

AS/400e™



ILE COBOL for AS/400® Reference Summary

Version 4

AS/400e™



ILE COBOL for AS/400® Reference Summary

Version 4

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Second Edition (May 1999)

This edition applies to Version 4, Release 4, Modification Level 0, of IBM Application System/400 Integrated Language Environment COBOL for AS/400 (Program 5769-CB1) and to all subsequent releases and modifications until otherwise indicated in new editions. This edition applies only to reduced instruction set computer (RISC) systems.

This edition replaces SX09-1317-00.

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

IBM welcomes your comments. You can send your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/KB7/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "How to Send Your Comments" for a description of the methods.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1993, 1999. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Programming Interface Information	viii
Trademarks and Service Marks	viii
Acknowledgements	viii
About This Summary	ix
Who Should Use This Summary	ix
Prerequisite and Related Information	ix
How to Send Your Comments	ix
ILE COBOL Syntax Notation	x
How to Read the Syntax Diagrams	xi
IBM Extensions	xii
Documentary Syntax	xiii
CL Entry Codes	xiii
Industry Standard	xiii
Creating Module and Program Objects	1
CRTCLMOD Command Syntax	2
CRTBNDCBL Command Syntax	4
PROCESS Statement	6
COBOL Source Program—General Structure	11
Identification Division	13
Environment Division	15
Configuration Section	15
SOURCE-COMPUTER Paragraph	15
OBJECT-COMPUTER Paragraph	15
SPECIAL-NAMES Paragraph	16
Input-Output Section	19
FILE-CONTROL Paragraph	19
I-O-CONTROL Paragraph	21
Data Division	23
File Section	23
Working-Storage Section	29
Linkage Section	32
Procedure Division	33
Procedure Division Statements	34
ACCEPT Statement	34
ACQUIRE Statement	36
ADD Statement	36
ALTER Statement	37

CALL Statement	37
CANCEL Statement	39
CLOSE Statement	39
COMMIT Statement	40
COMPUTE Statement	40
CONTINUE Statement	40
DELETE Statement	40
DISPLAY Statement	41
DIVIDE Statement	43
DROP Statement	44
ENTER Statement	44
EVALUATE Statement	44
EXIT Statement	45
EXIT PROGRAM Statement	45
GOBACK Statement	45
GO TO Statement	45
IF Statement	46
INITIALIZE Statement	46
INSPECT Statement	46
MERGE Statement	48
MOVE Statement	48
MULTIPLY Statement	48
OPEN Statement	49
PERFORM Statement	49
READ Statement	50
RELEASE Statement	52
RETURN Statement	53
REWRITE Statement	53
ROLLBACK Statement	54
SEARCH Statement	54
SET Statement	55
SORT Statement	56
START Statement	57
STOP Statement	57
STRING Statement	58
SUBTRACT Statement	58
UNSTRING Statement	59
WRITE Statement	60
Intrinsic Functions	62
Function-Identifier	62
ACOS Function	62
ADD-DURATION Function	62
ASIN Function	62
ATAN Function	62
CHAR Function	62
CONVERT-DATE-TIME Function	63
COS Function	63
CURRENT-DATE Function	63

DATE-OF-INTEGER Function	63
DAY-OF-INTEGER Function	63
DATE-TO-YYYYMMDD Function	63
DAY-TO-YYYYDDD Function	64
EXTRACT-DATE-TIME Function	64
FIND-DURATION Function	64
INTEGER-OF-DATE Function	64
INTEGER-OF-DAY Function	64
LENGTH Function	65
LOCALE-DATE Function	65
LOCALE-TIME Function	65
LOG Function	65
LOG10 Function	65
LOWER-CASE Function	65
MEAN Function	65
NUMVAL Function	66
NUMVAL-C Function	66
ORD Function	67
REVERSE Function	67
SIN Function	67
SQRT Function	67
SUBTRACT-DURATION Function	67
TAN Function	67
TEST-DATE-TIME Function	68
UPPER-CASE Function	68
WHEN-COMPILED Function	68
YEAR-TO-YYYY Function	68
Conditional Expressions	69
Qualifying Data Reference Formats	71
Qualification	71
Reference Modification	72
Subscripting	72
Compiler-Directing Statements	73
CONTROL Statement	73
COPY Statement	73
EJECT Statement	75
REPLACE Statement	75
SKIP Statement	75
TITLE Statement	75
USE Statement	76
Symbols, Names, and Figurative Constants	77
Assignment-Names in the ASSIGN Clause	78
Environment-Names in the SPECIAL-NAMES Paragraph	79
Figurative Constants	80

File Structure Support Summary and Status Key Values	81
File Structure Support Tables	81
File Status Key Values and Meanings	86
ILE COBOL Function-Name and Context-Sensitive Word List	91
Visual Key	91
Function-Names	91
Context-Sensitive Words	91
ILE COBOL Reserved Word List	93
Visual Key	93
Reserved Words	93

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Ltd.
Department 071
1150 Eglinton Avenue East
North York, Ontario M3C 1H7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Programming Interface Information

This summary is intended to help you write ILE COBOL programs. It contains information necessary for you to use the ILE COBOL compiler. This summary documents no programming interfaces for use in writing programs that request or receive the services of the ILE COBOL compiler.

Trademarks and Service Marks

The following terms are trademarks of International Business Machines Corporation in the United States or other countries or both:

400	IBM
Application System/400	IBMLink
AS/400	Integrated Language Environment
AS/400e	Operating System/400
COBOL/400	OS/400
GDDM	

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Other company, product, and service names may be the trademarks or service marks of others.

Registered trademarks and unregistered trademarks are denoted by ® and ™ respectively.

Acknowledgements

IBM acknowledges the use of the following research product in the ILE COBOL compiler:

S/SL ©Copyright 1981 by the University of Toronto

About This Summary

This summary contains all the COBOL statements and related information you may need to refer to when programming in the Integrated Language Environment® (ILE) COBOL for AS/400® language.

Before using this summary, you should have a basic understanding of the ILE COBOL for AS/400 (ILE COBOL) language and of the Operating System/400® (OS/400®) operating system Control Language (CL).

Who Should Use This Summary

This publication is for programmers familiar with the COBOL language. The purpose of this publication is to summarize the formats of the COBOL language as it is used on the ILE COBOL compiler.

In order to use this summary effectively, you should be familiar with the *ILE COBOL for AS/400 Programmer's Guide* and the *ILE COBOL for AS/400 Reference*. If you need reference information for the Control Language, see the *CL Reference (Abridged)*. You should also be familiar with data management concepts as described in the *Data Management* manual.

Prerequisite and Related Information

Use the AS/400 Information Center as your starting point for looking up AS/400 technical information. You can access the Information Center from the AS/400e Information Center CD-ROM (English version: SK3T-2027-01) or from one of these Web sites:

<http://www.as400.ibm.com/infocenter>
<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm>

The AS/400 Information Center contains important topics such as logical partitioning, clustering, Java, TCP/IP, Web serving, and secured networks. It also contains Internet links to Web sites such as the AS/400 Online Library and the AS/400 Technical Studio. Included in the Information Center is a link that describes at a high level the differences in information between the Information Center and the Online Library.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. IBM welcomes any comments about this book or any other AS/400 documentation.

- If you prefer to send comments by mail, use the the following address:

IBM Canada Ltd. Laboratory
Information Development
2G/KB7/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada M3C 1H7

If you are mailing a readers' comment form from a country other than the United States, you can give the form to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by FAX, use the following number:
 - 1-416-448-6161
- If you prefer to send comments electronically, use one of these e-mail addresses:
 - Comments on books:
torrcf@ca.ibm.com
IBMLink: toribm(torrcf)
 - Comments on the AS/400 Information Center:
RCHINFOC@us.ibm.com

Be sure to include the following:

- The name of the book.
- The publication number of the book.
- The page number or topic to which your comment applies.

ILE COBOL Syntax Notation

ILE COBOL basic formats are presented in a uniform system of syntax notation. This notation, designed to assist you in writing COBOL source statements, is explained in the following paragraphs:

- COBOL keywords and optional words appear in uppercase letters; for example:
MOVE

They must be spelled exactly as shown. If any keyword is missing, the compiler considers it as an error.
- Variables representing user-supplied names or values appear in all lowercase italic letters; for example:
parmx
- For easier text reference, some words are followed by a hyphen and a digit or a letter, as in:
identifier-1

This suffix does not change the syntactical definition of the word.
- If punctuation marks, parentheses, arithmetic operators, logical operators, or such symbols are shown, they must be entered as part of the syntax.
- Arithmetic and logical operators (+, -, *, /, **, >, <, =, ≥, and ≤) are *special character* reserved words. For a complete listing of reserved ILE COBOL words, see "ILE COBOL Reserved Word List" on page 93.

How to Read the Syntax Diagrams

Throughout this book, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line:

- ▶▶— indicates the beginning of a statement.
- ▶ indicates that the statement syntax is continued on the next line.
- ▶— indicates that a statement is continued from the previous line.
- ▶◀ indicates the end of a statement.

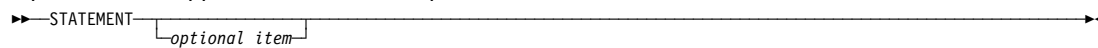
Diagrams of syntactical units other than statements, such as clauses, phrases, and paragraphs, also start with the ▶▶— symbol and end with the —▶◀ symbol.

Note: Statements within a diagram of an entire paragraph will not start with ▶▶— and end with —▶◀ unless their beginning or ending coincides with that of the paragraph.

- Required items appear on the horizontal line (the main path).

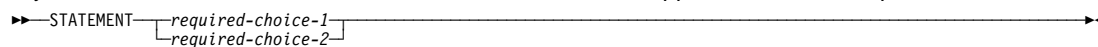


- Optional items appear below the main path.



- When you can choose from two or more items, they appear vertically, in a stack.

If you must choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.



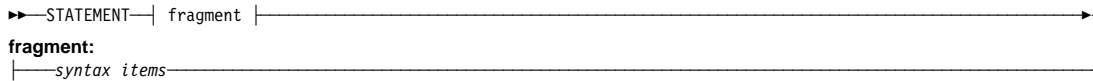
- An arrow returning to the left above an item indicates that the item can be repeated.



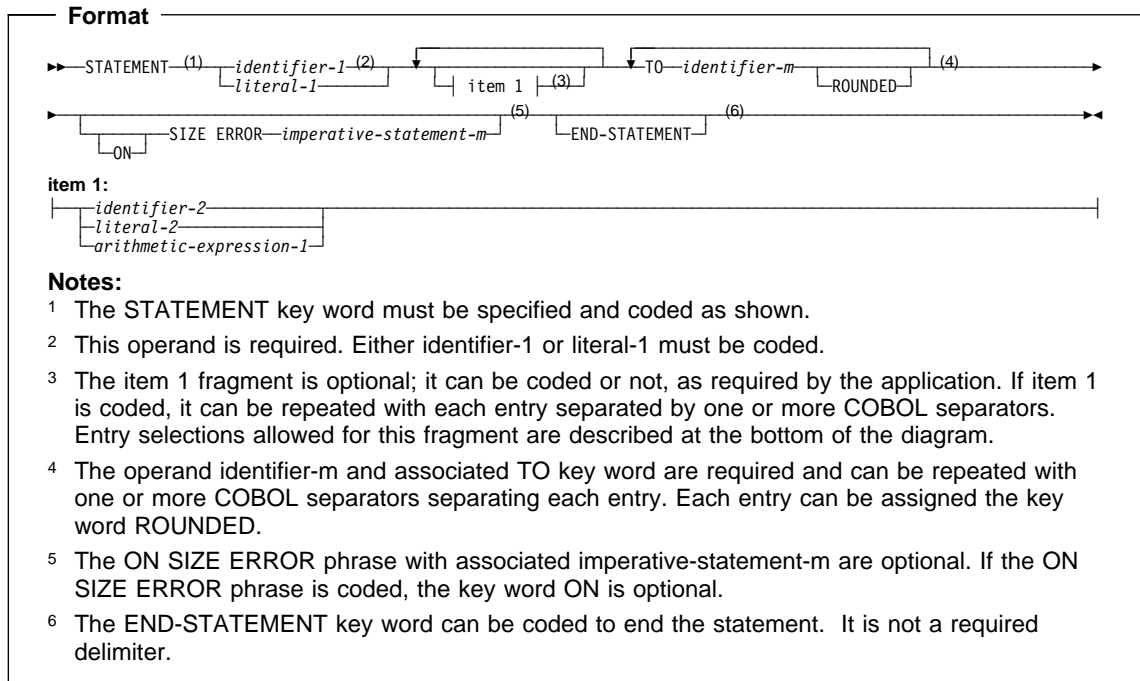
A repeat arrow above a stack of required or optional choices indicates that you can make more than one choice from the stacked items, or repeat a single choice.



- A syntax fragment is delimited in the main syntax diagram by a set of vertical lines. The corresponding definition of the fragment begins with the name of the fragment followed by the syntax, which starts and ends with a vertical line.



The following example shows how the syntax is used:



IBM Extensions

An IBM® extension generally modifies a rule or restriction that immediately precedes it. The standard is presented first, because some programmers use the ILE COBOL language without IBM extensions. The extensions are then presented for those who **do** use them.

Clauses and statements illustrated within syntax diagrams that are ILE COBOL language extensions to the American National Standards Institute (ANSI) standard *X3.23b-1993, American National Standard for Information Systems - Programming Language - COBOL* are identified by footnotes.

IBM Extension

ILE COBOL language extensions to ANSI X3.23b-1993 COBOL that are part of the text description are enclosed in IBM Extension bars, like this paragraph.

End of IBM Extension

Documentary Syntax

Some COBOL clauses and statements are syntax checked and treated as documentation by the ILE COBOL compiler. Such clauses and statements are identified with a footnote in syntax diagrams.

CL Entry Codes

The code that appears in the upper right corner of each CL syntax diagram contains the entry codes that specify the environment in which the command can be entered. The codes indicate whether or not the command can be:

- Used in a batch or interactive job (outside a compiled program; Job:B or I)
- Used in a batch or interactive compiled program (Pgm:B or I)
- Used in a batch or interactive REXX procedure (REXX:B or I)
- Used as a parameter for the CALL CL command, or passed as a character string to the system program QCMDEXC (Exec).

Industry Standard

Standard COBOL refers to the COBOL programming language as defined in the document entitled American National Standard for Information Systems - Programming Language - COBOL, ANSI X3.23-1985, ISO 1989:1985, updated with the content of the following documents, in the order they are listed:

- ANSI X3.23a-1989, American National Standard for Information Systems - Programming Language - Intrinsic Function Module for COBOL and ISO 1989:1985/ Amd.1:1992
- Programming Languages - COBOL, AMENDMENT 1: Intrinsic function module
- ANSI X3.23b-1993, American National Standard for Information Systems - Programming Language - Correction Amendment for COBOL
- ISO/IEC 1989 DAM2 Programming Languages - COBOL, AMENDMENT 2: Correction and clarification amendment for COBOL.

From this point on, the term Standard COBOL will be used to refer to the ANSI standard just described.

Creating Module and Program Objects

Creating Module and Program Objects

Use the CRTCLMOD (Create COBOL Module) command to create one or more module objects from ILE COBOL source members. Use the CRTPGM (Create Program) command to bind the module objects created by the CRTCLMOD command into one or more program objects.

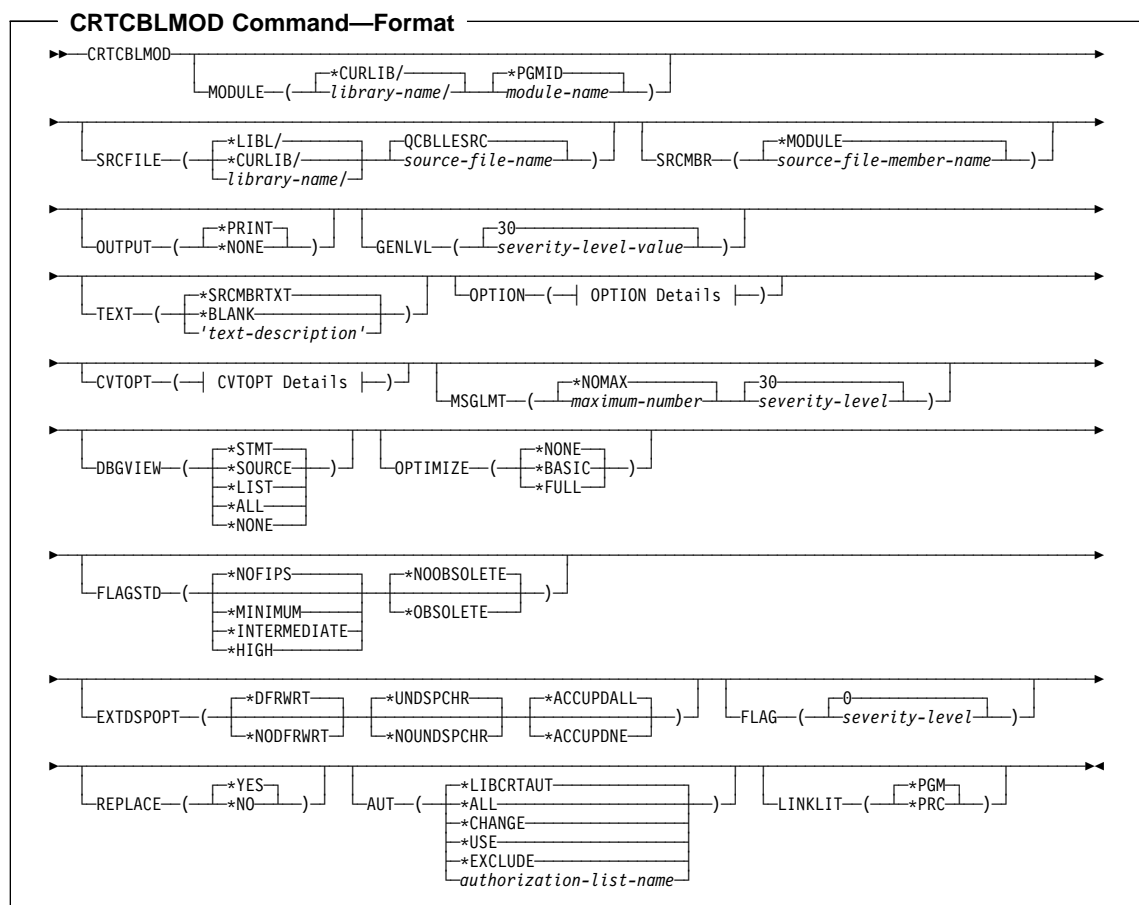
Use the CRTBNDCL (Create Bound COBOL) command to create one or more program objects directly from ILE COBOL source members.

You can use the PROCESS statement in your source member to override the options specified for the CRTCLMOD or CRTBNDCL command. The options of the PROCESS statement are covered in "PROCESS Statement" on page 6.

Creating Module and Program Objects

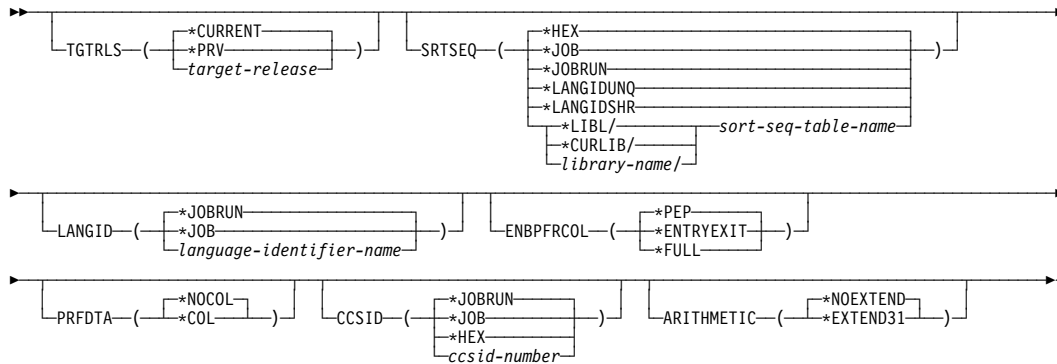
CRTCBLMOD Command Syntax

The following diagram shows the syntax of the CRTCBLMOD command:

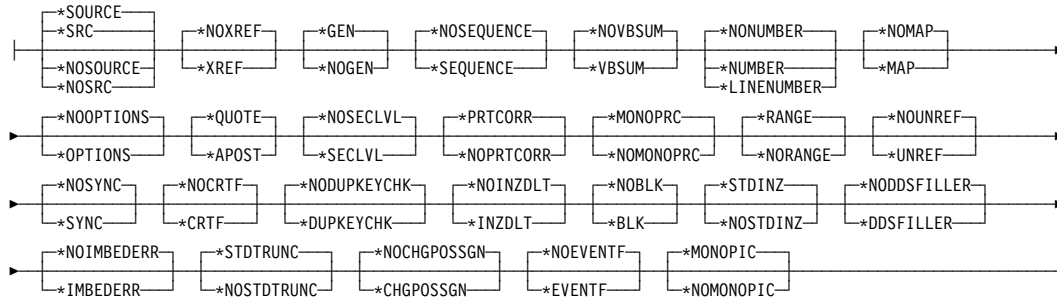


Creating Module and Program Objects

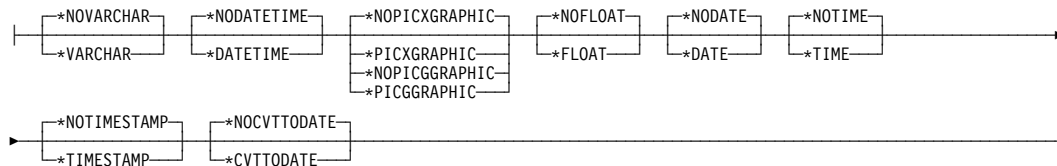
CRTCLMOD Command - Format (continued)



OPTION Details:



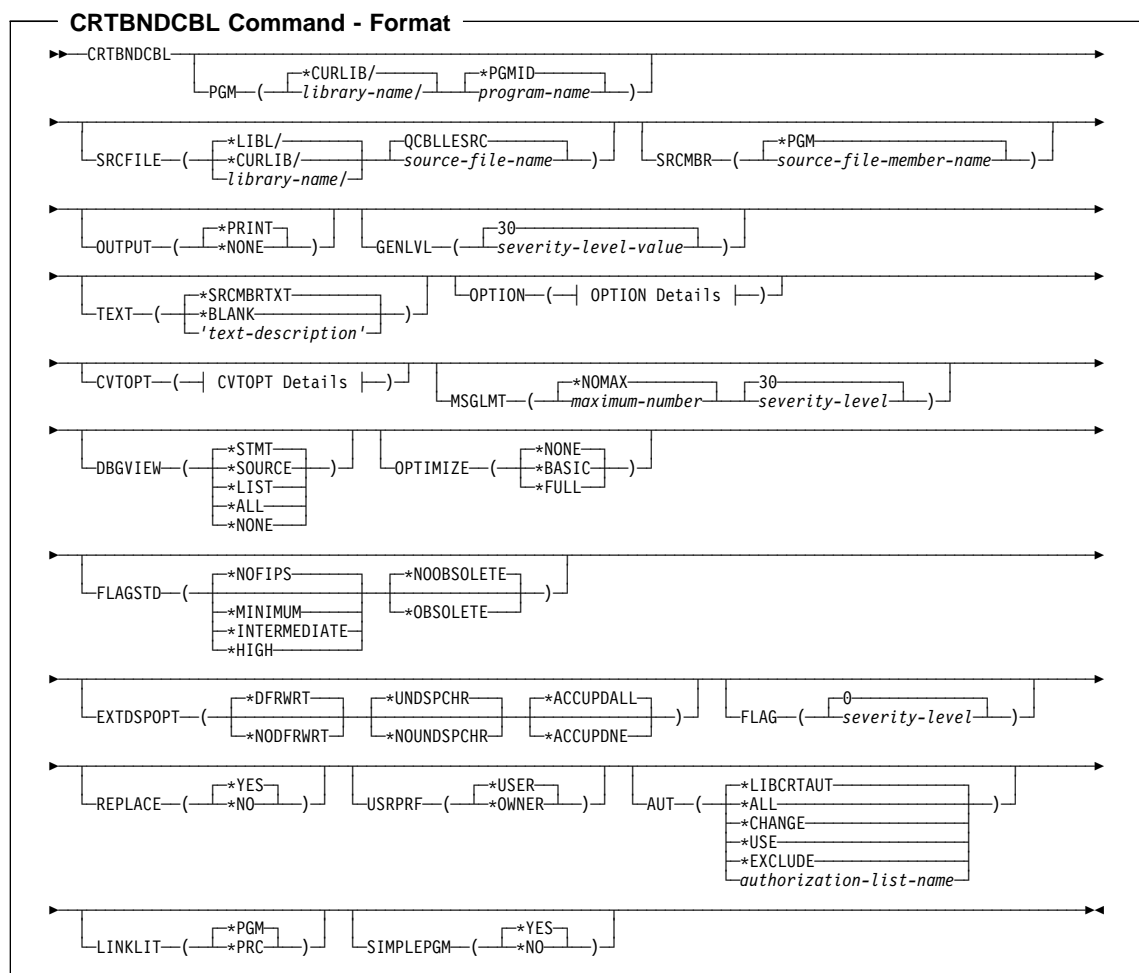
CVTOPT Details:



Creating Module and Program Objects

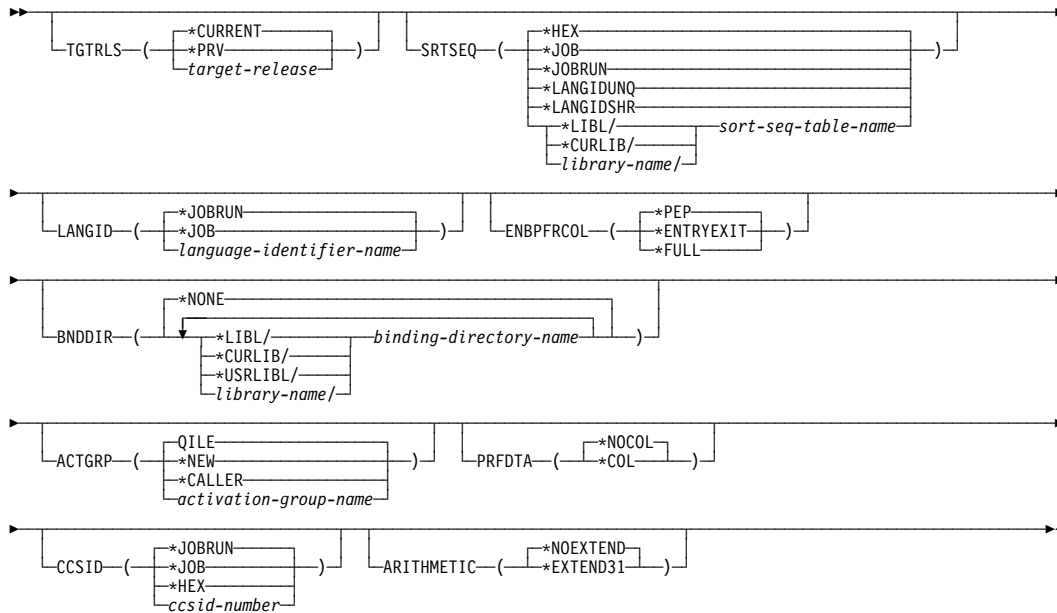
CRTBNDCBL Command Syntax

The following diagram shows the syntax of the CRTBNDCBL command:

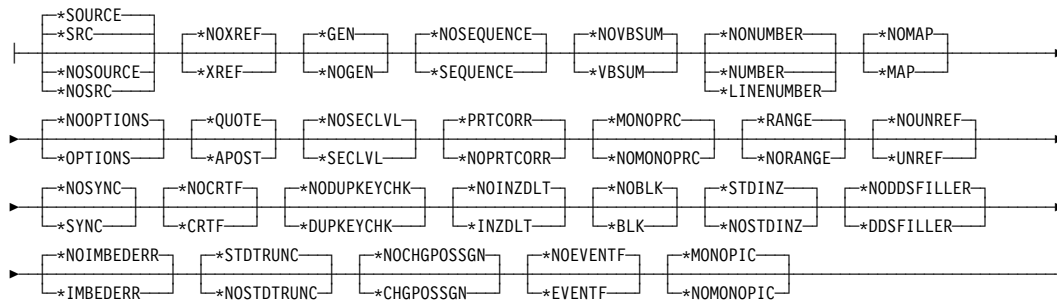


Creating Module and Program Objects

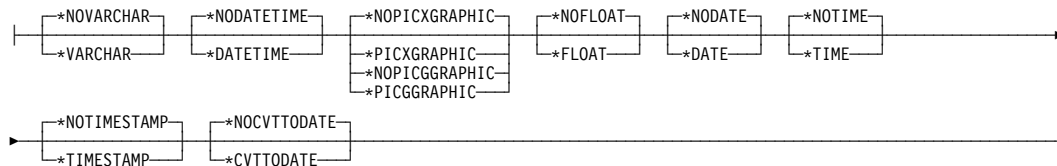
CRTBNDCBL Command - Format (continued)



OPTION Details:



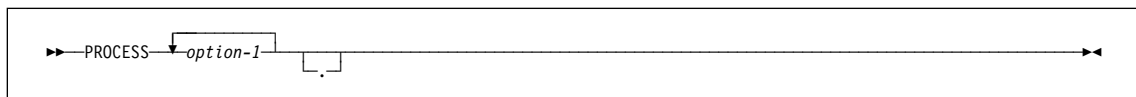
CVTOPT Details:



Creating Module and Program Objects

PROCESS Statement

The syntax of the PROCESS statement is:



PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	OUTPUT Parameter Options
<u>OUTPUT</u> NOOUTPUT	<u>*PRINT</u> <u>*NONE</u>

PROCESS Statement Option	CRTCBLMOD/CRTBND CBL
	GENLVL Parameter Option
GENLVL(nn)	nn

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	OPTION Parameter Options
<u>SOURCE</u> <u>SRC</u> NOSOURCE NOSRC	<u>*SOURCE</u> <u>*SRC</u> <u>*NOSOURCE</u> <u>*NOSRC</u>
<u>NOXREF</u> XREF	<u>*NOXREF</u> <u>*XREF</u>
<u>GEN</u> NOGEN	<u>*GEN</u> <u>*NOGEN</u>
<u>NOSEQUENCE</u> SEQUENCE	<u>*NOSEQUENCE</u> <u>*SEQUENCE</u>
<u>NOVBSUM</u> VBSUM	<u>*NOVBSUM</u> <u>*VBSUM</u>
<u>NONUMBER</u> NUMBER LINENUMBER	<u>*NONUMBER</u> <u>*NUMBER</u> <u>*LINENUMBER</u>
<u>NOMAP</u> MAP	<u>*NOMAP</u> <u>*MAP</u>
<u>NOOPTIONS</u> OPTIONS	<u>*NOOPTIONS</u> <u>*OPTIONS</u>
<u>QUOTE</u> APOST	<u>*QUOTE</u> <u>*APOST</u>
<u>NOSECLVL</u> SECLVL	<u>*NOSECLVL</u> <u>*SECLVL</u>

Creating Module and Program Objects

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	OPTION Parameter Options
PRTCORR NOPRTCORR	* PRTCORR *NOPRTCORR
MONOPRC NOMONOPRC	* MONOPRC *NOMONOPRC
RANGE NORANGE	* RANGE *NORANGE
NOUNREF UNREF	* NOUNREF *UNREF
NOSYNC SYNC	* NOSYNC *SYNC
NOCRTF CRTF	* NOCRTF *CRTF
NODUPKEYCHK DUPKEYCHK	* NODUPKEYCHK *DUPKEYCHK
NOINZDLT INZDLT	* NOINZDLT *INZDLT
NOBLK BLK	* NOBLK *BLK
STDINZ NOSTDINZ	* STDINZ *NOSTDINZ
NODDSFILLER DDSFILLER	* NODDSFILLER *DDSFILLER
Not applicable	* NOIMBEDERR *IMBEDERR
STDTRUNC NOSTDTRUNC	* STDTRUNC *NOSTDTRUNC
CHGPOSSGN NOCHGPOSSGN	* CHGPOSSGN *NOCHGPOSSGN
Not applicable	* NOEVENTF *EVENTF
MONOPIC NOMONOPIC	* MONOPIC *NOMONOPIC

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	CVTOPT Parameter Options
NOVARCHAR VARCHAR	* NOVARCHAR *VARCHAR
NODATETIME DATETIME	* NODATETIME *DATETIME

Creating Module and Program Objects

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	CVTOPT Parameter Options
<u>NOCVTPIXGRAPHIC</u> CVTPIXGRAPHIC CVTPIXGGRAPHIC NOCVTPIXGGRAPHIC	<u>*NOPIXGRAPHIC</u> *PIXGRAPHIC *PIXGGRAPHIC *NOPIXGGRAPHIC
<u>NOFLOAT</u> FLOAT	<u>*NOFLOAT</u> *FLOAT
<u>NODATE</u> DATE	<u>*NODATE</u> *DATE
<u>NOTIME</u> TIME	<u>*NOTIME</u> *TIME
<u>NOTIMESTAMP</u> TIMESTAMP	<u>*NOTIMESTAMP</u> *TIMESTAMP
<u>NOCVTTODATE</u> CVTTODATE	<u>*NOCVTTODATE</u> *CVTTODATE

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	OPTIMIZE Parameter Options
<u>NOOPTIMIZE</u> BASICOPT FULLOPT	<u>*NONE</u> *BASIC *FULL

PROCESS Statement Options	CRTCBLMOD/CRTBND CBL
	FLAGSTD Parameter Options
<u>NOFIPS</u> MINIMUM INTERMEDIATE HIGH	<u>*NOFIPS</u> *MINIMUM *INTERMEDIATE *HIGH
<u>NOOBSOLETE</u> OBSOLETE	<u>*NOOBSOLETE</u> *OBSOLETE

PROCESS Statement Options EXTDSPOPT(a b c)	CRTCBLMOD/CRTBND CBL
	EXTDSPOPT Parameter Options
<u>DFRWRT</u> NODFRWRT	<u>*DFRWRT</u> *NODFRWRT
<u>UNDSPCHR</u> NOUNDSPCHR	<u>*UNDSPCHR</u> *NOUNDSPCHR
<u>ACCUPDALL</u> ACCUPDNE	<u>*ACCUPDALL</u> *ACCUPDNE

Creating Module and Program Objects

PROCESS Statement Option	CRTCBMOD/CRTBND CBL
	FLAG Parameter Option
FLAG(nn)	nn

PROCESS Statement Options	CRTCBMOD/CRTBND CBL
	LINKLIT Parameter Options
<u>LINKPGM</u> LINKPRC	<u>*PGM</u> <u>*PRC</u>

PROCESS Statement Options SRTSEQ(a)	CRTCBMOD/CRTBND CBL
	SRTSEQ Parameter Options
<u>HEX</u> JOB JOBRUN LANGIDUNQ LANGIDSHR "LIBL/sort-seq-table-name" "CURLIB/sort-seq-table-name" "library-name/sort-seq-table-name" "sort-seq-table-name"	<u>*HEX</u> *JOB *JOBRUN *LANGIDUNQ *LANGIDSHR *LIBL/sort-seq-table-name *CURLIB/sort-seq-table-name library-name/sort-seq-table-name sort-seq-table-name

PROCESS Statement Options LANGID(a)	CRTCBMOD/CRTBND CBL
	LANGID Parameter Options
<u>JOBRUN</u> JOB "language-identifier-name"	<u>*JOBRUN</u> *JOB language-identifier-name

PROCESS Statement Options ENBPFCOL(a)	CRTCBMOD/CRTBND CBL
	ENBPFCOL Parameter Options
<u>PEP</u> ENTRYEXIT FULL	<u>*PEP</u> *ENTRYEXIT *FULL

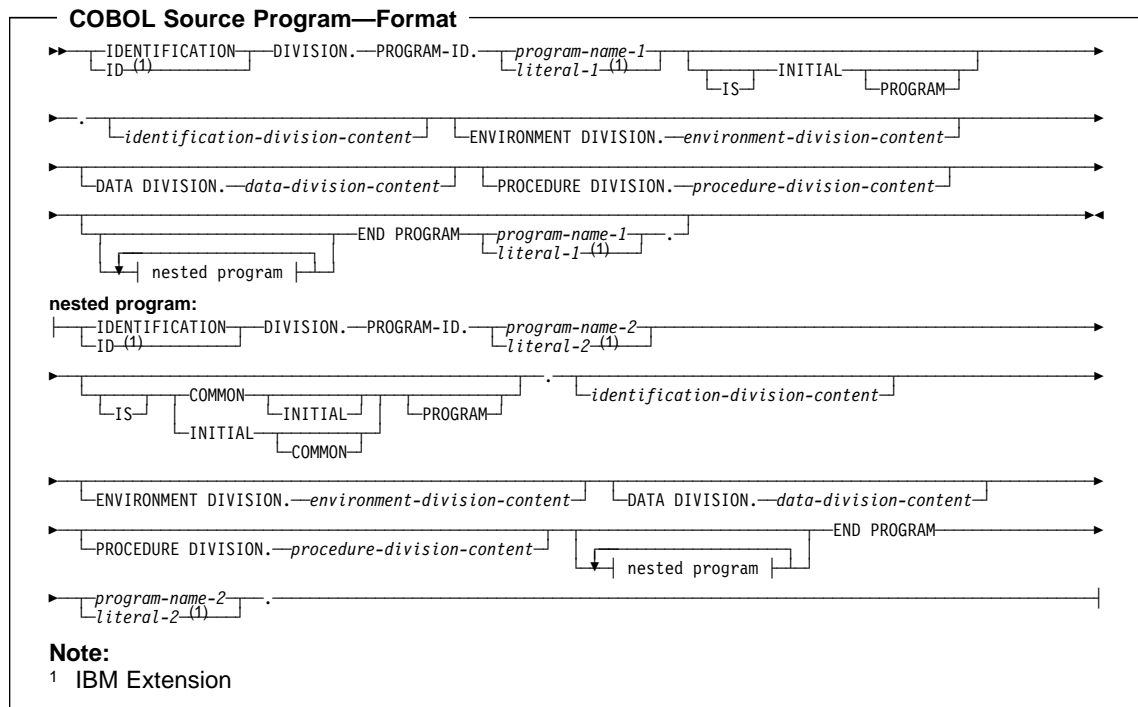
PROCESS Statement Options PRFDTA(a)	CRTCBMOD/CRTBND CBL
	PRFDTA Parameter Options
<u>NOCOL</u> COL	<u>*NOCOL</u> *COL

Creating Module and Program Objects

PROCESS Statement Options CCSID(<i>a b c</i>)	CRTCBLMOD/CRTBND CBL CCSID Parameter Options
<i>a</i> = Locale single-byte data CCSID	
JOBRUN JOB HEX <i>coded-character-set-identifier</i>	*JOBRUN *JOB *HEX <i>coded-character-set-identifier</i>
<i>b</i> = Non-locale single-byte data CCSID	
JOBRUN JOB HEX <i>coded-character-set-identifier</i>	Not applicable
<i>c</i> = Non-locale double-byte data CCSID	
JOBRUN JOB HEX <i>coded-character-set-identifier</i>	Not applicable
PROCESS Statement Options DATTIM(<i>a b</i>)	CRTCBLMOD/CRTBND CBL
<i>4-digit base century</i> (default 1900) <i>2-digit base year</i> (default 40)	Not applicable
PROCESS Statement Options THREAD(<i>a</i>)	CRTCBLMOD/CRTBND CBL
NOTHREAD SERIALIZE	Not applicable
PROCESS Statement Options ARITHMETIC(<i>a</i>)	CRTCBLMOD/CRTBND CBL ARITHMETIC Parameter Options
NOEXTEND EXTEND31	*NOEXTEND *EXTEND31
PROCESS Statement Option	CRTCBLMOD/CRTBND CBL
NOGRAPHIC GRAPHIC	Not applicable

COBOL Source Program—General Structure

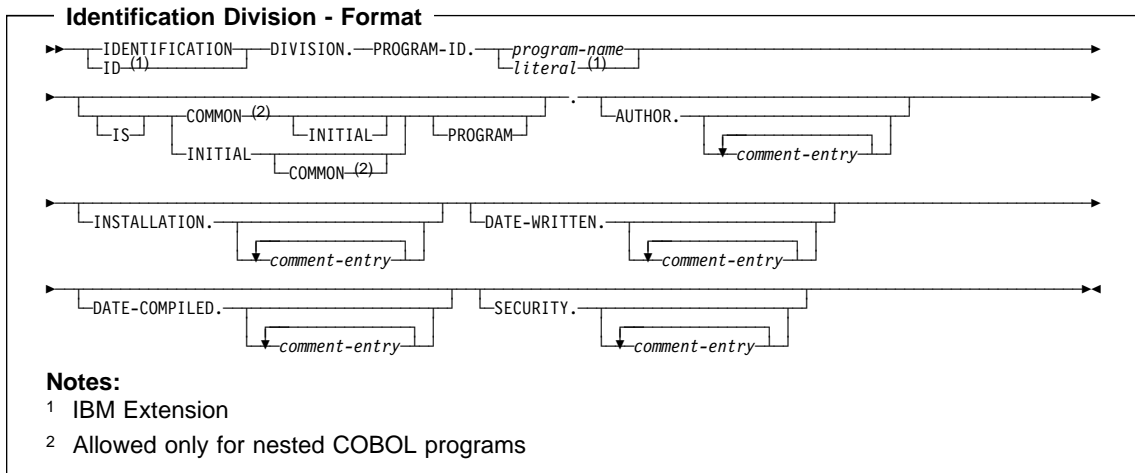
COBOL Source Program—General Structure



COBOL Source Program—General Structure

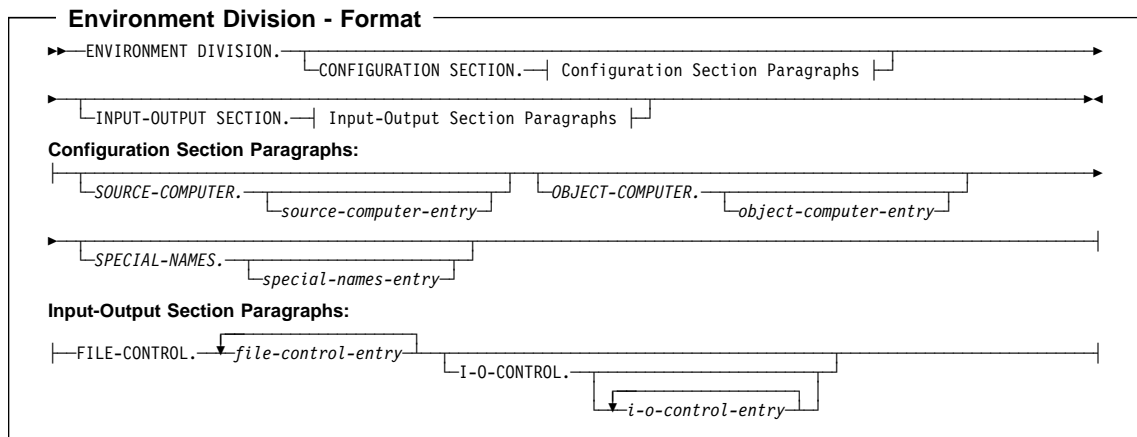
Identification Division

Identification Division



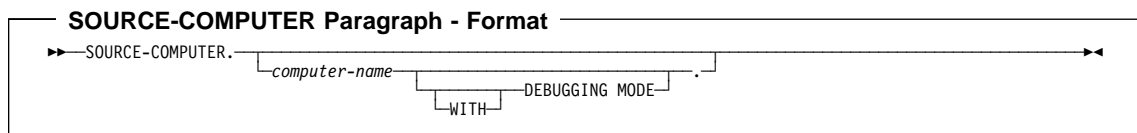
Identification Division

Environment Division

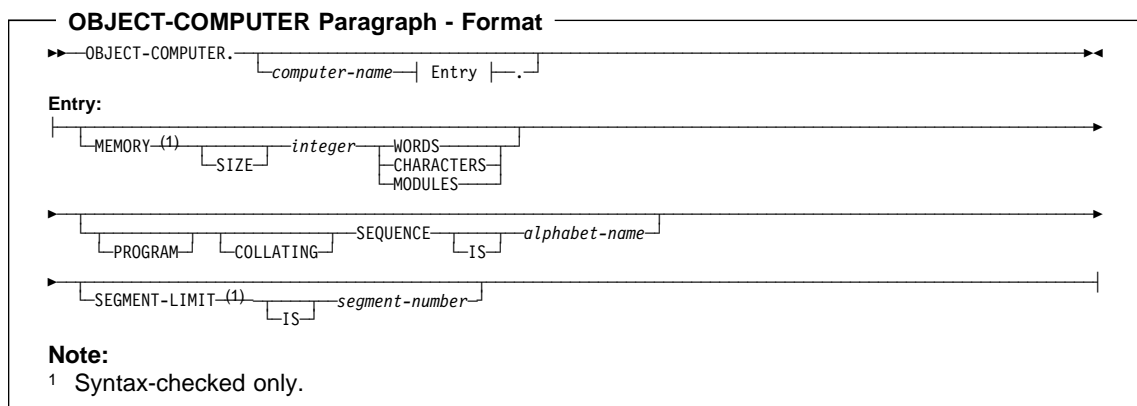


Configuration Section

SOURCE-COMPUTER Paragraph

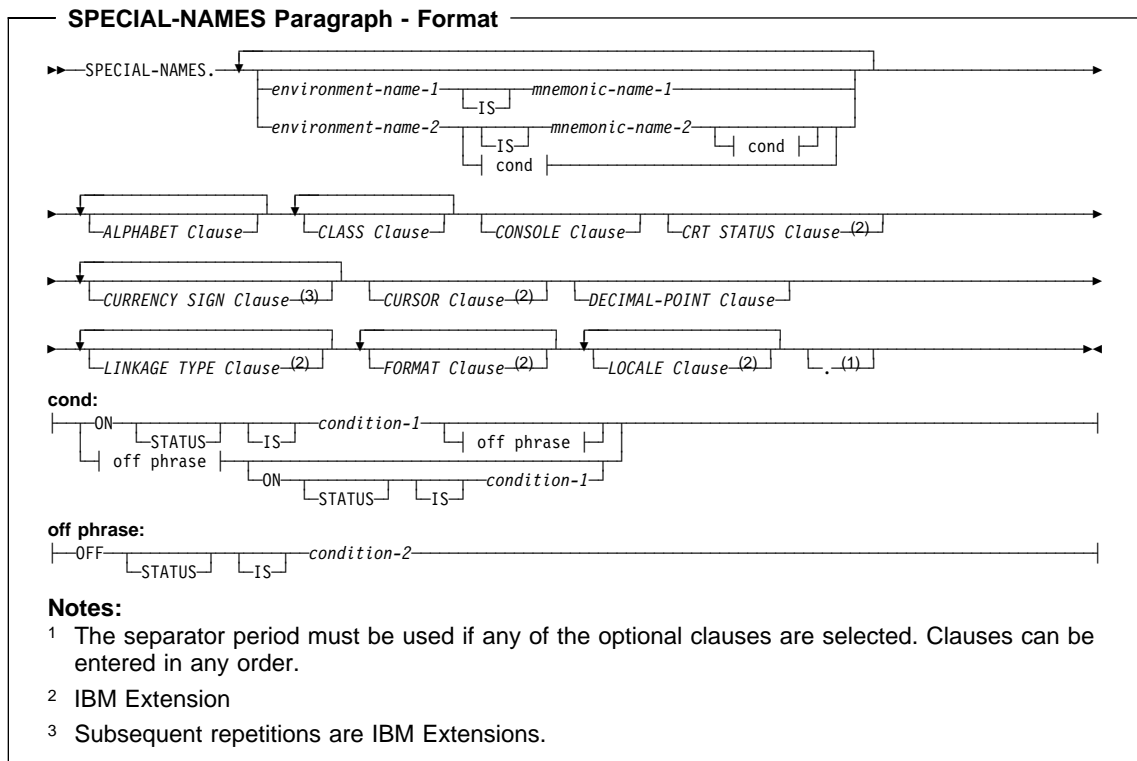


OBJECT-COMPUTER Paragraph

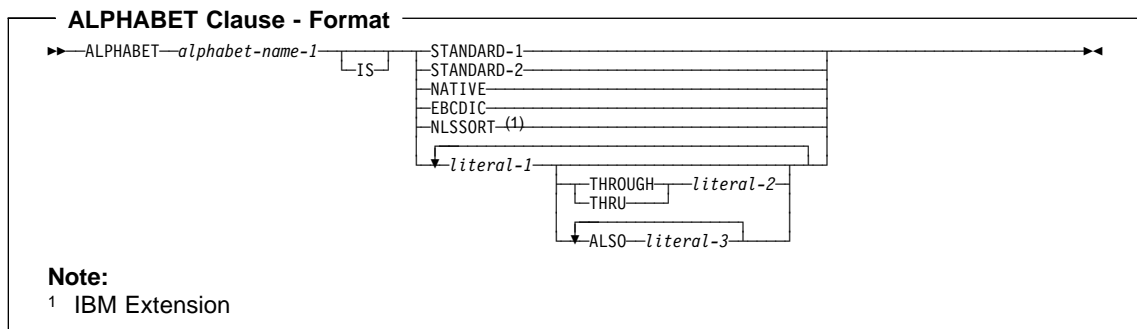


Environment Division

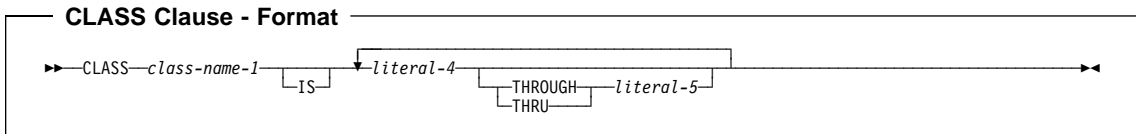
SPECIAL-NAMES Paragraph



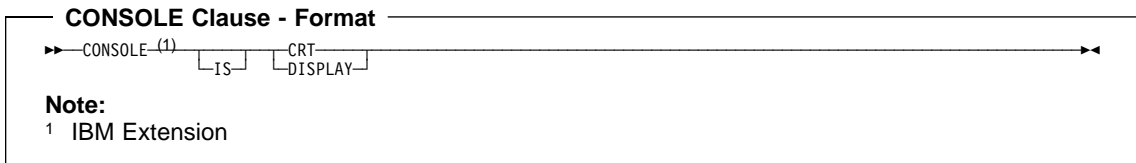
ALPHABET Clause



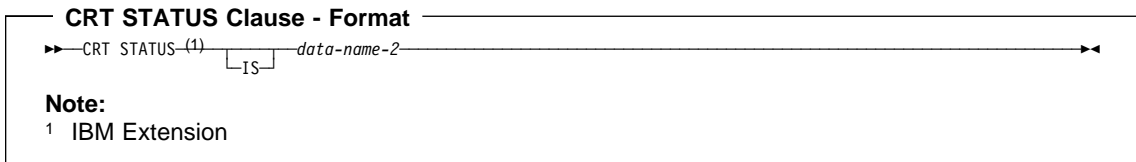
CLASS Clause



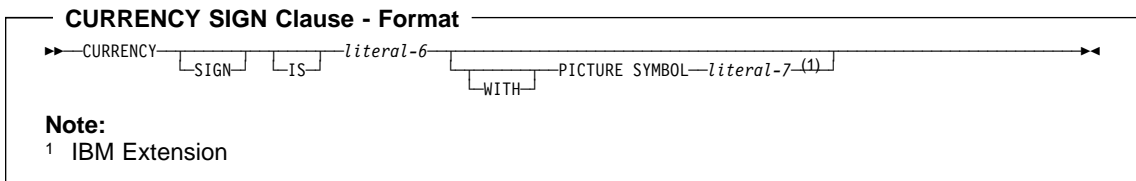
CONSOLE Clause



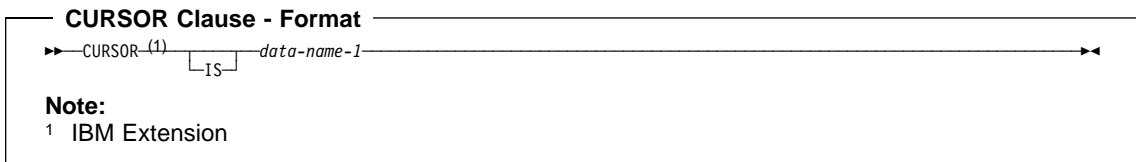
CRT STATUS Clause



CURRENCY Clause

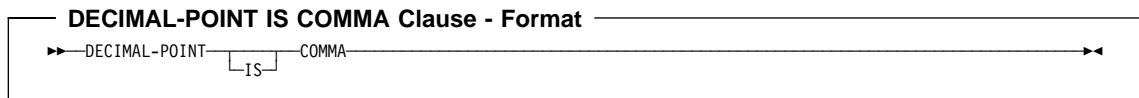


CURSOR Clause



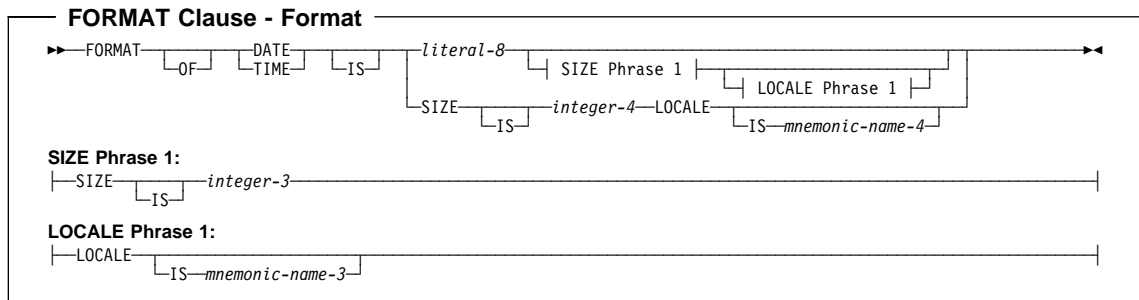
Environment Division

DECIMAL-POINT Clause



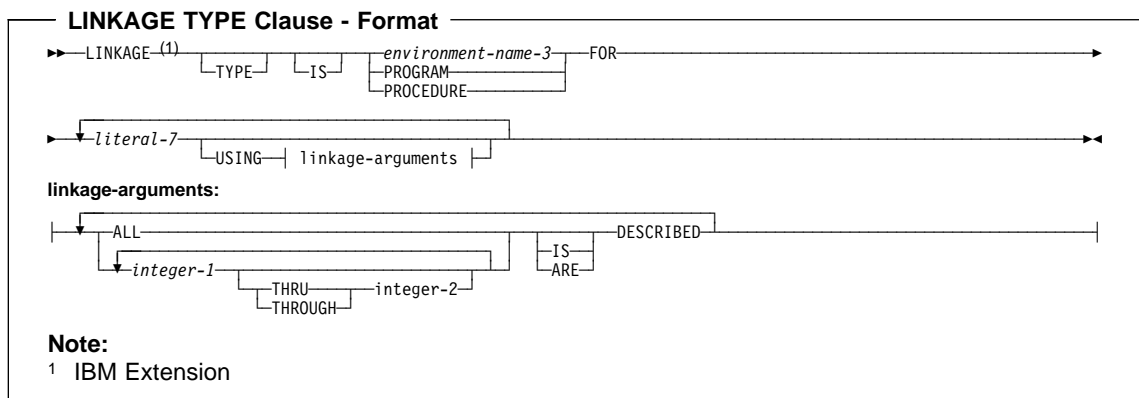
IBM Extension

FORMAT Clause



End of IBM Extension

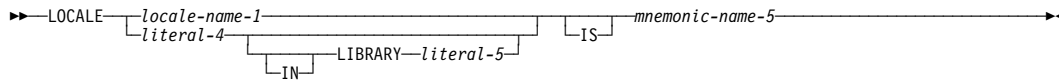
LINKAGE-TYPE Clause



IBM Extension

LOCALE Clause

LOCALE Clause - Format



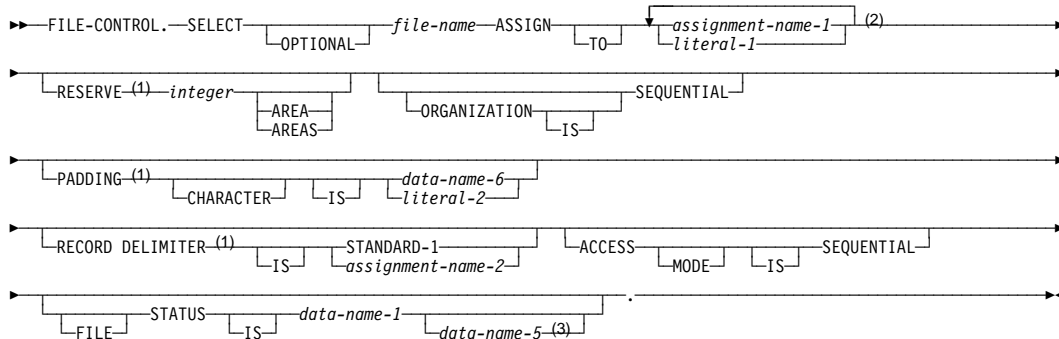
End of IBM Extension

Input-Output Section

Note: The keyword FILE-CONTROL appears only once at the beginning of the paragraph before the first file-control entry. The keyword I-O-CONTROL appears only once at the beginning of the paragraph before the first input-output-control entry.

FILE-CONTROL Paragraph

FILE-CONTROL Paragraph - Format 1 - Sequential

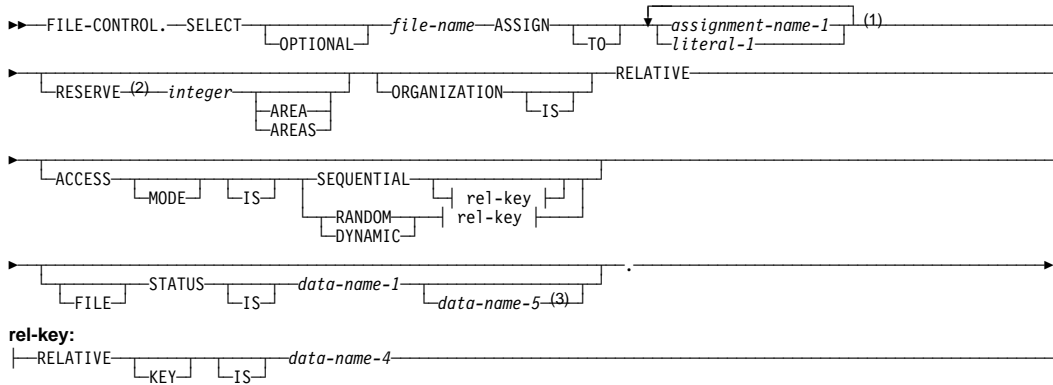


Notes:

- 1 Syntax-checked only.
- 2 Subsequent repetitions syntax-checked only.
- 3 IBM Extension

Environment Division

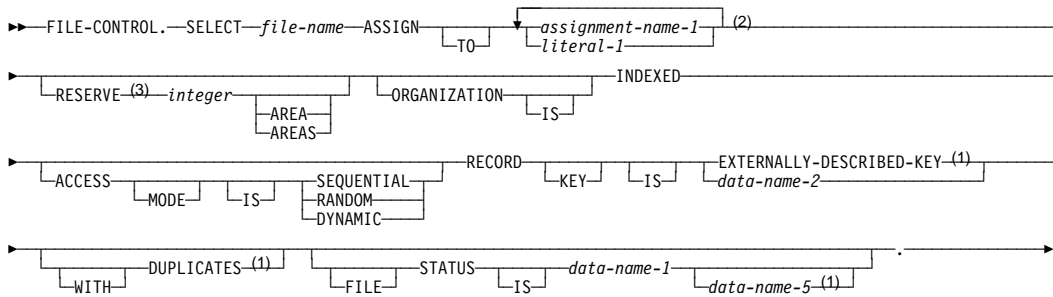
FILE-CONTROL Paragraph - Format 2 - Relative



Notes:

- 1 Subsequent repetitions syntax-checked only.
- 2 Syntax-checked only.
- 3 IBM Extension

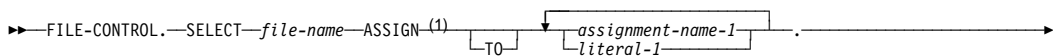
FILE-CONTROL Paragraph - Format 3 - Indexed



Notes:

- 1 IBM Extension
- 2 Subsequent repetitions syntax-checked only.
- 3 Syntax-checked only.

FILE-CONTROL Paragraph - Format 4 - Sort or Merge



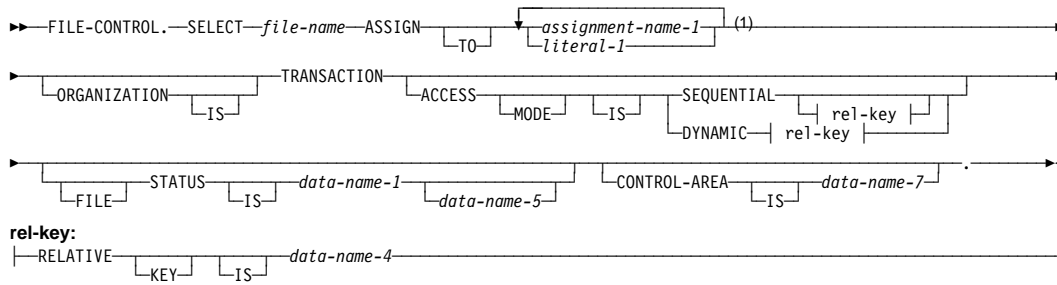
Note:

- 1 Syntax checked only.

Environment Division

IBM Extension

FILE-CONTROL Paragraph - Format 5 - Transaction



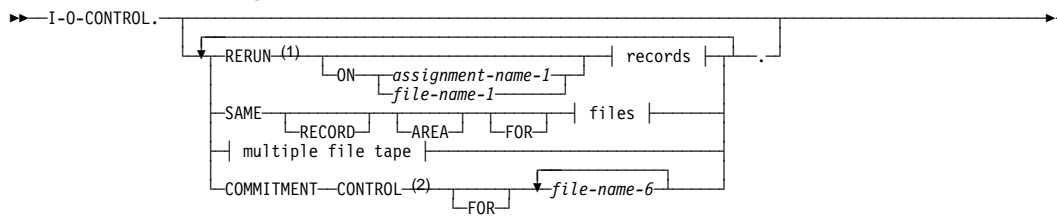
Note:

1 Subsequent repetitions syntax checked only.

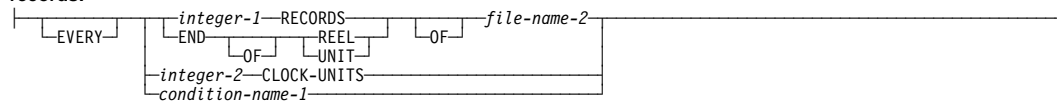
End of IBM Extension

I-O-CONTROL Paragraph

I-O-CONTROL Paragraph - Format 1 - Sequential



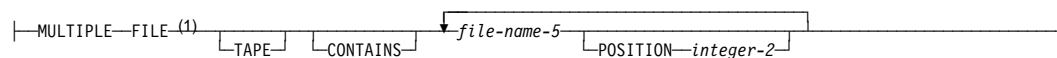
records:



files:



multiple file tape:



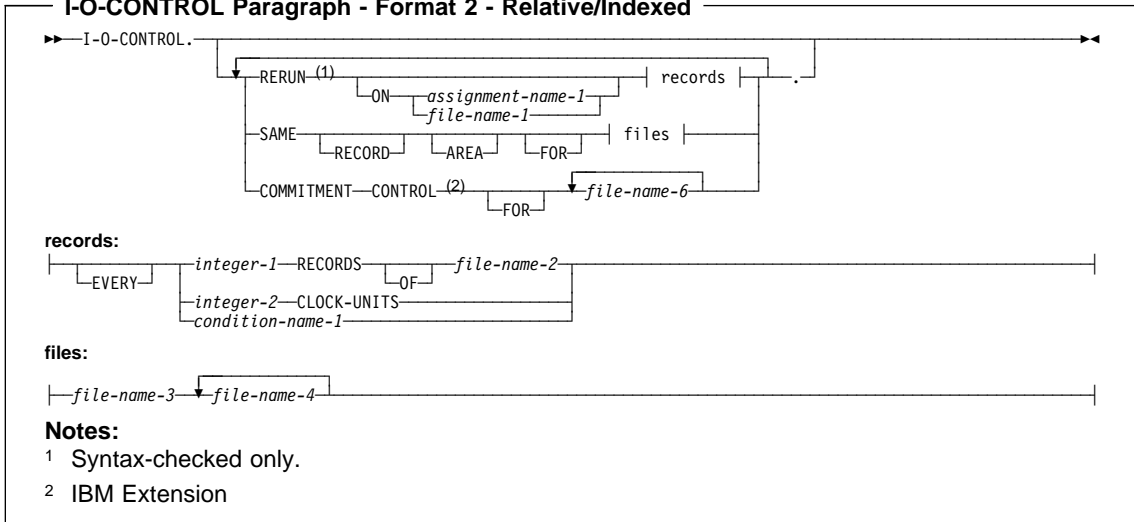
Notes:

1 Syntax-checked only.

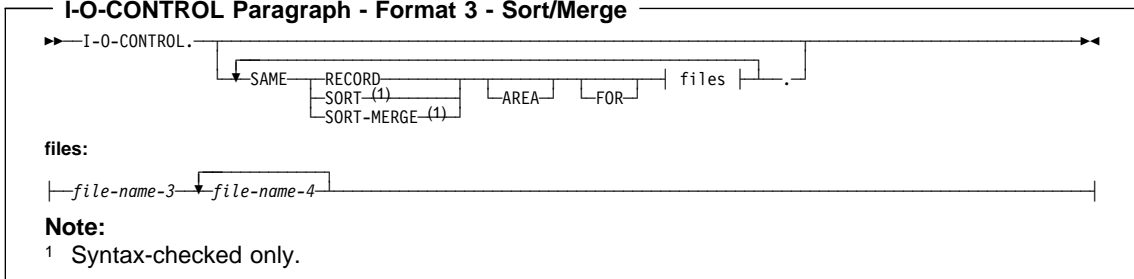
2 IBM Extension

Environment Division

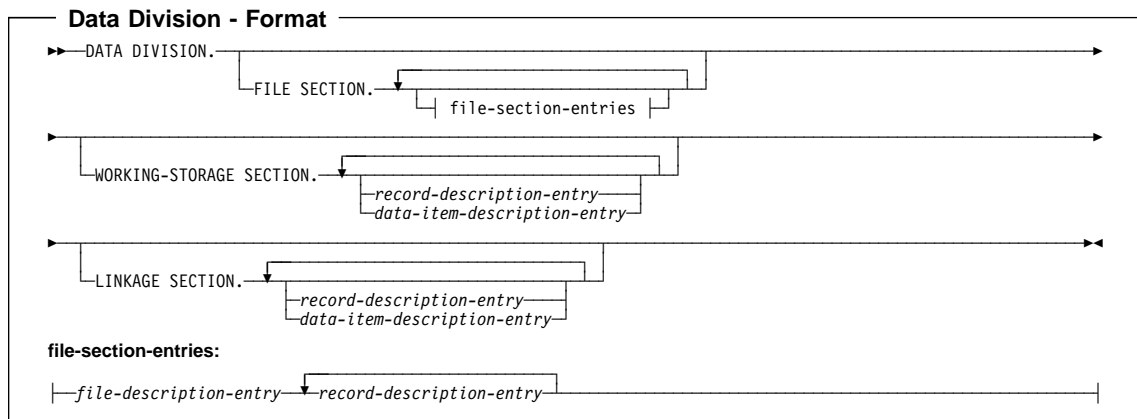
I-O-CONTROL Paragraph - Format 2 - Relative/Indexed



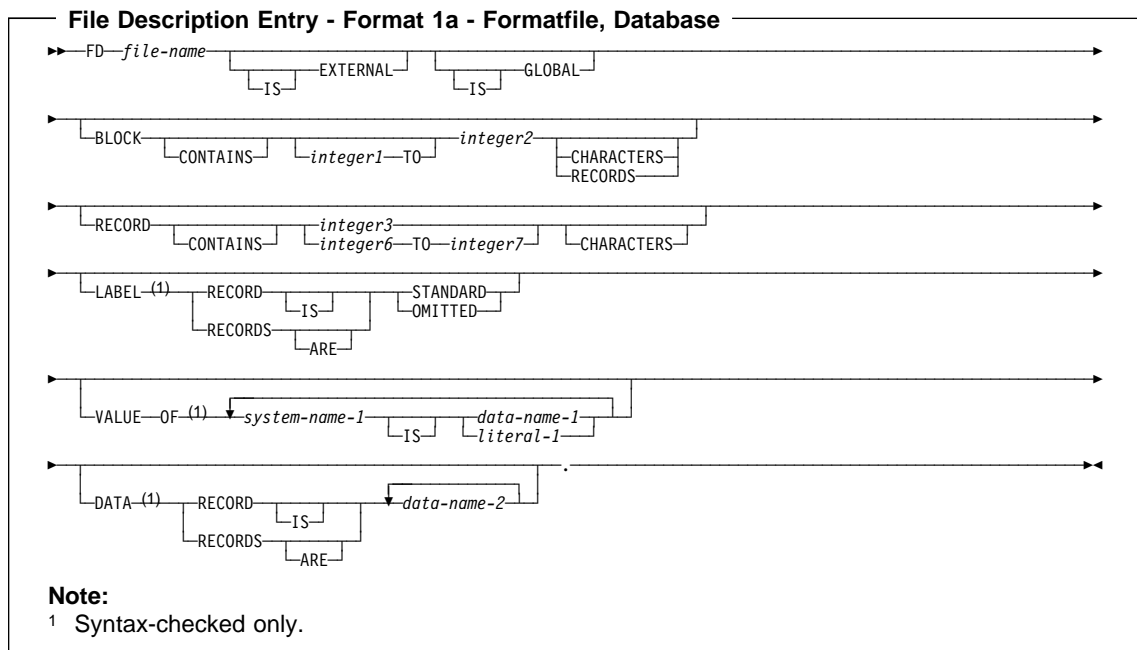
I-O-CONTROL Paragraph - Format 3 - Sort/Merge



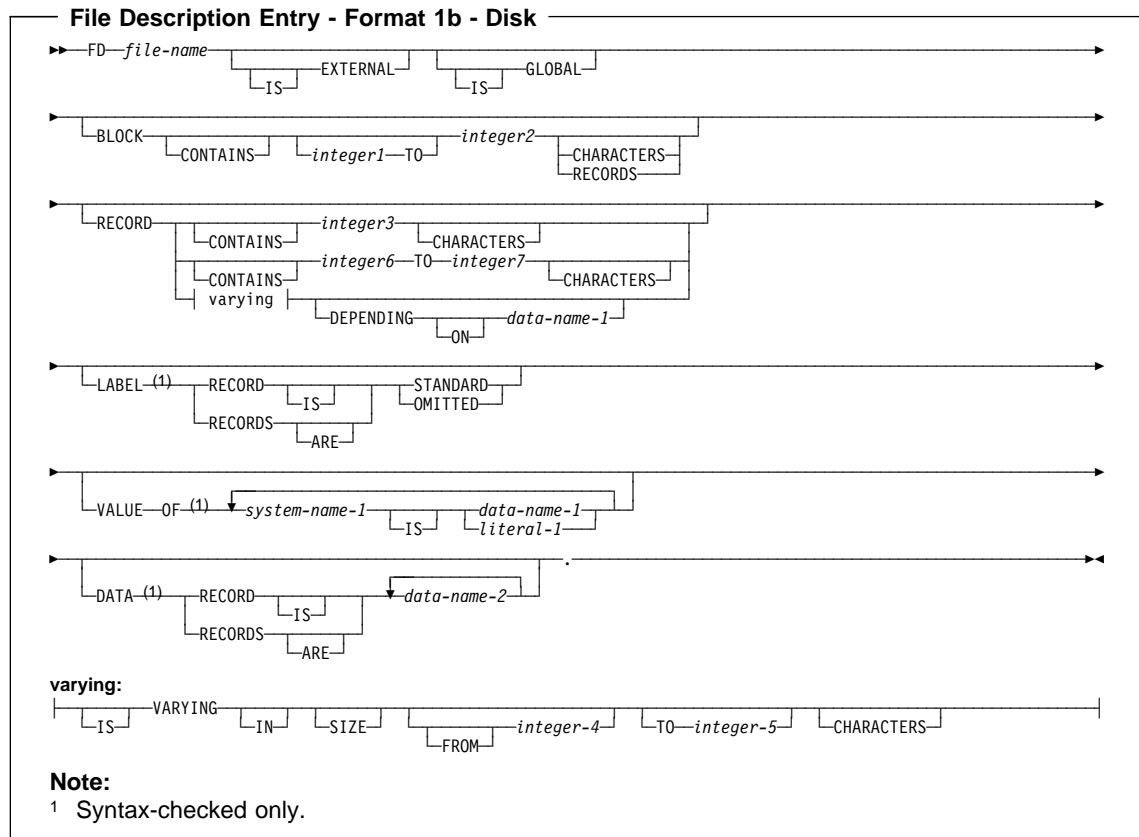
Data Division



File Section

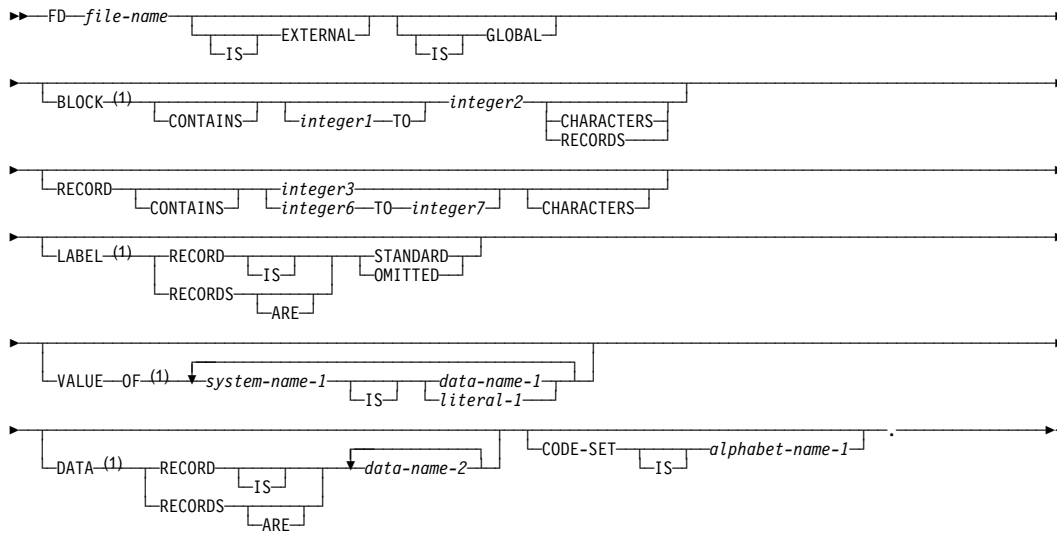


Data Division



Data Division

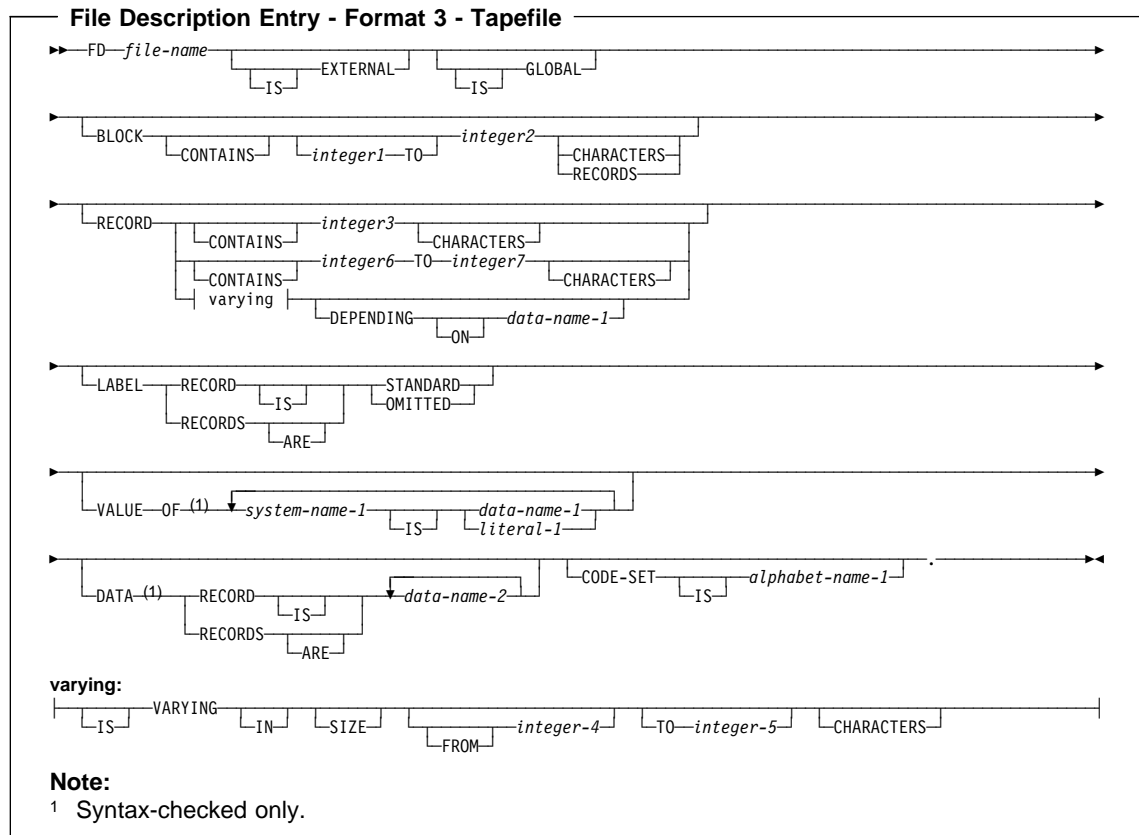
File Description Entry - Format 2 - Diskette



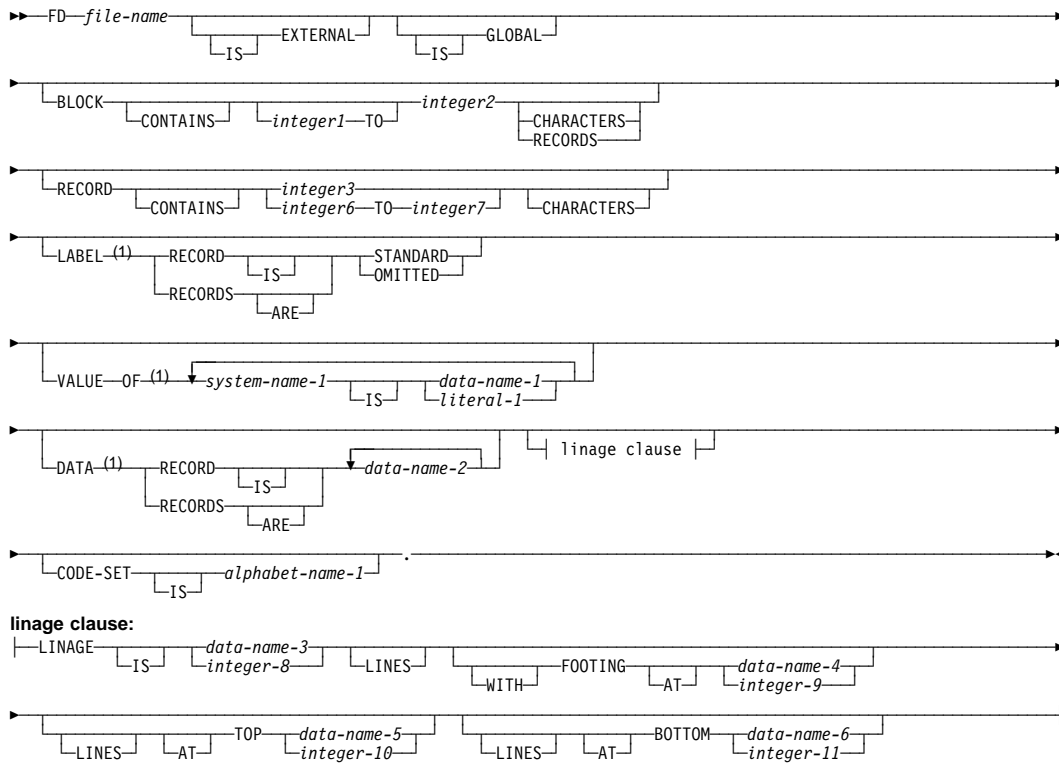
Note:

¹ Syntax-checked only.

Data Division

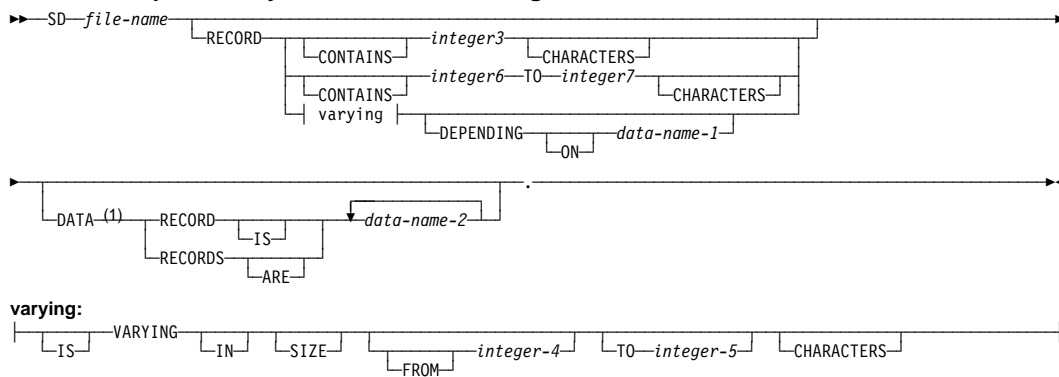


File Description Entry - Format 4 - Printer



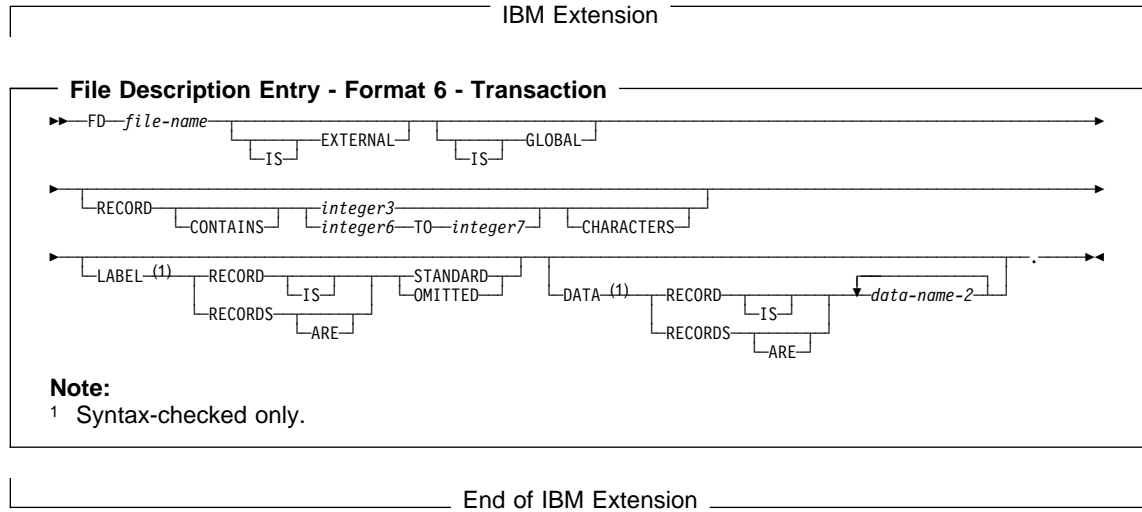
Note:
 1 Syntax-checked only.

File Description Entry - Format 5 - Sort/Merge



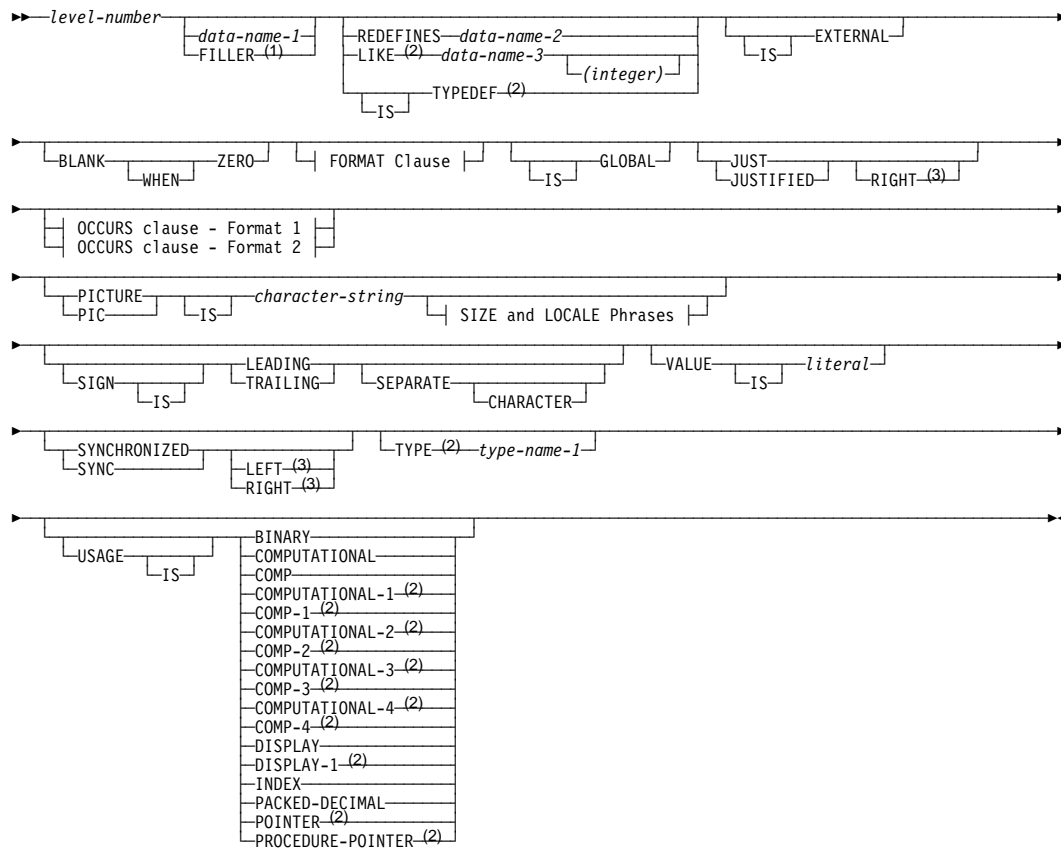
Note:
 1 Syntax-checked only.

Data Division



Working-Storage Section

Data Description Entry - General Format 1



Notes:

- 1 Cannot be used with the TYPEDEF clause.
- 2 IBM Extension
- 3 Syntax-checked only

Data Division

Data Description Entry - General Format 1 (continued)

OCCURS clause - Format 1:

| OCCURS *integer-2* TIMES | key-indexed-by phrase |

OCCURS clause - Format 2:

| OCCURS *integer-1* TO *integer-2* TIMES DEPENDING ON *data-name-1* | key-indexed-by phrase |

key-indexed-by phrase:

| ASCENDING (1) | DESCENDING (1) | KEY | IS | *data-name-2* | INDEXED | BY | *index-name-1* |

Size and Locale Phrases:

| SIZE (2) | IS | *integer-3* | LOCALE | IS | *mnemonic-name-1* |

FORMAT Clause:

| FORMAT (2) | OF | DATE | TIME | TIMESTAMP | IS | *literal-2* | phrase 1 | phrase 2 |

phrase 1:

| SIZE | IS | *integer-4* | LOCALE | IS | *mnemonic-name-2* |

phrase 2:

| SIZE | IS | *integer-5* | LOCALE | IS | *mnemonic-name-3* |

Notes:

- 1 Cannot be used with boolean data type
- 2 IBM Extension

Data Description Entry - General Format 2

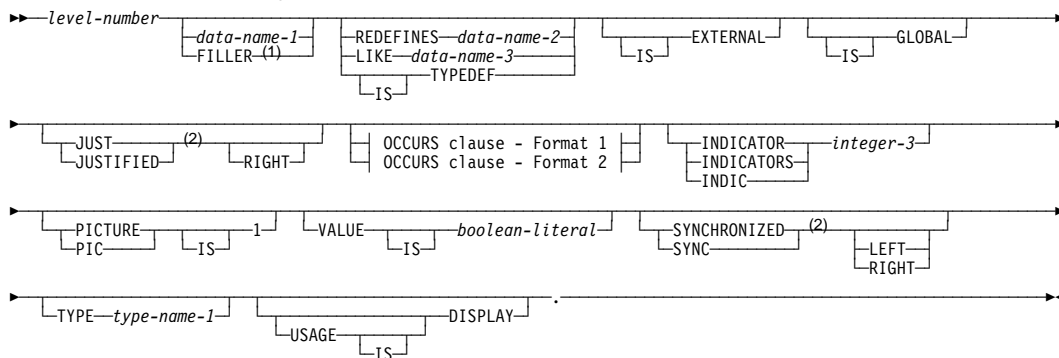
► 66 *data-name-1* RENAMES *data-name-2* THROUGH *data-name-3* .

Data Description Entry - General Format 3

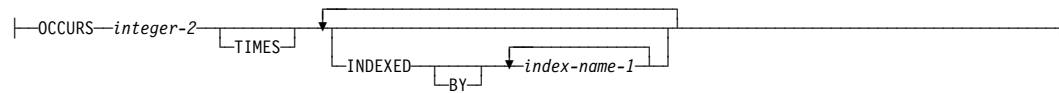
► 88 *condition-name* VALUE IS *literal-1* THROUGH *literal-2* .
VALUES ARE

IBM Extension

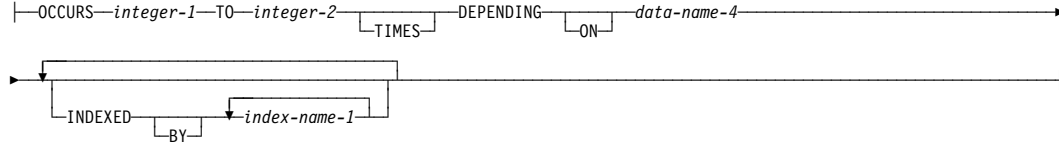
Data Description Entry - Format 4 - Boolean Data



OCCURS clause - Format 1:



OCCURS clause - Format 2:



Notes:

- 1 Cannot be used with the TYPEDEF clause.
- 2 Syntax-checked only

End of IBM Extension

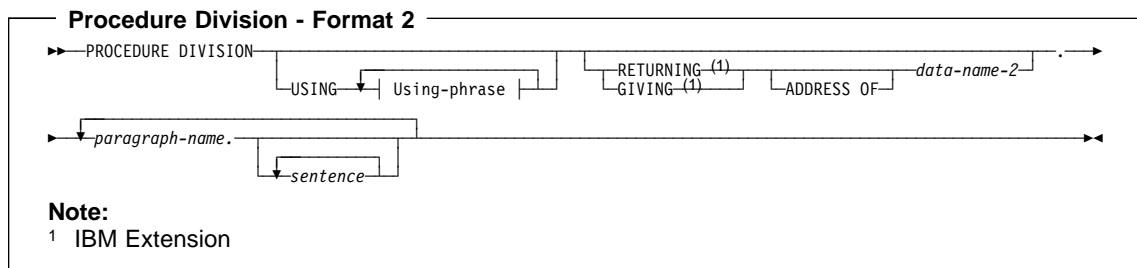
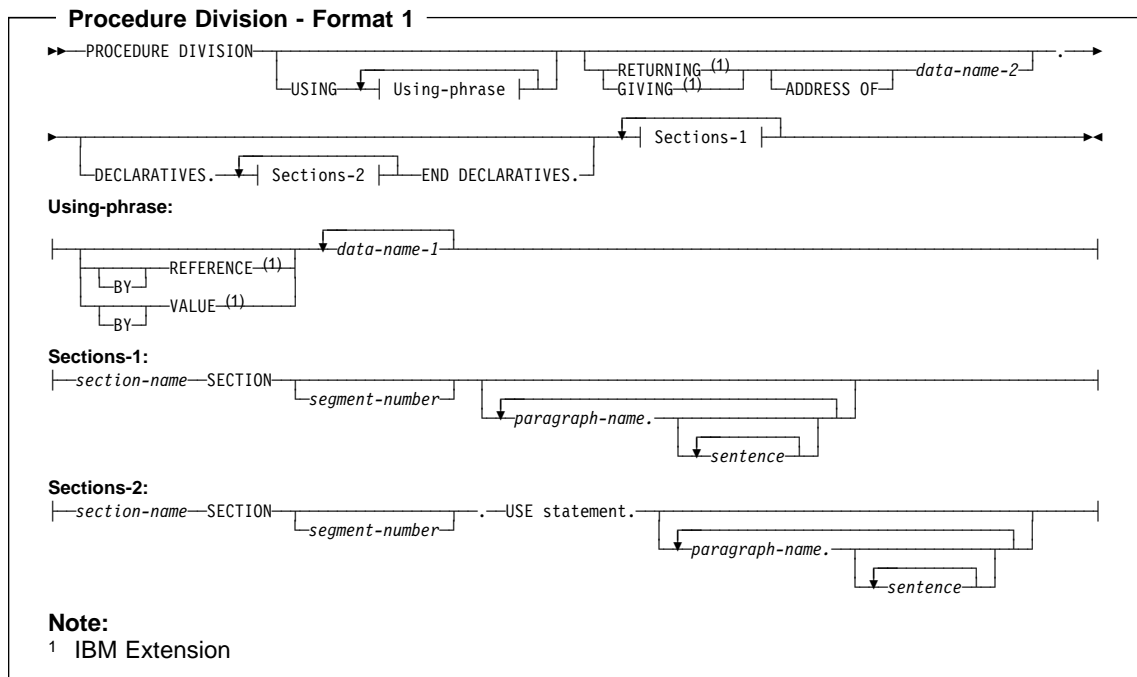
Data Division

Linkage Section

See “Working-Storage Section” on page 29 for data description entry clause formats.

The EXTERNAL clause cannot be specified in the Linkage Section.

Procedure Division



Procedure Division Statements

Procedure Division Statements

These statements are presented in alphabetical order.

ACCEPT Statement

ACCEPT Statement - Format 1 - Data Transfer

```

ACCEPT—identifier-1 FROM mnemonic-name environment-name (1) END-ACCEPT—(1)

```

Note:

¹ IBM Extension

ACCEPT Statement - Format 2 - System Info Transfer

```

ACCEPT—identifier-1 FROM DATE YYYYMMDD (1) END-ACCEPT—(1)
                        DAY YYYYDDD (1)
                        DAY-OF-WEEK
                        TIME

```

Note:

¹ IBM Extension

IBM Extension

ACCEPT Statement - Format 3 - Feedback

```

ACCEPT—identifier-1 FROM mnemonic-name FOR file-name-1 END-ACCEPT—(1)

```

Note:

¹ IBM Extension

ACCEPT Statement - Format 4 - Local Data Area

```

ACCEPT—identifier-1 FROM mnemonic-name FOR (1) identifier-2 literal-1 END-ACCEPT—

```

Note:

¹ Syntax-checked only.

ACCEPT Statement - Format 5 - PIP Data Area

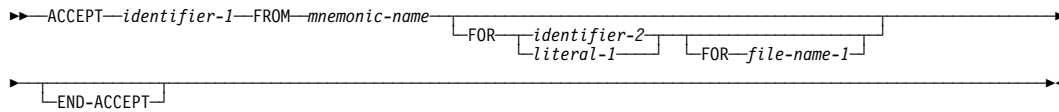
```

ACCEPT—identifier-1 FROM mnemonic-name ON EXCEPTION imperative-statement-1
NOT ON EXCEPTION imperative-statement-2 END-ACCEPT—

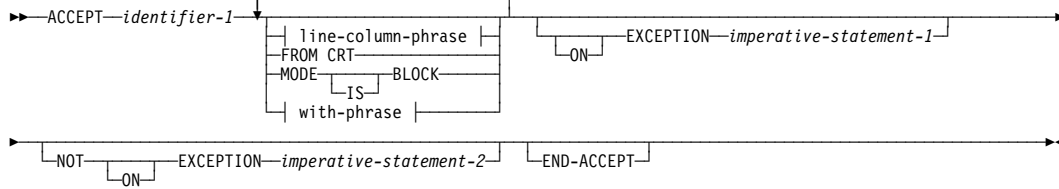
```

Procedure Division Statements

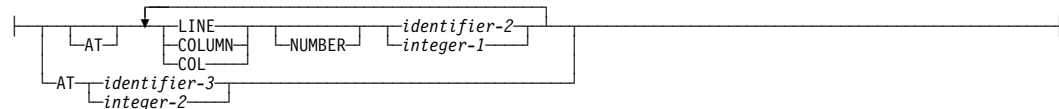
ACCEPT Statement - Format 6 - Attribute Data



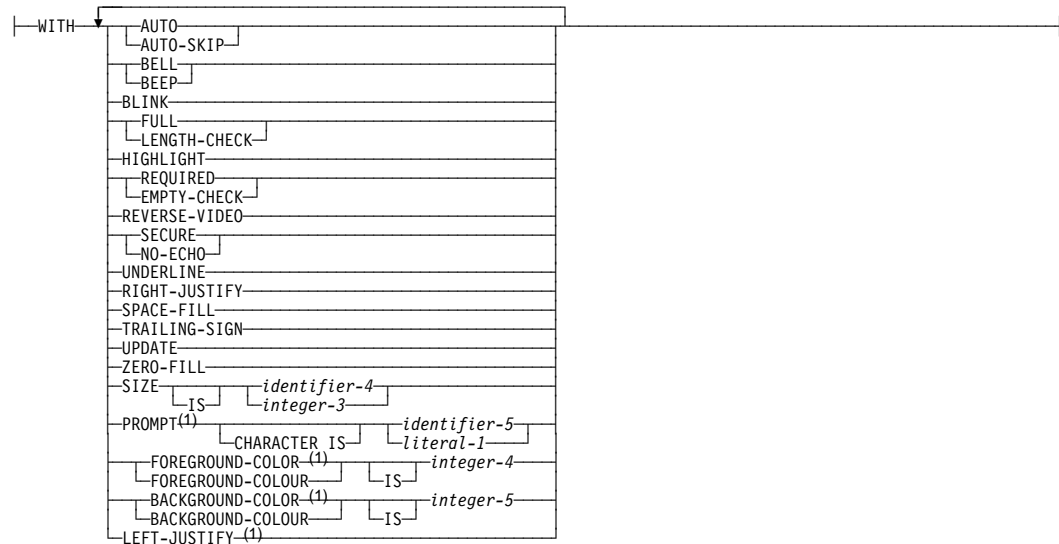
ACCEPT Statement - Format 7 - Workstation I/O



line-column-phrase:



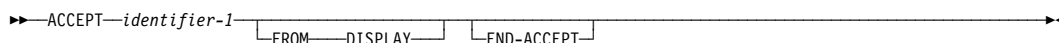
with-phrase:



Note:

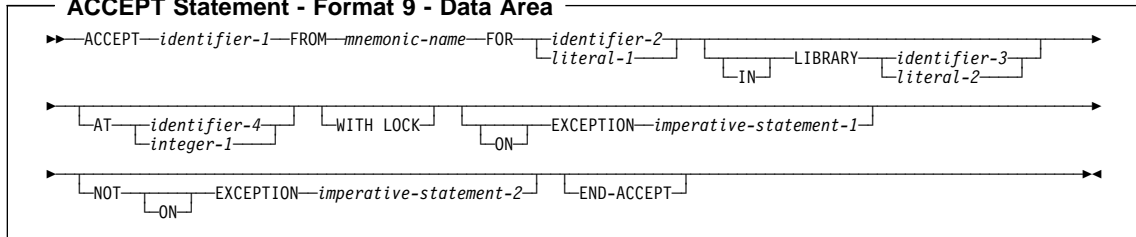
1 Syntax-checked only.

ACCEPT Statement - Format 8 - Session I/O



Procedure Division Statements

ACCEPT Statement - Format 9 - Data Area

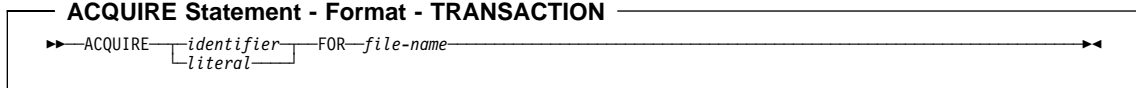


End of IBM Extension

IBM Extension

ACQUIRE Statement

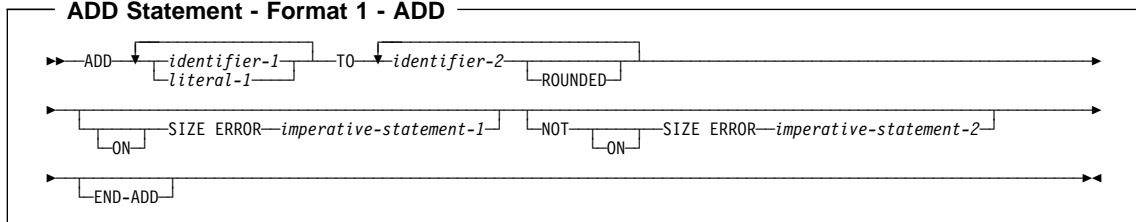
ACQUIRE Statement - Format - TRANSACTION



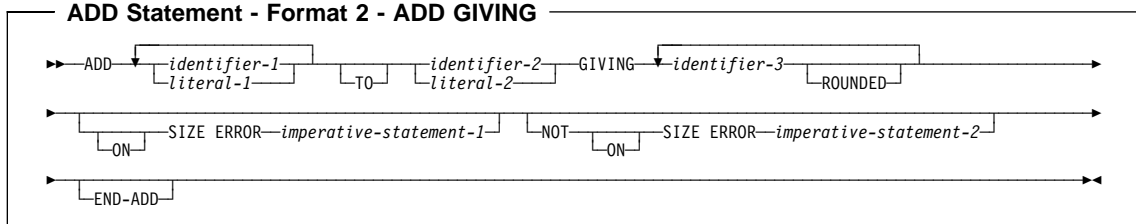
End of IBM Extension

ADD Statement

ADD Statement - Format 1 - ADD

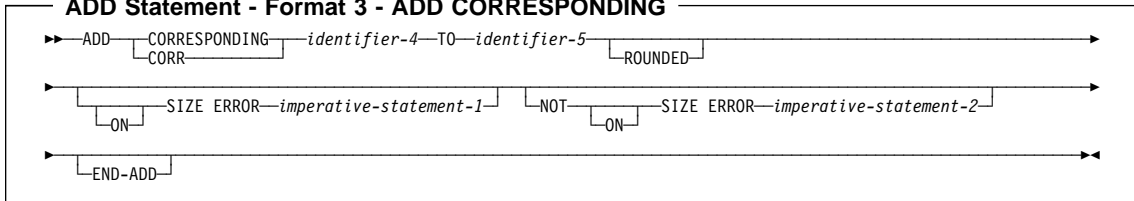


ADD Statement - Format 2 - ADD GIVING



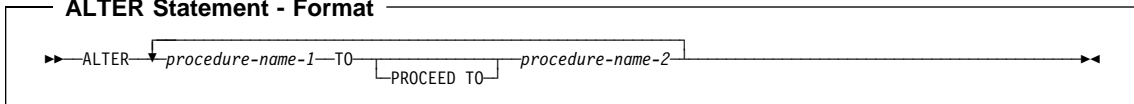
Procedure Division Statements

ADD Statement - Format 3 - ADD CORRESPONDING



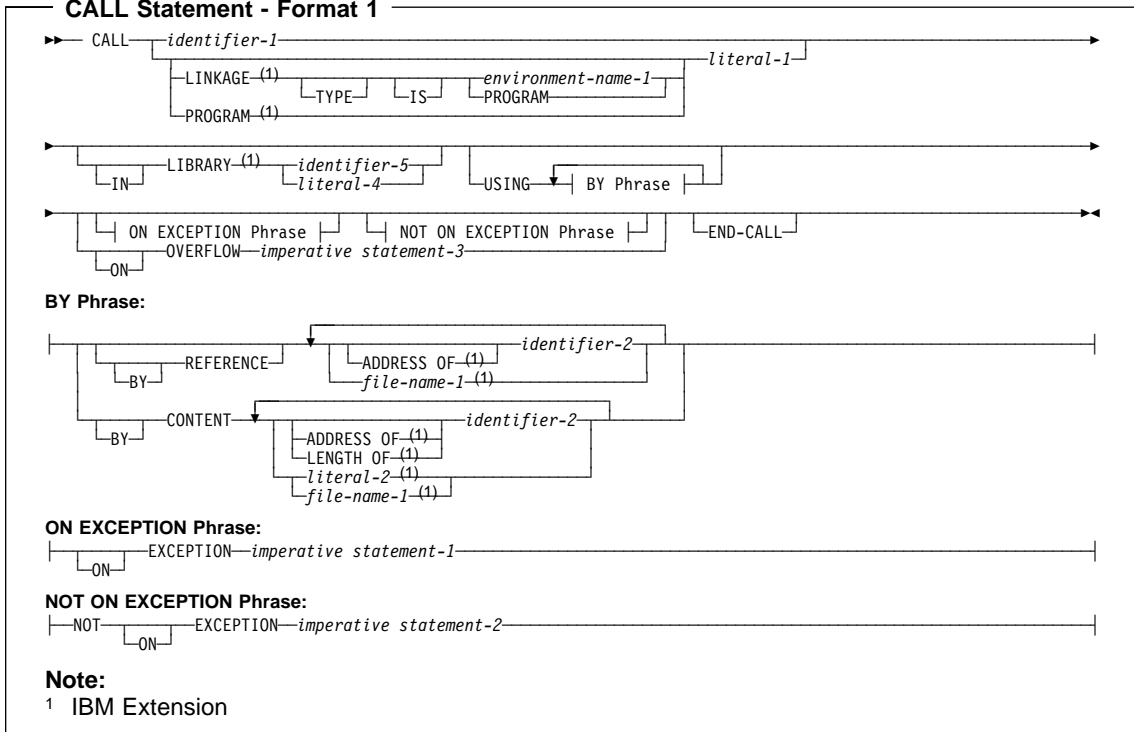
ALTER Statement

ALTER Statement - Format



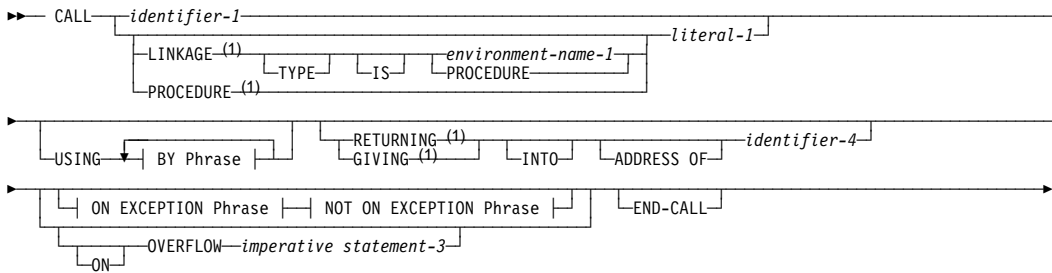
CALL Statement

CALL Statement - Format 1

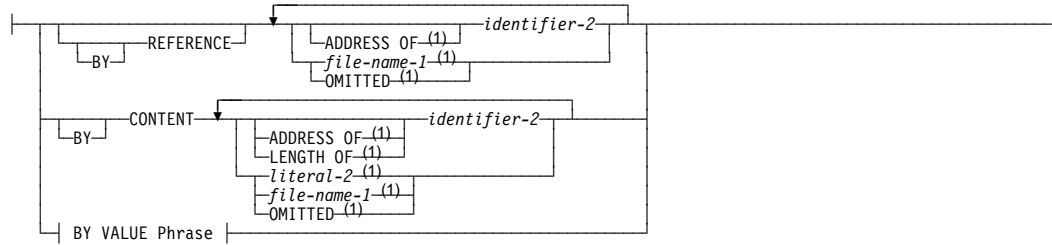


Procedure Division Statements

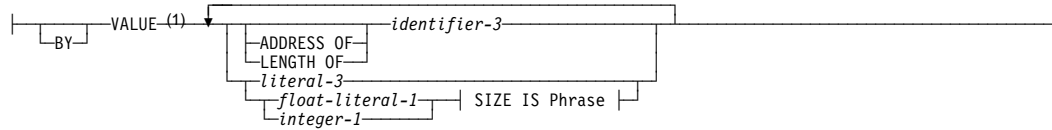
CALL Statement - Format 2



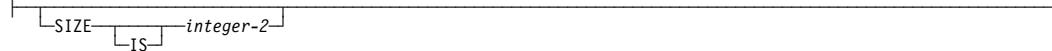
BY Phrase:



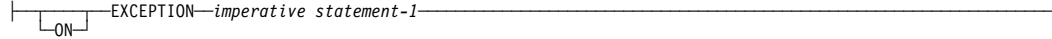
BY VALUE Phrase:



SIZE IS Phrase:



ON EXCEPTION Phrase:



NOT ON EXCEPTION Phrase:



Note:

¹ IBM Extension

Procedure Division Statements

IBM Extension

CALL GDDM Statement - Format

CALL "GDDM" USING (1) routine-name data-name-1

Note:

1 IBM Extension

End of IBM Extension

CANCEL Statement

CANCEL Statement - Format

CANCEL identifier-1 literal-1 In Library Phrase

Linkage Phrase

Linkage Phrase:

LINKAGE (1) environment-name-1
 TYPE IS PROGRAM PROCEDURE
 PROGRAM (1)
 PROCEDURE (1)

In Library Phrase:

IN LIBRARY (1) identifier-2
 literal-2

Note:

1 IBM Extension

CLOSE Statement

CLOSE Statement - Format 1

CLOSE file-name-1 WITH LOCK

CLOSE Statement - Format 2 - Tape Files

CLOSE file-name-1 REEL UNIT FOR REMOVAL
 WITH NO REWIND LOCK

Procedure Division Statements

IBM Extension

COMMIT Statement

COMMIT Statement - Format

COMMIT

End of IBM Extension

COMPUTE Statement

COMPUTE Statement - Format

COMPUTE *identifier-1* [ROUNDED] [EQUAL] *arithmetic-expression*
[ON] SIZE ERROR *imperative-statement-1* [NOT] [ON] SIZE ERROR *imperative-statement-2*
END-COMPUTE

CONTINUE Statement

CONTINUE Statement - Format

CONTINUE

DELETE Statement

DELETE Statement - Format

DELETE *file-name* [RECORD] [FORMAT⁽¹⁾] [IS] *identifier-1* [literal-1]
[NULL-KEY-MAP⁽¹⁾] [IS] *identifier-2* [INVALID] [KEY] *imperative-statement-1*
[NOT INVALID] [KEY] *imperative-statement-2* [END-DELETE]

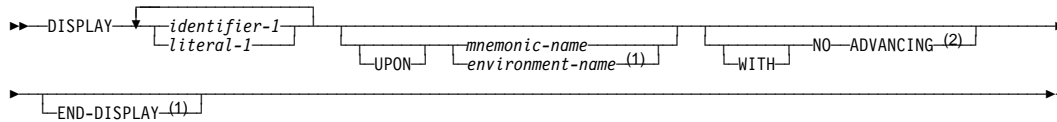
Note:

¹ IBM Extension

Procedure Division Statements

DISPLAY Statement

DISPLAY Statement - Format 1 - Data Transfer

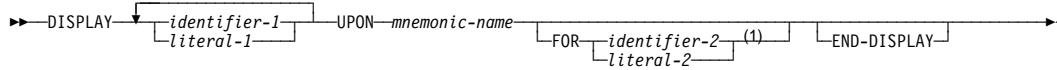


Notes:

- 1 IBM Extension
- 2 Syntax-checked only.

IBM Extension

DISPLAY Statement - Format 2 - Local Data Area

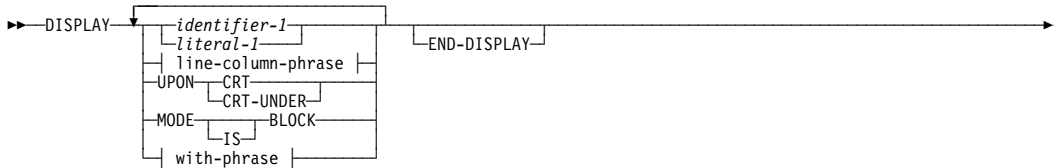


Note:

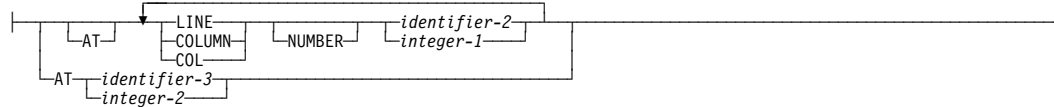
- 1 Syntax-checked only.

Procedure Division Statements

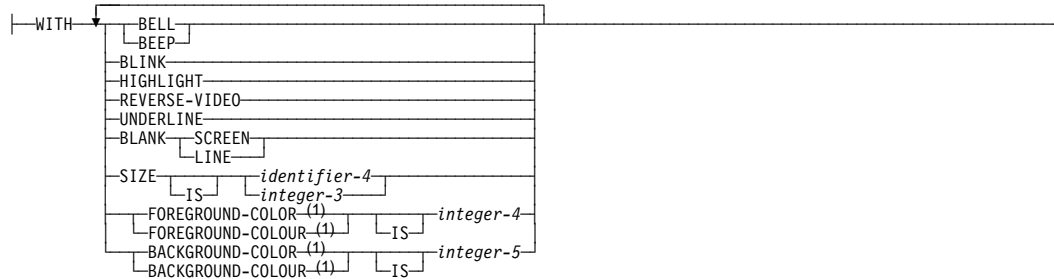
DISPLAY Statement - Format 3 - Workstation I/O



line-column-phrase:



with-phrase:



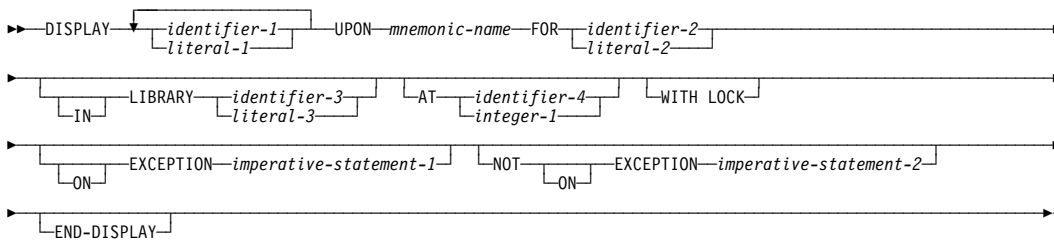
Note:

1 Syntax-checked only.

DISPLAY Statement - Format 4 - Session I/O



DISPLAY Statement - Format 5 - Data Area

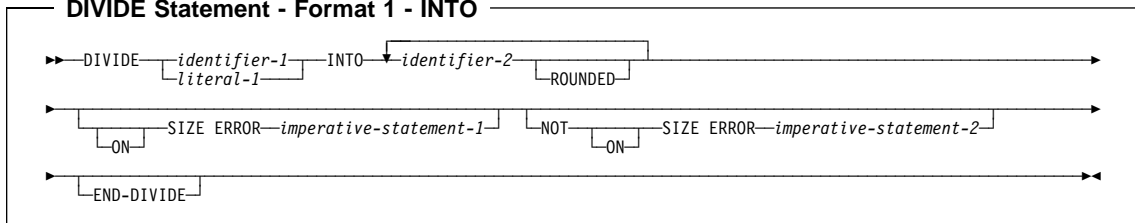


End of IBM Extension

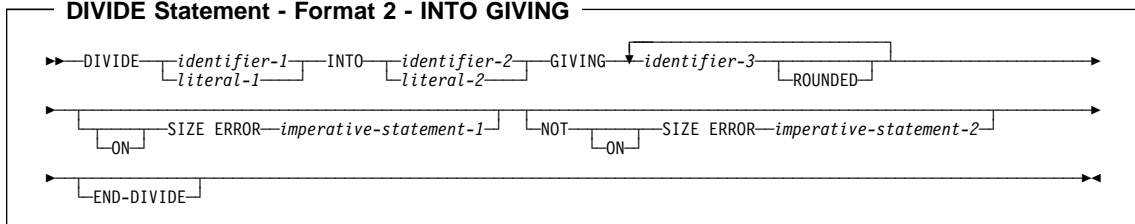
Procedure Division Statements

DIVIDE Statement

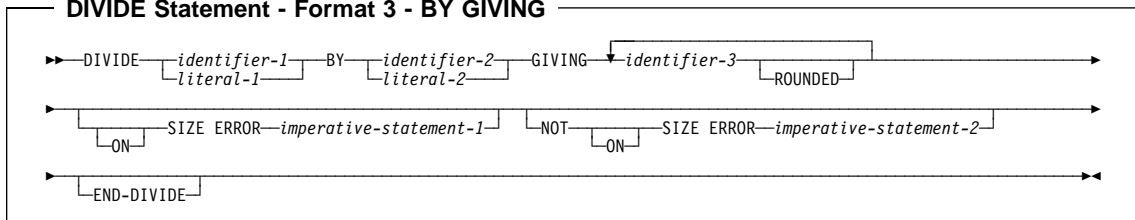
DIVIDE Statement - Format 1 - INTO



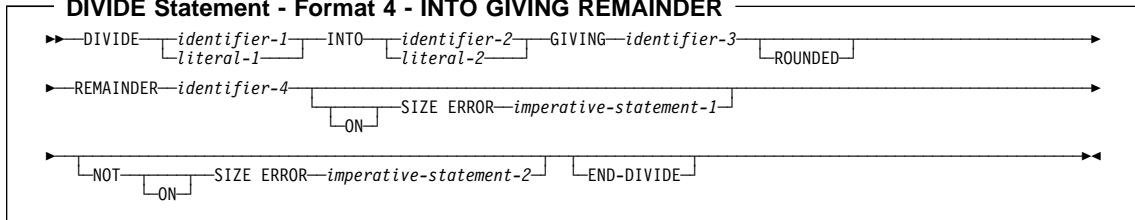
DIVIDE Statement - Format 2 - INTO GIVING



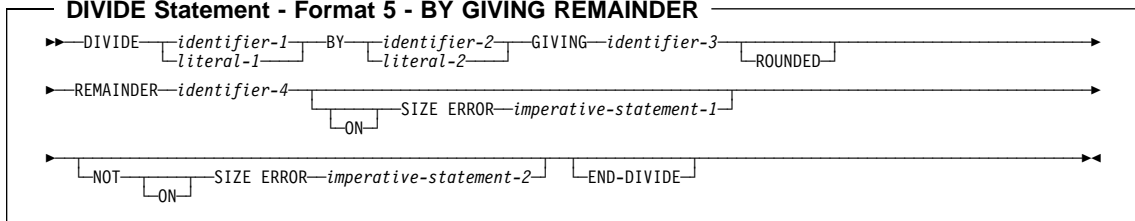
DIVIDE Statement - Format 3 - BY GIVING



DIVIDE Statement - Format 4 - INTO GIVING REMAINDER



DIVIDE Statement - Format 5 - BY GIVING REMAINDER



Procedure Division Statements

IBM Extension

DROP Statement

DROP Statement - Format

→ DROP

identifier
literal

 FROM *file-name* →

End of IBM Extension

ENTER Statement

ENTER Statement - Format

→ ENTER ⁽¹⁾ *language-name*

routine-name

 . →

Note:

¹ Syntax-checked only.

EVALUATE Statement

EVALUATE Statement - Format

→ EVALUATE

identifier-1
literal-1
expression-1
TRUE
FALSE

ALSO

identifier-2
literal-2
expression-2
TRUE
FALSE

 →

→

WHEN phrase

ALSO phrase

imperative-stmt-1

WHEN OTHER

imperative-stmt-2 →

→

END-EVALUATE

 →

WHEN phrase:

→ WHEN

ANY
condition-1
TRUE
FALSE

NOT

identifier-3
literal-3
arith-expr-1

THROUGH
THRU

identifier-4
literal-4
arith-expr-2

 →

ALSO phrase:

→ ALSO

ANY
condition-2
TRUE
FALSE

NOT

identifier-5
literal-5
arith-expr-3

THROUGH
THRU

identifier-6
literal-6
arith-expr-4

 →

Procedure Division Statements

EXIT Statement

EXIT Statement - Format

▶ EXIT ▶▶

EXIT PROGRAM Statement

EXIT PROGRAM Statement

▶ EXIT PROGRAM [AND CONTINUE RUN UNIT-(1)] ▶▶

Note:

¹ IBM Extension

IBM Extension

GOBACK Statement

GOBACK Statement - Format

▶ GOBACK ▶▶

End of IBM Extension

GO TO Statement

GO TO Statement - Format 1 - Unconditional

▶ GO [TO] *procedure-name* ▶▶

GO TO Statement - Format 2 - Conditional

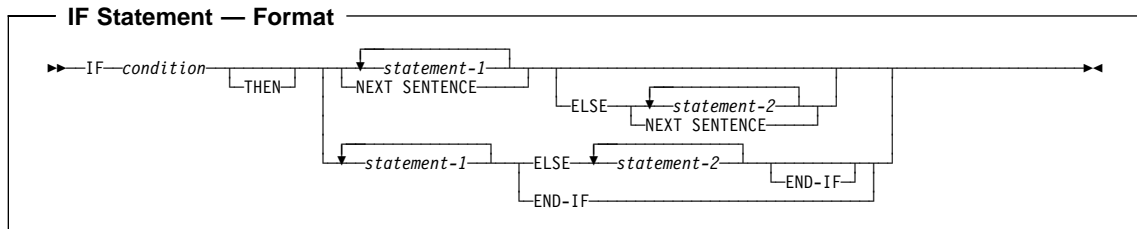
▶ GO [TO] *procedure-name-1* DEPENDING [ON] *identifier* ▶▶

GO TO Statement - Format 3 - Altered

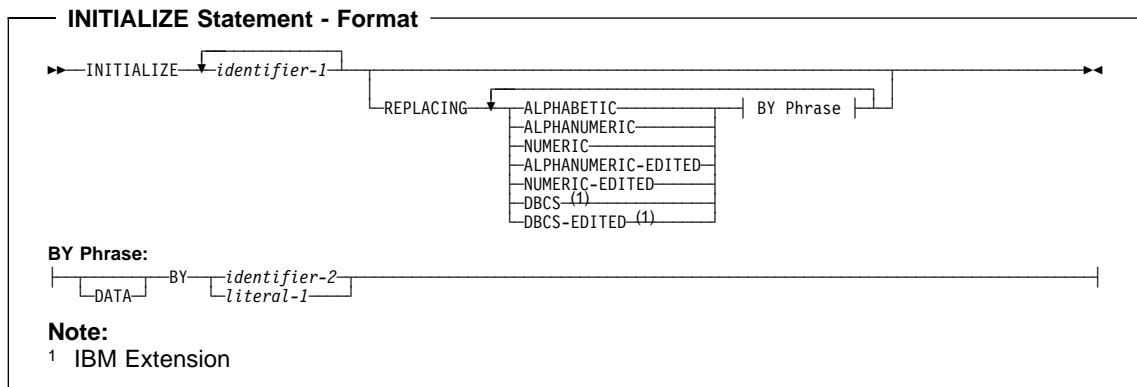
▶ GO [TO] . ▶▶

Procedure Division Statements

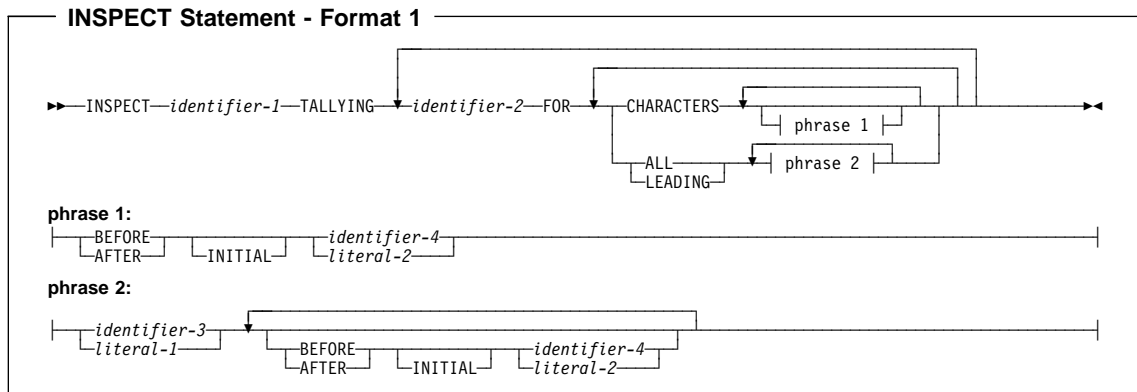
IF Statement



INITIALIZE Statement

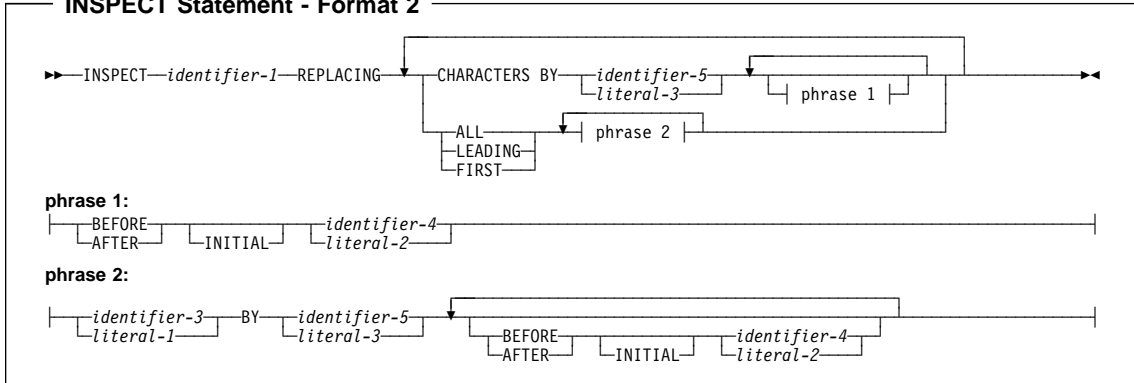


INSPECT Statement

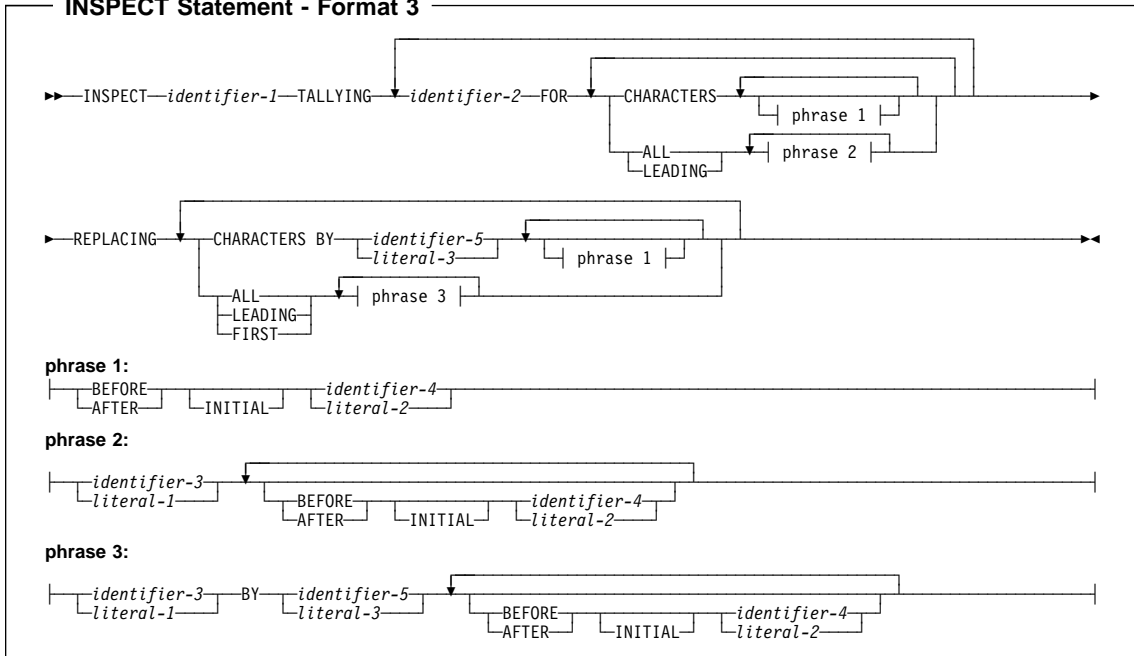


Procedure Division Statements

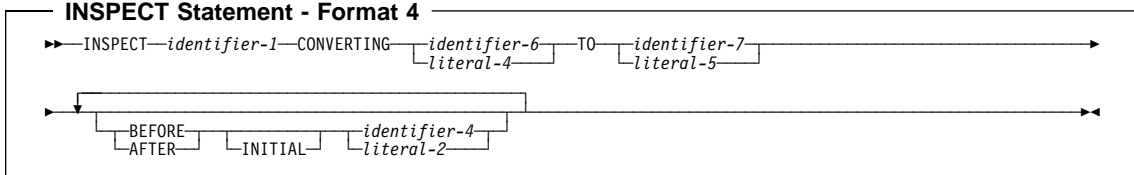
INSPECT Statement - Format 2



INSPECT Statement - Format 3

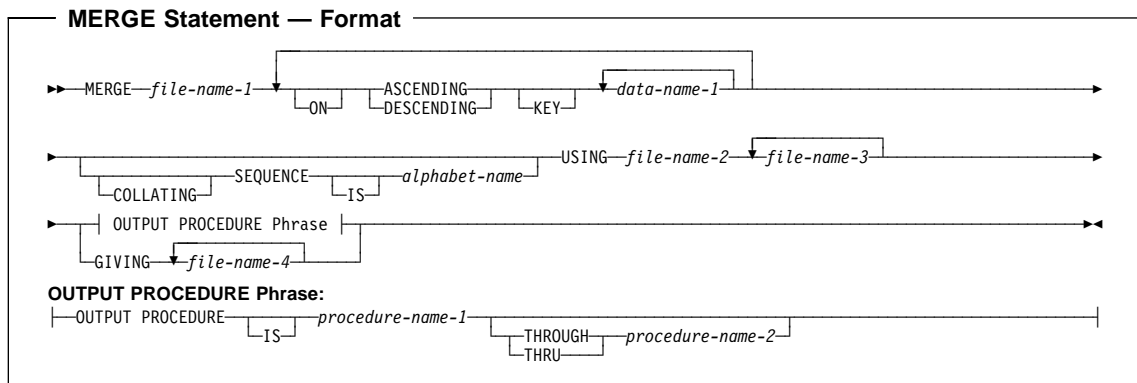


INSPECT Statement - Format 4

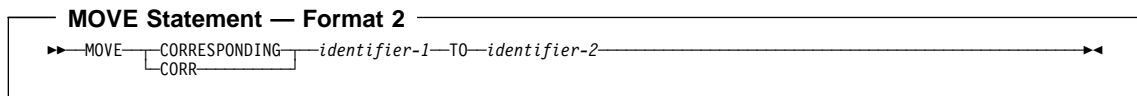
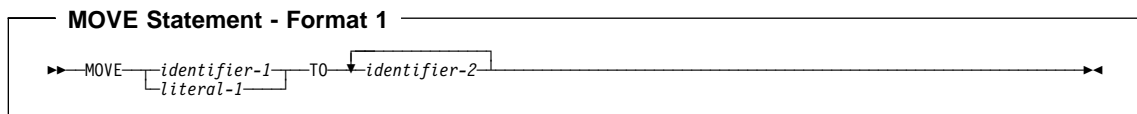


Procedure Division Statements

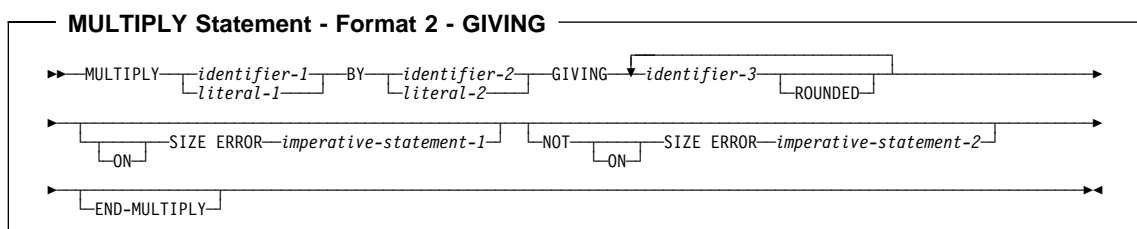
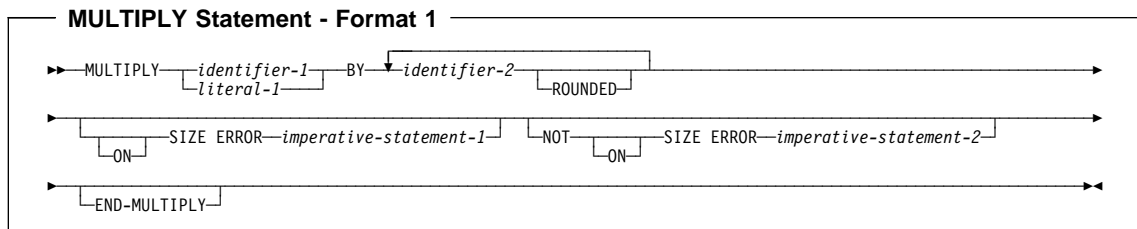
MERGE Statement



MOVE Statement

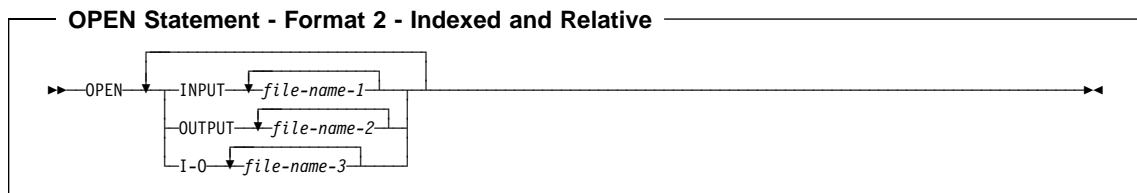
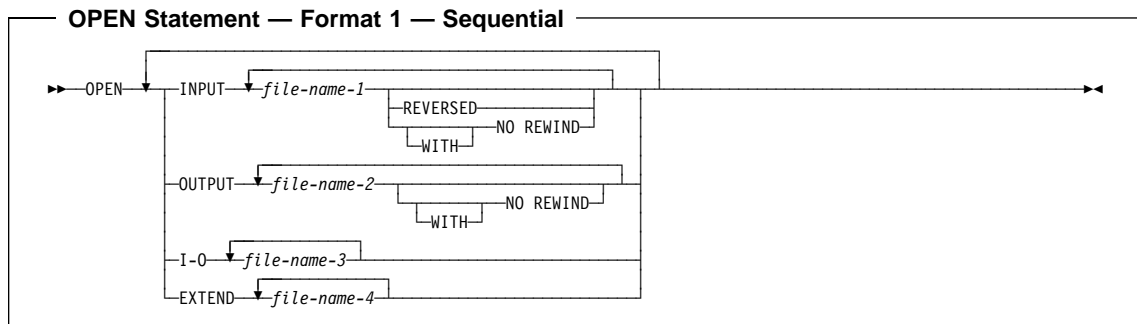


MULTIPLY Statement

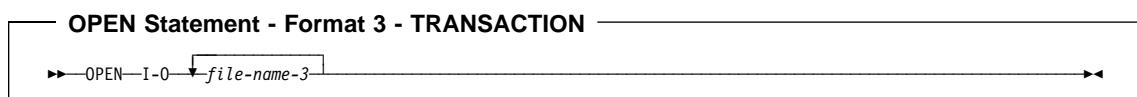


Procedure Division Statements

OPEN Statement

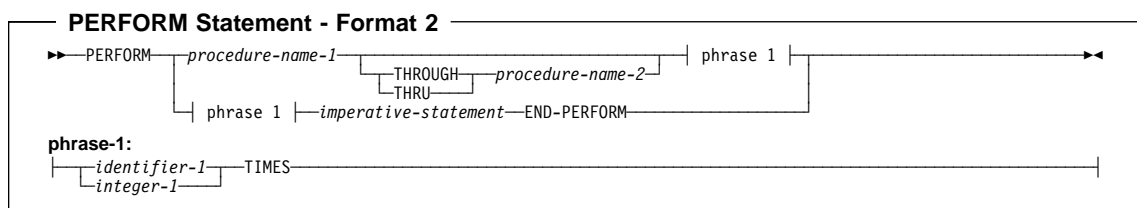
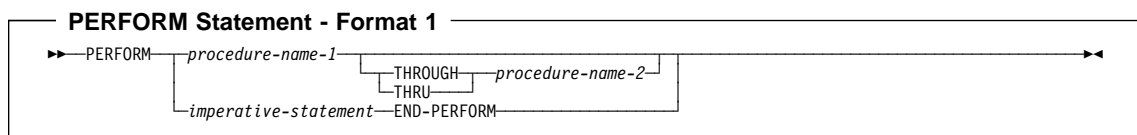


IBM Extension

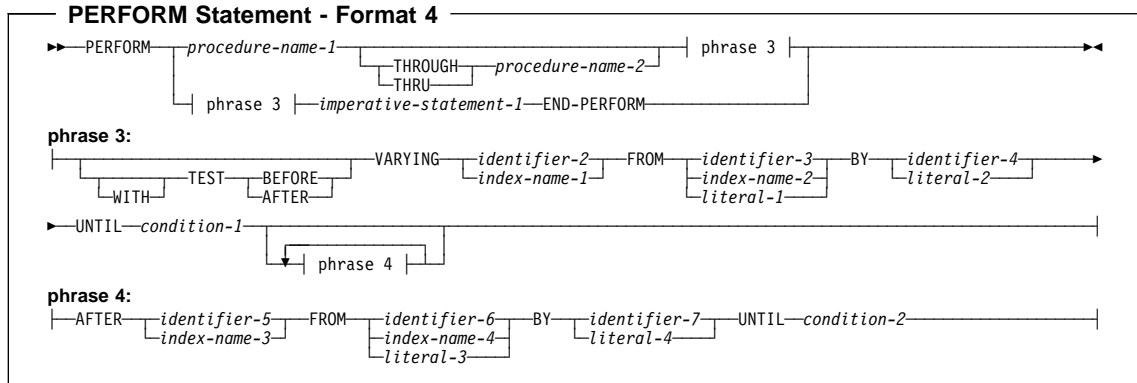
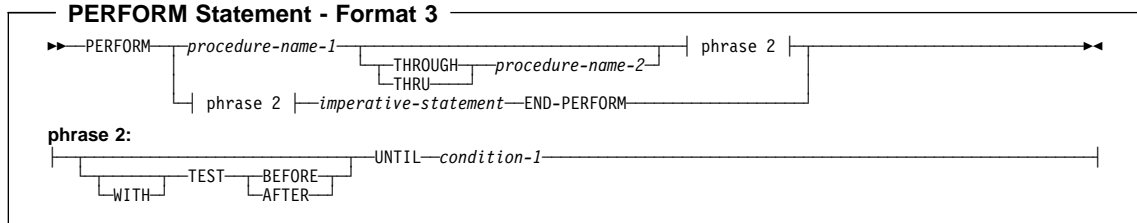


End of IBM Extension

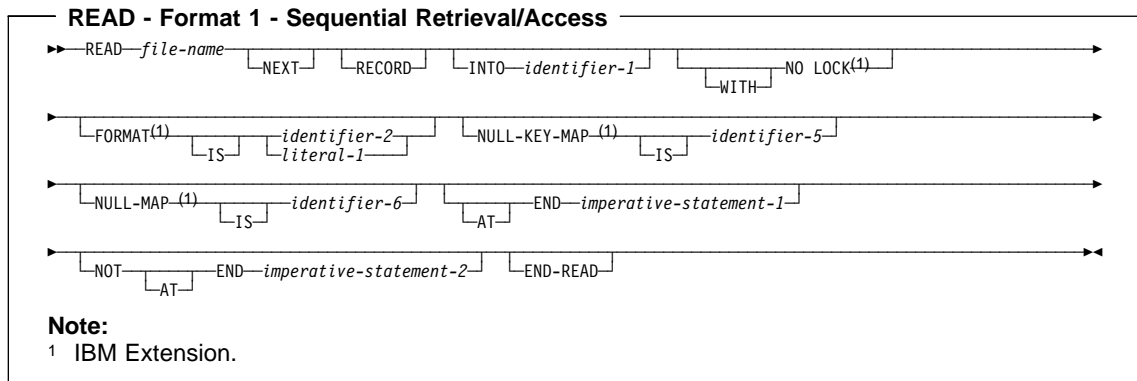
PERFORM Statement



Procedure Division Statements



READ Statement



Procedure Division Statements

READ - Format 2 - Sequential Ret./Dynamic Access

```

READ file-name
  NEXT
  FIRST (1)
  LAST (1)
  PRIOR (1)
  RECORD INTO identifier-1
  WITH NO LOCK(1)
  FORMAT(1) IS identifier-2
  NULL-KEY-MAP(1) IS identifier-5
  NULL-MAP(1) IS identifier-6
  AT END imperative-statement-1
  NOT AT END imperative-statement-2
  END-READ
  
```

Note:

1 IBM Extension

READ Statement - Format 3 - Random Retrieval

```

READ file-name
  RECORD INTO identifier-1
  WITH NO LOCK(1)
  KEY(2) IS data-name-1
  FORMAT(1) IS identifier-2
  NULL-KEY-MAP(1) IS identifier-5
  NULL-MAP(1) IS identifier-6
  INVALID KEY imperative-statement-1
  NOT INVALID KEY imperative-statement-2
  END-READ
  
```

Notes:

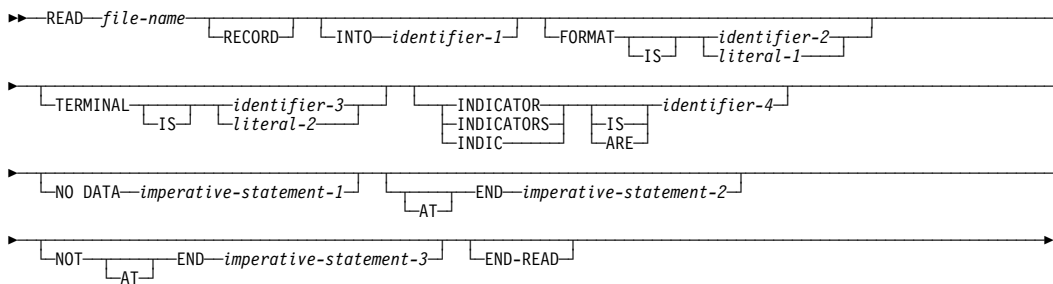
1 IBM Extension

2 Syntax-checked only.

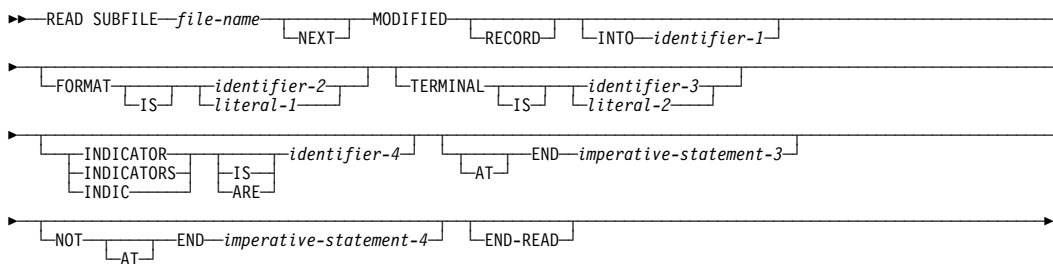
Procedure Division Statements

IBM Extension

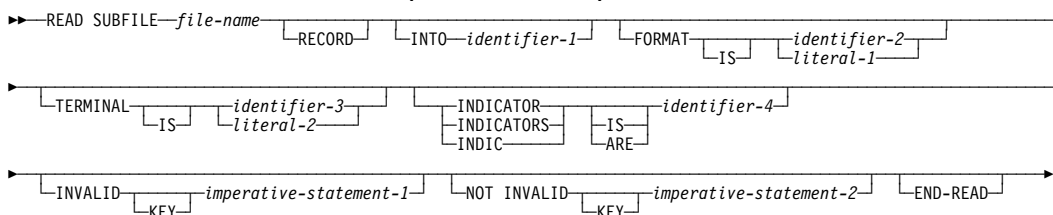
READ Statement - Format 4 - Transaction (Nonsubfile)



READ - Format 5a - Transaction (Subfile Sequential)



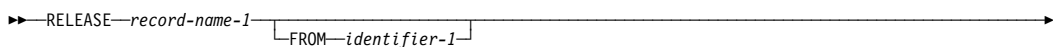
READ - Format 5b - Transaction (Subfile Random)



End of IBM Extension

RELEASE Statement

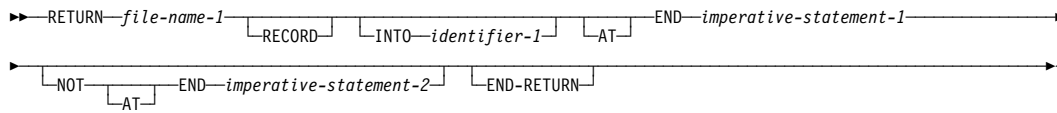
RELEASE Statement - Format



Procedure Division Statements

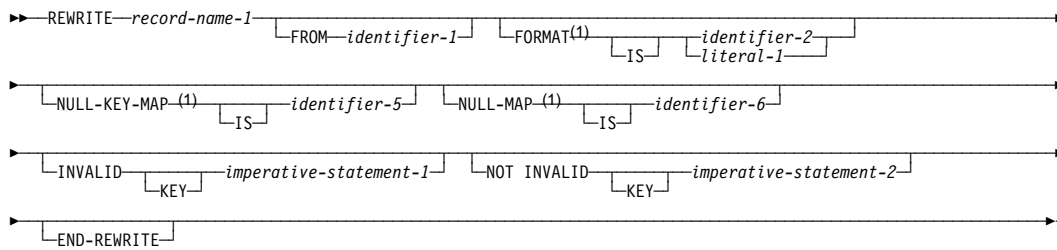
RETURN Statement

RETURN Statement - Format



REWRITE Statement

REWRITE Statement - Format 1

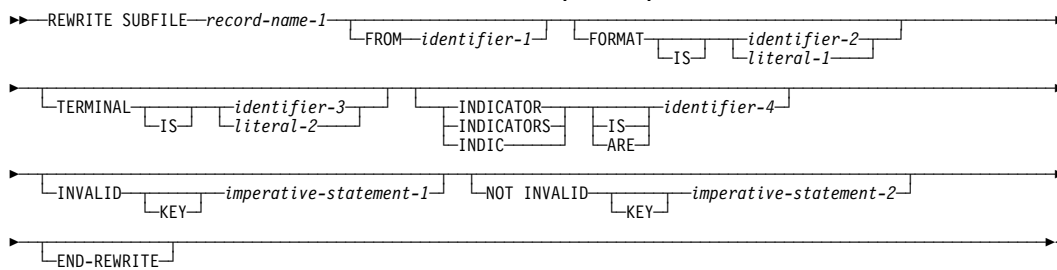


Note:

¹ IBM Extension

IBM Extension

REWRITE Statement - Format 2 - Transaction (Subfile)



End of IBM Extension

Procedure Division Statements

IBM Extension

ROLLBACK Statement

ROLLBACK Statement - Format

▶ ROLLBACK

End of IBM Extension

SEARCH Statement

SEARCH Statement - Format 1 - Serial Search

▶ SEARCH *identifier-1* VARYING *identifier-2* AT *index-name-1* END *imperative-statement-1*

▶ WHEN *condition-1* *imperative-statement-2* NEXT SENTENCE

▶ WHEN *condition-1* *imperative-statement-2* END-SEARCH

SEARCH Statement - Format 2 - Binary Search

▶ SEARCH ALL *identifier-1* AT *index-name-1* END *imperative-statement-1* WHEN *condition-name-1* equal phrase 1

▶ AND *condition-name-2* equal phrase 2 *imperative-statement-2* NEXT SENTENCE END-SEARCH

equal phrase 1:

data-name-1 IS EQUAL TO *identifier-3* *literal-1* *arithmetic-expression-1*

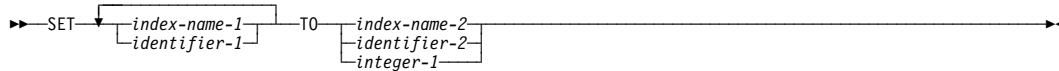
equal phrase 2:

data-name-2 IS EQUAL TO *identifier-4* *literal-2* *arithmetic-expression-2*

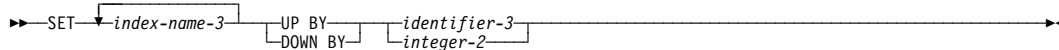
Procedure Division Statements

SET Statement

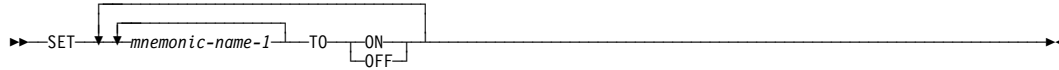
SET Statement - Format 1



SET Statement - Format 2



SET Statement - Format 3

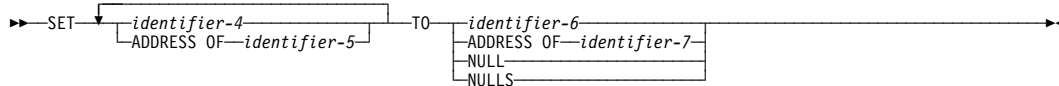


SET Statement - Format 4

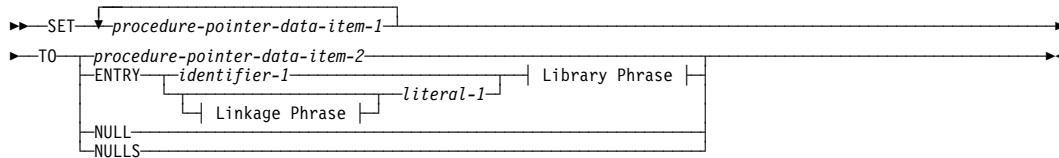


IBM Extension

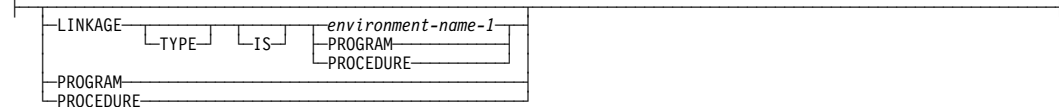
SET Statement - Format 5



SET Statement - Format 6



Linkage Phrase:

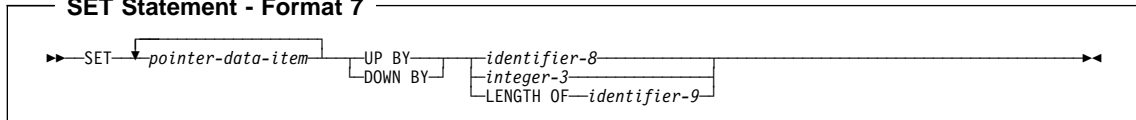


Library Phrase:

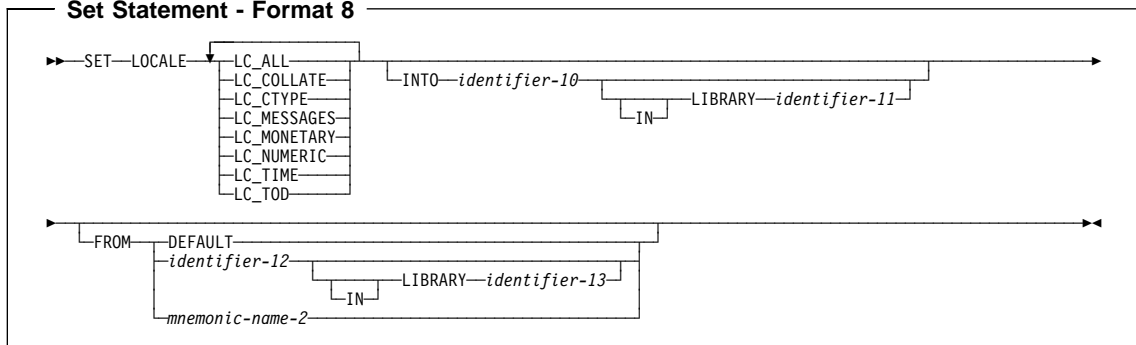


Procedure Division Statements

SET Statement - Format 7



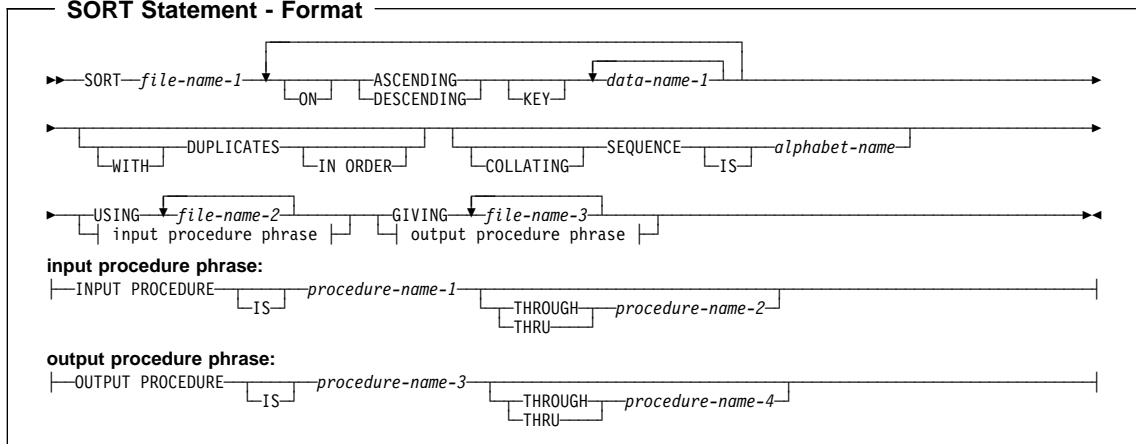
Set Statement - Format 8



End of IBM Extension

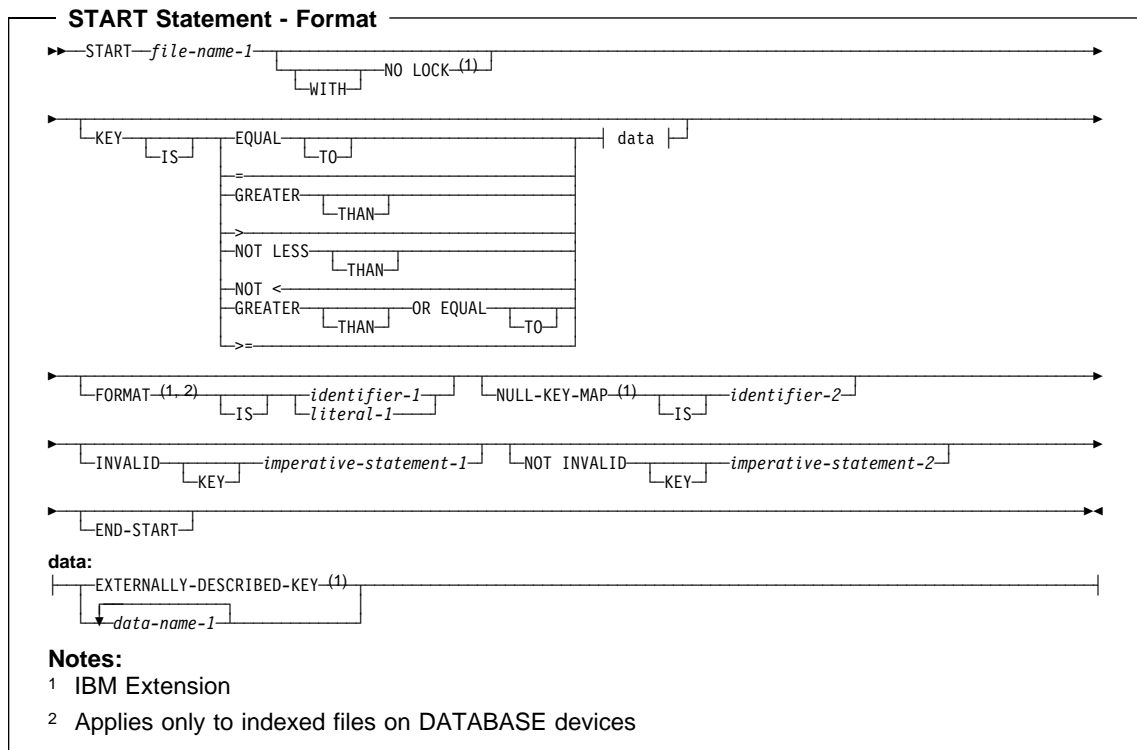
SORT Statement

SORT Statement - Format

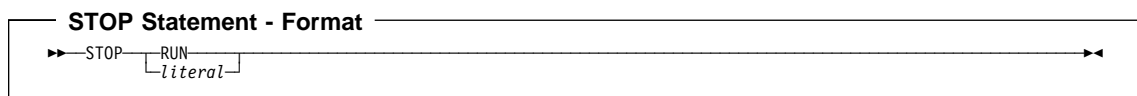


Procedure Division Statements

START Statement

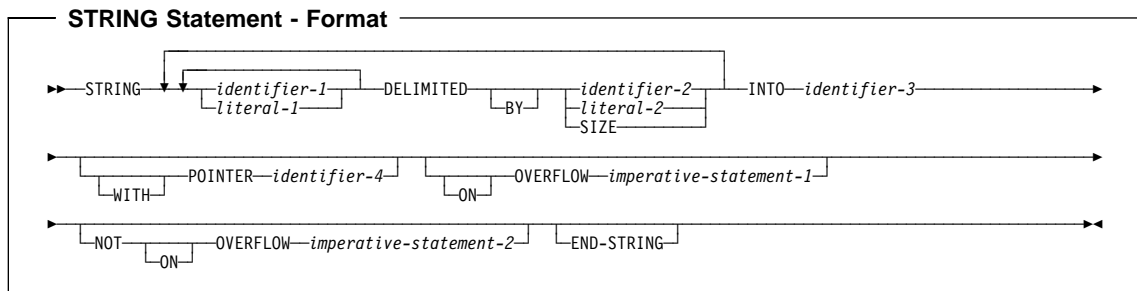


STOP Statement

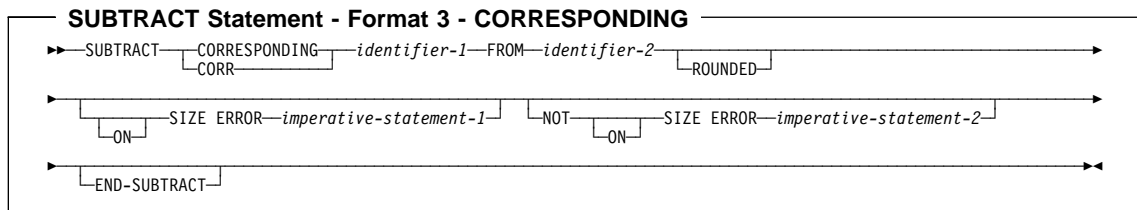
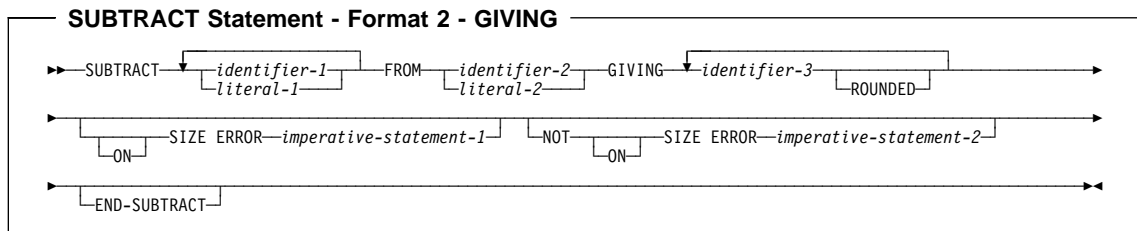
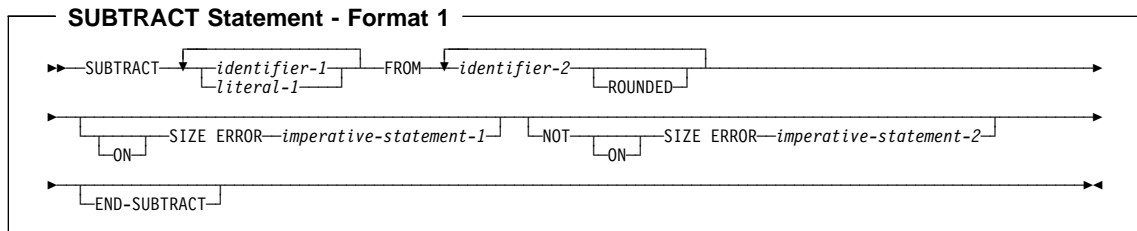


Procedure Division Statements

STRING Statement

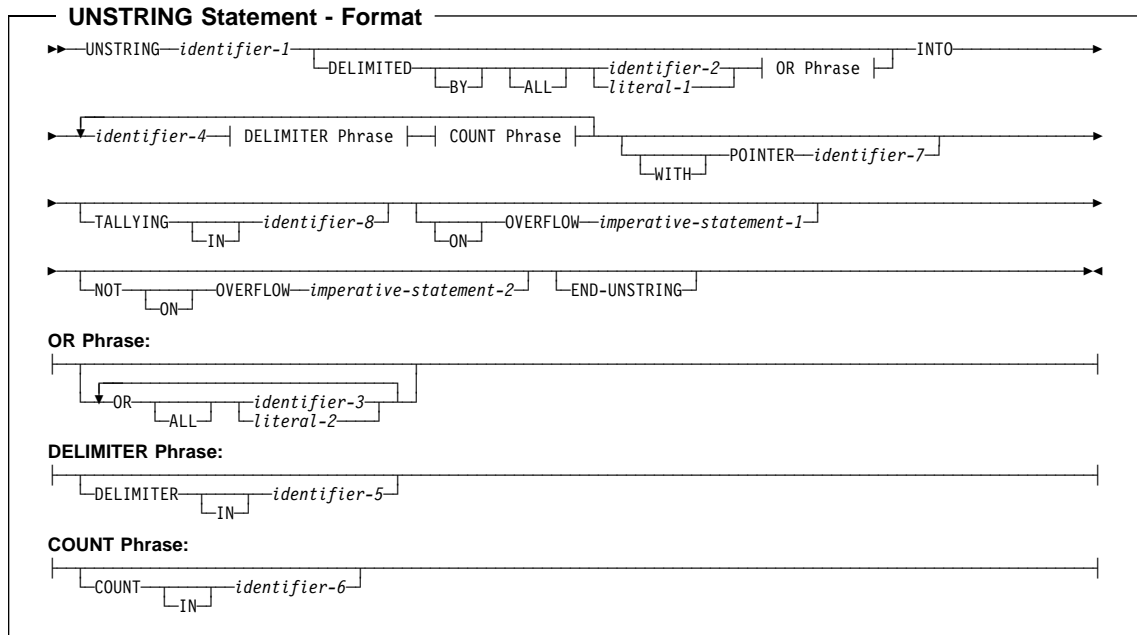


SUBTRACT Statement



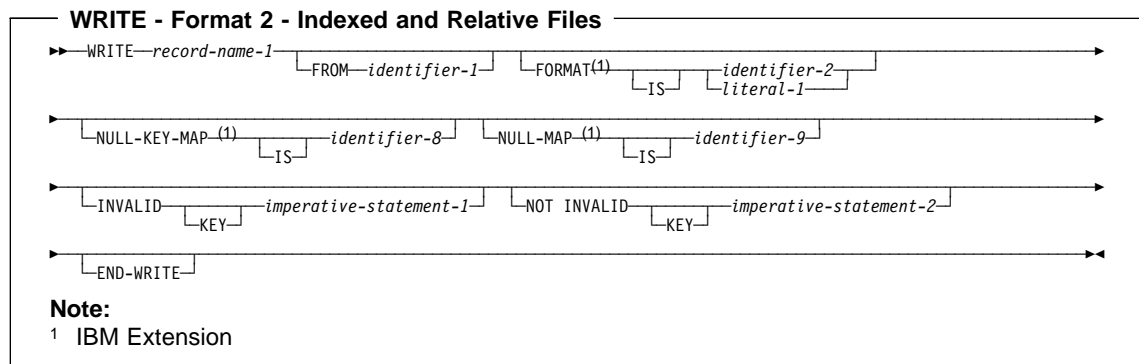
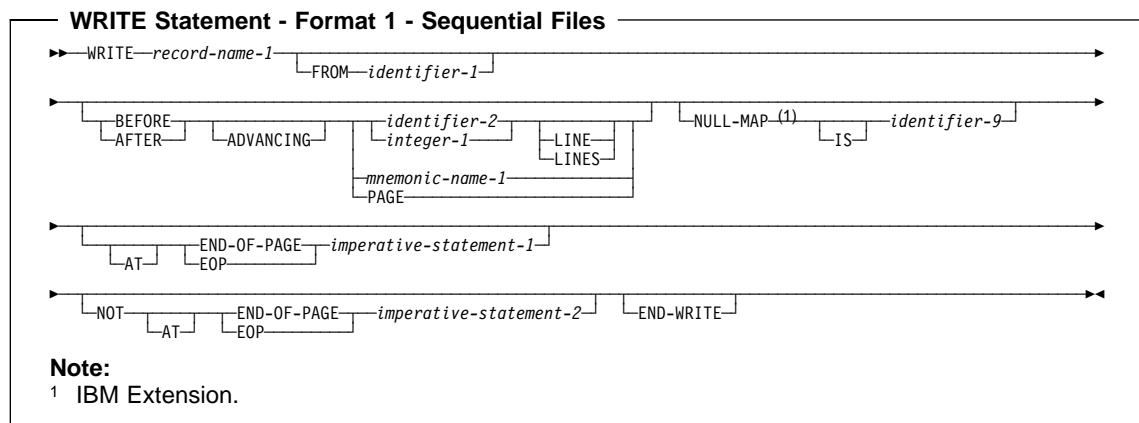
Procedure Division Statements

UNSTRING Statement



Procedure Division Statements

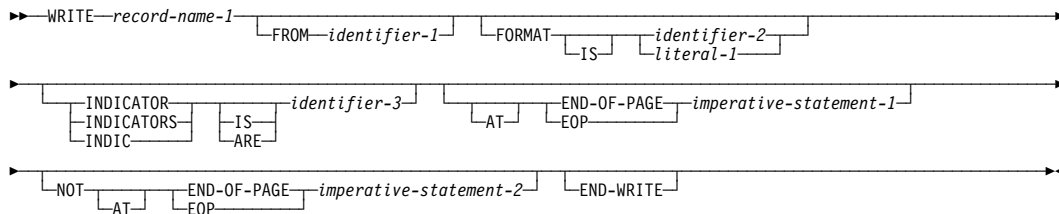
WRITE Statement



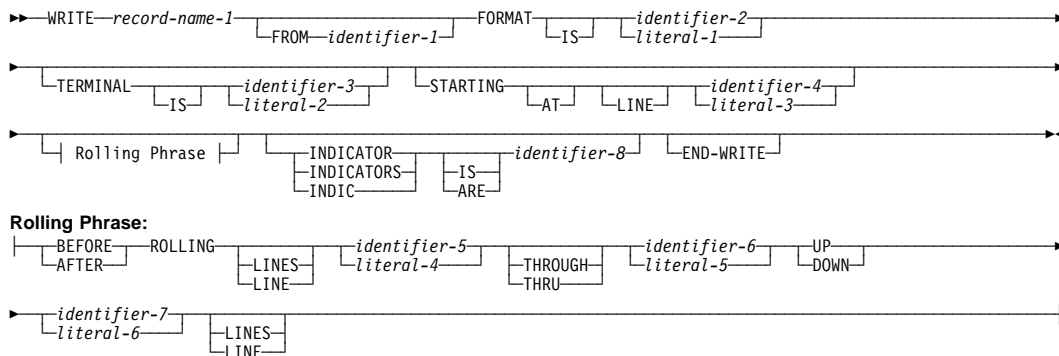
Procedure Division Statements

IBM Extension

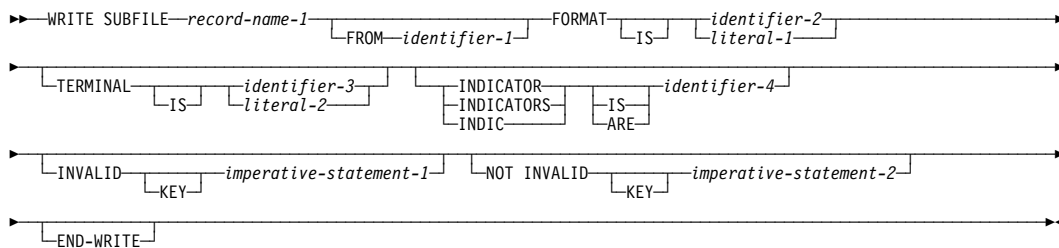
WRITE Statement - Format 3 - FORMATFILE



WRITE - Format 4 - TRANSACTION (Nonsubfile)



WRITE Statement - Format 5 - TRANSACTION (Subfile)



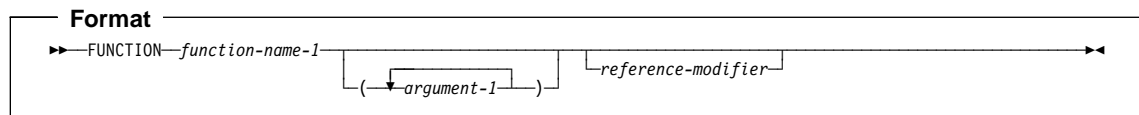
End of IBM Extension

Intrinsic Functions

Intrinsic Functions

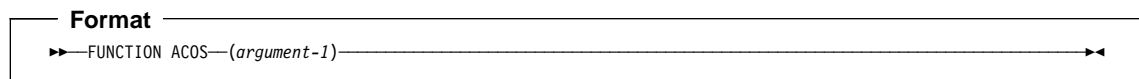
This is the general syntax for intrinsic functions.

Function-Identifier



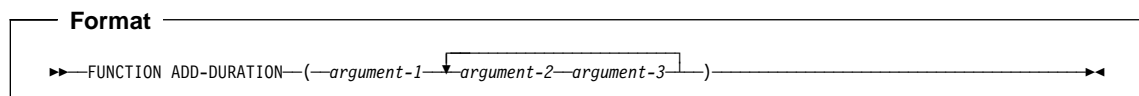
These following intrinsic functions are presented in alphabetical order.

ACOS Function



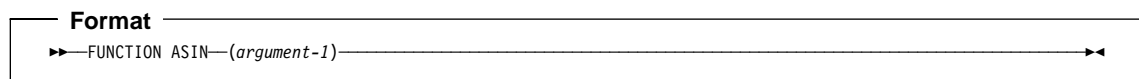
IBM Extension

ADD-DURATION Function

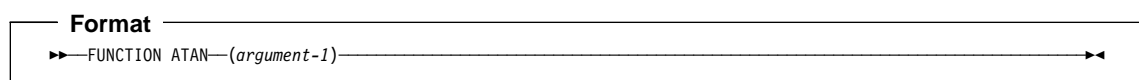


End of IBM Extension

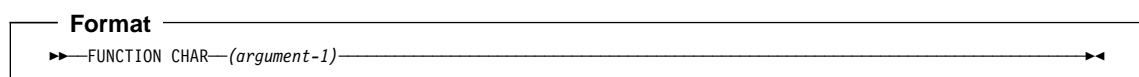
ASIN Function



ATAN Function



CHAR Function



Intrinsic Functions

IBM Extension

CONVERT-DATE-TIME Function

Format
FUNCTION CONVERT-DATE-TIME (argument-1 argument-2 argument-3 argument-4)

End of IBM Extension

COS Function

Format
FUNCTION COS (argument-1)

CURRENT-DATE Function

Format
FUNCTION CURRENT-DATE

DATE-OF-INTEGER Function

Format
FUNCTION DATE-OF-INTEGER (argument-1)

DAY-OF-INTEGER Function

Format
FUNCTION DAY-OF-INTEGER (argument-1)

IBM Extension

DATE-TO-YYYYMMDD Function

Format
FUNCTION DATE-TO-YYYYMMDD (argument-1 argument-2)

End of IBM Extension

Intrinsic Functions

IBM Extension

DAY-TO-YYYYDDD Function

Format

►►FUNCTION DAY-TO-YYYYDDD—(*argument-1*—*argument-2*—)

End of IBM Extension

IBM Extension

EXTRACT-DATE-TIME Function

Format

►►FUNCTION EXTRACT-DATE-TIME—(*argument-1*—*argument-2*—)

End of IBM Extension

IBM Extension

FIND-DURATION Function

Format

►►FUNCTION FIND-DURATION—(*argument-1*—*argument-2*—*argument-3*—)

End of IBM Extension

INTEGER-OF-DATE Function

Format

►►FUNCTION INTEGER-OF-DATE—(*argument-1*)

INTEGER-OF-DAY Function

Format

►►FUNCTION INTEGER-OF-DAY—(*argument-1*)

Intrinsic Functions

LENGTH Function

Format

►►FUNCTION LENGTH(*argument-1*)◄◄

IBM Extension

LOCALE-DATE Function

►►FUNCTION LOCALE-DATE(*argument-1* [*mnemonic-name-1*])◄◄

End of IBM Extension

IBM Extension

LOCALE-TIME Function

►►FUNCTION LOCALE-TIME(*argument-1* [*mnemonic-name-1*])◄◄

End of IBM Extension

LOG Function

Format

►►FUNCTION LOG(*argument-1*)◄◄

LOG10 Function

Format

►►FUNCTION LOG10(*argument-1*)◄◄

LOWER-CASE Function

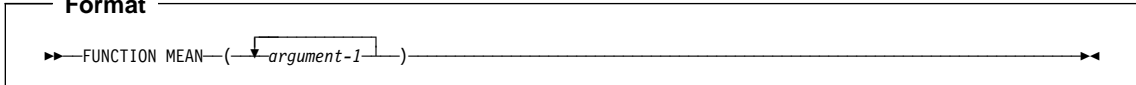
Format

►►FUNCTION LOWER-CASE(*argument-1*)◄◄

MEAN Function

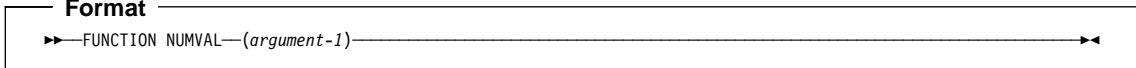
Intrinsic Functions

Format

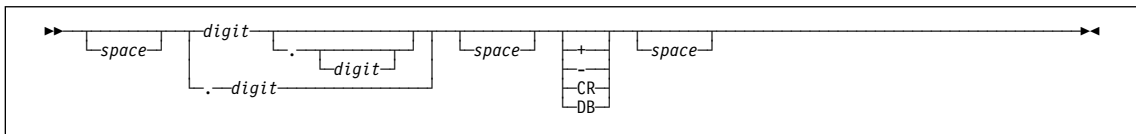
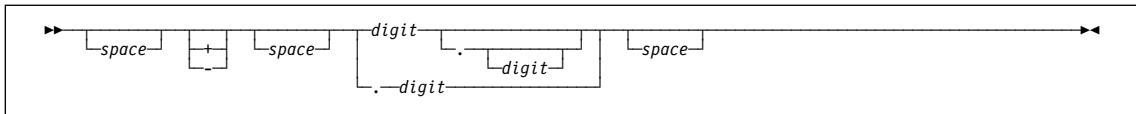


NUMVAL Function

Format

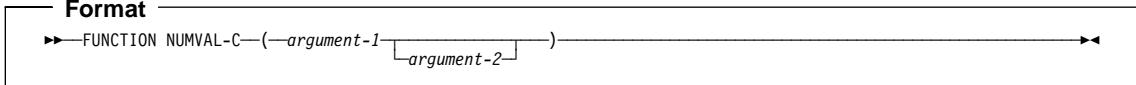


Argument-1 can have one of the following formats:

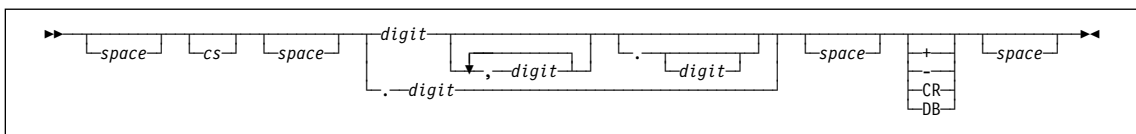
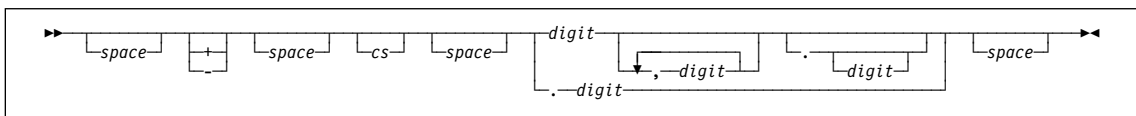


NUMVAL-C Function

Format



The format for argument-1 is one of the following two formats, where cs is the currency sign specified in argument-2:



Intrinsic Functions

ORD Function

Format
▶—FUNCTION ORD—(*argument-1*)—————▶◀

REVERSE Function

Format
▶—FUNCTION REVERSE—(*argument-1*)—————▶◀

SIN Function

Format
▶—FUNCTION SIN—(*argument-1*)—————▶◀

SQRT Function

Format
▶—FUNCTION SQRT—(*argument-1*)—————▶◀

_____ IBM Extension _____

SUBTRACT-DURATION Function

Format
▶—FUNCTION SUBTRACT-DURATION—(*argument-1*—*argument-2*—*argument-3*)—————▶◀

_____ End of IBM Extension _____

TAN Function

Format
▶—FUNCTION TAN—(*argument-1*)—————▶◀

Intrinsic Functions

IBM Extension

TEST-DATE-TIME Function

Format

►►FUNCTION TEST-DATE-TIME—(*argument-1*—*argument-2*—*argument-3*—*argument-4*)—►►

End of IBM Extension

UPPER-CASE Function

Format

►►FUNCTION UPPER-CASE—(*argument-1*)—►►

WHEN-COMPILED Function

Format

►►FUNCTION WHEN-COMPILED—►►

IBM Extension

YEAR-TO-YYYY Function

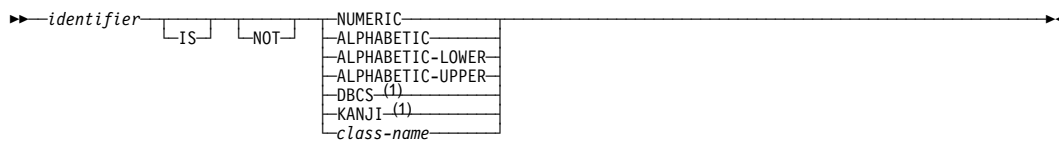
Format

►►FUNCTION YEAR-TO-YYYY—(*argument-1*—*argument-2*)—►►

End of IBM Extension

Conditional Expressions

Class Condition - Format



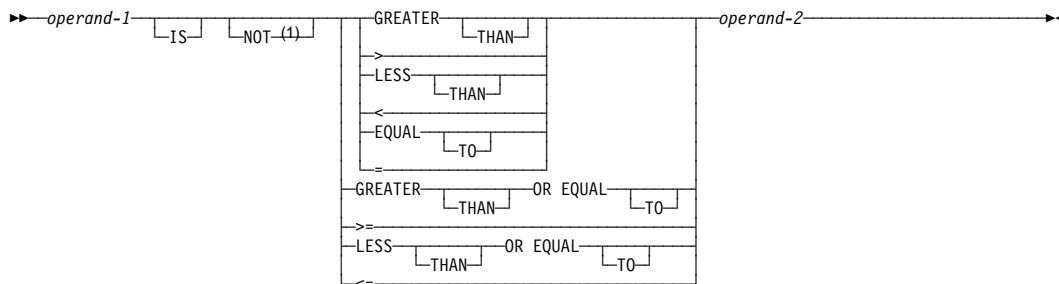
Note:

¹ IBM Extension

Condition-Name Condition - Format



Relation Condition - Format



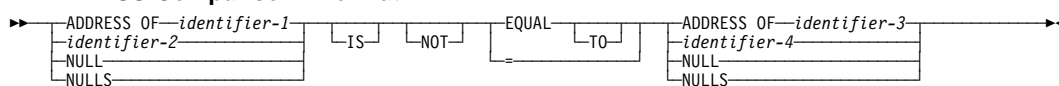
Note:

¹ NOT GREATER THAN OR EQUAL TO, NOT >=, NOT LESS THAN OR EQUAL TO, and NOT <=, are IBM Extensions.

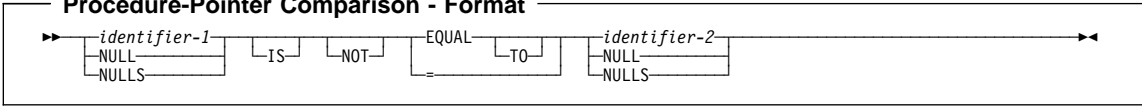
IBM Extension

The following two syntax diagrams apply to pointer data items and procedure-pointer data items, respectively.

ADDRESS Comparison - Format

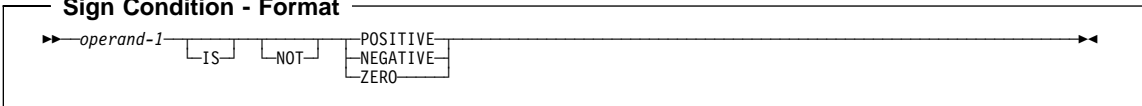


Procedure-Pointer Comparison - Format

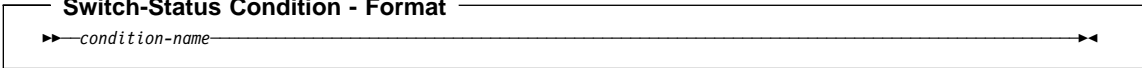


End of IBM Extension

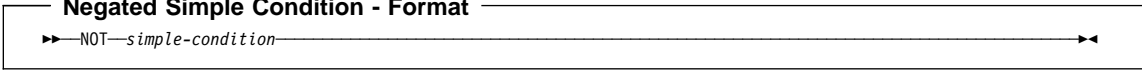
Sign Condition - Format



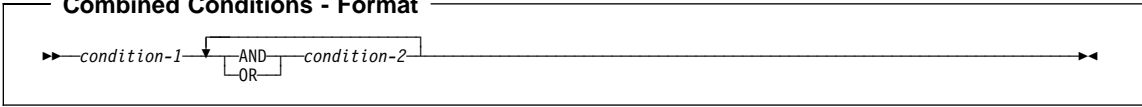
Switch-Status Condition - Format



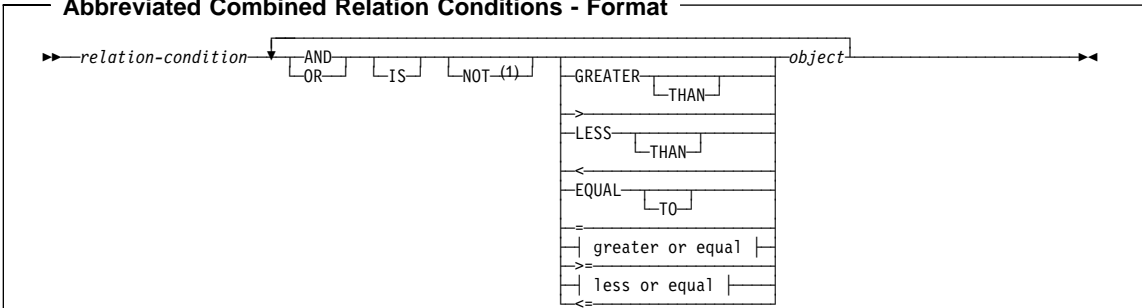
Negated Simple Condition - Format



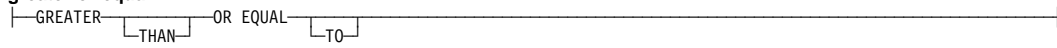
Combined Conditions - Format



Abbreviated Combined Relation Conditions - Format



greater or equal:



less or equal:

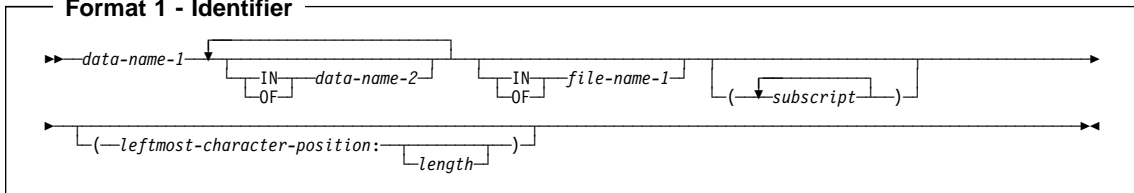


Note:

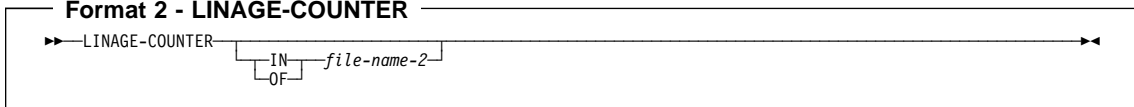
¹ NOT GREATER THAN OR EQUAL TO, NOT >=, NOT LESS THAN OR EQUAL TO, and NOT <=, are IBM Extensions.

Qualifying Data Reference Formats

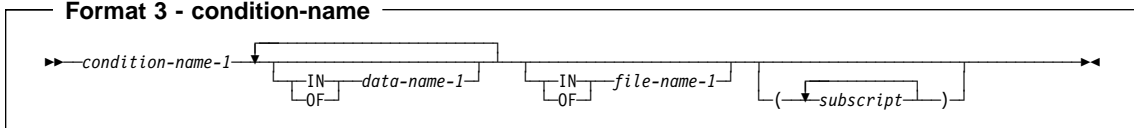
Format 1 - Identifier



Format 2 - LINAGE-COUNTER

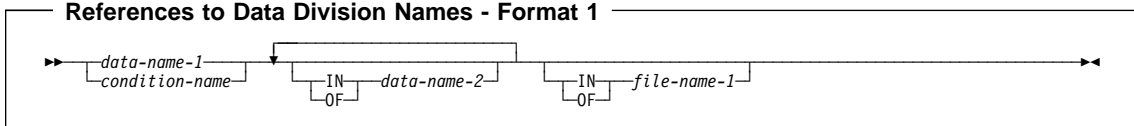


Format 3 - condition-name

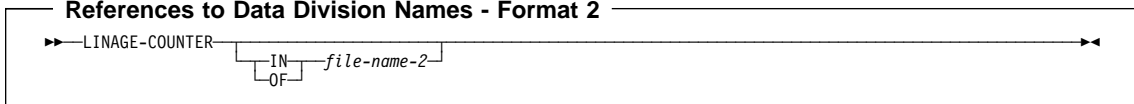


Qualification

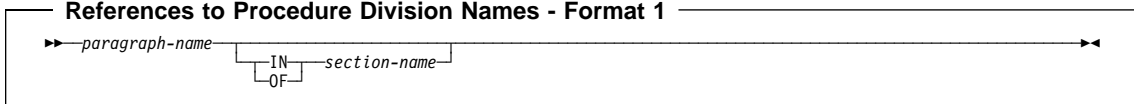
References to Data Division Names - Format 1



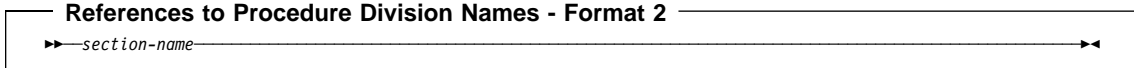
References to Data Division Names - Format 2



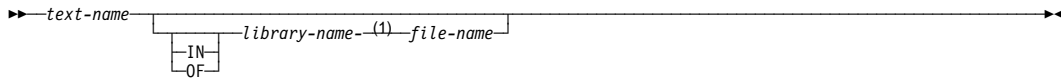
References to Procedure Division Names - Format 1



References to Procedure Division Names - Format 2



References to COPY Libraries - Format 3

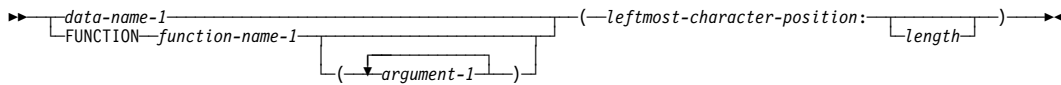


Note:

¹ Required hyphen between library-name-file-name to qualify

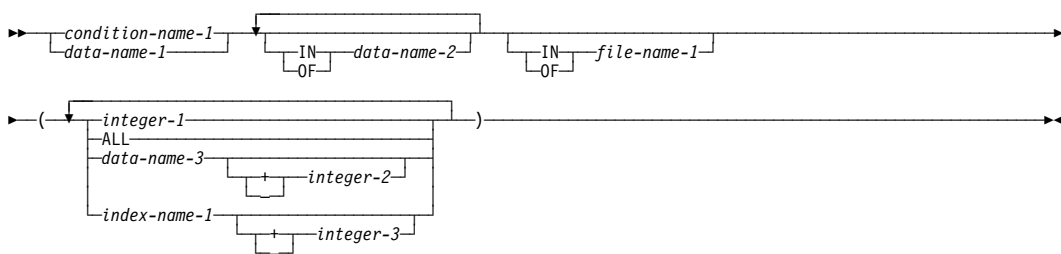
Reference Modification

Format



Subscripting

Subscripting - Format

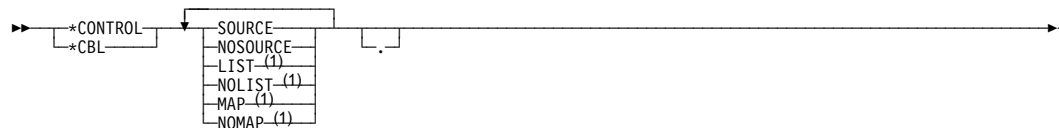


Compiler-Directing Statements

IBM Extension

CONTROL Statement

CONTROL (*CBL) Statement - Format



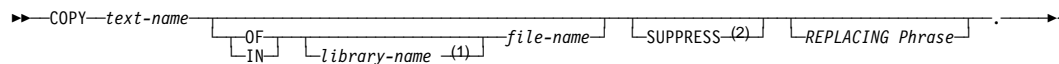
Note:

1 Syntax-checked only.

End of IBM Extension

COPY Statement

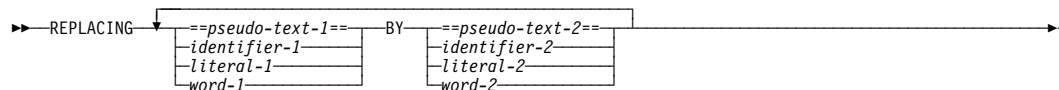
COPY Statement - Format 1



Notes:

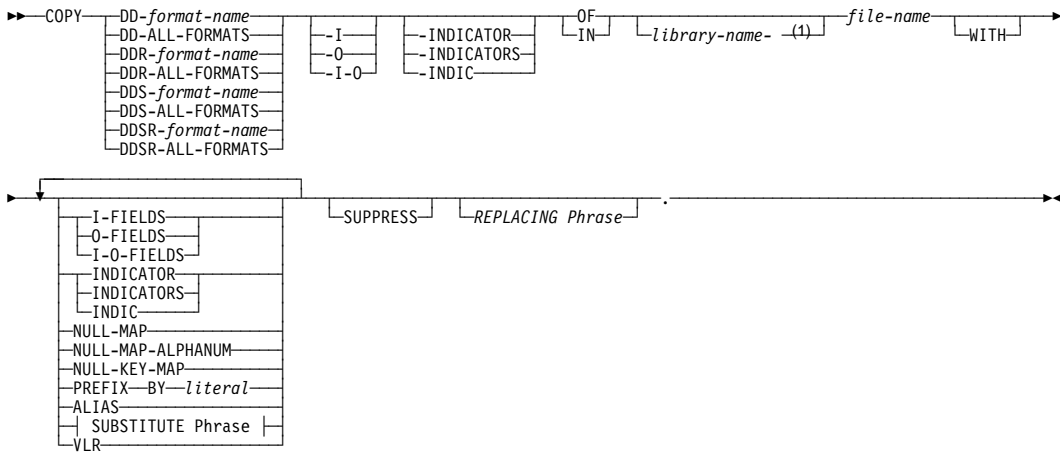
- 1 Required hyphen between library-name-file-name to qualify.
- 2 IBM Extension

REPLACING Phrase - Format

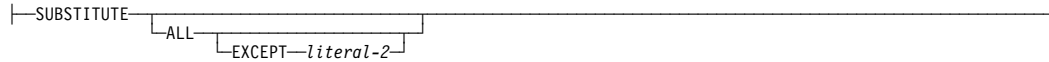


IBM Extension

COPY Statement - Format 2 - DDS Translate



SUBSTITUTE Phrase:



Note:

¹ Required hyphen between library-name-file-name to qualify.

End of IBM Extension

IBM Extension

EJECT Statement

EJECT Statement - Format

▶ EJECT [] ▶▶

End of IBM Extension

REPLACE Statement

REPLACE Statement - Format 1

▶ REPLACE [*==pseudo-text-1==* BY *==pseudo-text-2==*] . ▶▶

REPLACE Statement - Format 2

▶ REPLACE OFF . ▶▶

IBM Extension

SKIP Statement

SKIP1/2/3 Statements - Format

▶ SKIP1 []
 SKIP2 []
 SKIP3 [] . ▶▶

End of IBM Extension

IBM Extension

TITLE Statement

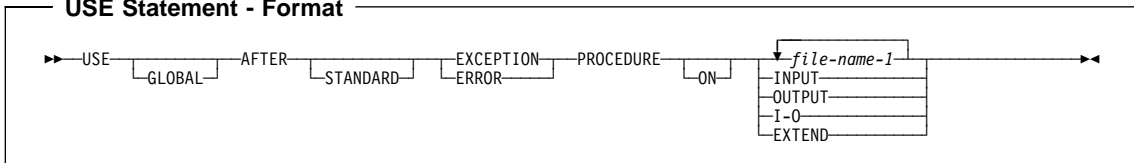
TITLE Statement - Format

▶ TITLE *literal* [] ▶▶

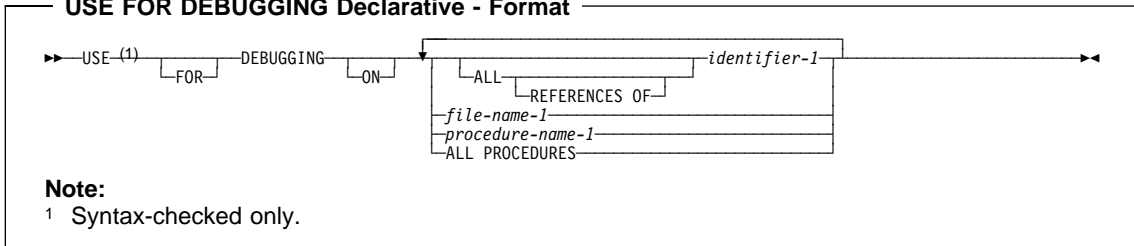
End of IBM Extension

USE Statement

USE Statement - Format



USE FOR DEBUGGING Declarative - Format



Symbols, Names, and Figurative Constants

Symbol	Meaning
A	Alphabetic character or space
B	Space insertion character
P	Decimal scaling position (not counted in size of data item)
S	Operational sign (not counted in size of data item unless a SIGN clause with optional SEPARATE CHARACTER phrase is specified)
V	Assumed decimal point (not counted in size of data item)
X	Alphanumeric character (any from the EBCDIC set)
Z	Zero suppression character
9	Numeric character

IBM Extension

1	Boolean character
E	Floating-point data
G	DBCS character
N	DBCS character

End of IBM Extension

0	Zero insertion character
/	Slash insertion character
,	Comma insertion character
.	Decimal point or period editing control character
+	Plus sign insertion editing control character
-	Minus sign editing control character
CR	Credit editing control character
DB	Debit editing control character
\$	Currency symbol insertion character (\$ is default).

Assignment-Names in the ASSIGN Clause

Format

► *device* — *file-name* [*-attribute*] ◄◄

device:	PRINTER FORMATFILE TAPEFILE DISKETTE DISK DATABASE WORKSTATION
file-name:	1-10 character name
attribute:	SI (separate indicator area) (allow null fields)

Environment-Names in the SPECIAL-NAMES Paragraph

<i>Table 1. Choices of Environment-Name-1 and Action Taken</i>	
Environment-name-1	Usage
CONSOLE, SYSTEM-CONSOLE	Communicate with the system operator's message queue (QSYSOPR).
REQUESTOR	Communicate with the user work station (interactive jobs) or the batch input stream or job log (batch jobs).
CSP	Suppress spacing when printing a line. Use only when PRINTER is the device.
C01	Skip to the next page. Use only when PRINTER is the device.
OPEN-FEEDBACK	Give information about a file, but only when the file is open.
I-O-FEEDBACK	Give information about the last I-O operation on a file, but only when the file is open.
DATA-AREA	Retrieves or updates an AS/400 data area.
ATTRIBUTE-DATA	Retrieve attribute data about a program device acquired by a transaction file, but only when the file is open.
LOCAL-DATA	Retrieve data from, or move data to the local data area created by the system for every job.
PIP-DATA	Retrieve data from the Program Initialization Parameters (PIP) data area for programs running as part of a prestart job.
SYSIN	The equivalent of REQUESTOR (for the ACCEPT statement only)
SYSOUT	The equivalent of REQUESTOR (for the DISPLAY statement only)

<i>Table 2. Choices of Environment-Name-2 and Action Taken</i>	
Environment-name-2	Usage
UPSI-0 through UPSI-7	Program switches associated with condition-names
SYSTEM-SHUTDOWN	Internal switches associated with condition-names

Figurative Constants

The following figurative constants can be used:

ALL "literal"
HIGH-VALUE
HIGH-VALUES
LOW-VALUE
LOW-VALUES

IBM Extension

NULL
NULLS

End of IBM Extension

QUOTE
QUOTES
SPACE
SPACES
ZERO
ZEROES
ZEROS

File Structure Support Summary

File Structure Support Summary and Status Key Values

File Structure Support Tables

Table 3 lists the required and optional entries for various types of file structures supported. Any file with a device type of disk can be assigned to a database or non-database auxiliary storage file. The codes used are as follows:

- . Not applicable
- B** Optional for a work station that supports subfiles
- C** Optional entry, treated as comments only
- D** Optional for file assigned to DATABASE-, not allowed if not assigned to a database file
- I** Optional for a file opened for input or input-output
- J** Optional for a file opened for input-output
- O** Optional
- R** Required
- S** Required for a work station that supports subfiles
- X** Required; syntax checked, but treated as documentation

Table 4 on page 85 and Table 5 on page 86 contain status key values and their meanings.

<i>Table 3 (Page 1 of 4). File Structure Support</i>												
Device Type	Printer	Tape	DiskSeq	Disk Rel Seq	Disk Rel Random	Disk Rel Dynamic	Disk IDX Seq	Disk IDX Random	Disk IDX Dynamic	Workstation	Diskette	Format File
Environment Division												
RERUN...RECORDS	C	C	C	C	C	C	C	C	C	C	C	C
SAME	O	O	O	O	O	O	O	O	O	O	O	O
AREA	C	C	C	C	C	C	C	C	C	C	C	C
RECORD AREA	O	O	O	O	O	O	O	O	O	O	O	O
SORT AREA	.	C	C
SORT MERGE AREA	.	C	C
MULTIPLE FILE TAPE	.	C
COMMITMENT CONTROL	.	.	D	D	D	D	D	D	D	.	.	.
SELECT	R	R	R	R	R	R	R	R	R	R	R	R
ASSIGN	R	R	R	R	R	R	R	R	R	R	R	R
OPTIONAL	.	.	I	I	I	I

File Structure Support Summary

Table 3 (Page 2 of 4). File Structure Support												
Device Type	Printer	Tape	DiskSeq	Disk Rel Seq	Disk Rel Random	Disk Rel Dynamic	Disk IDX Seq	Disk IDX Random	Disk IDX Dynamic	Workstation	Diskette	Format File
ORGANIZATION	O	O	O	R	R	R	R	R	R	R	O	O
SEQUENTIAL	O	O	O	O	O
RELATIVE	.	.	.	R	R	R
INDEXED	R	R	R	.	.	.
TRANSACTION	R	.	.
ACCESS	O	O	O	O	R	R	O	R	R	O	O	O
SEQUENTIAL	O	O	O	O	.	.	O	.	.	O	O	O
RANDOM	R	.	.	R
DYNAMIC	R	.	.	R	S	.	.
RESERVE	C	C	C	C	C	C	C	C	C	.	C	C
RELATIVE KEY	.	.	.	O	R	R	.	.	.	S	.	.
RECORD KEY	R	R	R	.	.	.
DUPLICATES	D	D	D	.	.	.
FILE STATUS	O	O	O	O	O	O	O	O	O	O	O	O
CONTROL-AREA	O	.	.
Data Division												
LABEL RECORDS	X	R	X	X	X	X	X	X	X	X	X	X
STANDARD	.	O	R	R	R	R	R	R	R	O	R	R
OMITTED	R	O	O	.	.
VALUE OF	C	C	C	C	C	C	C	C	C	C	C	C
BLOCK CONTAINS	O	O	O	O	O	O	O	O	O	O	O	O
RECORD CONTAINS	O	O	O	O	O	O	O	O	O	O	O	O
DATA RECORDS	O	O	O	O	O	O	O	O	O	O	O	O
CODE-SET	.	O	O	.
LINAGE	O
Procedure Division												
OPEN	R	R	R	R	R	R	R	R	R	R	R	R
INPUT	.	O	O	O	O	O	O	O	O	.	O	.
OUTPUT	R	O	O	O	O	O	O	O	O	.	O	O
I-O	.	.	O	O	O	O	O	O	O	R	.	.
NO REWIND	.	I
REVERSED	.	I
EXTEND	.	O	O	O
CLOSE	R	R	R	R	R	R	R	R	R	R	R	R

File Structure Support Summary

Table 3 (Page 3 of 4). File Structure Support

Device Type	Printer	Tape	DiskSeq	Disk Rel Seq	Disk Rel Random	Disk Rel Dynamic	Disk IDX Seq	Disk IDX Random	Disk IDX Dynamic	Workstation	Diskette	Format File
REEL/UNIT	.	O
REMOVAL	.	O
NO REWIND	.	O
NO REWIND	.	O
WITH LOCK	O	O	O	O	O	O	O	O	O	O	O	O
READ	.	I	I	I	I	I	I	I	I	I	I	.
NEXT	I	.	.	I	.	.	.
FIRST	D	.	.	.
LAST	D	.	.	.
PRIOR	D	.	.	.
INTO	.	I	I	I	I	I	I	I	I	I	I	.
WITH NO LOCK	.	.	J	J	J	J	J	J	J	.	.	.
KEY IS	I	I	.	.	.
AT END	.	I	I	I	.	I	I	.	I	I	I	.
NOT AT END	.	I	I	I	.	I	I	.	I	I	I	.
INVALID KEY	I	I	.	I	I	B	.	.
NOT INVALID KEY	I	I	.	I	I	B	.	.
FORMAT	.	.	D	.	.	.	D	D	D	J	.	R
NULL-KEY-MAP	D	D	D	.	.	.
NULL-MAP	.	.	D	D	D	D	D	D	D	.	.	.
NEXT MODIFIED	B	.	.
SUBFILE	B	.	.
INDICATORS	J	.	.
TERMINAL	O	.	.
NO DATA	O	.	.
WRITE	O	O	O	O	O	O	O	O	O	O	O	O
FROM	O	O	O	O	O	O	O	O	O	O	O	O
INVALID KEY	.	.	.	O	O	O	O	O	O	B	.	.
NOT INVALID KEY	.	.	.	O	O	O	O	O	O	B	.	.
ADVANCING	O
AT END-OF-PAGE	O
NOT AT END-OF-PAGE	O
FORMAT	.	.	D	.	.	.	D	D	D	R	.	R
NULL-KEY-MAP	D	D	D	.	.	.

File Structure Support Summary

<i>Table 3 (Page 4 of 4). File Structure Support</i>												
Device Type	Printer	Tape	DiskSeq	Disk Rel Seq	Disk Rel Random	Disk Rel Dynamic	Disk IDX Seq	Disk IDX Random	Disk IDX Dynamic	Workstation	Diskette	Format File
NULL-MAP	.	.	D	D	D	D	D	D	D	.	.	.
STARTING	O	.	.
ROLLING	O	.	.
INDICATORS	O	.	.
SUBFILE	B	.	.
TERMINAL	O	.	.
START	.	.	.	O	.	O	O	.	O	.	.	.
KEY	.	.	.	O	.	O	O	.	O	.	.	.
INVALID KEY	.	.	.	O	.	O	O	.	O	.	.	.
NOT INVALID KEY	.	.	.	O	.	O	O	.	O	.	.	.
FORMAT	D	D	D	.	.	.
NULL-KEY-MAP	D	D	D	.	.	.
REWRITE	.	.	O	O	O	O	O	O	O	B	.	.
FROM	.	.	O	O	O	O	O	O	O	B	.	.
INVALID KEY	O	O	.	O	O	B	.	.
NOT INVALID KEY	O	O	.	O	O	B	.	.
FORMAT	D	D	D	B	.	.
NULL-KEY-MAP	D	D	D	.	.	.
NULL-MAP	.	.	D	D	D	D	D	D	D	.	.	.
INDICATORS	B	.	.
SUBFILE	S	.	.
TERMINAL	O	.	.
DELETE	.	.	.	O	O	O	O	O	O	.	.	.
NULL-KEY-MAP	D	D	D	.	.	.
INVALID KEY	O	O	.	O	O	.	.	.
NOT INVALID KEY	O	O	.	O	O	.	.	.
FORMAT	D	D	D	.	.	.
USE	O	O	O	O	O	O	O	O	O	O	O	O
EXCEPTION/ERROR	O	O	O	O	O	O	O	O	O	O	O	O
FOR DEBUGGING	O	O	O	O	O	O	O	O	O	O	O	O
COMMIT	.	.	D	D	D	D	D	D	D	.	.	.
ROLLBACK	.	.	D	D	D	D	D	D	D	.	.	.
ACQUIRE	O	.	.
DROP	O	.	.

File Structure Support Summary

Return codes are set by the system after transaction I-O, which involves ICF files or DISPLAY files.

For more information about return codes, see the *ILE COBOL for AS/400 Programmer's Guide*.

<i>Table 4. File Status Keys and Corresponding Return Codes</i>			
File Status Key	Major Return Code	Minor Return Code	Explanation
00	00 03 08 09	xx xx except 09) 00 00	Normal completion (operation was successful). No data received. Acquire operation attempted to acquire an already active session or device. File has been dynamically created for OPEN OUTPUT. (See the OPTION(*CRTF) parameter description on the CRTCLMOD command in the <i>ILE COBOL for AS/400 Programmer's Guide</i> for further information about dynamic file creation.)
0A	02 03	xx 09	Job being cancelled (controlled).
10	11	00	Read-from-invited-program-device rejected; no invites outstanding.
30	80	xx	Permanent system error. The session has been ended.
92	81	xx	Permanent device or session error.
9C	82	xx	Open or acquire failed; session was not started.
9G	34	xx	Output exception to device or session.
9I	04	xx	Output exception to device or session.
9K	83	E0	Format not found.
9N	83	xx (except E0)	Session error. Session is still active.

File Status Key Values

File Status Key Values and Meanings

For information about **error handling**, refer to the “Error and Exception Handling” section in the *ILE COBOL for AS/400 Programmer's Guide*.

Table 5 (Page 1 of 5). File Status Key Values			
High Order Digit	Meaning	Low Order Digit	Meaning
0	Successful Completion	0	Nofurther information
		2	The READ statement was successfully executed, but a duplicate key was detected. That is, the key value for the current key of reference was equal to the value of the key in the next record. For information about enabling file status 02 see the accompanying notes under the READ statement.
		4	An attempt was made to read a record that is larger than the largest, or smaller than the smallest record allowed by the RECORD IS VARYING clause of the associated file-name.
		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 or EXTEND, the file has been created. CPF4101, CPF4102, CPF4103, CPF4207, CPF9812.
		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.
		A	Job ended in a controlled manner by CL command ENDJOB, PWRDWN SYS, ENDSYS, or ENDSBS CPF4741. Escape message sent during an accept input operation, READ from invited program device (multiple device listings only).
		M	Last record written to a subfile. CPF5003
		P	The file has been opened successfully, but it contains null-capable fields and the ASSIGN clause does not specify ALWNULL and device-type DATABASE.
		Q	A CLOSE statement for a sequentially-processed relative file was successfully executed. The file was created with the *INZDLT and *NOMAX options, so its boundary has been set to the number of records written.
1	At end conditions	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 (no invites outstanding) CPF4740, CPF5001, CPF5025.
		2	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center;">IBM Extension</p> </div> <p>No modified subfile record found. CPF5037</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center;">End of IBM Extension</p> </div>
		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.

File Status Key Values

<i>Table 5 (Page 2 of 5). File Status Key Values</i>			
High Order Digit	Meaning	Low Order Digit	Meaning
2	Invalid key	1	<p>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.</p> <p>Alternatively, the program has changed the record key value between a successful READ and subsequent REWRITE or DELETE operation on a randomly or dynamically-accessed file with duplicate keys.</p>
		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 in an indexed file. CPF4759, CPF5008, CPF5026, CPF5034, CPF5084, CPF5085.
		3	An attempt was made to randomly access a record that does not exist in the file. CPF5001, CPF5006, CPF5013, CPF5020, CPF5025.
		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 record key data item described for the file. CPF5006, CPF5018, CPF5021, CPF5043, CPF5272.
3	Permanent error condition	0	No further information CPF4192, CPF5101, CPF5102, CPF5129, CPF5030, CPF5143.
		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. CPF5116, CPF5018, CPF5272 if organization is sequential.
		5	An OPEN statement with the INPUT, I-O, or EXTEND phrase was attempted on a non-optional file that was not present. CPF4101, CPF4102, CPF4103, CPF4207, CPF9812.
		7	<p>An OPEN statement was attempted on a file that would not support the open mode specified in the OPEN statement. Possible violations are:</p> <ul style="list-style-type: none"> • The EXTEND or OUTPUT phrase was specified but the file would not support write operations. • The I-O phrase was specified but the file would not support the input and output operations permitted. • The INPUT phrase was specified but the file would not support read operations. <p>CPF4194.</p>
		8	An OPEN statement was attempted on a file previously closed with lock.
		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. The minimum record length specified by the program is less than the minimum record length required for the file. Level check error. CPF4131.

File Status Key Values

<i>Table 5 (Page 3 of 5). File Status Key Values</i>			
High Order Digit	Meaning	Low Order Digit	Meaning
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 that was already closed.
		3	For a sequential 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. An attempt was made to write or rewrite a record that is larger than the largest, or smaller than the smallest record allowed by the RECORD IS VARYING clause of the associated file-name.
		6	A sequential READ, READ NEXT or READ PRIOR statement was attempted on a file open in the input or I-O mode and no valid next record had been established because the preceding START statement was unsuccessful, or the preceding READ statement was unsuccessful or caused an at end condition. CPF5001, CPF5025, CPF5183.
		7	The execution of a READ or START 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 sequential file not open in the output, or extend mode. The execution of a WRITE statement was attempted on an indexed or relative 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.

File Status Key Values

Table 5 (Page 4 of 5). File Status Key Values			
High Order Digit	Meaning	Low Order Digit	Meaning
9	Other errors	0	<p>Other errors:</p> <ul style="list-style-type: none"> • File not found • Member not found • Unexpected I-O exceptions <p>CPF4101, CPF4102, CPF4103 if a USE is applicable for the file (on OPEN OUTPUT, non-optional file).</p> <p>The following exceptions are monitored generically:</p> <ul style="list-style-type: none"> • CPF4101 through CPF4399 • CPF4501 through CPF4699 • CPF4701 through CPF4899 • CPF5001 through CPF5099 • CPF5101 through CPF5399 • CPF5501 through CPF5699 <p>These exceptions are caught, and FILE STATUS is set to 90.</p>
		1	Undefined or unauthorized access type CPF2207, CPF4104, CPF4236, CPF4238, CPF5057, CPF5109, CPF5134, CPF5279.
		2	<p>Logic error:</p> <ul style="list-style-type: none"> • File locked • File already open • I-O to closed file • READ after end of file • CLOSE on unopened file <p>CPF4106, CPF4132, CPF4740, CPF5067, CPF5070, CPF5119, CPF5145, CPF5146, CPF5149, CPF5176, CPF5209.</p>
		4	No file position indicator REWRITE/DELETE when <i>not</i> sequential access, and last operation was not a successful READ.
9	Other errors	5	Invalid or incomplete file information (1) Duplicate keys specified in COBOL program. The file has been successfully opened, but indexed database file created with unique key; or (2) Duplicate keys not specified in COBOL program, and indexed database file created allowing duplicate keys.
		9	Undefined (display or ICF).
		C	Acquire failed; session was not started.
		D	Record is locked CPF5027, CPF5032.
		G	Output exception to device or session.
		H	ACQUIRE operation failed. Resource owned by another program, or unavailable. (9H is the result when an ACQUIRE operation causes any of the OS/400 exceptions monitored for 90, or 9N to occur.)
		I	WRITE operation failed CPF4702, CPF4737, CPF5052, CPF5076.
		K	Invalid format-name; format not found. CPF5022, CPF5023, CPF5053, CPF5054, CPF5121, CPF5152, CPF5153, CPF5186, CPF5187.

File Status Key Values

<i>Table 5 (Page 5 of 5). File Status Key Values</i>			
High Order Digit	Meaning	Low Order Digit	Meaning
9	Other errors	N	Temporary (potentially recoverable) hardware I-O error. (Error during communication session.) CPF4145, CPF4146, CPF4193, CPF4229, CPF4291, CPF4299, CPF4354, CPF4526, CPF4542, CPF4577, CPF4592, CPF4602, CPF4603, CPF4611, CPF4612, CPF4616, CPF4617, CPF4622, CPF4623, CPF4624, CPF4625, CPF4628, CPF4629, CPF4630, CPF4631, CPF4632, CPF4705, CPF5013, CPF5107, CPF5128, CPF5166, CPF5198, CPF5280, CPF5282, CPF5287, CPF5293, CPF5352, CPF5353, CPF5517, CPF5524, CPF5529, CPF5530, CPF5532, CPF5533.
		P	OPEN failed because file cannot be placed under commitment control CPF4293, CPF4326, CPF4327, CPF4328, CPF4329.
		Q	An OPEN statement for a randomly- or dynamically-accessed relative file failed because its size was *NOMAX. Change the file size (for example, using CHGPF) to the size you expect, and submit the program again.
		R	Referential integrity error. CPF502D, CPF502E, CPF503A.
		S	REWRITE or DELETE failed because last READ operation specified NO LOCK.
		T	Trigger program exception. CPF502B
		U	Cannot complete READ PRIOR because records are left in block from READ NEXT, or vice versa. CPF5184. Close the file, then open it again.
		W	Check constraint exception. CPF502F.
X	OPEN failed because the file type is not supported in a multithreaded job. Change the file type to DATABASE, PRINTER (spool file only), or a DDM file of type *IP and submit the program again. CPF4380.		

F-Name and Context-Sensitive Word List

ILE COBOL Function-Name and Context-Sensitive Word List

The following sections list all of the context-sensitive words and function-names in ILE COBOL.

Visual Key

The following key identifies the function-names and context-sensitive words in the ILE COBOL language:

- Blank** An ILE COBOL function-name or context-sensitive word from Standard COBOL.
- (1)** An ILE COBOL function-name or context-sensitive word that is an IBM extension to Standard COBOL.
- (2)** A COBOL function-name from the 1985 (revised 1989) ANSI Standard that is not used by the ILE COBOL compiler.

Function-Names

Function-Name	Function-Name	Function-Name
ACOS	ADD-DURATION (1)	ANNUITY (2)
ASIN	ATAN	CHAR
CONVERT-DATE-TIME (1)	COS	CURRENT-DATE
DATE-OF-INTEGERS	DATE-TO-YYYYMMDD (1)	DAY-OF-INTEGERS
DAY-TO-YYYYDDD (1)	EXTRACT-DATE-TIME (1)	FACTORIAL (2)
FIND-DURATION (1)	INTEGER (2)	INTEGER-OF-DATE
INTEGER-OF-DAY	INTEGER-PART (2)	LENGTH
LOCALE-DATE (1)	LOCALE-TIME (1)	LOG
LOG10	LOWER-CASE	MAX (2)
MEAN	MEDIAN (2)	MIDRANGE (2)
MIN (2)	MOD (2)	NUMVAL
NUMVAL-C	ORD	ORD-MAX (2)
ORD-MIN (2)	PRESENT-VALUE (2)	RANGE (2)
REM (2)	REVERSE	SIN
SQRT	STANDARD-DEVIATION (2)	SUBTRACT-DURATION (1)
SUM (2)	TAN	TEST-DATE-TIME (1)
UPPER-CASE	VARIANCE (2)	WHEN-COMPILED
YEAR-TO-YYYY (1)		

IBM Extension

Context-Sensitive Words

Context-Sensitive Word	Context
DAYS	MOVE FUNCTION ADD-DURATION(date-1 DAYS 90) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)

F-Name and Context-Sensitive Word List

Context-Sensitive Word	Context
DEFAULT	SET LOCALE LC_ALL FROM DEFAULT
HOURS	MOVE FUNCTION ADD-DURATION(time-1 HOURS 90) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
LC_ALL	SET LOCALE LC_ALL FROM DEFAULT
LC_COLLATE	SET LOCALE LC_COLLATE FROM DEFAULT
LC_CURRENCY	SET LOCALE LC_CURRENCY FROM DEFAULT
LC_MESSAGES	SET LOCALE LC_MESSAGES FROM DEFAULT
LC_MONETARY	SET LOCALE LC_MONETARY FROM DEFAULT
LC_NUMERIC	SET LOCALE LC_NUMERIC FROM DEFAULT
LC_TIME	SET LOCALE LC_TIME FROM DEFAULT
LC_TYPE	SET LOCALE LC_TYPE FROM DEFAULT
MICROSECONDS	MOVE FUNCTION ADD-DURATION(time-1 MICROSECONDS 30) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
MINUTES	MOVE FUNCTION ADD-DURATION(time-1 MINUTES 35) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
MONTHS	MOVE FUNCTION ADD-DURATION(date-1 MONTHS 12) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
SECONDS	MOVE FUNCTION ADD-DURATION(time-1 SECONDS 30) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
SYMBOL	CURRENCY IS "EUR" PICTURE SYMBOL "\$"
TIMESTAMP	05 date-1 FORMAT TIMESTAMP (Also found in SPECIAL-NAMES paragraph, intrinsic functions TEST-DATE-TIME and CONVERT-DATE-TIME.)
YEARS	MOVE FUNCTION ADD-DURATION(date-1 YEARS 2) (Also can be used in SUBTRACT-DURATION, FIND-DURATION, and EXTRACT-DATE-TIME.)
YYYYDDD	ACCEPT id-1 FROM DATE YYYYDDD
YYYYMMDD	ACCEPT id-1 FROM DATE YYYYMMDD
End of IBM Extension	

ILE COBOL Reserved Word List

ILE COBOL Reserved Word List

The following sections list all of the reserved words in ILE COBOL.

Visual Key

The following key identifies the reserved words in the ILE COBOL language:

- Blank** An ILE COBOL reserved word from Standard COBOL.
- (1)** An ILE COBOL reserved word that is an IBM extension to the Standard COBOL.
- (2)** A COBOL reserved word from Standard COBOL that is not used by the ILE COBOL compiler. These words should not be used if compatibility is important to an installation. If used, a diagnostic message will be issued.
- (3)** A COBOL reserved word that is not in Standard COBOL and is not supported by the ILE COBOL compiler. If used, a diagnostic message will be issued.

Reserved Words

Reserved Word	Reserved Word	Reserved Word
ACCEPT	ACCESS	ACQUIRE (1)
ADD	ADDRESS (1)	ADVANCING
AFTER	ALIAS (1)	ALL
ALPHABET	ALPHABETIC	ALPHABETIC-LOWER
ALPHABETIC-UPPER	ALPHANUMERIC	ALPHANUMERIC-EDITED
ALSO	ALTER	ALTERNATE
AND	ANY (2)	ARE
AREA	AREAS	ARITHMETIC (3)
ASCENDING	ASSIGN	AT
ATTRIBUTE (1)	AUTHOR	AUTO (1)
AUTO-SKIP (1)	AUTOMATIC (3)	BACKGROUND-COLOR (1)
BACKGROUND-COLOUR (1)	B-AND (3)	BEEP (1)
BEFORE	BELL (1)	B-EXOR (3)
BINARY	BIT (3)	BITS (3)
BLANK	B-LESS (3)	BLINK (1)
BLOCK	B-NOT (3)	BOOLEAN (3)
B-OR (3)	BOTTOM	BY
CALL	CANCEL	CD (2)
CF (2)	CH (2)	CHARACTER
CHARACTERS	CLASS	CLOCK-UNITS
CLOSE	COBOL (2)	CODE
CODE-SET	COL (1)	COLLATING
COLUMN	COMMA	COMMIT (1)
COMMITMENT (1)	COMMON	COMMUNICATION (2)
COMP	COMP-0 (3)	COMP-1 (1)
COMP-2 (1)	COMP-3 (1)	COMP-4 (1)
COMP-5 (3)	COMP-6 (3)	COMP-7 (3)
COMP-8 (3)	COMP-9 (3)	COMPUTATIONAL
COMPUTATIONAL-0 (3)	COMPUTATIONAL-1 (1)	COMPUTATIONAL-2 (1)

ILE COBOL Reserved Word List

Reserved Word	Reserved Word	Reserved Word
COMPUTATIONAL-3 (1)	COMPUTATIONAL-4 (1)	COMPUTATIONAL-5 (3)
COMPUTATIONAL-6 (3)	COMPUTATIONAL-7 (3)	COMPUTATIONAL-8 (3)
COMPUTATIONAL-9 (3)	COMPUTE	CONFIGURATION
CONNECT (3)	CONSOLE (1)	CONTAINED (3)
CONTAINS	CONTENT	CONTINUE
CONTROL	CONTROL-AREA (1)	CONTROLS
CONVERTING	COPY	CORR
CORRESPONDING	COUNT	CRT (1)
CRT-UNDER (1)	CURRENCY	CURRENT (3)
CURSOR (1)	DATA	DATE
DATE-COMPILED	DATE-WRITTEN	DAY
DAY-OF-WEEK	DB (3)	DB-ACCESS-CONTROL-KEY (3)
DB-DATA-NAME (3)	DB-EXCEPTION (3)	DB-FORMAT-NAME (1)
DB-RECORD-NAME (3)	DB-SET-NAME (3)	DB-STATUS (3)
DBCS (1)	DBCS-EDITED (1)	DE (2)
DEBUG-CONTENTS	DEBUG-ITEM	DEBUG-LINE
DEBUG-NAME	DEBUG-SUB-1	DEBUG-SUB-2
DEBUG-SUB-3	DEBUGGING	DECIMAL-POINT
DECLARATIVES	DEFAULT (3)	DELETE
DELIMITED	DELIMITER	DEPENDING
DESCENDING	DESCRIBED (1)	DESTINATION (2)
DETAIL (2)	DISABLE (2)	DISCONNECT (3)
DISPLAY	DISPLAY-1 (1)	DISPLAY-2 (3)
DISPLAY-3 (3)	DISPLAY-4 (3)	DISPLAY-5 (3)
DISPLAY-6 (3)	DISPLAY-7 (3)	DISPLAY-8 (3)
DISPLAY-9 (3)	DIVIDE	DIVISION
DOWN	DROP (1)	DUPLICATE (3)
DUPLICATES	DYNAMIC	EBCDIC (1)
EGI (2)	EJECT (1)	ELSE
EMI (2)	EMPTY (3)	EMPTY-CHECK (1)
ENABLE (2)	END	END-ACCEPT (1)
END-ADD	END-CALL	END-COMPUTE
END-DELETE	END-DISPLAY (1)	END-DIVIDE
END-EVALUATE	END-IF	END-INVOKE (1)
END-MULTIPLY	END-OF-PAGE	END-PERFORM
END-READ	END-RECEIVE (2)	END-RETURN
END-REWRITE	END-SEARCH	END-START
END-STRING	END-SUBTRACT	END-UNSTRING
END-WRITE	ENTER	ENTRY (1)
ENVIRONMENT	EOP	EQUAL
EQUALS (3)	ERASE (3)	ERROR
ESI (2)	EVALUATE	EVERY
EXCEEDS (3)	EXCEPTION	EXCLUSIVE (3)
EXIT	EXTEND	EXTERNAL
EXTERNALLY-DESCRIBED-KEY (1)	FALSE	FD
FETCH (3)	FILE	FILE-CONTROL
FILES (3)	FILLER	FINAL (2)
FIND (3)	FINISH (3)	FIRST
FOOTING	FOR	FOREGROUND-COLOR (1)
FOREGROUND-COLOUR (1)	FORMAT (1)	FREE (3)
FROM	FULL (1)	FUNCTION
GENERATE	GET (3)	GIVING
GLOBAL	GO	GOBACK (1)

ILE COBOL Reserved Word List

Reserved Word	Reserved Word	Reserved Word
GREATER	GROUP (2)	HEADING (2)
HIGHLIGHT (1)	HIGH-VALUE	HIGH-VALUES
I-O	I-O-CONTROL	ID (1)
IDENTIFICATION	IF	IN
INDEX	INDEXED	INDEX-1 (3)
INDEX-2 (3)	INDEX-3 (3)	INDEX-4 (3)
INDEX-5 (3)	INDEX-6 (3)	INDEX-7 (3)
INDEX-8 (3)	INDEX-9 (3)	INDIC (1)
INDICATE	INDICATOR (1)	INDICATORS (1)
INITIAL	INITIALIZE	INITIATE
INPUT	INPUT-OUTPUT	INSPECT
INSTALLATION	INTO	INVALID
INVOKE (1)	IS	JUST
JUSTIFIED	KANJI (1)	KEEP (3)
KEY	LABEL	LAST
LD (3)	LEADING	LEFT
LEFT-JUSTIFY (1)	LENGTH	LENGTH-CHECK (1)
LESS	LIBRARY (1)	LIKE (1)
LIMIT (2)	LIMITS (2)	LINAGE
LINAGE-COUNTER	LINE	LINE-COUNTER (2)
LINES	LINKAGE	LOCALE (1)
LOCALLY (3)	LOCAL-STORAGE (1)	LOCK
LOW-VALUE	LOW-VALUES	MEMBER (3)
MEMORY	MERGE	METACLASS (1)
MODE	MODIFIED (1)	MODIFY (3)
MODULES	MOVE	MULTIPLE
MULTIPLY	MESSAGE (2)	NATIVE
NEGATIVE	NEXT	NO
NO-ECHO (1)	NONE (3)	NOT
NULL-KEY-MAP (1)	NULL-MAP (1)	NULL (1)
NULLS (1)	NUMBER	NUMERIC
NUMERIC-EDITED	OBJECT (1)	OBJECT-COMPUTER
OCCURS	OF	OFF
OMITTED	ON	ONLY (3)
OPEN	OPTIONAL	OR
ORDER	ORGANIZATION	OTHER
OUTPUT	OVERFLOW	OWNER (3)
PACKED-DECIMAL	PADDING	PAGE
PAGE-COUNTER (2)	PERFORM	PF (2)
PH (2)	PICTURE	PLUS (2)
PIC	POINTER	POSITION
POSITIVE	PREFIX (1)	PRESENT (3)
PRINTING	PRIOR (1)	PROCEDURE
PROCEDURE-POINTER (1)	PROCEDURES	PROCEED
PROCESS (1)	PROGRAM-ID	PROMPT (1)
PROGRAM	PROTECTED (3)	PURGE (2)
QUEUE (2)	QUOTE	QUOTES
RANDOM	RD (2)	READ
READY (3)	REALM (3)	RECEIVE (2)
RECURSIVE (1)	RECONNECT (3)	RECORD
RECORD-NAME (3)	RECORDS	REDEFINES
REEL	REFERENCE	REFERENCE-MONITOR (3)
REFERENCES	RELATION (3)	RELATIVE

ILE COBOL Reserved Word List

Reserved Word	Reserved Word	Reserved Word
RELEASE	REMAINDER	REMOVAL
RENAMES	REPEATED (3)	REPLACE
REPLACING	REPORT (2)	REPORTING (2)
REPORTS (2)	REPOSITORY (1)	REQUIRED (1)
RERUN	RESERVE	RESET
RETAINING (3)	RETRIEVAL (3)	RETURN
RETURNING (1)	RETURN-CODE (1)	REVERSED
REVERSE-VIDEO (1)	REWIND	REWRITE
RF (2)	RH (2)	RIGHT
RIGHT-JUSTIFY (1)	ROLLBACK (1)	ROLLING (1)
ROUNDED	RUN	SAME
SCREEN (1)	SD	SEARCH
SECTION	SECURE (1)	SECURITY
SEGMENT (2)	SEGMENT-LIMIT	SELECT
SEND (2)	SENTENCE	SEPARATE
SEQUENCE	SEQUENTIAL	SET
SHARED (3)	SIGN	SIZE
SKIP1 (1)	SKIP2 (1)	SKIP3 (1)
SORT	SORT-MERGE	SORT-RETURN (1)
SOURCE (2)	SOURCE-COMPUTER	SPACE
SPACE-FILL (1)	SPACES	SPECIAL-NAMES
STANDARD	STANDARD-1	STANDARD-2
START	STARTING (1)	STATUS
STOP	STORE (3)	STRING
SUB-QUEUE-1 (2)	SUB-QUEUE-2 (2)	SUB-QUEUE-3 (2)
SUB-SCHEMA (3)	SUBFILE (1)	SUBSTITUTE (1)
SUBTRACT	SUM (2)	SUPPRESS
SYMBOLIC	SYNC	SYNCHRONIZED
SYSIN (1)	SYSOUT (1)	TABLE (2)
TALLYING	TAPE	TENANT (3)
TERMINAL	TERMINATE (2)	TEST
TEXT (2)	THAN	THEN
THROUGH	THRU	TIME
TIMES	TITLE (1)	TO
TOP	TRAILING	TRAILING-SIGN (1)
TRANSACTION (1)	TRUE	TYPE
TYPEDEF (1)	UNDERLINE (1)	UNEQUAL (3)
UNIT	UNSTRING	UNTIL
UP	UPDATE (1)	UPON
USAGE	USAGE-MODE (3)	USE
USING	VALID (3)	VALIDATE (3)
VALUE	VALUES	VARYING
VLR (1)	WAIT (3)	WHEN
WHEN-COMPILED (1)	WITH	WITHIN (3)
WORDS	WORKING-STORAGE	WRITE
ZERO	ZEROS	ZERO-FILL (1)
ZEROS	<	<=
+	*	**
-	/	>
>=	=	



Program Number: 5769-CB1

Printed in U.S.A.

SX09-1317-01

