



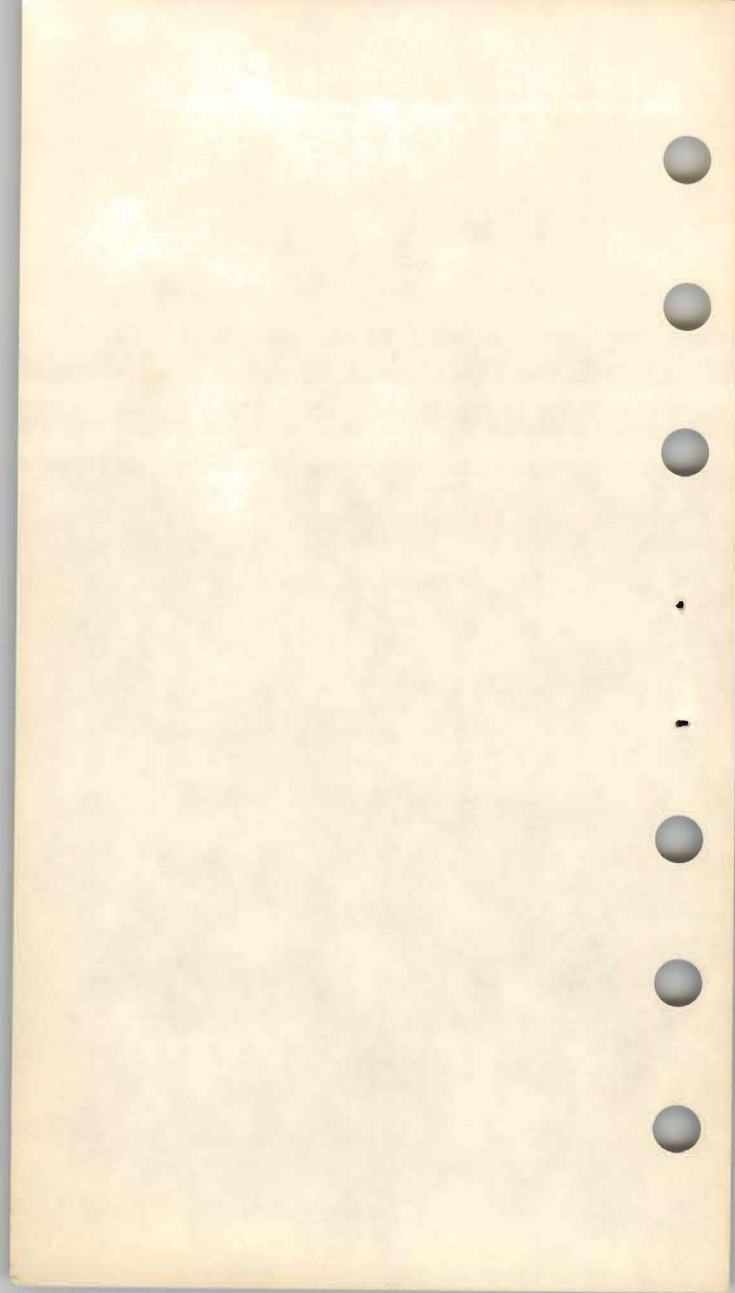
SC21-7781-4

File No. S38-24

IBM System/38

IBM System/38  
COBOL  
Reference Summary

Program Number 5714-CB1





SC21-7781-4

File No. S38-24

**IBM System/38**

**IBM System/38**

**COBOL**

**Reference Summary**

Program Number 5714-CB1

**Fifth Edition (April 1985)**

This major revision obsoletes SC21-7781-3. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to release 7, modification 0 of the IBM System/38 COBOL Program Product (Program 5714-CB1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Changes are periodically made to the information herein; changes will be reported in new editions of this publication. The *IBM System/38 COBOL Reference Manual Programmer's Guide*, SC21-7718, is the authoritative source and will be the first to reflect any changes.

Use this publication only for the purposes stated in *About This Manual*.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Canada Ltd., Information Development, Dept. 849, 1150 Eglinton Ave. E., Don Mills, Ontario, Canada M3C 1H7. IBM may use and distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1981, 1982, 1983, 1985

## Contents

ABOUT THIS MANUAL . . . . .	ix
CONVENTIONS IN FORMAT NOTATION . . . . .	1
CREATE COBOL PROGRAM COMMAND . . . . .	3
COBOL PROGRAM STRUCTURE . . . . .	7
Process Statement . . . . .	7
Identification Division . . . . .	8
Environment Division . . . . .	9
Data Division . . . . .	10
Procedure Division—Format 1—Section, Declaratives . . . . .	11
Procedure Division—Format 2 . . . . .	12
IDENTIFICATION DIVISION FORMAT . . . . .	13
ENVIRONMENT DIVISION FORMATS . . . . .	14
Configuration Section . . . . .	14
Input-Output Section . . . . .	16
FILE-CONTROL Paragraph—Sequential File Entries (READER, PUNCH, PUNCHPRINT, PRINT, PRINTER, TAPEFILE, DISKETTE, FORMATFILE, DISK, DATABASE) . . . . .	17
FILE-CONTROL Paragraph—Relative (Direct) File Entries (DISK, DATABASE) . . . . .	18
FILE-CONTROL Paragraph—Indexed File Entries (DISK, DATABASE) . . . . .	19
FILE-CONTROL Paragraph—Sort or Merge File Entries . . . . .	20
FILE-CONTROL Paragraph—TRANSACTION File Entries (WORKSTATION) . . . . .	20
I-O-CONTROL Paragraph . . . . .	21
DATA DIVISION FORMATS . . . . .	22
File Section Formats . . . . .	22
FD Entry—Files (FORMATFILE, DATABASE, DISK, READER, PUNCH, PUNCHPRINT, PRINT) . . . . .	22

FD Entry—Files (DISKETTE) . . . . .	24
FD Entry—Files (TAPEFILE) . . . . .	26
FD Entry—Files (PRINTER) . . . . .	28
FD Entry—TRANSACTION File . . . . .	30
SD Entry . . . . .	31
Working-Storage Section . . . . .	32
Data Description Entry—Format 1—Item Description . . . . .	32
Data Description Entry—Format 2—Regroup or Rename Data Items . . . . .	34
Data Description Entry—Format 3— Condition-name Description . . . . .	35
Data Description Entry—Format 4— Boolean Data Description . . . . .	36
Linkage Section . . . . .	38
 PROCEDURE DIVISION FORMATS . . . . .	 39
Procedure Division Header . . . . .	39
Procedure Division Statements . . . . .	39
ACCEPT Statement—Format 1 . . . . .	39
ACCEPT Statement—Format 2 . . . . .	40
ACCEPT Statement—Format 3 . . . . .	40
ACCEPT Statement—Format 4—Local Data Area . . . . .	41
ACCEPT Statement—Format 5—TRANSACTION Attributes . . . . .	41
ACQUIRE Statement—TRANSACTION File . . . . .	42
ADD Statement—Format 1 . . . . .	43
ADD Statement—Format 2—Giving . . . . .	43
ADD Statement—Format 3—Corresponding . . . . .	44
ALTER Statement . . . . .	44
CALL Statement . . . . .	45
CANCEL Statement . . . . .	45
CLOSE Statement—Format 1 . . . . .	46
CLOSE Statement—Format 2—TRANSACTION File . . . . .	47
COMMIT Statement . . . . .	47
COMPUTE Statement . . . . .	47
Declarative Procedures . . . . .	48
DELETE Statement . . . . .	48
DISPLAY Statement—Format 1 . . . . .	49

DISPLAY Statement—Format 2— Local Data Area . . . . .	49
DIVIDE Statement—Format 1 . . . . .	49
DIVIDE Statement—Format 2—Giving . . . . .	50
DIVIDE Statement—Format 3—Giving, Remainder . . . . .	50
DROP Statement—TRANSACTION File . . . . .	50
ENTER Statement . . . . .	51
EXIT Statement . . . . .	51
GO TO Statement—Format 1 . . . . .	51
GO TO Statement—Format 2—Depending On (Conditional) . . . . .	52
IF Statement . . . . .	52
INSPECT Statement—Format 1—Tallying . . . . .	53
INSPECT Statement—Format 2—Replacing . . . . .	54
INSPECT Statement—Format 3—Tallying and Replacing . . . . .	56
MERGE Statement . . . . .	58
MOVE Statement—Format 1 . . . . .	59
MOVE Statement—Format 2—Corresponding . . . . .	59
MULTIPLY Statement—Format 1 . . . . .	60
MULTIPLY Statement—Format 2—Giving . . . . .	60
OPEN Statement—Sequential Files . . . . .	61
OPEN Statement—Indexed and Relative Files . . . . .	62
OPEN Statement—TRANSACTION File . . . . .	62
PERFORM Statement—Format 1 . . . . .	62
PERFORM Statement—Format 2—Multiple Times . . . . .	63
PERFORM Statement—Format 3—Unit Condition Satisfied . . . . .	63
PERFORM Statement—Format 4—Varying Index or Identifier . . . . .	64
READ Statement—Format 1—Sequential Retrieval using SEQUENTIAL Access . . . . .	65
READ Statement—Format 2—Sequential Retrieval using DYNAMIC Access . . . . .	66
READ Statement—Format 3—Random Retrieval . . . . .	66
READ Statement—Format 4—TRANSACTION File (Nonsubfile) . . . . .	67
READ Statement—Format 5—TRANSACTION File (Subfile) . . . . .	68

RELEASE Statement . . . . .	69
RETURN Statement . . . . .	69
REWRITE Statement—Format 1 . . . . .	69
REWRITE Statement—Format 2—TRANSACTION File (Subfile) . . . . .	70
ROLLBACK Statement . . . . .	70
SEARCH Statement—Format 1—Selective Table Search . . . . .	71
SEARCH Statement—Format 2—Key Table Search . . . . .	72
SET Statement—Format 1—Switches . . . . .	73
SET Statement—Format 2—Condition Values . . . . .	73
SET Statement—Format 3—Index Save/Restore . . . . .	73
SET Statement—Format 4—Index Adjustment . . . . .	73
SORT Statement . . . . .	74
START Statement . . . . .	75
STOP Statement . . . . .	75
STRING Statement . . . . .	76
SUBTRACT Statement—Format 1 . . . . .	76
SUBTRACT Statement—Format 2—Giving . . . . .	77
SUBTRACT Statement—Format 3—Corresponding . . . . .	77
UNSTRING Statement . . . . .	78
USE Statement—EXCEPTION/ERROR Procedure— Format 1 . . . . .	79
USE Statement—EXCEPTION/ERROR Procedure (TRANSACTION)—Format 2 . . . . .	79
USE Statement—FOR DEBUGGING . . . . .	80
WRITE Statement—Format 1—Sequential Files . . . . .	81
WRITE Statement—Format 2—Indexed and Relative Files . . . . .	82
WRITE Statement—Format 3—FORMATFILE Files . . . . .	82
WRITE Statement—Format 4—TRANSACTION File (Nonsubfile) . . . . .	83
WRITE Statement—Format 5—TRANSACTION File (Subfile) . . . . .	84
CONDITIONAL EXPRESSIONS . . . . .	85
Class Condition . . . . .	85
Condition-Name Condition . . . . .	85
Relation Condition . . . . .	85
Sign Condition . . . . .	86



	Switch-Status Condition . . . . .	86
	Combined Condition . . . . .	86
	Negated Simple Condition . . . . .	86
	Abbreviated Combined Relation Condition . . . . .	87
	<b>QUALIFICATION OF DATA REFERENCE FORMATS</b>	<b>88</b>
	Data Item Reference . . . . .	88
	Procedure-Name Reference . . . . .	88
	COPY Library Reference . . . . .	88
	Subscripting . . . . .	89
	Indexing . . . . .	89
	<b>ALL DIVISIONS</b> . . . . .	<b>90</b>
	COPY Statement—Format 1 . . . . .	90
	COPY Statement—Format 2—DDS Translate . . . . .	91
	<b>PROCESS STATEMENT</b> . . . . .	<b>92</b>
	<b>SYMBOLS IN THE PICTURE CLAUSE</b> . . . . .	<b>93</b>
	<b>ASSIGNMENT-NAMES IN THE ASSIGN CLAUSE</b> . . . . .	<b>94</b>
	<b>FUNCTION-NAMES IN THE SPECIAL-NAMES</b>	
	<b>PARAGRAPH</b> . . . . .	95
	<b>FIGURATIVE CONSTANTS</b> . . . . .	<b>96</b>
	<b>STATUS KEY VALUES AND MEANINGS</b> . . . . .	<b>97</b>
	<b>COBOL RESERVED WORDS</b> . . . . .	<b>106</b>





### PURPOSE OF THIS MANUAL



The purpose of this publication is to summarize the formats of the IBM System/38 COBOL language.

This reference summary illustrates the following:



- The general outline of a COBOL program
- A detailed format of each clause and statement in each division of a COBOL program
- The PROCESS statement
- The symbols in the PICTURE clause
- The assignment-names in the ASSIGN clause
- The function-names in the SPECIAL-NAMES paragraph
- Status key values and meanings
- COBOL reserved words



### WHAT YOU SHOULD KNOW



Before reading this manual, you should be familiar with the *IBM System/38 COBOL Reference Manual and Programmer's Guide*, SC21-7718, which more fully documents all information contained in this summary.



**Note:** The separately orderable IBM binder, SR30-0700, and set of binder tabs, SX21-7884, can be used with this manual as well as with the *IBM System/38 Programming Reference Summary*, SC21-7734.





## Conventions In Format Notation

In this manual, the COBOL statement formats are presented in a uniform system of notation where capitalized words, underlined words, lowercase words, braces, square brackets, and ellipses have the following meanings:

- Reserved words are printed entirely in capital letters:
  - Required reserved words (key words) are also underlined.
  - Optional reserved words are not underlined.
- Programmer-defined words are printed in lowercase letters.
- Braces { } enclosing listed items indicate (1) that exactly one of the enclosed stacked items must be specified, and/or (2) when followed by an ellipsis, that the enclosed item or unit must be specified at least once.
- Square brackets [ ] indicate that the enclosed item or unit can be specified or omitted as required for the program. If items are stacked, only one of the items can be specified. When followed by an ellipsis, the item or unit can be repeated.
- An ellipsis ( . . . ) indicates that the immediately preceding unit can occur one or more times in succession.

2 The arithmetic and logical operators (+ - <> =) appearing in formats are required although they are not underlined. All punctuation and other special characters (except braces, square brackets, ellipses, commas, and semicolons) appearing in formats are required. Additional punctuation can be specified.

The required clauses and any specified optional clauses must be written in the sequence shown in the format except where accompanying text states otherwise.

**IBM extensions to American National Standard (ANS) COBOL, X3.23-1974, are boxed like this sentence.**

COBOL clauses and statements that are syntax-checked by the System/38 COBOL compiler, but that generate no code, are boxed like this sentence.

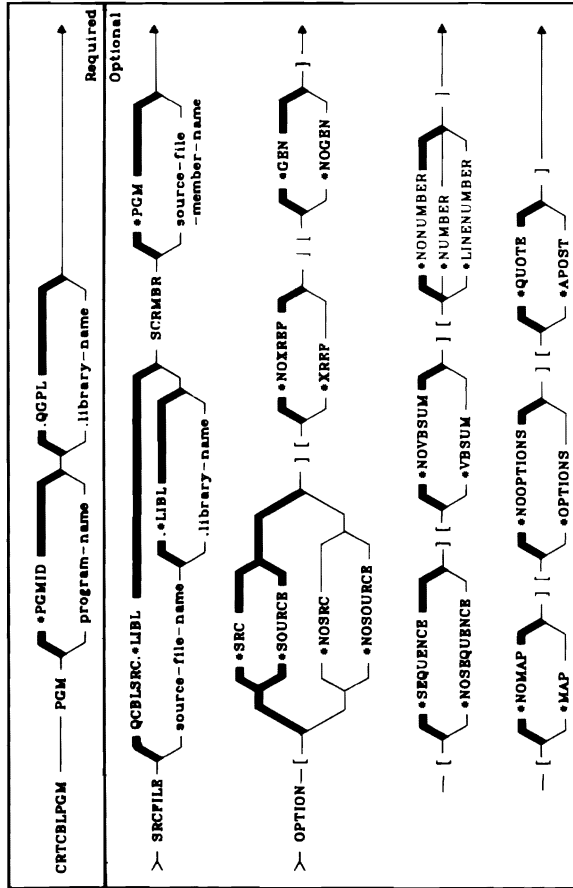




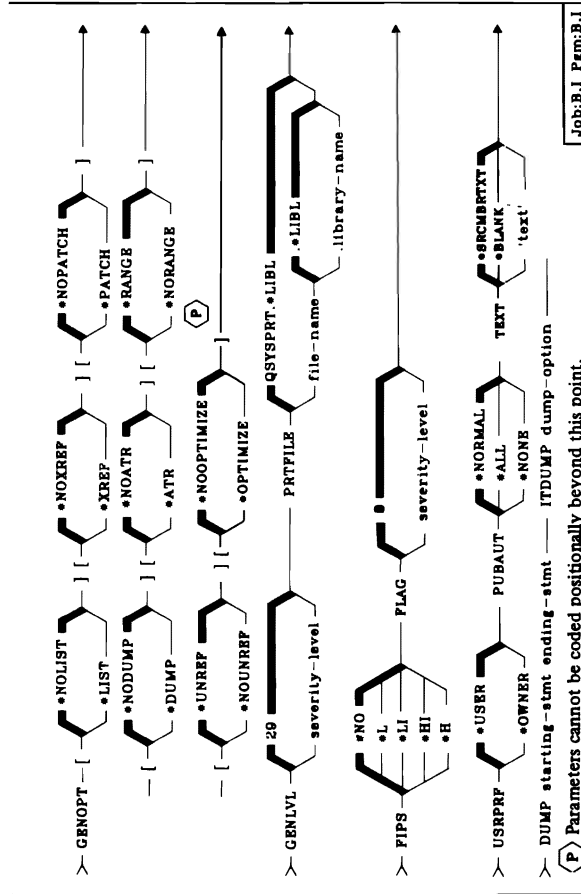
### **Create COBOL Program Command**

To compile a COBOL source program into an executable program, you must enter the Create COBOL Program (CRTCLPGM) command that invokes the COBOL compiler. The command is valid in batch and interactive jobs, or from other programs.

The command syntax is as follows. The defaults are indicated by heavier branch lines. The brackets indicate that the options can be specified in any order.







Job:B,I Pg:m:B,I

6 Notes:





## COBOL Program Structure

### PROCESS STATEMENT

The PROCESS statement specifies compile-time options. It is not a COBOL source statement. This statement must precede the Identification Division header if any options other than the default options of the PROCESS statement are desired. PROCESS statement options and defaults are listed later in this manual.

```
PROCESS option-1 [option-2] . . . [option-n] [.]
```

**Note:** A COPY statement can be specified within a PROCESS statement.

∞ IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name.

[AUTHOR. [comment-entry] . . .]

[INSTALLATION. [comment-entry] . . .]

[DATE WRITTEN. [comment-entry] . . .]

[DATE COMPILED. [comment-entry] . . .]

[SECURITY. [comment-entry] . . .]





**ENVIRONMENT DIVISION**

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER, source-computer-entry

OBJECT-COMPUTER, object-computer-entry

[ SPECIAL-NAMES, special-names-entry ]

[ INPUT-OUTPUT SECTION.

FILE-CONTROL, {file-control-entry} . . .

[ I-O-CONTROL, input-output-control-entry ]

## 10 DATA DIVISION

### DATA DIVISION.

#### [ FILE SECTION.

[ file-description-entry, { record-description-entry } . . . ]

[ sort-merge-file-description-entry, { record-description-entry } . . . ]]

#### [ WORKING-STORAGE SECTION.

[ data-description-entry ] . . .

[ record-description-entry ] . . . ]

#### [ LINKAGE SECTION.

[ data-description-entry ] . . .

[ record-description-entry ] . . . ]

**PROCEDURE DIVISION—FORMAT 1—SECTION, DECLARATIVES**

```
PROCEDURE DIVISION [ USING data-name-1 [ , data-name-2 ] . . . ] .  
[ DECLARATIVES.  
  {section-name SECTION [ segment-number ] . use-sentence  
  [ paragraph-name. [ sentence ] . . . } . . .  
  END DECLARATIVES.  
  {section-name SECTION [ segment-number ] .  
  [ paragraph-name. [ sentence ] . . . } . . .
```

**PROCEDURE DIVISION--FORMAT 2**

PROCEDURE DIVISION [ USING data-name-1 [ , data-name-2 ] . . . ]  
{ paragraph-name. [ sentence ] . . . } . . .







## Identification Division Format

IDENTIFICATION DIVISION.

PROGRAM-ID. program-name.

[ AUTHOR. [ comment-entry ] . . . ]

[ INSTALLATION. [ comment-entry ] . . . ]

[ DATE WRITTEN. [ comment-entry ] . . . ]

[ DATE COMPILED. [ comment-entry ] . . . ]

[ SECURITY. [ comment-entry ] . . . ]

14 Environment Division Formats

CONFIGURATION SECTION

CONFIGURATION SECTION.

SOURCE-COMPUTER, computer-name [WITH DEBUGGING MODE] .

OBJECT-COMPUTER, computer-name

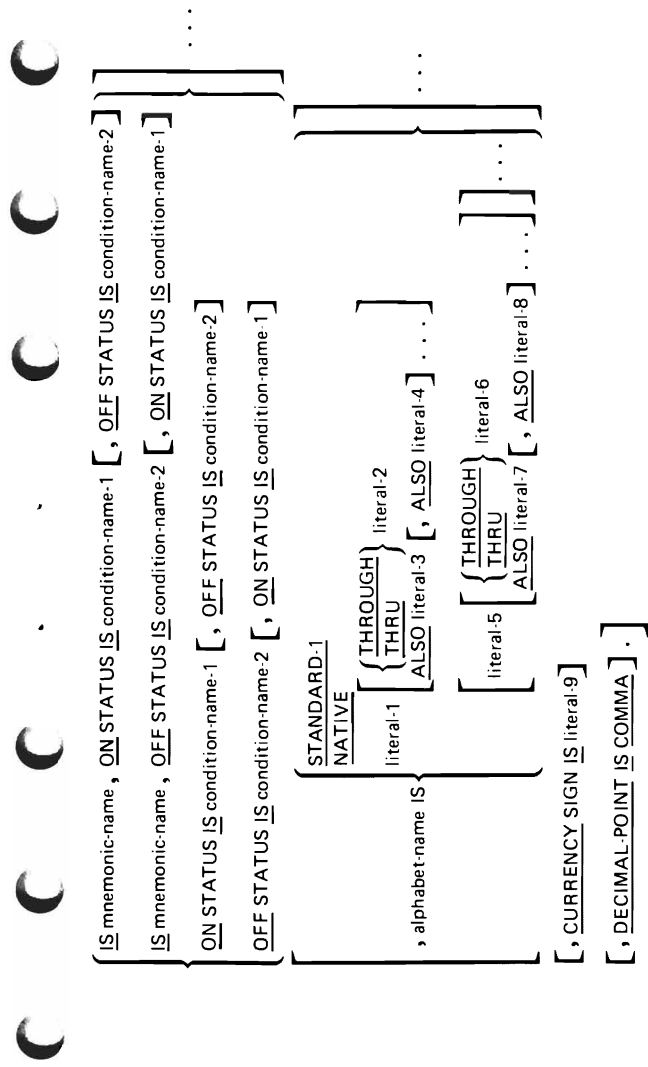
[ MEMORY SIZE integer }  
WORDS  
CHARACTERS  
MODULES ]

[ , PROGRAM COLLATING SEQUENCE IS alphabet-name ]

[ , SEGMENT LIMIT IS segment-number ] .

[ SPECIAL NAMES, [ function-name-1 IS mnemonic-name ] . . .

[ function-name-2



**INPUT-OUTPUT SECTION**

The key word **FILE-CONTROL** appears only once—at the beginning of the paragraph and before the first file-control entry. The key word **I-O-CONTROL** appears only once—at the beginning of the paragraph and before the input-output-control entry.

[ INPUT-OUTPUT SECTION,  
FILE-CONTROL, { file-control-entry } . . .  
[ I-O-CONTROL, input-output-control-entry ] ] .



**FILE-CONTROL Paragraph—Sequential File Entries (READER, PUNCH, PUNCHPRINT, PRINT, PRINTER, TAPEFILE, DISKETTE, FORMATFILE, DISK, DATABASE)**

SELECT [ OPTIONAL ] file-name  
ASSIGN TO assignment-name-1 [ [ , assignment-name-2 ] ... ]  
[ RESERVE integer-1 [ AREA  
[ AREAS ] ]  
[ ORGANIZATION IS SEQUENTIAL ]  
[ ACCESS MODE IS SEQUENTIAL ]  
[ FILE STATUS IS data-name-1 ] .

18 FILE-CONTROL Paragraph—Relative (Direct) File Entries (DISK, DATABASE)

SELECT file-name

ASSIGN TO assignment-name-1 [ , assignment-name-2 ] ...

[ RESERVE integer-1 [ AREA AREAS ] ]

ORGANIZATION IS RELATIVE

ACCESS MODE IS { SEQUENTIAL [ , RELATIVE KEY IS data-name-3 ]  
                  { RANDOM  
                  { DYNAMIC } , RELATIVE KEY IS data-name-3 } }

[ FILE STATUS IS data-name-1 ] .

**FILE-CONTROL Paragraph--Indexed File Entries (DISK, DATABASE)**

SELECT file-name

ASSIGN TO assignment-name-1 [assignment-name-2] ...

[RESERVE integer-1 [AREA AREAS]]

ORGANIZATION IS INDEXED

[ACCESS MODE IS {SEQUENTIAL  
RANDOM  
DYNAMIC}]

RECORD KEY IS {EXTERNALLY-DESCRIBED-KEY} [WITH DUPLICATES]  
data-name-2

[FILE STATUS IS data-name-1].

**FILE-CONTROL Paragraph—Sort or Merge File Entries**

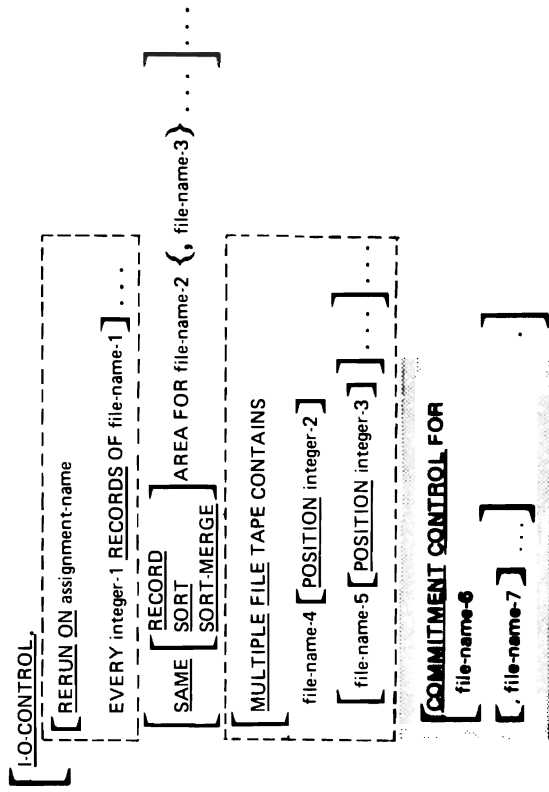
SELECT file-name/ASSIGN TO assignment-name-1 [ , assignment-name-2 ] ...

**FILE-CONTROL Paragraph—TRANSACTION File Entries (WORKSTATION)**

SELECT file-name  
ASSIGN TO assignment-name-1 [ , assignment-name-2 ] ...  
ORGANIZATION IS TRANSACTION  
ACCESS MODE IS { SEQUENTIAL  
DYNAMIC, RELATIVE KEY IS data-name-3 }  
FILE STATUS IS data-name-1 { , data-name-5 }  
CONTROL-AREA IS data-name-6 .



**I-O-CONTROL Paragraph**



**Data Division Formats**

**FILE SECTION FORMATS**

The order of the clauses specified after the FD file-name or SD file-name clause is optional.

**FD Entry—Files (FORMATFILE, DATABASE, DISK, READER, PUNCH, PUNCHPRINT, PRINT)**

[ FD file-name ]

[ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS } { CHARACTERS } ]

[ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

[ LABEL { RECORD IS } { STANDARD } { RECORDS ARE } { OMITTED } ]

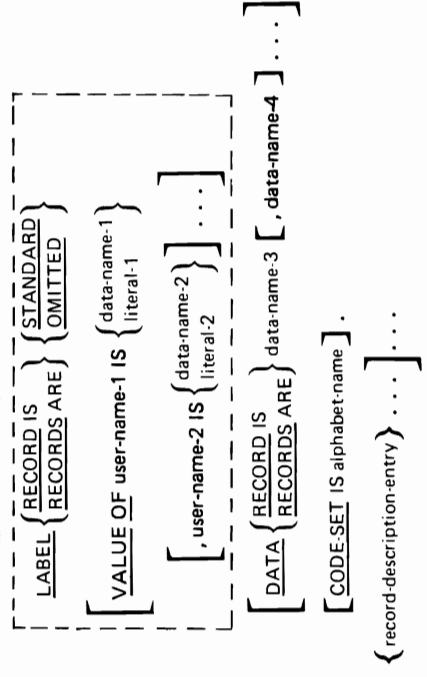


[ VALUE OF user-name-1 IS { data-name-1  
 literal-1 }  
 [ , user-name-2 IS { data-name-2  
 literal-2 } ] ... ]  
 [ DATA { RECORD IS  
 RECORDS ARE } data-name-3 [ , data-name-4 ] ... ]  
 { record-description-entry } ... ] ...

24 FD Entry—Files (DISKETTE)

[ FD file-name  
[ BLOCK CONTAINS [ integer-1 IO ] integer-2 { RECORDS  
CHARACTERS }  
[ RECORD CONTAINS [ integer-3 IO ] integer-4 CHARACTERS ]

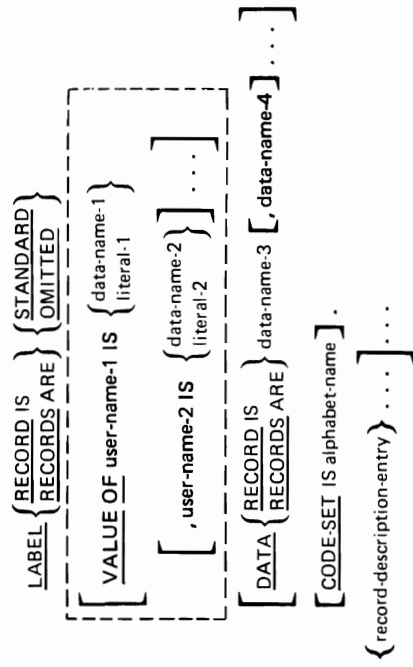




26 **FD Entry—Files (TAPEFILE)**

[ FD file-name  
[ BLOCK CONTAINS [ integer-1 IO] integer-2 { RECORDS  
CHARACTERS } ]  
[ RECORD CONTAINS [ integer-3 IO] integer-4 CHARACTERS ]





[ FD file-name ]

[ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS } { CHARACTERS } ]

[ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

[ LABEL { RECORD IS } { STANDARD } { RECORDS ARE } { OMITTED } ]

[ VALUE OF user-name-1 IS { data-name-1 } { literal-1 } ]

[ , user-name-2 IS { data-name-2 } { literal-2 } ] . . . ]





[ DATA { RECORD IS } data-name-3 [ , data-name-4 ] . . . ]  
 [ LINAGE IS { integer-5 } LINES [ , WITH FOOTING AT { data-name-6 } ]  
 [ , LINES AT TOP { data-name-7 } ] [ , LINES AT BOTTOM { data-name-8 } ] ] ]  
 { record-description-entry } . . .

FD Entry—TRANSACTION File

[ FD file-name  
RECORD CONTAINS [ integer-3 IO ] integer-4 CHARACTERS  
LABEL { RECORD IS } (STANDARD)  
          { RECORDS ARE } (OMITTED)  
DATA { RECORD IS } data-name-3 [ , data-name-4 ] . . .  
      { RECORDS ARE }  
      { record-description-entry } . . . ]

C C C C C C C C

**SD Entry**

```

[ SD file-name
  [ RECORD CONTAINS [ integer-1 TO ] integer-2 CHARACTERS
    [ DATA { RECORD IS } data-name-1 [ , data-name-2 ] . . . ] .
    { record-description-entry } . . . ]
  
```

**WORKING-STORAGE SECTION**

**Data Description Entry—Format 1—Item Description**

level-number { data-name-1  
                  FILLER }

[ REDEFINES data-name-2 ]

[ { PICTURE } IS character-string  
  PIC ]

[ [ USAGE IS ]  
  [ DISPLAY  
    COMPUTATIONAL  
    COMP  
    COMPUTATIONAL-3  
    COMP-3  
    COMPUTATIONAL-4  
    COMP-4  
    INDEX ] ] ]





[[ SIGN IS ] { LEADING } { TRAILING } ] [ SEPARATE CHARACTER ] ]

[ OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-3 } integer-2 TIMES ]

[ { { ASCENDING } { DESCENDING } } KEY IS data-name-4 [ , data-name-5 ] . . . ]

[ INDEXED BY index-name-1 [ , index-name-2 ] . . . ] ]

[ { { SYNCHRONIZED } { LEFT } { RIGHT } } ]

[ { { JUSTIFIED } { RIGHT } } { JUST } ]

[ BLANK WHEN ZERO ]

[ VALUE IS literal ] .

The clauses of the data description entry can be specified in any order except that the level-number and data-name/FILLER clause must be specified first, and if the REDEFINES clause is specified, it must immediately follow the data-name/FILLER clause.

**Data Description Entry—Format 2—Regroup or Rename Data Items**

66 data-name.1 RENAMES data-name.2 [ { THROUGH } data-name.3 ]





**Data Description Entry—Format 3—Condition-name Description**

88 condition-name { VALUE IS } literal-1 [ { THROUGH } literal-2  
VALUES ARE ]  
[ literal-3 [ { THROUGH } literal-4 ] ] . . . . .

36 Data Description Entry—Format 4—Boolean Data Description

level-number { data-name-1 }  
                  { FILLER }

[ REDEFINES data-name-2 ]

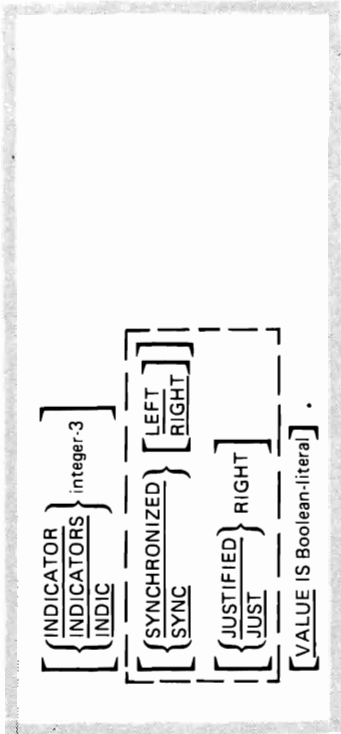
[ { PICTURE } IS 1 ]  
   { PIC }

[ [ USAGE IS ] DISPLAY ]

[ OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-3 }  
          integer-2 TIMES ]

[ INDEXED BY index-name-1 [ , index-name-2 ] . . . ]





**LINKAGE SECTION**

Any data description entry clause, except the VALUE clause, can describe items in the Linkage Section. The VALUE clause can be specified only for level-88 items. See *Working-Storage Section* earlier in this manual for data description entry clause formats.





**Procedure Division Formats**

**PROCEDURE DIVISION HEADER**

PROCEDURE DIVISION [ USING data-name-1 [ , data-name-2 ] . . . ] .

**PROCEDURE DIVISION STATEMENTS**

**ACCEPT Statement—Format 1**

ACCEPT identifier [ FROM mnemonic-name ]

**ACCEPT Statement—Format 2**

ACCEPT identifier FROM { DATE  
DAY  
TIME }

**ACCEPT Statement—Format 3**

ACCEPT identifier FROM mnemonic-name  
[ FOR file-name ]





**ACCEPT Statement – Format 4 – Local Data Area**

```
ACCEPT identifier-1 FROM mnemonic-name  
[FOR (literal) identifier-2]
```

**ACCEPT Statement – Format 5 – TRANSACTION Attributes**

```
ACCEPT identifier-1 FROM mnemonic-name  
[FOR (literal) identifier-2 (FOR file-name)]
```

ACQUIRE Statement—TRANSACTION File

42



**ADD Statement—Format 1**

ADD { identifier-1 } [ , identifier-2 ] . . . TO identifier-m [ ROUNDED ]  
[ , identifier-n [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]

**ADD Statement—Format 2—Giving**

ADD { identifier-1 } { identifier-2 } [ , identifier-3 ]  
[ , identifier-1 ] [ , identifier-2 ] [ , identifier-3 ] . . .  
GIVING identifier-m [ ROUNDED ] [ , identifier-n [ ROUNDED ] ] . . .  
[ ON SIZE ERROR imperative-statement ]

**4 ADD Statement—Format 3—Corresponding**

ADD { CORRESPONDING } identifier-1 TO identifier-2 [ ROUNDED ]  
[ ON SIZE ERROR imperative-statement ]

**ALTER Statement**

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





**CALL Statement**

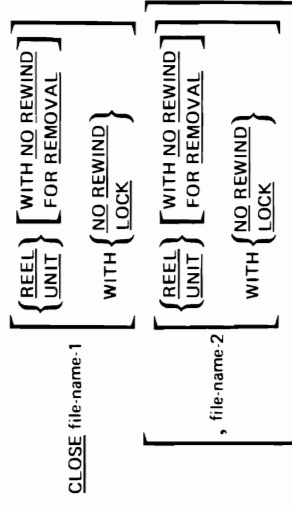
CALL { identifier-1 } [ USING data-name-1 [ , data-name-2 ] . . . ]  
[ ON OVERFLOW imperative-statement ]

**CANCEL Statement**

CANCEL { identifier-1 } [ , identifier-2 ] [ , literal-2 ] . . .

**CLOSE Statement—Format 1**

46





**CLOSE Statement—Format 2—TRANSACTION File**

CLOSE file-name-1 [ WITH LOCK ]  
[ file-name-2 [ WITH LOCK ] ] . . .

**COMMIT Statement**

COMMIT

**COMPUTE Statement**

COMPUTE identifier-1 [ ROUNDED ] [ , identifier-2 [ ROUNDED ] ] . . .  
= arithmetic-expression [ ON SIZE ERROR imperative-statement ]

**Declarative Procedures**

[ DECLARATIVES.  
{ section-name SECTION [ segment-number ] . **USE statement**  
[ paragraph-name, [ sentence ] . . . } . . .  
END DECLARATIVES. ]

**DELETE Statement**

DELETE file-name RECORD [ FORMAT IS { identifier }  
[ INVALID KEY imperative-statement ]

**DISPLAY Statement—Format 1**

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

**DISPLAY Statement—Format 2—Local Data Area**

DISPLAY { identifier-1 } [ literal-1 ] { identifier-2 } [ literal-2 ] . . .

UPON mnemonic-name

[ FOR { identifier-3 } [ literal-3 ] ]

**DIVIDE Statement—Format 1**

DIVIDE { identifier-1 } [ literal-1 ] INTO identifier-2 [ ROUNDED ]

[ identifier-3 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]

**DIVIDE Statement—Format 2—Giving**

$\underline{\text{DIVIDE}}$  {  $\frac{\text{INTO}}{\text{BY}}$  } {  $\frac{\text{identifier-2}}{\text{literal-2}}$  }  $\frac{\text{GIVING identifier-3}}{\text{literal-2}}$  [ ROUNDED ]  
 [ , identifier-4 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]

**DIVIDE Statement—Format 3—Giving, Remainder**

$\underline{\text{DIVIDE}}$  {  $\frac{\text{INTO}}{\text{BY}}$  } {  $\frac{\text{identifier-1}}{\text{literal-1}}$  } {  $\frac{\text{INTO}}{\text{BY}}$  } {  $\frac{\text{identifier-2}}{\text{literal-2}}$  }  $\frac{\text{GIVING identifier-3}}{\text{literal-2}}$  [ ROUNDED ]  
REMAINDER identifier-4 [ ON SIZE ERROR imperative-statement ]

**DROP Statement—TRANSACTION File**

$\underline{\text{DROP}}$  {  $\frac{\text{identifier}}{\text{literal}}$  } FROM file-name



### ENTER Statement

```
ENTER language-name [ routine-name ] .
```

### EXIT Statement

The EXIT statement must be preceded by a paragraph-name, and it must be the only statement in the paragraph.

```
paragraph-name. EXIT [ PROGRAM ] .
```

### GO TO Statement – Format 1

```
GO TO [ procedure-name-1 ]
```







**INSPECT Statement – Format 1 – Tallying**

INSPECT identifier-1 TALLYING

{ , identifier-2 FOR { { ALL LEADING } { identifier-3 } } { CHARACTERS } { literal-1 } }  
[ { BEFORE } INITIAL { identifier-4 } ] { literal-2 } ] { ... } { ... }

54 **INSPECT Statement—Format 2—Replacing**

INSPECT identifier-1 REPLACING

$\left\{ \left\{ \begin{array}{l} \text{ALL} \\ \text{LEADING} \\ \text{FIRST} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-5} \\ \text{literal-3} \end{array} \right\} \text{BY} \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-4} \end{array} \right\} \left\{ \begin{array}{l} \text{CHARACTERS} \\ \text{BY} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-4} \end{array} \right\} \left[ \left\{ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \text{identifier-7} \\ \text{literal-5} \end{array} \right\} \right] \right\}$

$\left[ \left\{ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \text{identifier-7} \\ \text{literal-5} \end{array} \right\} \right] \dots \left. \right\}$



Notes:

56 **INSPECT Statement—Format 3—Tallying and Replacing**

INSPECT identifier-1 TALLYING  
{ , identifier-2 FOR { { ALL { LEADING } { CHARACTERS } { identifier-3 } { literal-1 } } }  
[ { BEFORE } { AFTER } { INITIAL } { identifier-4 } { literal-2 } } { ..... } { ..... }

REPLACING

$\left\{ \begin{array}{l} \text{CHARACTERS BY } \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-4} \end{array} \right\} \left[ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right] \left[ \begin{array}{l} \text{identifier-7} \\ \text{literal-5} \end{array} \right] \\ \left\{ \begin{array}{l} \text{ALL} \\ \text{LEADING} \\ \text{FIRST} \end{array} \right\} \left\{ \begin{array}{l} \text{identifier-5} \\ \text{literal-3} \end{array} \right\} \text{BY } \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-4} \end{array} \right\} \\ \left[ \begin{array}{l} \text{BEFORE} \\ \text{AFTER} \end{array} \right] \left[ \begin{array}{l} \text{identifier-7} \\ \text{literal-5} \end{array} \right] \dots \end{array} \right\}$

**MERGE Statement**

```
MERGE file-name-1 ON { ASCENDING } KEY data-name-1 [ , data-name-2 ] . . .  
[ ON { ASCENDING } KEY data-name-3 [ , data-name-4 ] . . . ] . . .  
[ COLLATING SEQUENCE IS alphabet-name ]  
USING file-name-2, file-name-3 [ , file-name-4 ] . . .  
{ OUTPUT PROCEDURE IS section-name-1 [ { THROUGH } section-name-2 ] }  
[ GIVING file-name-5 ]
```



**MOVE Statement—Format 1**

MOVE { identifier.1 } TO identifier.2 [ , identifier.3 ] . . .

**MOVE Statement—Format 2—Corresponding**

MOVE { CORRESPONDING } identifier.1 TO identifier.2

**MULTIPLY Statement—Format 1**

MULTIPLY {  $\left. \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$  BY identifier-2 [ ROUNDED ]  
[ , identifier-3 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]

**MULTIPLY Statement—Format 2—Giving**

MULTIPLY {  $\left. \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$  BY {  $\left. \begin{array}{l} \text{identifier-2} \\ \text{literal-2} \end{array} \right\}$  GIVING identifier-3 [ ROUNDED ]  
[ , identifier-4 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]





**OPEN Statement—Sequential Files**

OPEN INPUT file-name-1 [ REVERSED [ WITH NO REWIND ] ]  
[ , file-name-2 [ REVERSED [ WITH NO REWIND ] ] ] ...  
OUTPUT file-name-3 [ WITH NO REWIND ]  
[ , file-name-4 [ WITH NO REWIND ] ] ...  
I/O file-name-5 [ , file-name-6 ] ...  
EXTEND file-name-7 [ , file-name-8 ] ...

82 OPEN Statement—Indexed and Relative Files

$$\text{OPEN } \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \\ \text{I-O} \end{array} \right\} \text{file-name-1 } [ \text{, file-name-2} ] \dots$$

OPEN Statement—TRANSACTION File

$$\text{OPEN I-O file-name-1 } [ \text{file-name-2} ] \dots$$

PERFORM Statement—Format 1

$$\text{PERFORM procedure-name-1 } [ \left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{procedure-name-2} ]$$



84 **PERFORM Statement—Format 4—Varying Index or Identifier**

PERFORM procedure-name-1 [ { THROUGH } procedure-name-2 ]

VARYING { identifier-1 } FROM { identifier-2 }  
                  { index-name-1 }            { literal-2 }

BY { identifier-3 } UNTIL condition-1  
      { literal-3 }

[ AFTER { identifier-4 } FROM { identifier-5 }  
          { index-name-4 }        { index-name-5 }  
                                  { literal-5 }

BY { identifier-6 } UNTIL condition-2  
      { literal-6 }

[ AFTER { identifier-7 } FROM { identifier-8 }  
 { index-name-7 } { index-name-8 }  
 { literal-8 } ]  
BY { identifier-9 } UNTIL condition-3 ] ]

**Read Statement—Format 1—Sequential Retrieval using SEQUENTIAL Access**

READ file-name RECORD

[ INTO identifier-1 ]

[ FORMAT IS { identifier-2 }  
 { literal-1 } ]

[ AT END imperative-statement ]

**6 READ Statement—Format 2—Sequential Retrieval using DYNAMIC Access**

READ file-name { FIRST  
LAST  
NEXT  
PRIOR } RECORD

[ INTO identifier-1 ]

[ FORMAT IS { identifier-2  
literal-1 } ]

[ AT END imperative-statement ]

**Read Statement—Format 3—Random Retrieval**

READ file-name RECORD [ INTO identifier-1 ]

[ FORMAT IS { identifier-2  
literal-1 } ]

[ INVALID KEY imperative-statement ]

**READ Statement—Format 4—TRANSACTION File (Nonserialfile)**

```
READ file-name RECORD  
  [ INTO identifier-1 ]  
  [ FORMAT IS { literal-1 } ]  
  [ TERMINAL IS { literal-2 } ]  
  [ INDICATOR IS { identifier-4 } ]  
  [ INDICATORS [ ARE ] ]  
  [ NO DATA imperative-statement-1 ]  
  [ AT END imperative-statement-2 ]
```





**RELEASE Statement**

RELEASE record-name [FROM identifier]

**RETURN Statement**

RETURN file-name RECORD [INTO identifier] AT END imperative-statement

**REWRITE Statement—Format 1**

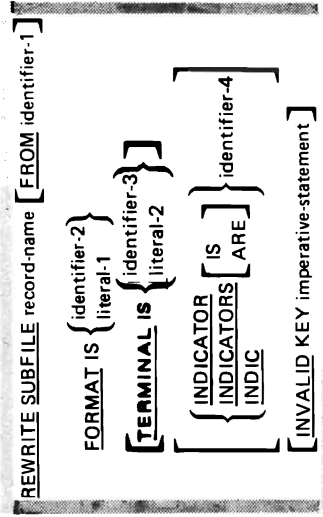
REWRITE record-name [FROM identifier.1]

[FORMAT IS { identifier.2  
literal.1 }]

[INVALID KEY imperative-statement]

**REWRITE Statement—Format 2—TRANSACTION File (Subfile)**

70



**ROLLBACK Statement**





**SEARCH Statement—Format 1—Selective Table Search**

SEARCH identifier-1 [ VARYING { identifier-2  
index-name.1 } ] [ AT END imperative-statement-1 ]

WHEN condition-1 { imperative-statement-2  
NEXT SENTENCE }

[ WHEN condition-2 { imperative-statement-3  
NEXT SENTENCE } ] . . .

**SEARCH Statement—Format 2—Key Table Search**

SEARCH ALL identifier-1 [AT END imperative-statement-1]

WHEN { data-name-1 { IS EQUAL TO { identifier-3  
literal-1  
arithmetic-expression-1 } }  
condition-name-1 }

[ AND { data-name-2 { IS EQUAL TO { identifier-4  
literal-2  
arithmetic-expression-2 } }  
condition-name-2 }  
imperative-statement-2  
NEXT SENTENCE }  
... ]



**SET Statement—Format 1—Switches**

SET mnemonic-name-1 [, mnemonic-name-2] . . . IO { ON / OFF }

**SET Statement—Format 2—Condition Values**

SET condition-name-1 [, condition-name-2] . . . IO TRUE

**SET Statement—Format 3—Index Save/Restore**

SET { identifier-1 [, identifier-2] . . . } IO { identifier-3 / index-name-3 / integer-1 }

**SET Statement—Format 4—Index Adjustment**

SET index-name-4 [, index-name-5] . . . { UP BY / DOWN BY } { identifier-4 / integer-2 }

**SORT Statement**

```

SORT file-name-1 ON {ASCENDING | DESCENDING} KEY data-name-1 [, data-name-2] . . .
[ ON {ASCENDING | DESCENDING} KEY data-name-3 [, data-name-4] . . . ] . . .

[COLLATING SEQUENCE IS alphabet-name]

{INPUT PROCEDURE IS section-name-1 [ {THROUGH | THRU} section-name-2] }
{USING file-name-2 [, file-name-3] . . . }
{OUTPUT PROCEDURE IS section-name-3 [ {THROUGH | THRU} section-name-4] }
{GIVING file-name-4 }

```

**START Statement**

START file-name KEY IS  $\left\{ \begin{array}{l} \text{EQUAL TO} \\ = \\ \text{GREATER THAN} \\ > \\ \text{NOT LESS THAN} \\ \text{NOT} < \end{array} \right\}$   $\left\{ \begin{array}{l} \text{EXTERNALLY-DESCRIBED-KEY} \\ \text{data-name-1} [ , \text{data-name-2} ] \dots \end{array} \right\}$

FORMAT IS  $\left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}$

INVALID KEY imperative-statement

**STOP Statement**

STOP  $\left\{ \begin{array}{l} \text{RUN} \\ \text{literal} \end{array} \right\}$

**STRING Statement**

$$\text{STRING } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} [ , \text{identifier-2} ] [ , \text{literal-2} ] \dots \text{DELIMITED BY } \left\{ \begin{array}{l} \text{identifier-3} \\ \text{literal-3} \\ \text{SIZE} \end{array} \right\}$$

$$\left[ , \left\{ \begin{array}{l} \text{identifier-4} \\ \text{literal-4} \end{array} \right\} [ , \text{identifier-5} ] [ , \text{literal-5} ] \dots \text{DELIMITED BY } \left\{ \begin{array}{l} \text{identifier-6} \\ \text{literal-6} \\ \text{SIZE} \end{array} \right\} \right] \dots$$
  
 INTO identifier-7 [ WITH POINTER identifier-8 ]  
 [ ON OVERFLOW imperative-statement ]

**SUBTRACT Statement—Format 1**

$$\text{SUBTRACT } \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} [ , \text{identifier-2} ] [ , \text{literal-2} ] \dots \text{FROM identifier-3 [ ROUNDED ]}$$

$$\left[ , \text{identifier-4 [ ROUNDED ]} \right] \dots [ \text{ON SIZE ERROR imperative-statement} ]$$





**SUBTRACT Statement—Format 2—Giving**

SUBTRACT { identifier-1 } [ , identifier-2 ] [ , literal-2 ] . . . FROM { identifier-3 }  
literal-3 }  
GIVING identifier-4 [ ROUNDED ] [ , identifier-5 [ ROUNDED ] ] . . .  
[ ON SIZE ERROR imperative-statement ]

**SUBTRACT Statement—Format 3—Corresponding**

SUBTRACT { CORRESPONDING } identifier-1 FROM identifier-2 [ ROUNDED ]  
CORR }  
[ ON SIZE ERROR imperative-statement ]

**UNSTRING Statement**UNSTRING identifier-1

[ DELIMITED BY [ ALL ] { identifier-2 } [ , OR [ ALL ] { literal-2 } ] . . . ]  
INTO identifier-4 [ , DELIMITER IN identifier-5 ] [ , COUNT IN identifier-6 ]  
[ , identifier-7 ] [ , DELIMITER IN identifier-8 ] [ , COUNT IN identifier-9 ] ] . . .  
[ WITH POINTER identifier-10 ] [ TALLYING IN identifier-11 ]  
[ ON OVERFLOW imperative-statement ]

**USE Statement—EXCEPTION/ERROR Procedure—Format 1**

USE AFTER STANDARD { EXCEPTION } PROCEDURE ON { file-name-1 [ file-name-2 ] . . . }  
ERROR { INPUT }  
I/O { OUTPUT }  
EXTEND { }

**USE Statement—EXCEPTION/ERROR Procedure (TRANSACTION)—Format 2**

USE AFTER STANDARD { ERROR }  
EXCEPTION { file-name-1 [ file-name-2 ] . . . }  
PROCEDURE ON { I/O }

### 80 USE Statement— FOR DEBUGGING

USE FOR DEBUGGING ON  $\left\{ \begin{array}{l} \text{[ ALL REFERENCES OF ] identifier-1} \\ \text{file-name-1} \\ \text{procedure-name-1} \\ \text{ALL PROCEDURES} \end{array} \right\}$

$\left[ \begin{array}{l} \text{[ ALL REFERENCES OF ] identifier-2} \\ \text{file-name-2} \\ \text{procedure-name-2} \\ \text{ALL PROCEDURES} \end{array} \right] \dots$





**WRITE Statement—Format 1—Sequential Files**

WRITE record-name [ FROM identifier-1 ]  
[ { BEFORE } ADVANCING { identifier-2 } { LINE }  
  { AFTER }                   integer        LINES }  
  { mnemonic-name }  
  PAGE }  
[ AT { END-OF-PAGE } imperative-statement ]  
      { EOP }

**32 WRITE Statement—Format 2—Indexed and Relative Files**

WRITE record-name [ FROM identifier-1 ]

[ FORMAT IS { identifier-2  
literal-1 } ]

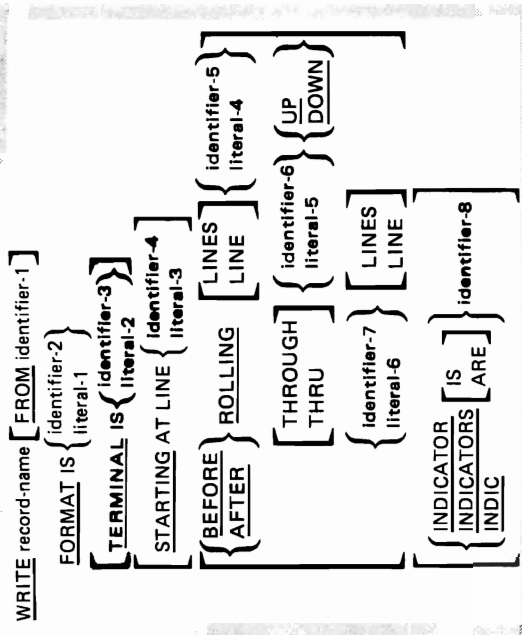
[ INVALID KEY imperative-statement ]

**WRITE Statement—Format 3—FORMATFILE Files**

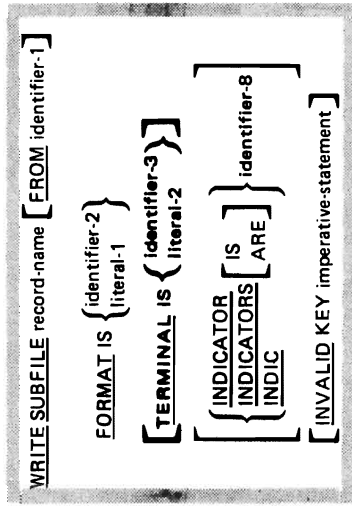
WRITE record-name [ FROM identifier-1 ]

[ FORMAT IS { identifier-2  
literal-1 } ]  
[ { INDICATOR  
INDICATORS [ IS  
INDIC ARE ] identifier-3 } ]  
[ AT { END-OF-PAGE  
EOP } imperative-statement ]

**WRITE Statement—Format 4—TRANSACTION File (Nonsubfile)**



**WRITE Statement—Format 5—TRANSACTION File (Subfile)**







## Conditional Expressions

### Class Condition

identifier IS [ NOT ] { NUMERIC / ALPHABETIC }

### Condition-Name Condition

condition.name

### Relation Condition

operand-1 IS [ NOT ] { GREATER THAN / LESS THAN / EQUAL TO } > < = operand-2

**Sign Condition**

operand IS [ NOT ]

{ POSITIVE  
NEGATIVE  
ZERO }

**Switch-Status Condition**

condition-name

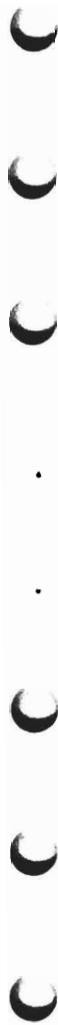
**Combined Condition**

condition { { AND }  
                  { OR } } condition . . . .

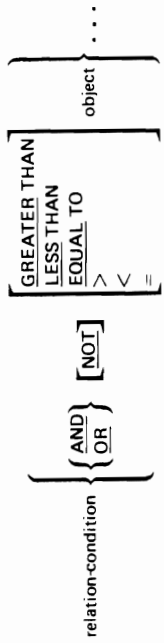
**Negated Simple Condition**

NOT simple-condition

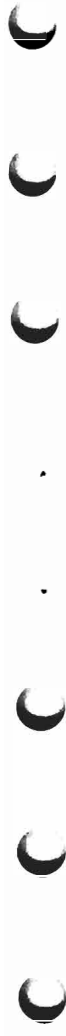




**Abbreviated Combined Relation Condition**



**Qualification of Data Reference Formats****Data Item Reference**
$$\left\{ \begin{array}{l} \text{data-name:1} \\ \text{condition-name} \end{array} \right\} \left[ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \left\{ \begin{array}{l} \text{data-name:2} \\ \dots \end{array} \right\}$$
**Procedure-Name Reference**
$$\text{paragraph-name} \left[ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{section-name}$$
**COPY Library Reference**
$$\text{text-name} \left[ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{System/38 file name} \left[ - \text{System/38 library name} \right]$$



**Subscripting**

{data-name-1} [ {OE/IN} data-name-2 ] . . . ( subscript 1 [ , subscript-2 [ , subscript-3 ] ] )

**Indexing**

{data-name-1} [ {OE/IN} data-name-2 ] ( { {index-name 1 [ { {literal-2} } ] } } {literal-1} )  
[ { {index-name-2 [ { {literal-4} } ] } } [ { {index-name 3 [ { {literal-6} } ] } } ] ] ] )

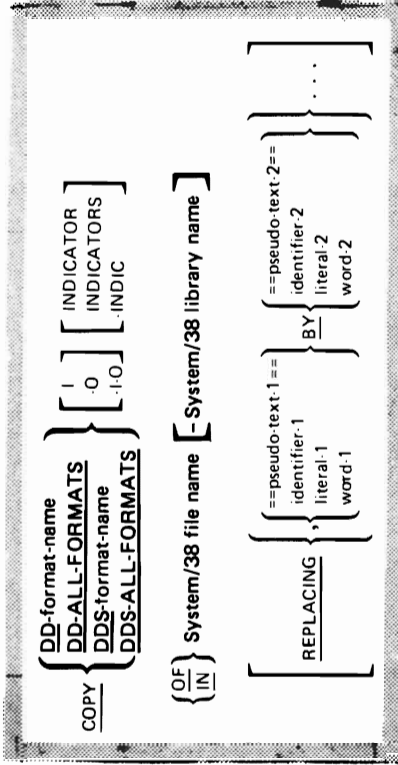
COPY Statement—Format 1

$$\text{COPY } \text{text-name} \left[ \begin{matrix} \text{OF} \\ \text{IN} \end{matrix} \right] \text{System/38 file name} \left[ - \text{System/38 library name} \right]$$

$$\left[ \begin{matrix} \text{REPLACING} \\ \text{BY} \end{matrix} \right] \left\{ \begin{matrix} \text{==pseudo;text-1==} \\ \text{identifier-1} \\ \text{literal-1} \\ \text{word-1} \end{matrix} \right\} \left\{ \begin{matrix} \text{==pseudo;text-2==} \\ \text{identifier-2} \\ \text{literal-2} \\ \text{word-2} \end{matrix} \right\} \dots \right]$$



**COPY Statement—Format 2—DDS Translate**



**PROCESS Statement**

The PROCESS statement allows you to specify compile-time options unique to COBOL. The PROCESS statement must immediately precede the IDENTIFICATION DIVISION header. The format of the PROCESS statement is as follows:

PROCESS option 1 [option 2] . . . [option n] [.]

The following list identifies PROCESS statement options. The options can appear in any order. Defaults are underlined.

SOURCE or SRC	LINENUMBER	LIST
<u>NOSOURCE</u> or <u>NOSRC</u>	NUMBER	<u>NOLIST</u>
XREF	<u>NONUMBER</u>	GENVL(n) (n = 0 through 29)
<u>NOXREF</u>	MAP	<u>GENVL(29)</u>
<u>GEN</u>	<u>NOMAP</u>	FIPS( <u>NO</u> ,L,L,I,HI,H)
NOGEN	OPTIONS	
SEQUENCE	<u>NOOPTIONS</u>	
<u>NOSEQUENCE</u>	APOST	FLAG(n) (n = 0 through 99)
VBSUM	<u>QUOTE</u>	<u>FLAG(00)</u>
<u>NOVBSUM</u>		





### Symbols in the PICTURE Clause

#### 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
1	Boolean character
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
*	Check protect insertion character
`cs`	Currency sign insertion character (default is \$)

Assignment-Names in the ASSIGN Clause

device[System/38 file name [- attribute]]

where attribute can be one of the following:

- - hopper [- association]
- - SI

System/38

file name: 1-10 character unhyphenated system-name

attribute: - hopper [- association]  
- SI

hopper: P or S (for card devices only)

association: 0-9 (for card devices only)

Device	Name and Default	Hopper	Association
READER	O QCARD96	O	O
PUNCH	O QCARD96	O	O
PUNCHPRINT	O QCARD96	O	O
PRINT	O QCARD96	O	O
PRINTER	O QPRINT	N	N
TAPEFILE	O QTAPE	N	N
DISKETTE	O QDKT	N	N
DISK	R	N	N
DATABASE	R	N	N
WORKSTATION	R	N	N
FORMATFILE	R	N	N
R = Required O = Optional N = Not allowed			

## Function-Names in the SPECIAL-NAMES Paragraph

### function-name-1

CONSOLE	System console
SYSTEM CONSOLE	System console
REQUESTOR	Work station or batch input stream
CSP	Suppress spacing after printing a line
C01	Skip to next page
S01 through S05	Stacker selection on a card punch file
OPEN FEEDBACK	Information about a file
I-O-FEEDBACK	Information about the last I-O operation
LOCAL-DATA	Data area created by the system for every job
ATTRIBUTE-DATA	Information about a work station or communications device

### function-name-2

UPI-0 through UPI-7	Program switches associated with condition-names
SYSTEM-SHUTDOWN	Internal switches associated with condition-names

## 96 **Figurative Constants**

The following figurative constants can be used:

ALL 'literal'  
HIGH-VALUE  
HIGH-VALUES  
LOW-VALUE  
LOW-VALUES  
QUOTE  
QUOTES  
SPACE  
SPACES  
ZERO  
ZEROS  
ZEROS





### Status Key Values and Meanings

See the *System/38 Messages Guide: CPF, RPG III, and IDU, SC21-7736*, for information on CPF exceptions.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
0	<b>Successful Completion</b> Successful completion	No error condition occurred during the I-O operation.
1	<b>At End of File</b> End of file	CPF4740, CPF5001, CPF5025, CPF5070.
2	No modified subfile record found (IBM extension)	CPF5037.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
1	Invalid Key Sequence error	REWRITE to an indexed file with sequential access and key for REWRITE ≠ key from previous READ. WRITE to an indexed file with sequential access and key values for succeeding writes are not in ascending sequence.
2	Duplicate key when duplicates are not allowed	CPF5008, CPF5009, CPF5010, CPF5026, CPF5034, CPF5084, CPF5085; or WRITE to an indexed file with sequential access and key values for succeeding writes are not in ascending sequence.
3	No record found	CPF5006, CPF5013.
4	Boundary violation	CPF5018, CPF5020, CPF5043, CPF5272, CPF5305 if organization is not sequential.





Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
3	<b>Permanent Error</b>	
0	Permanent error	CPF5030, CPF5036, CPF5101, CPF5102, CPF5113, CPF5120, CPF5129, CPF5143, CPF5155.
4	Boundary violation	CPF5116. CPF5018, CPF5272, CPF5305 if organization is sequential.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<p><b>Other Errors</b></p> <p>Other errors:</p> <ul style="list-style-type: none"> <li>• File not found</li> <li>• Member not found</li> <li>• Level check error</li> <li>• Unexpected I-O exceptions</li> </ul>	<p>CPF4101 if a USE is applicable for the file.                      CPF4102 if a USE is applicable for the file.                      CPF4131.</p> <p>The following exceptions are monitored generically:                      CPF 4101 through CPF 4399                      CPF 4501 through CPF 4699                      CPF 4701 through CPF 4899                      CPF 5001 through CPF 5099                      CPF 5101 through CPF 5399</p> <p>These exceptions are caught, and FILE STATUS is set to 90. If a USE procedure is applicable, it is executed. Otherwise, the program terminates and gives the operator the exception and the option to cancel, take a partial dump, or take a full dump.</p>





Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<b>Other Errors (continued)</b> Undefined or unauthorized access type	CPF2207, CPF4104, CPF5057, CPF5109, CPF5134, CPF5279.
1	Logic error:	CPF4102, CPF4106, CPF5013, CPF5119, CPF5132, CPF5145, CPF5146, CPF5176, CPF5183.
2	<ul style="list-style-type: none"> <li>● File locked</li> <li>● File already open</li> <li>● I-O to closed file</li> <li>● READ after end of file</li> <li>● CLOSE on unopened file</li> </ul>	
4	No current record pointer	REWRITE/DELETE with sequential access, and last operation was not a successful READ.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<b>Other Errors (continued)</b> Invalid or incomplete file information	(1) Duplicate keys specified in COBOL program, but indexed data base file created with unique key; or (2) Duplicate keys not specified in COBOL program, and indexed data base file created allowing duplicate keys.
A	Job has been cancelled in a controlled manner by CL command CNLJOB, TRMSBS, TRMCPF, or PWRDWN SYS	CPF4741
D	Record is locked	CPF 5027



Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<b>Other Errors (continued)</b> ACQUIRE operation failed	Resource owned by another program, or unavailable. (9H is the result when an ACQUIRE operation causes any of the CPF exceptions monitored for 90, 9J, or 9N to occur). CPF5105, CPF5127, CPF5128, CPF5165, CPF5166, CPF5198, CPF5217, CPF5219, CPF5254, CPF5260, CPF5265, CPF5269, CPF5274, CPF5278, CPF5279, CPF5351, CPF5355, CPF5358, CPF5410, CPF5411, CPF5412, CPF5413, CPF5424, CPF5447, CPF5449, CPF5450, CPF5454, CPF5507, CPF5508, CPF5509, CPF5510, CPF5511, CPF5512, CPF5513, CPF5514, CPF5515, CPF5516, CPF5517, CPF5518, CPF5519, CPF5521, CPF5537, CPF5538, CPF5539, CPF5540, CPF5541, CPF5542, CPF5543, CPF5544, CPF5545, CPF5547, CPF5552.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<b>Other Errors (continued)</b> WRITE operation failed Invalid format-name; format not found Last record written to subfile	CPF5044. CPF5022, CPF5023, CPF5053, CPF5054, CPF5121, CPF5152, CPF5153, CPF5186, CPF5187. CPF5003, CPF5043.

Status Key 1 2	Meaning	When Set (CPF exceptions monitored, condition detected)
9	<p><b>Other Errors (continued)</b></p> <p>Temporary (potentially re-coverable) hardware I-O error</p>	<p>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, CPF5107, CPF5128, CPF5166, CPF5198, CPF5280, CPF5282, CPF5287, CPF5293, CPF5352, CPF5353, CPF5517, CPF5524, CPF5529, CPF5530, CPF5532, CPF5533.</p> <p>CPF4285, CPF4293, CPF4326, CPF4327, CPF4328, CPF4329, CPF4330, CPF4331.</p>
P	<p>OPEN failed because file cannot be placed under commitment control</p>	

**COBOL Reserved Words**

No word in the following list should appear as a programmer-defined name.

Each word in the list is preceded by an identifier associated with one of the following meanings:

(blank) A System/38 COBOL reserved word from American National Standard COBOL, X3.23-1974.

1 A System/38 COBOL reserved word that is an IBM extension to the 1974 ANS standard.

2 A COBOL reserved word from the 1974 ANS standard not used by System/38 COBOL. If used, a diagnostic message is sent.

3 A CODASYL COBOL reserved word (from the *CODASYL COBOL Journal of Development*, December, 1978) that is not included in the 1974 ANS standard and is not supported by System/38 COBOL as an extension. If used, a diagnostic message is sent. These words are included for compatibility.

ACCEPT	AREAS	CALL	COMP
ACCESS	ASCENDING	CANCEL	1 COMP-3
1 ACQUIRE	ASSIGN	2 CD	1 COMP-4
ADD	AT	2 CF	COMPUTATIONAL
ADVANCING	1 ATTRIBUTE	2 CH	1 COMPUTATIONAL-3
AFTER	AUTHOR	CHARACTER	1 COMPUTATIONAL-4
ALL		CHARACTERS	COMPUTE
3 ALPHABET		2 CLOCK-UNITS	CONFIGURATION
ALPHABETIC	BEFORE	CLOSE	3 CONNECT
3 ALPHANUMERIC	3 BEGINNING	2 COBOL	CONTAINS
3 ALPHANUMERIC-EDITED	3 BINARY	2 CODE	3 CONTENT
ALSO	3 BIT	CODE-SET	3 CONTINUE
ALTER	3 BITS	COLLATING	1 CONTROL
ALTERNATE	BLANK	2 COLUMN	1 CONTROL-AREA
AND	BLOCK	COMMA	2 CONTROLS
3 ANY	3 BOOLEAN	1 COMMIT	3 CONVERSION
ARE	BOTTOM	1 COMMITMENT	3 CONVERTING
AREA	BY	3 COMMON	COPY
		2 COMMUNICATION	

CORR	3 DB-RECORD-NAME	3 DEBUG-SUB-NUM	1 DROP
CORRESPONDING	3 DB-SET-NAME	DEBUGGING	3 DUPLICATE
COUNT	3 DB-STATUS	DECIMAL-POINT	DUPLICATES
CURRENCY	2 DE	DECLARATIVES	DYNAMIC
3 CURRENT	DEBUG-CONTENTS	DELETE	
	DEBUG-ITEM	DELIMITED	2 EGI
	3 DEBUG-LENGTH	DELIMITER	ELSE
	DEBUG-LINE	DEPENDING	2 EMI
DATA	DEBUG-NAME	DESCENDING	3 EMPTY
DATE	3 DEBUG-NUMERIC-CONTENTS	2 DESTINATION	2 ENABLE
DATE-COMPILED	3 DEBUG-SIZE	2 DETAIL	END
DATE-WRITTEN	3 DEBUG-START	2 DISABLE	3 END-ADD
DAY	3 DEBUG-SUB	3 DISCONNECT	3 END-CALL
3 DAY-OF-WEEK	DEBUG-SUB-1	DISPLAY	3 END-COMPUTE
3 DB	3 DEBUG-SUB-2	3 DISPLAY-n	3 END-DELETE
3 DB-ACCESS-CONTROL-KEY	DEBUG-SUB-3	DIVIDE	3 END-DIVIDE
3 DB-DATA-NAME	3 DEBUG-SUB-ITEM	DIVISION	3 END-EVALUATE
3 DB-EXCEPTION	3 DEBUG-SUB-N	DOWN	3 END-IF
1 DB-FORMAT-NAME			



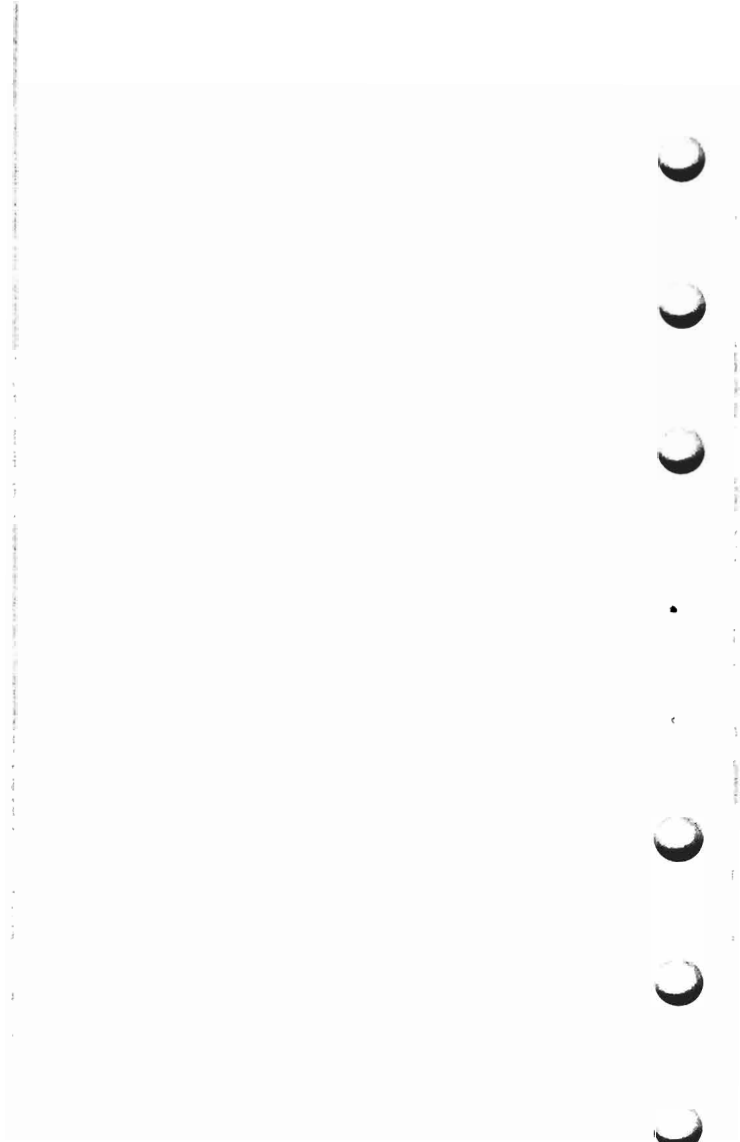
3 END-MULTIPLY	3 EQUALS	FILE	GO
END-OF-PAGE	3 ERASE	FILE-CONTROL	GREATER
3 END-PERFORM	ERROR	FILLER	2 GROUP
3 END-READ	2 ESI	2 FINAL	
3 END-RECEIVE	3 EVALUATE	3 FIND	2 HEADING
3 END-RETURN	EVERY	3 FINISH	HIGH-VALUE
3 END-REWRITE	3 EXCEEDS	FIRST	HIGH-VALUES
3 END-SEARCH	EXCEPTION	FOOTING	
3 END-START	3 EXCLUSIVE	FOR	
3 END-STRING	EXIT	1 FORMAT	
3 END-SUBTRACT	3 EXOR	3 FREE	I-O
3 END-UNSTRING	EXTEND	FROM	I-O-CONTROL
3 END-WRITE	3 EXTERNAL		IDENTIFICATION
3 ENDING	1 EXTERNALLY-DESCRIBED-KEY	2 GENERATE	IF
ENTER		3 GET	IN
ENVIRONMENT		GIVING	INDEX
EOP	3 FALSE	3 GLOBAL	3-INDEX-n
EQUAL	FD		INDEXED

1 INDIC			
2 INDICATE			
1 INDICATOR			
1 INDICATORS			
INITIAL			
3 INITIALIZE			
2 INITIATE			
INPUT			
INPUT-OUTPUT			
INSPECT			
INSTALLATION			
INTO			
INVALID			
IS			
JUST			
JUSTIFIED			
3 KEEP			
KEY			
LABEL			
LAST			
3 LD			
LEADING			
LEFT			
2 LENGTH			
LESS			
2 LIMIT			
2 LIMITS			
LINAGE			
LINAGE-COUNTER			
LINE			
2 LINE-COUNTER			
LINES			
LINKAGE			
3 LOCALLY			
LOCK			
LOW-VALUE			
LOW-VALUES			
3 MEMBER			
MEMORY			
MERGE			
2 MESSAGE			
MODE			
1 MODIFIED			
3 MODIFY			
MODULES			
MOVE			
MULTIPLE			
MULTIPLY			
NATIVE			
NEGATIVE			
NEXT			
NO			
3 NON-NULL			
NOT			
3 NULL			
2 NUMBER			
NUMERIC			
3 NUMERIC-EDITED			
OBJECT-COMPUTER			
OCCURS			
OF			
OFF			
OMITTED			
ON			

OPEN	2 PH	3 REFERENCE
OPTIONAL	PIC	3 REFERENCE-MODIFIER
OR	PICTURE	REFERENCES
3 ORDER	2 PLUS	RELATIVE
ORGANIZATION	POINTER	RELEASE
3 OTHER	POSITION	REMAINDER
OUTPUT	POSITIVE	REMOVAL
OVERFLOW	2 PRINTING	RENAMES
3 OWNER	1 PRIOR	3 REPLACE
	PROCEDURE	REPLACING
	PROCEDURES	2 REPORT
3 PACKED-DECIMAL	PROCEED	2 REPORTING
3 PADDING	1 PROCESS	2 REPORTS
PAGE	PROGRAM	RERUN
2 PAGE-COUNTER	PROGRAM-ID	RESERVE
PERFORM	3 PROTECTED	2 RESET
2 PF	3 PURGE	3 RETAINING
		3 RETRIEVAL
		2 QUEUE
		QUOTE
		QUOTES
		RANDOM
		2 RD
		READ
		3 READY
		3 REALM
		3 REALMS
		2 RECEIVE
		3 RECONNECT
		RECORD
		3 RECORD-NAME
		RECORDS
		REDEFINES
		REEL

RETURN	SEGMENT-LIMIT	STANDARD	SYNC
REVERSED	SELECT	STANDARD-1	SYNCHRONIZED
REWIND	2 SEND	3 STANDARD-2	
REWRITE	SENTENCE	START	2 TABLE
2 RF	SEPARATE	1 STARTING	TALLYING
2 RH	SEQUENCE	STATUS	TAPE
RIGHT	SEQUENTIAL	STOP	3 TENANT
1 ROLLBACK	SET	3 STORE	TERMINAL
1 ROLLING	3 SETS	STRING	2 TERMINATE
ROUNDED	SIGN	2 SUB-QUEUE-1	3 TEST
RUN	SIZE	2 SUB-QUEUE-2	2 TEXT
	SORT	2 SUB-QUEUE-3	THAN
SAME	SORT-MERGE	3 SUB-SCHEMA	1 THEN
SD	2 SOURCE	1 SUBFILE	THROUGH
SEARCH	SOURCE-COMPUTER	SUBTRACT	THRU
SECTION	SPACE	2 SUM	TIME
SECURITY	SPACES	2 SUPPRESS	TIMES
2 SEGMENT	SPECIAL-NAMES	2 SYMBOLIC	TO

TOP	VALUE	-
TRAILING	VALUES	*
1 TRANSACTION	VARYING	/
1 TRUE		**
2 TYPE		>
		<
		=
UNIT	WHEN	
UNSTRING	WITH	
UNTIL	3 WITHIN	
UP	WORDS	
	WORKING-STORAGE	
3 UPDATE	WRITE	
UPON		
USAGE	ZERO	
3 USAGE-MODE	ZEROES	
USE	ZEROS	
USING		



READER'S COMMENT FORM

IBM System/38  
COBOL  
Reference Summary

SC21-7781-4

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

**Note:** Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

**Please use this form only to identify publication errors or to request changes in publications.**

Possible topics for comment are:

Clarity	Accuracy	Completeness	Organization	Retrieval	Legibility
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): \_\_\_\_\_ Comments(s): \_\_\_\_\_

Please print

Name \_\_\_\_\_  
 Company or Organization \_\_\_\_\_  
 Address \_\_\_\_\_

No postage necessary if mailed in the U.S.A. City State Zip Code



NO POSTAGE  
NECESSARY IF  
MAILED IN THE  
UNITED STATES

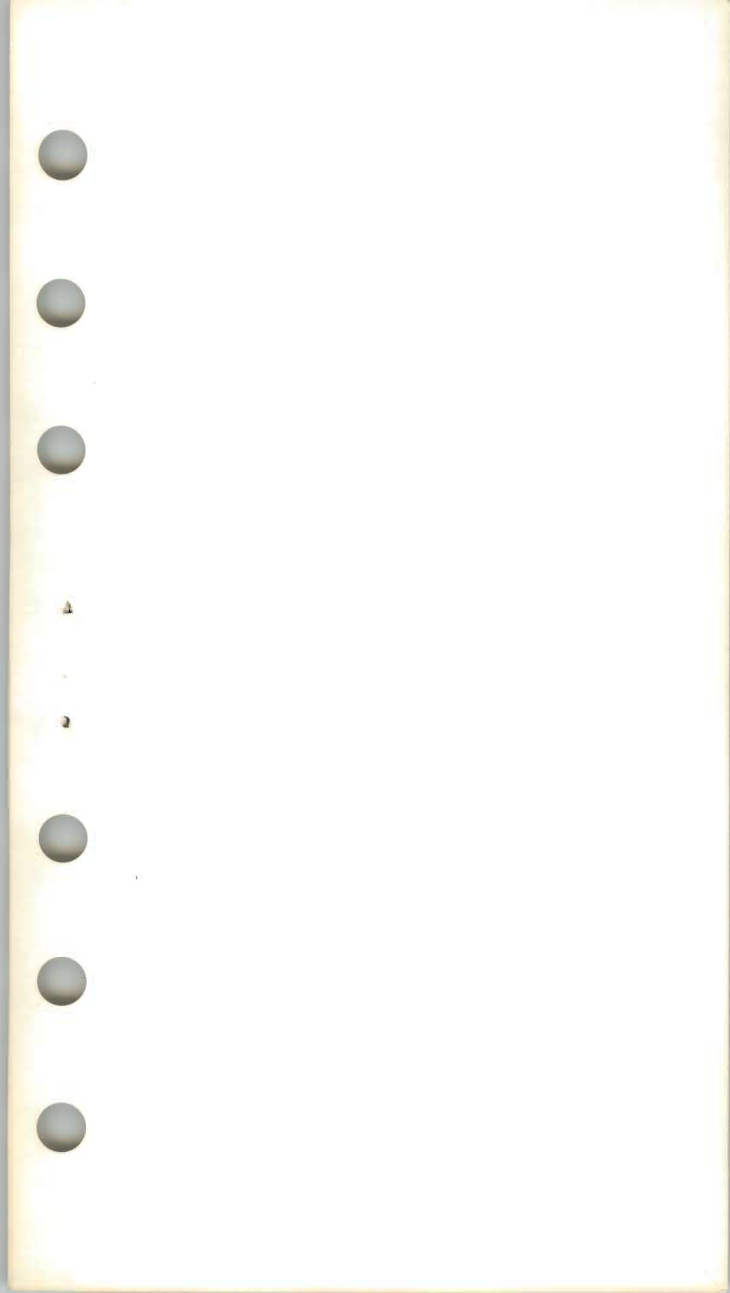


**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY

International Business Machines Corporation  
Development Laboratory  
Information Development, Department 532  
Rochester, Minnesota 55901







International Business Machines Corporation

IBM System/38 COBOL Reference Summary Printed in U.S.A. SC21-7781-4

SC21-7781-4

SC21-7781-04

