REMAINDER			
REMOVAL			
RENAMES			
REPEATED			X
REPLACE	X		
REPLACING			
REPORT		X	
REPORTING		X	
REPORTS		X	
REQUIRED			
RERUN			
RESERVE			
RESET	X		
RETAINING			X
RETRIEVAL			X
RETURN			
RETURN-CODE			
REVERSE-VIDEO			
REVERSED			
REWIND			
REWRITE			
RF		X	
RH		X	
RIGHT			
RIGHT-JUSTIFY			
ROLLBACK			X
ROLLING			
ROUNDED			
RUN			
SAME			
SCREEN			
SD			
SEARCH			
SECTION			
SECURE			
SECURITY			
SEGMENT		X	
SEGMENT-LIMIT			
SELECT			
SEND		X	
SENTENCE			
SEPARATE			
SEQUENCE			
SEQUENTIAL			

SERVICE	X		
SESSION-ID			X
SET			
SHARED			X
SHIFT-IN			
SHIFT-OUT			
SIGN			
SIZE			
SKIP1	X		
SKIP2	X		
SKIP3	X		
SORT			
SORT-CONTROL			
SORT-CORE-SIZE			
SORT-FILE-SIZE			
SORT-MERGE			
SORT-MESSAGE			
SORT-MODE-SIZE			
SORT-RETURN			
SOURCE		X	
SOURCE-COMPUTER			
SPACE			
SPACE-FILL			
SPACES			
SPECIAL-NAMES			
STANDARD			
STANDARD-1			
STANDARD-2			
STANDARD-3			X
STANDARD-4			X
START			
STARTING			
STATUS			
STOP			
STORE			X
STRING			
SUB-QUEUE-1		X	
SUB-QUEUE-2		X	
SUB-QUEUE-3		X	
SUB-SCHEMA			X
SUBFILE			
SUBPROGRAM			
SUBTRACT			
SUM			
SUPPRESS			
SWITCH			

SWITCH-1			
SWITCH-2			
SWITCH-3			
SWITCH-4			
SWITCH-5			
SWITCH-6			
SWITCH-7			
SWITCH-8			
SYMBOLIC			
SYNC			
SYNCHRONIZED			
TABLE		X	
TALLY			
TALLYING			
TAPE			
TENANT			X
TERMINAL		X	
TERMINATE		X	
TEST			
TEXT		X	
THAN			
THEN			
THROUGH			
THRU			
TIME			
TIMEOUT			X
TIMES			
TITLE	X		
TO			
TOP			
TRACE	X		
TRAILING			
TRAILING-SIGN			
TRANSACTION			
TRANSCEIVE			X
TRUE			
TYPE		X	
UNDERLINE			
UNEQUAL			X
UNIT			
UNLOCK			
UNSTRING			
UNTIL			
UP			
UPDATE			X

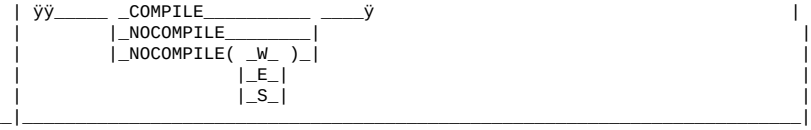
UPON			
USAGE			
USAGE-MODE			X
USE	X		
USER			
USING			
VALID			X
VALIDATE			X
VALUE			
VALUES			
VARIABLE			
VARYING			
WAIT			X
WHEN			
WHEN-COMPILED			
WITH			
WITHIN			X
WORDS			
WORKING-STORAGE			
WRITE			
WRITE-ONLY			
ZERO			
ZERO-FILL			
ZEROES			
ZEROS			
<			
<=			
+			
*			
**			
-			
/			
>			
>=			
=			

APPENDIX1.5 Appendix E. Compiler Options

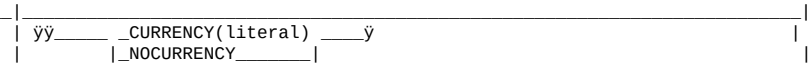
ADATA Option	
ADV Option	
AWO Option	
BUFSIZE Option	

CMPR2 Option

COMPILE Option



CURRENCY Option



DATA Option

DBCS Option

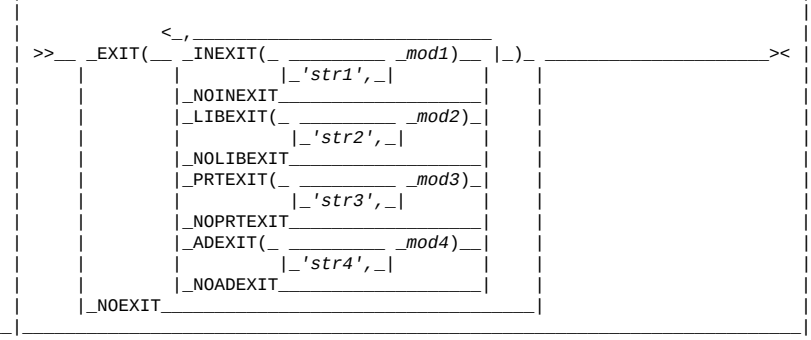
DECK Option

Note: This option may only be specified via the JCL OPTION statement.

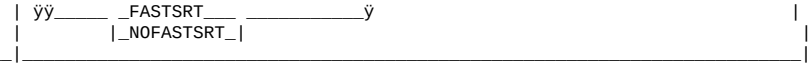
DUMP Option

DYNAM Option

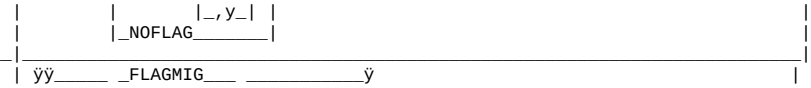
EXIT Option



FASTSRT Option



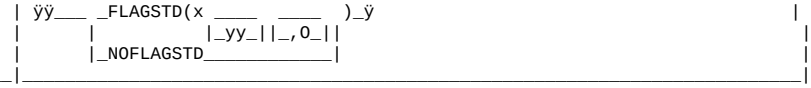
FLAG Option



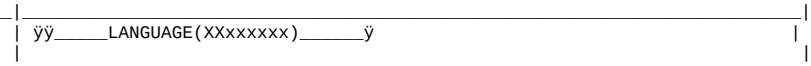
FLAGMIG Option

FLAGSAA Option

FLAGSTD Option



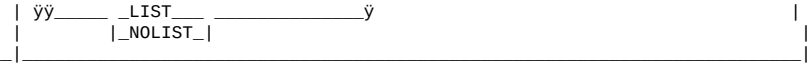
LANGUAGE Option



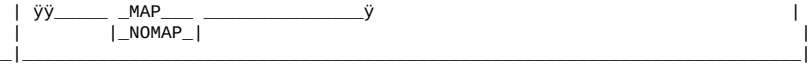
LIB Option

LINECOUNT Option

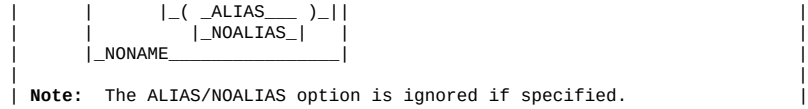
LIST Option



MAP Option



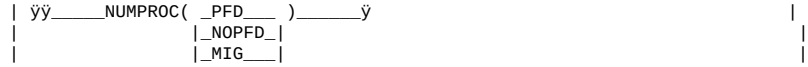
NAME Option

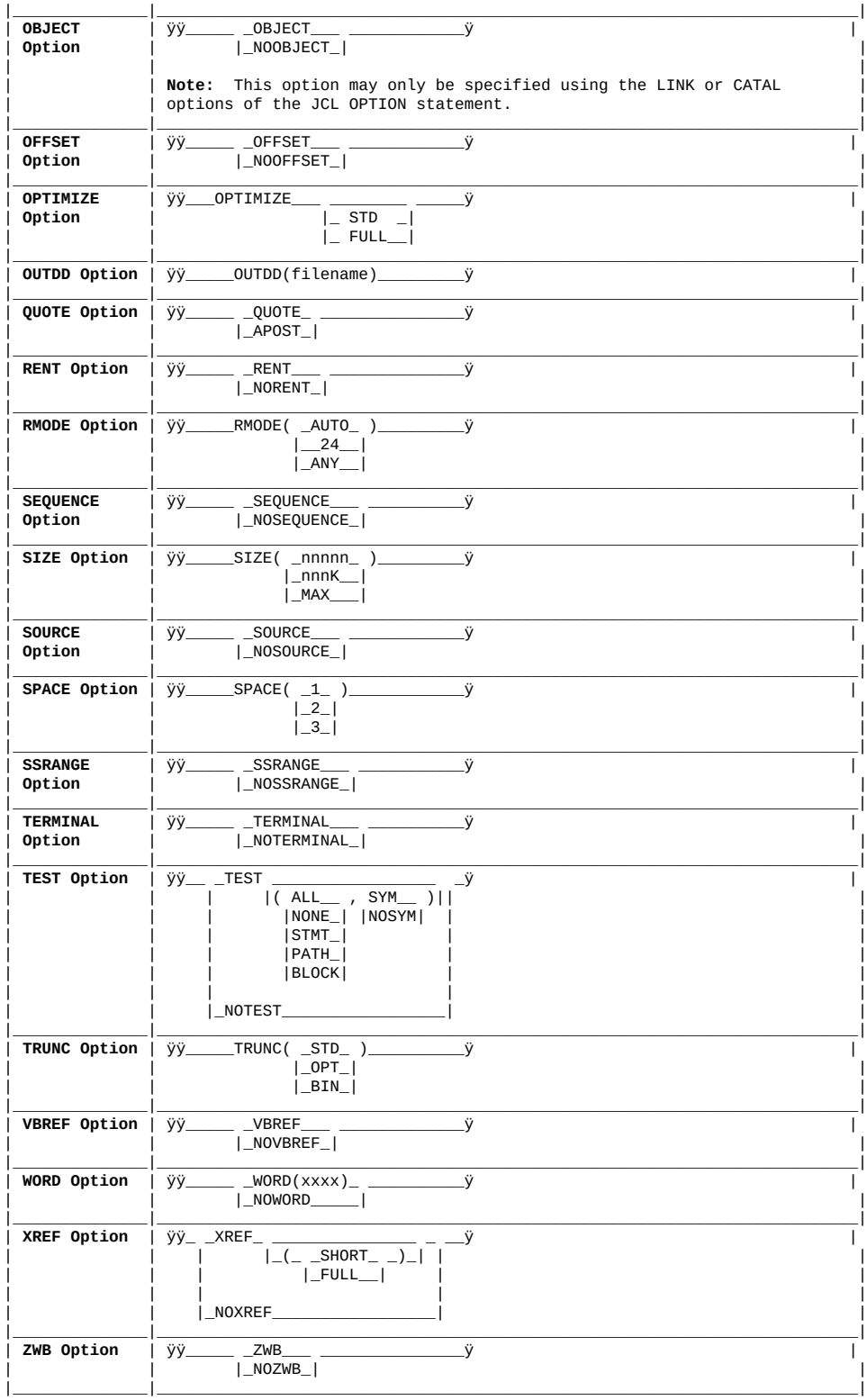


Note: The ALIAS/NOALIAS option is ignored if specified.

NUMBER Option

NUMPROC Option





Subtopics:

- APPENDIX1.5.1 Required Compiler Options for Conformance with the COBOL 85 Standard

---

APPENDIX1.5.1 Required Compiler Options for Conformance with the COBOL 85 Standard

```
ADV
NOCMPR2
NODBCS
NOFASTSRT
NOFLAGMIG
NOFLAGSAA
FLAGSTD(H)
LIB
NAME(ALIAS) or NAME(NOALIAS)
NONUMBER
NUMPROC(NOPFD) or NUMPROC(MIG)
QUOTE
NOSEQUENCE
TRUNC(STD)
NOWORD
ZWB
```

Notes:

1. The option AIXBLD must also be specified.

2. The ALIAS/NOALIAS option is ignored if specified.
