New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

<table>
<thead>
<tr>
<th>REVISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original printing.</td>
</tr>
<tr>
<td>(4-2-71)</td>
<td></td>
</tr>
</tbody>
</table>

Publication No. 60327600

Additional copies of this manual may be obtained from the nearest Control Data Corporation sales office. Address comments concerning this manual to:

CONTROL DATA CORPORATION
Software Documentation
215 Moffett Park Drive
Sunnyvale, California 94086

©1971
Control Data Corporation
Printed in the United States of America
CONTROL DATA 6400/6500/6600 COBOL

The COBOL language is designed to simplify the programming of business data processing operations; it produces easily modifiable source programs that result in shorter program development time and low program conversion costs. COBOL source and object programs run under the control of SCOPE Version 3.3.

COBOL for the CONTROL DATA® 6000 series is upwards compatible with ANSI COBOL and also with 3000 COBOL.

Special Features:

Mass Storage input and output including Indexed Sequential file processing

SORT verb sorts files within COBOL program

Automatic table search using index names and the SEARCH and SET statements

Report Writer to produce printed reports automatically, or report page may be produced by user with LINAGE clause and WRITE statement

Full arithmetic facility including:
   18-digit operands
   DIVIDE with REMAINDER
   COMPUTE with exponentiation
   CORRESPONDING option with ADD and SUBTRACT

Segmentation and overlay of object program

ENTER and CALL verbs, Common-Storage section provide communication with separately compiled COBOL programs and also with FORTRAN or COMPASS programs

COPY and INCLUDE provide access to COBOL source library

RERUN provides memory dumps with restart at specified checkpoints

Remote Interactive capability for remote terminal input/output
PROGRAM EFFICIENCY HINTS

To reduce key punching:
   Use abbreviations where permitted.
   Use PIC clause rather than SIZE, CLASS, USAGE clauses.

To increase compilation efficiency:
   Restrict data and paragraph names to 9 characters or less.
   Eliminate unnecessary paragraph names.
   Reduce forward references.

To increase execution efficiency:
   Use same size sending and receiving fields.
   Make table and item sizes a multiple of 10 characters.
   Reduce subscripting.
   Subscript with literals instead of variables.
   Use COMPUTATIONAL-1 items or index-names as subscripts.
   Use COMPUTATIONAL-1 items as arithmetic variables.
   Restrict arithmetic items to 9 digits or less.
   Use standard labels.
   Use SYNCHRONIZED RIGHT clause for data frequently referenced.
   Use SAME RECORD AREA to save moves; SAME AREA to save space.

COBOL NOTATION

[ ] Enclosed elements are optional.
{ } { } Only one element may be selected.
... Repeat preceding bracketed material as needed.
COBOL words have preassigned meanings and appear in capitals.
COBOL words not underlined may be omitted.
Terms in small letters are supplied by the programmer.
Punctuation and special characters are required where shown.
IDENTIFICATION DIVISION

PROGRAM-ID. program-name.
[AUTHOR. [comment-sentences.]]
[INSTALLATION. [comment-sentences.]]
[DATE-WRITTEN. [comment-sentences.]]
[DATE-Compiled. [current-date supplied by compiler.]]
[SECURITY. [comment-sentences.]]
[REMARKS. [comment-sentences.]]

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER.

format 1:

COPY library-name [REPLACING {literal-1
word-1
identifier-1} BY

{literal-2
word-2
identifier-2}

{literal-3
word-3
identifier-3} BY {literal-4
word-4
identifier-4} ] ... ].

format 2:

{6400}
{6500}
{6600}

OBJECT-COMPUTER.

format 1:

COPY library-name [REPLACING {literal-1
word-1
identifier-1} BY

{literal-2
word-2
identifier-2}

{literal-3
word-3
identifier-3} BY {literal-4
word-4
identifier-4} ] ... ].

format 2:

{6400}
{6500}
{6600} [SEGMENT-LIMIT IS priority-number].
SPECIAL-NAMES.

format 1:

COPY library-name [ REPLACING { literal-1 word-1 identifier-1 } BY

{ literal-2 word-2 identifier-2 } { literal-3 word-3 identifier-3 } BY { literal-4 word-4 identifier-4 } ] ... ].

format 2:

[ SWITCH integer-1

ON STATUS IS switch-status-name-1

[ OFF STATUS IS switch-status-name-2 ]

OFF STATUS IS switch-status-name-2

[ ON STATUS IS switch-name-1 ]

[ non-numeric-literal IS mnemonic-name-1 ] ...

[ implementor-name IS mnemonic-name-2 ] ...

[ CURRENCY SIGN IS literal ]

[ DECIMAL-POINT IS COMMA ]

[ CONSOLE IS mnemonic-name-3 ]

[ TERMINAL IS mnemonic-name-4 ].

INPUT-OUTPUT SECTION.

FILE-CONTROL.

format 1:

COPY library-name [ REPLACING { literal-1 word-1 identifier-1 } BY

{ literal-2 word-2 identifier-2 } { literal-3 word-3 identifier-3 } BY { literal-4 word-4 identifier-4 } ] ... ].

4
FORMAT 2:

SELECT [OPTIONAL] file-name-1 [RENAMEING file-name-2]

Assign TO implementor-name-1 [FOR MULTIPLE {REEL UNIT}]

[RESERVE {NO \{integer-1\} ALTERNATE \{AREA \{AREAS\}\}]

[FILE-LIMIT IS literal-1]

[ORGANIZATION IS \{INDEXED SEQUENTIAL\} \{STANDARD\}]

[ACCESS MODE IS \{SEQUENTIAL\} \{RANDOM\}]

(PROCESSING MODE IS SEQUENTIAL)

[\{ACTUAL SYMBOLIC\} KEY IS data-name-2]

[SELECT ...].

I-O-CONTROL:

FORMAT 1:

COPY library-name

[REPLACING \{\{literal-1 word-1 identifier-1\}\} BY \{\{literal-2 word-2 identifier-2\}\}]

\{\{literal-3 word-3 identifier-3\}\} BY \{\{literal-4 word-4 identifier-4\}\} ...]

FORMAT 2:

[RERUN [ON file-name-1] EVERY \{END OF REEL \{integer-1 RECORDS condition-name\}\}

OF file-name-2]

[SAME [SORT RECORD\{\{\} AREA FOR file-name-3,file-name-4

[\{\{\}\ file-name-5\}\ ...]\}

[MULTIPLE FILE TAPE CONTAINS file-name-6

[POSITION integer-2] \{\{\}\ file-name-7

[POSITION integer-3] \}...]}
PICTURE DESCRIPTION CODES

Data Characters
A  Alphabetic character
X  Alphanumeric character
9  Numeric character

Operation Symbols
S  Signed
V  Assumed decimal point location
P  Assumed decimal point scaling position

Replacement Characters
Z  Leading zeros replaced by blanks
*  Leading zeros replaced by *(check protection symbol)

Insertion Characters
$  Dollar sign; floating when more than one (dollar sign may be replaced by currency sign defined in SPECIAL-NAMES)
,  Comma
/  Slash (instead of comma)
.  Actual decimal point
B  Blank
0  Zero
-  Minus sign when item is negative, blank when positive; floating when more than one
+  Plus sign when item is positive, minus when negative; floating when more than one
CR Credit symbol when item is negative, blank when positive
DB Debit symbol when item is negative, blank when positive
# DATA SPECIFICATIONS

<table>
<thead>
<tr>
<th></th>
<th>File Section</th>
<th>Common and Working Storage Sections</th>
<th>Constant Section</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>01 group</td>
<td>77 group</td>
<td>77 group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01 group</td>
<td>01 group</td>
</tr>
<tr>
<td>REDEFINES</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZE</td>
<td>R R R R R R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USAGE</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCCURS</td>
<td>I I I I I I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT LOCATION</td>
<td>J I J I J I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGNED</td>
<td>J I J I J I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JUSTIFIED</td>
<td>J I J I J I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNCHRONIZED</td>
<td>J I J I J I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PICTURE</td>
<td>J I J I J I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editing Clauses</td>
<td>J I J I J I I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE</td>
<td>K K K C V V V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILLER</td>
<td>I I I I I I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **C**: Legal only in defining values for condition names
- **I**: Illegal
- **R**: Required if PICTURE is not used
- **blank**: Optional
- **V**: Required
- **J**: Legal only on elementary 01 items
- **K**: Documentary only
DATA DIVISION

[FILE SECTION]
[COMMON-STORAGE SECTION]
[WORKING-STORAGE SECTION]
[CONSTANT SECTION]
[REPORT SECTION]

File Description Entry (File Section Only)

format 1:

FD file-name COPY library-name [REPLACING {literal-1
word-1
identifier-1}]

{literal-2
word-2
identifier-2} [{litera-3
word-3
identifier-3}]

BY {literal-4
word-4
identifier-4} ... ]

format 2:

FD file-name

[RECORDING MODE IS {BINARY
DECIMAL}]

{HIGH
LOW
HYPER} DENSITY]

[FILE CONTAINS ABOUT integer-1 RECORDS]

[BLOCK CONTAINS [integer-2 TO] integer-3 {RECORDS
CHARACTERS}]

[RECORD CONTAINS [integer-4 TO] integer-5 CHARACTERS

[DEPENDING ON {RECORD-MARK}]

LABEL {RECORDS ARE
RECORD IS}

{STANDARD
OMMITED}

data-name-1

data-name-2]
If label records are STANDARD:

\[
\text{VALUE OF \{ID IDENTIFICATION\} IS \{literal-1 data-name-3\}}
\]
\[
\text{DATE-WRITTEN IS \{literal-2 data-name-4\}}
\]
\[
\text{EDITION-NUMBER IS \{literal-3 data-name-5\}}
\]
\[
\text{REEL-NUMBER IS \{literal-4 data-name-6\}}
\]
\[
\text{RETENTION-CYCLE IS \{literal-5 data-name-7\}}
\]

If label records are data-name-2:

\[
\text{VALUE OF ENDING-TAPE-LABEL-IDENTIFIER IS \{literal-6 data-name-8\}}
\]
\[
\text{LINAGE IS \{integer-6 identifier-1\} LINES}
\]
\[
\text{DATA \{RECORDS ARE\} data-name-9 [data-name-10] ...}
\]
\[
\text{\{REPORTS ARE\} report-name-1 [report-name-2] ...}
\]
\[
\text{SEQUENCED ON data-name-11 [data-name-12] ... .}
\]

Sort File Description Entry (File Section Only)

format 1:

SD file-name COPY library-name \[REPLACING \{literal-1 word-1 identifier-1\} BY \{literal-2 word-2 identifier-2\} \]
\[\{literal-3 word-3 identifier-3\} BY \{literal-4 word-4 identifier-4\} ... \].

format 2:

SD file-name

\[\text{FILE CONTAINS ABOUT integer-1 RECORDS}\]
\[\text{RECORD CONTAINS [integer-2 TO] integer-3 CHARACTERS}\]
\[\text{DATA \{RECORDS ARE\} data-name-1 [data-name-2] ... .}\]
Record Description Entry (File, Common-Storage, Working-Storage and Constant Sections)

format 1:

level-number data-name-1 [REDEFINES data-name-2] COPY data-name-3 [FROM LIBRARY].

format 2:

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

[CLASS IS] {ALPHABETIC  
    NUMERIC  
    ALPHANUMERIC  
    AN}

SIZE IS integer-1 [{CHARACTERS} {DIGITS}]

[USAGE IS]  
  COMP  
  COMPUTATIONAL  
  COMP-1  
  COMPUTATIONAL-1  
  COMP-2  
  COMPUTATIONAL-2  
  DISPLAY  
  INDEX

[OCCURS integer-1 [TO integer-2] TIMES  

[DEPENDING ON data-name-1]  

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

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

[SIGNED]  

[SIGN IS data-name-6]  

[POINT LOCATION IS {LEFT} {RIGHT} integer-5 PLACES]  

{JUSTIFIED} {RIGHT}
[SYNC SYNCHRONIZED} [LEFT} ]

[VALUE IS literal-1]

[ PIC PICTURE IS character-string ]

[ ZERO SUPPRESS CHECK PROTECT FLOAT DOLLAR SIGN FLOAT CURRENCY SIGN ]
[ LEAVING integer-6 PLACES ]

[ BWZ BLANK WHEN ZERO ]

format 3:

66 data-name-1 RENAMES data-name-2 [THRU data-name-3].

format 4:

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

Report Description Entry (Report Section only)

format 1:

RD report-name [WITH CODE mnemonic-name-1]

COPY library-name [REPLACING {literal-1 word-1 identifier-1} BY
{l literal-2 word-2 identifier-2} {l literal-3 word-3 identifier-3} BY {l literal-4 word-4 identifier-4} ...].

format 2:

RD report-name [WITH CODE mnemonic-name-1]

[ CONTROL IS CONTROLS ARE ]
[ FINAL data-name-1 [data-name-2] ... ]
[PAGE \{LIMIT IS \{integer-1 \{LINE \{LINES\} LIMITS ARE\}\}\}\]\{heading-2\} [FIRST DETAIL integer-3] [LAST DETAIL integer-4] [FOOTING integer-5]

Report Group Description Entry (Report Section only)

format 1:

01 [data-name-1] COPY data-name-2 [FROM LIBRARY]

REPLACING \{literal-1 \{word-1 \{identifier-1\}\}\} BY \{literal-2 \{word-2 \{identifier-2\}\}\}

\{literal-3 \{word-3 \{identifier-3\}\}\} BY \{literal-4 \{word-4 \{identifier-4\}\}\} ... .

format 2:

01 [data-name-1]

\[CLASS IS \{ALPHABETIC \{NUMERIC \{ALPHANUMERIC \{AN\}\}\}\}\]

LINE NUMBER IS \{integer-1 \{PLUS integer-2\}\} NEXT PAGE

NEXT GROUP IS \{integer-3 \{PLUS integer-4\}\} NEXT PAGE

SIZE IS integer-5 \{CHARACTERS\} \{DIGITS\}

[USAGE IS] DISPLAY

12
REPORT HEADING
RH
PAGE HEADING
PH
OVERFLOW HEADING
OH
{ CONTROL HEADING } { data-name-3 } { FINAL }
{ CH }
DETAIL
DE
{ CONTROL FOOTING } { data-name-4 } { FINAL }
{ CF }
OVERFLOW FOOTING
OV
PAGE FOOTING
PF
REPORT FOOTING
RF

Report Element Description (Report Section only)

level number [data-name-1]

[ CLASS IS ]
{ ALPHABETIC }
{ NUMERIC }
{ ALPHANUMERIC }
{ AN }

[COLUMN NUMBER IS integer-1]

[ ZERO SUPPRESS ]
{ CHECK PROTECT }
{ FLOAT DOLLAR SIGN }
{ FLOAT CURRENCY SIGN }
{ LEAVING integer-2 PLACES }

{ BLANK WHEN ZERO }
{ BWZ }

[GROUP INDICATE]

{ JUSTIFIED }
{ JUST }
{ RIGHT }

 LINE NUMBER IS { integer-3 } { PLUS integer-4 } { NEXT PAGE }

13
{PIC 
    {PICTURE} IS character-string}

{POINT LOCATION IS \{LEFT \RIGHT\} integer-5 PLACES}

{RESET ON \{data-name-2\} \FINAL}

{SIGNED}

{SIGN IS data-name-3}

{SIZE IS integer-6 \{CHARACTERS\} \DIGITS}

\{SOURCE IS \SELECTED data-name-4 \LINE-COUNTER \PAGE-COUNTER\}

\SUM data-name-5{data-name-6} \UPON data-name-7 \VALUE IS literal-1

{[USAGE IS] DISPLAY].

TYPE clause allowed if level 01

NEXT GROUP clause allowed if level 01
### Usage Specifications

<table>
<thead>
<tr>
<th>Element</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-name</td>
<td>30 characters, 5 levels of qualifications</td>
</tr>
<tr>
<td>elementary item/literal</td>
<td>255 characters/digits</td>
</tr>
<tr>
<td>PERFORM nesting</td>
<td>15 levels in separate overlays, no limit in main overlay</td>
</tr>
<tr>
<td>level numbers</td>
<td>01-49, 66, 77, 88, FD, RD, SD</td>
</tr>
<tr>
<td>OCCURS...DEPENDING ON</td>
<td>1 per record description</td>
</tr>
<tr>
<td>library copies</td>
<td>5 levels of nesting</td>
</tr>
<tr>
<td>ACCEPT items</td>
<td>80 characters; 40 characters from console</td>
</tr>
<tr>
<td>PICTURE clause</td>
<td>30 symbols</td>
</tr>
<tr>
<td>arithmetic operand</td>
<td>18 digits</td>
</tr>
<tr>
<td>GO TO statement</td>
<td>100 procedure names</td>
</tr>
<tr>
<td>ALTER statement</td>
<td>100 procedure names</td>
</tr>
<tr>
<td>DISPLAY items</td>
<td>no limit</td>
</tr>
<tr>
<td>ENTER parameters</td>
<td>no limit</td>
</tr>
<tr>
<td>Total files, I/O devices, and reports</td>
<td>53</td>
</tr>
<tr>
<td>Total procedure names</td>
<td>depends on field length</td>
</tr>
<tr>
<td>Total external references</td>
<td>depends on field length</td>
</tr>
</tbody>
</table>
# VALID MOVE OPERATIONS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Elem. Alpha</td>
<td></td>
<td>X AN TD AN AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Elem. BCD Num.</td>
<td></td>
<td>Conv. Bin. TD AN Num. AN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN†</td>
</tr>
<tr>
<td>Elem. AN</td>
<td></td>
<td>X TD AN Num. AN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Elem. Edit Num.</td>
<td></td>
<td>X TD AN X AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Elem. Edit AN</td>
<td></td>
<td>X TD AN X AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Group AN</td>
<td></td>
<td>TD AN TD AN TD AN AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Group Binary &amp; Mixed</td>
<td></td>
<td>TD AN TD AN TD AN TD AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TD AN</td>
</tr>
<tr>
<td>Zero</td>
<td></td>
<td>Num. Bin. X Num. AN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Literal &amp; Fig. Cons. AN</td>
<td></td>
<td>X TD AN X AN X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
<tr>
<td>Literal Num.</td>
<td></td>
<td>Conv. Bin. X Num. AN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AN</td>
</tr>
</tbody>
</table>

† Valid only when source is integer; others PD.

Any move to a binary or mixed group is treated as an alphanumeric move; a precautionary diagnostic is issued.

A move to a figurative constant or literal is illegal.

- X: Illegal
- AN: Alphanumeric
- AN-Edit: Alphanumeric edited
- Conv.: Conversion prior to move
- Edit: Numeric edited
- Num.: Numeric
- Num. Bin.: Numeric binary
- TD: Trivial diagnostic issued
PROCEDURE DIVISION

DECLARATIVES.
Section-name SECTION. declarative-sentence.
Paragraph-name. sentence-1 [sentence-2] ...
END DECLARATIVES.

ACCEPT identifier-1 [FROM mnemonic-name-1]

ADD \{identifier-1\} \{identifier-2\} \{identifier-3\} \{identifier-n [ROUNDED]

[ON SIZE ERROR imperative-statement]

ADD \{identifier-1\} \{identifier-2\} \{identifier-3\} \{identifier-n [ROUNDED]

[ON SIZE ERROR imperative-statement]

ADD \{identifier-1\} \{identifier-2\} \{identifier-3\} \{identifier-n [ROUNDED]

[ON SIZE ERROR imperative-statement]

ADD CORR CORRESPONDING identifier-1 TO identifier-2 [ROUNDED]

[identifier-3 [ROUNDED] ] ...

[ON SIZE ERROR imperative-statement]

ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2

[procedure-name-3 TO [PROCEED TO] procedure-name-4] ...

CLOSE file-name-1 \{UNIT REEL\} \{UNIT REEL\} [WITH NO REWIND LOCK]

file-name-2 \{UNIT REEL\} \{UNIT REEL\} [WITH NO REWIND LOCK]

...
COMPUTE identifier-1 [ROUNDED] [identifier-2 [ROUNDED]] ... 

{ FROM } { literal 
= arithmetic-expression 
EQUALS } identifier-3 

[ON SIZE ERROR imperative-statement] 

{ COPY } { library-name [ REPLACING } literal-1 { word-1 } 
{ INCLUDE } { } { identifier-2 } BY { literal-3 
{ } word-2 { word-3 } 
{ } identifier-3 } BY { literal-4 
{ } word-4 { identifier-4 } } ... 

DISPLAY { identifier-1 } { identifier-2 } { identifier-2 } ... 

[UPON mnemonic-name] 

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

[identifier-3 [ROUNDED]] ... 

[ON SIZE ERROR imperative-statement] 

DIVIDE { identifier-1 } { BY } { identifier-2 } GIVING identifier-3 

[ROUNDED] [identifier-4 [ROUNDED]] ... 

[ON SIZE ERROR imperative-statement] 

DIVIDE { identifier-1 } { BY } { identifier-2 } GIVING identifier-3 [ROUNDED] 

REMAINDER identifier-4 

[ON SIZE ERROR imperative-statement] 

ENTER COBOL. 

ENTER LINKAGE. 

{ ENTER } { language-name } routine-name 

[USING parameter-list] .
EXAMINE identifier-1

TALLYING

{ ALL
  LEADING
  (UNTIL FIRST)
  literal-1 [REPLACING BY
  literal-2]
}

REPLACING

{ ALL
  LEADING
  (UNTIL) FIRST
}

literal-3 BY literal-4

EXIT.

{EXIT PROGRAM.
  RETURN.
}

GENERATE identifier-1

GO TO [procedure-name-1]

GO TO procedure-name-2 [procedure-name-3 ...]

DEPENDING ON identifier-1

IF conditional-expression [THEN]

{ NEXT SENTENCE
  statement-1
}

[THEN]

{OTHERWISE
  ELSE
  NEXT SENTENCE
  statement-2
}

Conditional expressions include:

GREATERTHAN

GR

>

LESSTHAN

LS

<

IS [NOT]

GREATER-EQUAL TO

GO

LESS-EQUAL TO

LO

EQUAL TO

EQ

= identifier-1

{ literal-1
  (formula-1)
}

identifier-2

{ literal-2
  (formula-2)

IS UNEQUAL TO

EQUALS

EXCEEDS

IS NO

IS NGR

IS NLS

19
\{identifier-3\} IS NOT \{POSITIVE\} \{NEGATIVE\} \{ZERO\} 

identifier-4 IS NOT \{NUMERIC\} \{ALPHABETIC\}

\{NOT\} \{condition-name\} \{switch-status-name\}

INITIATE \{report-name-1 [report-name-2] ... \} \{ALL\}

\{CORR\} \{CORRESPONDING\} \{identifier-2\} 

\{literal-1\} \{identifier-1\} \{TO\} 

identifier-3 \{identifier-4\} ...

MULTIPLY \{identifier-1\} \{literal-1\} \{BY\} \{identifier-2\} \{ROUNDED\} 

\{identifier-3 \{ROUNDED\}\} ...

[ON SIZE ERROR imperative-statement]

MULTIPLY \{identifier-1\} \{literal-1\} \{BY\} \{identifier-2\} \{literal-2\} 

GIVING \{identifier-3 \{ROUNDED\}\} \{identifier-4 \{ROUNDED\}\} ...

[ON SIZE ERROR imperative-statement]

NOTE character-string.

\{INPUT\} \{file-name-1\} \{REVERSED\} \{WITH NO REWIND\} \{cost\} 

\{file-name-2\} \{REVERSED\} \{WITH NO REWIND\} ...

\{OPEN\} \{OUTPUT\} \{file-name-3\} \{WITH NO REWIND\} 

\{file-name-4\} \{WITH NO REWIND\} ...

\{INPUT-OUTPUT\} \{I-O\} \{file-name-5\} \{file-name-6\} ...

PERFORM \{procedure-name-1\}[THRU \{procedure-name-2\}]

20
PERFORM procedure-name-1 [THRU] procedure-name-2

{identifier-1} TIMES

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

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

VARYING {index-name-1} FROM {index-name-2} BY

{l literal-2} UNTIL condition-2

{identifier-3} FROM {identifier-4} BY {identifier-5}

{identifier-6} UNTIL condition-3

{identifier-7} FROM {identifier-8} BY

{l literal-6} UNTIL condition-4

READ file-name-1 RECORD [INTO identifier-1] AT END

imperative-statement

READ file-name-1 RECORD [INTO identifier-2] INVALID KEY

imperative-statement

RELEASE record-name-1 [FROM identifier-1]

RETURN file-name-1 RECORD [INTO identifier-1] AT END

imperative-statement

SEARCH identifier-1 [VARYING {index-name-1}]

{identifier-2} [AT END imperative-statement-1]

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

{imperative-statement-3} NEXT SENTENCE
SEARCH ALL identifier-1 [AT END imperative-statement-1]

WHEN condition-1

{imperative-statement-2}

NEXT SENTENCE

SEEK file-name-1 RECORD [WITH KEY CONVERSION]

SET \{index-name-1\} \{identifier-1\} \{index-2\} \{identifier-2\} ... \{index-name-3\} \{identifier-3\} \{literal-1\}

SET index-name-1 [index-name-2] ...

\{UP BY \{identifier-1\} \{literal-1\} \{DOWN BY \}

SORT file-name-1 ON \{DESCENDING \{ASCENDING \{KEY identifier-1[identifier-2]...

\[ON \{DESCENDING \{ASCENDING \{KEY identifier-3 [identifier-4] ...

\}\}\}\}\]\}

\{INPUT PROCEDURE IS section-name-1[THRU section-name-2]\}

\{USING file-name-2\}

\{OUTPUT PROCEDURE IS section-name-3[THRU section-name-4]\}

\{GIVING file-name-3\}

STOP \{literal\} .

RUN

SUBTRACT \{identifier-1\} \{literal-1\} \{identifier-2\} \{literal-2\} ... FROM identifier-m

[ROUNDED] \{identifier-n [ROUNDED]\} ...

[ON SIZE ERROR imperative-statement]

SUBTRACT \{identifier-1\} \{literal-1\} \{identifier-2\} \{literal-2\} ... FROM \{identifier-m\} \{literal-m\}

GIVING identifier-n [ROUNDED]

\{identifier-o [ROUNDED]\} ...

[ON SIZE ERROR imperative-statement]
SUBTRACT \{CORR CORRESPONDING\} identifier-1 FROM
identifier-2 \{ROUNDED\} [identifier-3 \{ROUNDED\} ...]
[ON SIZE ERROR imperative-statement].

TERMINATE \{report-name-1 [report-name-2] ...\}
\{ALL\}

USE AFTER STANDARD ERROR PROCEDURE ON
\{file-name\}
\{INPUT\}
\{OUTPUT\}
\{INPUT-OUTPUT\}
\{I-O\}

USE \{BEFORE\} \{AFTER\} STANDARD \{BEGINNING\}
\{ENDING\} \{REEL\} \{FILE\}

LABEL \{PROCEDURE\} \{PROCEDURES\} ON
\{file-name\}
\{INPUT\}
\{OUTPUT\}
\{INPUT-OUTPUT\}
\{I-O\}

USE BEFORE REPORTING identifier-1 [identifier-2] ...

USE FOR KEY CONVERSION ON \{ALL\}
\{file-name-1 [file-name-2] ...\}

WRITE record-name-1 \{FROM identifier-1\}

\{BEFORE\} \{AFTER\} ADVANCING \{identifier-2 LINES\}
\{integer-1 LINES\}
\{mnemonic-name-1\}

\{AT END-OF-PAGE\}
\{EOP\}
imperative-statement

WRITE record-name-2 \{FROM identifier-2\} INVALID KEY
imperative-statement
COBOL CONTROL CARD

Ten parameters are used to select compilation options. All are optional and may be specified in any order.

\{(COBOL, \
\{COBOL \{p1,p2,p3,p4,p5,p6,p7,p8,p9,p10\}\}) \} [comments]

p1 (Source Input) \{absent \} \{I \} INPUT assumed
I = INPUT
I = fn source input on file fn

p2 (Binary Output) \{absent \} \{B \} relocatable binary file on file
B = LGO LGO
B = fn binary output on file fn
B = 0 suppress binary output

p3 (List) \{absent \} \{L \} normal listing on OUTPUT
LX extended diagnostics
LR cross reference pointers
LC copy from library
LO object code in octal
LM data map
L = fn output on file fn
L = 0 suppress list output

p4 (Source Library) \{absent \} \{S \} source library from file COLIB
S = COLIB
S = fn from file fn

p5 (Subcompile) SUB suppresses all data division binary output except from working and constant storage

p6 (Overlay Binary) \{OB \} \{OB = LGO2 \} binary output on LGO2
OB = fn binary output from overlay segments put on file fn
p7 (EDITLIB) E = fn add object code to system library using EDITLIB

p8 (ASCII Collating) U use ASCII collating sequence

p9 (Tape Sort) T sort requests tape sort

p10 (BCOMMON) H BCOMMON replaces blank common as buffer area
## COBOL CODING FORMAT

<table>
<thead>
<tr>
<th>Column</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 6</td>
<td>Sequence number</td>
</tr>
<tr>
<td>7</td>
<td>Hyphen for continuation of words and/or literals</td>
</tr>
<tr>
<td>8</td>
<td>Division name</td>
</tr>
<tr>
<td></td>
<td>Section name</td>
</tr>
<tr>
<td></td>
<td>Paragraph name</td>
</tr>
<tr>
<td></td>
<td>File description</td>
</tr>
<tr>
<td>12</td>
<td>Record description level number</td>
</tr>
<tr>
<td></td>
<td>Record description data name</td>
</tr>
<tr>
<td></td>
<td>First sentence of a paragraph</td>
</tr>
<tr>
<td></td>
<td>File name</td>
</tr>
<tr>
<td></td>
<td>Continuation of a data description or a sentence</td>
</tr>
<tr>
<td>73 - 80</td>
<td>Identification, optional</td>
</tr>
</tbody>
</table>

**Sequence number**: Optional, checked by the processor if used.

**Hyphen**: Indicates continuation of a word from the preceding line.

**Division name**: Terminated by period, remainder of line is blank.

**Section name**: Followed by optional priority number, terminated by period, remainder is blank.

**Paragraph name**: Terminated by period, and followed by at least one blank before text begins.

**File Description**: FD or SD followed by file name and at least one blank.

**Record Description**: Level number followed by at least one blank and data name.

**First Sentence**: Begins in or after column 12. Spaces may be used freely to avoid splitting a word. If a word is split, a hyphen must appear in column 7 of the next line.
COBOL SOURCE DECKS

PROCEDURE DIVISION.

DATA DIVISION.

ENVIRONMENT DIVISION.

IDENTIFICATION DIVISION.
COBOL COMPILATION

(end of job)

COBOL source deck

(end of record)

COBOL.

JCJ001,T264,CM55000.
EXECUTION

6 7 8 9 (end of job)

7 8 9
data deck (end of record)

7 8 9
COBOL source deck (end of record)

7 8 9
LGO.

COBOL(LX)

JCJ001,T264,CM55000.
EXECUTION WITH SEGMENTATION

(empty record = end of loader information)

Data Deck

Binary Subprogram

C

Binary Subprogram

B

COBOL Source Deck

LGO.

LOAD (INPUT)

COBOL

(job card)
LIBRARY UPDATE

6 7 8 9
(end of job)
data deck
(end of record)
COBOL source deck
(end of record)
*EDIT, CODECK1
*DELETE, 21, 36
insert cards to COBOL source library
*INSERT, 20
(end of record)
LGO.
COBOL(S=SCRLIB, B=O, LOC)
COPYCL(OPL=COBB, C=SCRLIB)
EDITSYM(OPL=LIBA, NPL=COBB, L=COLIST)
JLB, T128, CM55000.
COBOL RESERVED WORD LIST

*indicates word not implemented in 6000 COBOL.

ABOUT
ACCEPT
ACCESS
ACTUAL
ADD
*ADDRESS
ADVANCING
AFTER
ALL
ALPHABETIC
ALPHANUMERIC
ALTER
ALTERNATE
AN
AND
*APPLY
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUTHOR
BEFORE
BEGINNING
BEGINNING-FILE-LABEL
BEGINNING-TAPE-LABEL
BINARY
*BITS
BLANK
BLOCK
BWZ
BY

CALL
CF
CH
CHARACTER
CHARACTERS
CHECK
CLASS
*CLOCK-UNITS
CLOSE
COBOL

CODE
COLUMN
COMMA
COMMON-STORAGE
COMP
COMP-1
COMP-2
COMPASS
COMPUTATIONAL
COMPUTATIONAL-1
COMPUTATIONAL-2
COMPUTE
CONFIGURATION
CONSOLE
CONSTANT
CONTAINS
CONTROL
CONTROLS
*CONVERSION
COPY
CORR
CORRESPONDING
CURRENCY

DATA
DATE-COMPiled
DATE-WRITTEN
DE
DECIMAL
DECIMAL-POINT
DECLARATIVES
*DEFINE
DENSITY
DEPENDING
DESCENDING
DETAIL
DIGIT
DIGITS
DISPLAY
DIVIDE
DIVided
DIVISION
DOLLAR
*DOWN
EDITION-NUMBER
ELSE
END
END-OF-PAGE
ENDING
ENDING-FILE-LABEL
ENDING-TAPE-LABEL
ENDING-TAPE-LABEL-IDENTIFIER
ENTER
ENTRY
ENVIRONMENT
EOP
EQ
EQUAL
EQUALS
ERROR
EVERY
EXAMINE
EXCEEDS
EXIT
EXPONENTIATED
FD
FILE
FILE-CONTROL
FILE-LABEL
FILE-LIMIT
FILE-LIMITS
FILLER
FINAL
FIRST
FLOAT
FOOTING
FOR
*FORMAT
FORTRAN-R
FORTRAN-X
FROM

GENERATE
GIVING
GO
GQ
GR
GREATER
GREATER-EQUAL
GROUP

*HASHED
HEADING
HIGH
HIGH-VALUE
HIGH-VALUES
*HOLD
HYPER
ID
IDENTIFICATION
IF
IN
INCLUDE
INDEX
INDEXED
INDICATE
INITIATE
INPUT
INPUT-OUTPUT
INSTALLATION
INTO
INVALID
I-O
I-O-CONTROL
IS

JUST
JUSTIFIED

KEY
*KEYS
LABEL
LAST
LEADING
LEAVING
LEFT
LESS
LESS-EQUAL
LIBRARY
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
REPORTING  SUPPRESS
REPORTS  SWITCH
RERUN  SYMBOLIC
RESERVE  SYNC
RESET  SYNCHRONIZED
RETENTION  TALLY
RETENTION-CYCLE  TALLYING
RETURN  TAPE
REVERSED  TAPE-LABEL
REWIND  TERMINAL
RF  TERMINATE
RH  THAN
RIGHT  THEN
ROUNDED  THROUGH
RUN  THRU

*SA  TIMES
SAME  TO
SD  TODAYS-DATE
SEARCH  TYPE
SECTION  UNEQUAL
SECURITY  UNIT
SEEK  UNTIL
SEGMENT-LIMIT  *UP
SELECT  UPON
SELECTED  *UPPER-BOUND
SENTENCE  *UPPER-BOUNDS
SEQUENCED  USAGE
SEQUENTIAL  USE
SET  USING
SIGN  VALUE
SIGNED  VALUES
SIZE  VARYING
SORT
SOURCE  WHEN
SOURCE-COMPUTER  WITH
SPACE  *WORDS
SPACES  WORKING-STORAGE
SPECIAL-_NAMES  WRITE
STANDARD
STATUS
STOP
SUBTRACT
SUM
*SUPERVISOR

ZERO
ZEROES
ZEROS
## COLLATING SEQUENCE

<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Δ</td>
<td>55</td>
<td>space</td>
</tr>
<tr>
<td>01</td>
<td>≤*</td>
<td>74</td>
<td>8-5</td>
</tr>
<tr>
<td>02</td>
<td>[*</td>
<td>61</td>
<td>8-7</td>
</tr>
<tr>
<td>03</td>
<td>→*</td>
<td>65</td>
<td>0-8-5</td>
</tr>
<tr>
<td>04</td>
<td>≡*</td>
<td>60</td>
<td>0-8-6</td>
</tr>
<tr>
<td>05</td>
<td>∧*</td>
<td>67</td>
<td>0-8-7</td>
</tr>
<tr>
<td>06</td>
<td>↑*</td>
<td>70</td>
<td>11-8-5</td>
</tr>
<tr>
<td>07</td>
<td>↓*</td>
<td>71</td>
<td>11-8-6</td>
</tr>
<tr>
<td>08</td>
<td>&gt;</td>
<td>73</td>
<td>11-8-7</td>
</tr>
<tr>
<td>09</td>
<td>≥*</td>
<td>75</td>
<td>12-8-5</td>
</tr>
<tr>
<td>10</td>
<td>←*</td>
<td>76</td>
<td>12-8-6</td>
</tr>
<tr>
<td>11</td>
<td>.</td>
<td>57</td>
<td>12-8-3</td>
</tr>
<tr>
<td>12</td>
<td>)</td>
<td>52</td>
<td>12-8-4</td>
</tr>
<tr>
<td>13</td>
<td>:</td>
<td>77</td>
<td>12-8-7</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>45</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>$</td>
<td>53</td>
<td>11-8-3</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>47</td>
<td>11-8-4</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>46</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>/</td>
<td>50</td>
<td>0-1</td>
</tr>
<tr>
<td>19</td>
<td>,</td>
<td>56</td>
<td>0-8-3</td>
</tr>
<tr>
<td>20</td>
<td>(</td>
<td>51</td>
<td>0-8-4</td>
</tr>
<tr>
<td>21</td>
<td>=</td>
<td>54</td>
<td>8-3</td>
</tr>
<tr>
<td>22</td>
<td>≠†</td>
<td>64</td>
<td>8-4</td>
</tr>
<tr>
<td>23</td>
<td>&lt;</td>
<td>72</td>
<td>12-0</td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>01</td>
<td>12-1</td>
</tr>
<tr>
<td>25</td>
<td>B</td>
<td>02</td>
<td>12-2</td>
</tr>
<tr>
<td>26</td>
<td>C</td>
<td>03</td>
<td>12-3</td>
</tr>
<tr>
<td>27</td>
<td>D</td>
<td>04</td>
<td>12-4</td>
</tr>
<tr>
<td>28</td>
<td>E</td>
<td>05</td>
<td>12-5</td>
</tr>
<tr>
<td>29</td>
<td>F</td>
<td>06</td>
<td>12-6</td>
</tr>
<tr>
<td>30</td>
<td>G</td>
<td>07</td>
<td>12-7</td>
</tr>
<tr>
<td>31</td>
<td>H</td>
<td>10</td>
<td>12-8</td>
</tr>
</tbody>
</table>

*Not in COBOL character set; may be present in data
†COBOL quote character ("'') is output on printer as #
<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>I</td>
<td>11</td>
<td>12-9</td>
</tr>
<tr>
<td>33</td>
<td>V</td>
<td>66</td>
<td>11-0</td>
</tr>
<tr>
<td>34</td>
<td>J</td>
<td>12</td>
<td>11-1</td>
</tr>
<tr>
<td>35</td>
<td>K</td>
<td>13</td>
<td>11-2</td>
</tr>
<tr>
<td>36</td>
<td>L</td>
<td>14</td>
<td>11-3</td>
</tr>
<tr>
<td>37</td>
<td>M</td>
<td>15</td>
<td>11-4</td>
</tr>
<tr>
<td>38</td>
<td>N</td>
<td>16</td>
<td>11-5</td>
</tr>
<tr>
<td>39</td>
<td>O</td>
<td>17</td>
<td>11-6</td>
</tr>
<tr>
<td>40</td>
<td>P</td>
<td>20</td>
<td>11-7</td>
</tr>
<tr>
<td>41</td>
<td>Q</td>
<td>21</td>
<td>11-8</td>
</tr>
<tr>
<td>42</td>
<td>R</td>
<td>22</td>
<td>11-9</td>
</tr>
<tr>
<td>43</td>
<td>j††</td>
<td>62</td>
<td>0-8-2</td>
</tr>
<tr>
<td>44</td>
<td>S</td>
<td>23</td>
<td>0-2</td>
</tr>
<tr>
<td>45</td>
<td>T</td>
<td>24</td>
<td>0-3</td>
</tr>
<tr>
<td>46</td>
<td>U</td>
<td>25</td>
<td>0-4</td>
</tr>
<tr>
<td>47</td>
<td>V</td>
<td>26</td>
<td>0-5</td>
</tr>
<tr>
<td>48</td>
<td>W</td>
<td>27</td>
<td>0-6</td>
</tr>
<tr>
<td>49</td>
<td>X</td>
<td>30</td>
<td>0-7</td>
</tr>
<tr>
<td>50</td>
<td>Y</td>
<td>31</td>
<td>0-8</td>
</tr>
<tr>
<td>51</td>
<td>Z</td>
<td>32</td>
<td>0-9</td>
</tr>
<tr>
<td>52</td>
<td>:*</td>
<td>63</td>
<td>8-2</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>1</td>
<td>34</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>35</td>
<td>2</td>
</tr>
<tr>
<td>56</td>
<td>3</td>
<td>36</td>
<td>3</td>
</tr>
<tr>
<td>57</td>
<td>4</td>
<td>37</td>
<td>4</td>
</tr>
<tr>
<td>58</td>
<td>5</td>
<td>40</td>
<td>5</td>
</tr>
<tr>
<td>59</td>
<td>6</td>
<td>41</td>
<td>6</td>
</tr>
<tr>
<td>60</td>
<td>7</td>
<td>42</td>
<td>7</td>
</tr>
<tr>
<td>61</td>
<td>8</td>
<td>43</td>
<td>8</td>
</tr>
<tr>
<td>62</td>
<td>9</td>
<td>44</td>
<td>9</td>
</tr>
</tbody>
</table>

††COBOL record mark