COBOL-81

Pocket Guide

digital software

COBOL-81
Pocket Guide

Order No. AV-H630C-TC

May 1983

This guide contains quick-reference information about the COBOL-81 language and its interface with your operating system.

OPERATING SYSTEM AND VERSION	RSTS/E	V8
	RSX-11M	V4
	RSX-11M-PLUS	V2
SOFTWARE VERSION	COBOL-81	V2

digital equipment corporation, maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital™	DECwriter	RSTS
	DIBOL	RSX
DEC	MASSBUS	UNIBUS
DECmate	PDP	VAX
DECsystem-10	P/OS	VMS
DECSYSTEM-20	Professional	VT
DECUS	Rainbow	Work Processor

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726(Ottawa-Hull)
800-267-6146 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*
Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (CANADA)
Digital Equipment Corporation of Canada Ltd.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: A&SG Business Manager

DIRECT MAIL ORDERS (INTERNATIONAL)
Digital Equipment Corporation
A&SG Business Manager
c/o Digital's local subsidiary or
approved distributor

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575)

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

Contents

Conventions Used in this Manual

[ ]	Brackets enclose an optional part of a general format. When they enclose vertically stacked entries, brackets indicate that only one entry can be selected.
{ }	Braces indicate that you must select one (but no more than one) of the enclosed entries.
{| |}	Choice indicators allow you to select one or more of the enclosed entries. However, none can be used more than once.
...	Ellipses allow repetition of a part of the format.
.	Periods are required where shown in the format.
PICTURE IS char-string	Underlined uppercase words are key or required words; for example, PICTURE.
PICTURE IS char-string	Uppercase words not underlined are optional words; for example, IS.
PICTURE IS char-string	Lowercase words are generic terms supplied by the programmer; for example, *char-string*.

+ – < > = * ** / \	These special-character words are not underlined in the general formats, but are required where they appear.

This book lists only DIGITAL Command Language (DCL) command lines and qualifiers. The COBOL-81 User's Guide for your system (Appendix D) contains information on MCR (or CCL) commands and options.

Logging On to Your Operating System

Type HELLO. The system will prompt you for your name and password:

On RSTS/E:

```
Username:  { enter your username }
Password:  { enter your password }
```

On RSX-11M/M-PLUS:

```
Name or Account:  { enter your username }
Password:         { enter your password }
```

Logging Off of Your Operating System

```
LO[GOUT]
```

Getting Help from Your Operating System

HELP [subject]

Example

To get help for the COBOL-81 compiler, enter:

```
HELP  COBOL
```

File Specification

A fully qualified file specification, or file-spec, lets your COBOL program uniquely identify a file or device. The file-spec format for your system is one of the following.

- On RSTS/E:

  device:[directory]filename.type

- On RSX-11M/M-PLUS:

  device:[directory]filename.type;ver

where:

device is a unique hardware device name consisting of two alphabetic characters followed by a unit number.

directory is a file that contains the identification and location of your files. You must enclose directory names in square brackets ([ ]).

filename is a field that, combined with the file type and version number, identifies files in directories. The length of filename depends on your operating system:

- For RSTS/E, filename can be from one to six characters.
- For RSX-11M/M-PLUS, filename can be from one to nine characters.

type identifies a file by its contents. It can be from zero to three characters long. For example, the file type of an executable image (or task image) is usually TSK, and the file type for a COBOL-81 source program is CBL.

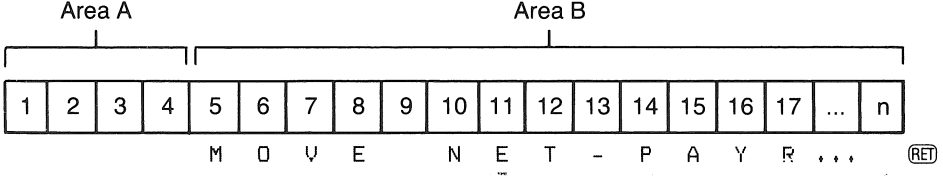
ver (if applicable) is a number assigned to different versions of the same file. If you duplicate a file name and type in the same directory, the system increments the version number by one.

COBOL Coding Format

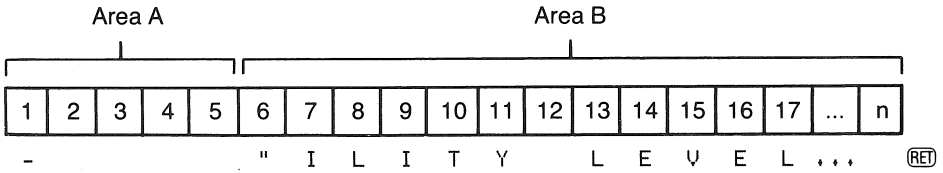
You can code COBOL-81 source programs using either terminal or ANSI format.

Terminal Format

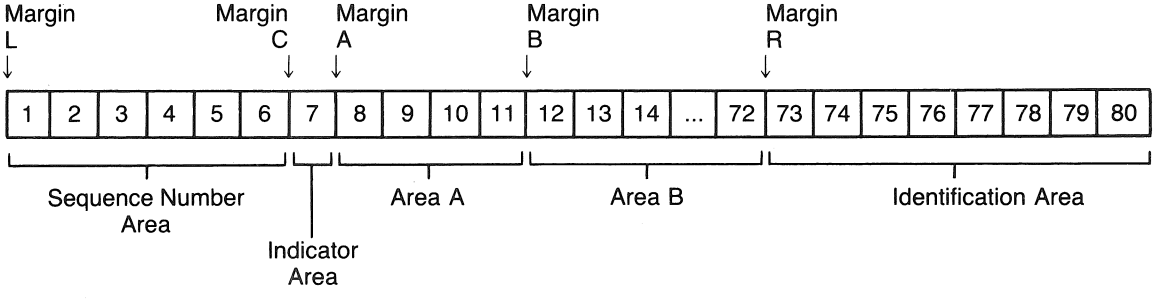
Without Indicator Character



With Indicator Character



ANSI Format



Format Legend:

Margin L	Immediately to the left of the leftmost character position.
Margin C	Between character positions 6 and 7.

(continued on next page)

Margin A	Between character positions 7 and 8.
Margin B	Between character positions 11 and 12.
Margin R	Between character positions 72 and 73.
Sequence Number Area	The six character positions between Margin L and Margin C. The contents can be any character(s) from the computer character set. The compiler does not check the uniqueness of the contents.
Indicator Area	The character in this position directs the compiler to interpret the source line in one of the following ways:

Character	Source Line Interpretation
space ( )	Default (ANSI format only). The compiler processes the line as normal COBOL text.
hyphen (-)	Continuation line. The compiler processes the line as a continuation of the previous source line.

asterisk (*) Comment line. The compiler ignores the contents of the line. However, the source line appears on the program listing.

slash (/) New listing page. The compiler treats the line as a comment line. However, it advances the program listing to the top of the next page before printing the line.

Area A Area A contains division headers, section headers, paragraph headers, paragraph-names, level indicators, and certain level-numbers.

Area B Area B contains all other COBOL text.

Identification Area The eight character positions immediately following Margin R. The compiler ignores the contents of the identification area. However, the contents appear on the source program listing.

n Can be a maximum of 200 characters. Only the first 125 characters are displayed on the listing.

Creating a COBOL Source Program

```
EDIT file-spec
    or
CREATE file-spec
```

Compiling, Linking, and Running a COBOL Program

```
COBOL[/qualifiers] file-spec[/qualifiers]...

LINK/C81[/qualifiers] file-spec[,...] [/qualifiers]...

RUN task-file
```

COBOL Command Qualifiers

Qualifier	Default
/[NO]ANSI_FORMAT	/NOANSI_FORMAT
/[NO]CHECK[:option]	/CHECK
:[NO]BOUNDS	
:[NO]PERFORM	

/CODE:[NO]CIS	
/[NO]CROSS_REFERENCE	/NOCROSS_REFERENCE
/[NO]DEBUG	/NODEBUG
/[NO]DIAGNOSTICS	/NODIAGNOSTICS
/[NO]LIST[=file-spec]	/NOLIST
/[NO]OBJECT[=file-spec]	/OBJECT
/[NO]SHOW[:option]	/NOSHOW, /SHOW:NOMAP
:[NO]MAP	
/[NO]SUBPROGRAM	/NOSUBPROGRAM
/[NO]TRUNCATE	/NOTRUNCATE
*/[NO]WARNINGS[:option]	/WARNINGS,
:[NO]INFORMATIONAL	/WARNINGS:INFORMATIONAL

* Only informational (I) diagnostics can be suppressed.

LINK/C81 Qualifiers

Library Qualifier	Default	
*/[NO]FMS[:option] :[NO]RESIDENT	/NOFMS	* [NO]RESIDENT option available only on RSTS/E systems
/OTS:[NO]RESIDENT	RSTS/E: /OTS:RESIDENT,if installed. Otherwise, /OTS:NORESIDENT RSX-11M/M-PLUS: /OTS:NORESIDENT	
/RMS:[NO]RESIDENT	RSTS/E: /RMS:RESIDENT, if installed. Otherwise, /RMS:NORESIDENT RSX-11M/M-PLUS: /RMS:NORESIDENT	

Output File Qualifier	Default	
/[NO]DEBUG	/NODEBUG	
**/[NO]MAP[=file-spec]	/NOMAP	** File-spec option available only on RSTS/E systems

COBOL-81 Source Program General Format

identification-division

[environment-division]

[data-division]

[procedure-division]

General Format of COBOL Programs

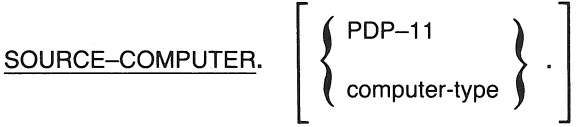
Identification Division Format

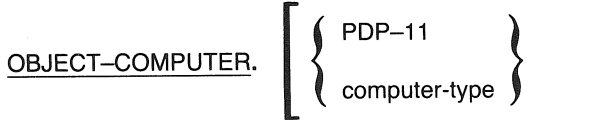
```
IDENTIFICATION DIVISION.
PROGRAM-ID. program-name.
[ AUTHOR. [ comment-entry ] ... ]
[ INSTALLATION. [ comment-entry ] ... ]
[ DATE-WRITTEN. [ comment-entry ] ... ]
[ DATE-COMPILED. [ comment-entry ] ... ]
[ SECURITY. [ comment-entry ] ... ]
```

Environment Division Format

```
[ 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. ] ] ] ]
```

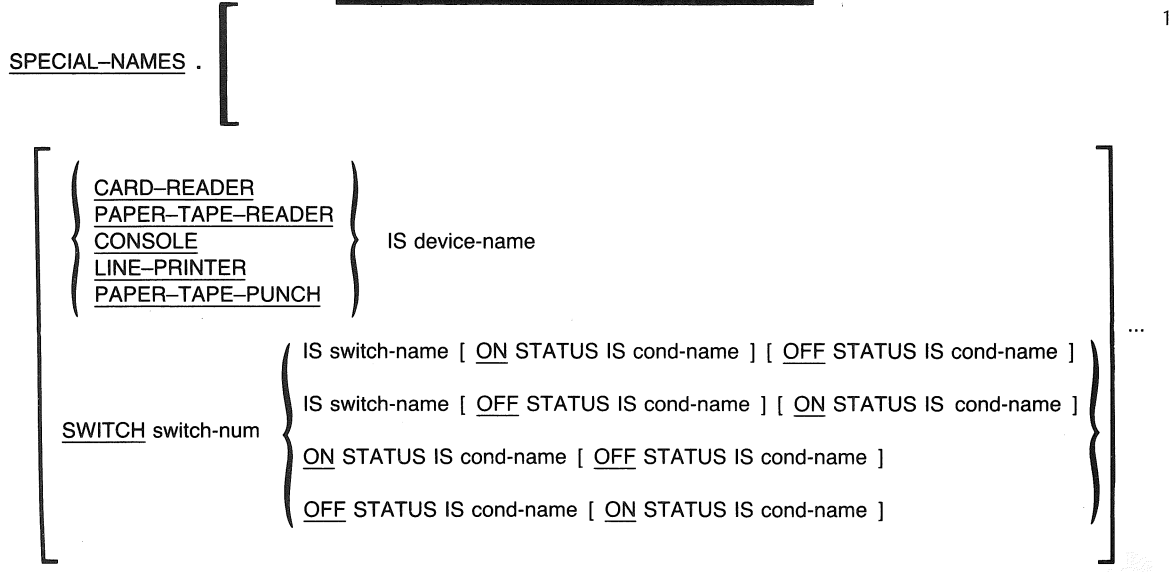
Configuration Section Entries

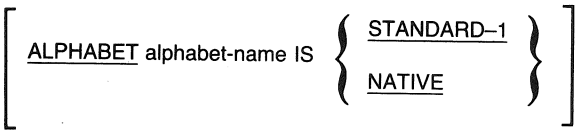




[PROGRAM COLLATING SEQUENCE IS alphabet-name]

[SEGMENT–LIMIT IS segment-number] .]


SPECIAL–NAMES .
CARD–READER
PAPER–TAPE–READER
CONSOLE
LINE–PRINTER
PAPER–TAPE–PUNCH
IS device-name
SWITCH switch-num
IS switch-name [ON STATUS IS cond-name] [OFF STATUS IS cond-name]
IS switch-name [OFF STATUS IS cond-name] [ON STATUS IS cond-name]
ON STATUS IS cond-name [OFF STATUS IS cond-name]
OFF STATUS IS cond-name [ON STATUS IS cond-name]
...



[CURRENCY SIGN IS char]

[DECIMAL–POINT IS COMMA] .

Input-Output Section Entries

FILE–CONTROL.

Format 1 – Sequential File

```
SELECT [ OPTIONAL ] file-name

    ASSIGN TO file-spec

[   RESERVE reserve-num [ AREA  ] ]
                        [ AREAS ]

[ [ ORGANIZATION IS ] SEQUENTIAL ]

[  ACCESS MODE IS SEQUENTIAL  ]
[  FILE STATUS IS file-stat ]  .
```

Format 2 – Relative File

```
SELECT file-name

    ASSIGN TO file-spec
[                                    [ AREA  ]  ]
    RESERVE reserve-num              [ AREAS ]
]

[ ORGANIZATION IS ] RELATIVE
```

(continued on next page)

Format 2 – Relative File (Cont.)

```
[                        { SEQUENTIAL [ RELATIVE KEY IS rel-key ]                 } ]
[ ACCESS MODE IS         {                                                        } ]
[                        { { RANDOM  }                                            } ]
[                        { {         }  RELATIVE KEY IS rel-key                   } ]
[                        { { DYNAMIC }                                            } ]
[ FILE STATUS IS file-stat ] .
```

Format 3 – Indexed File

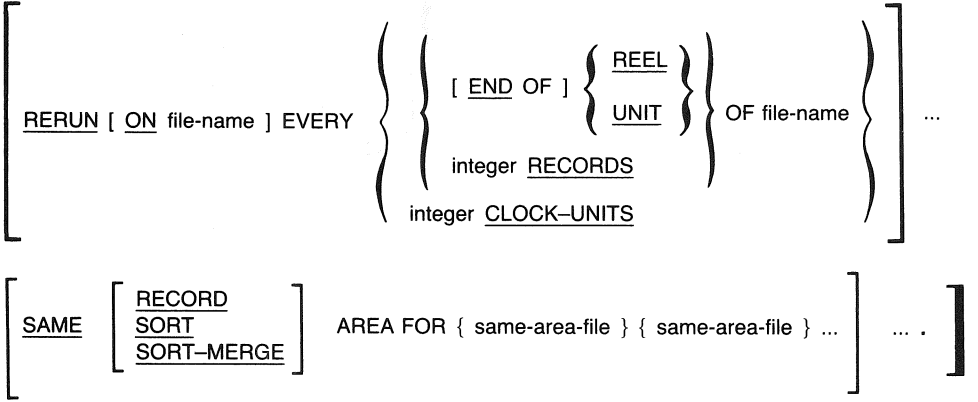
SELECT file-name

ASSIGN TO file-spec

```
[ RESERVE reserve-num [ AREA  ] ]
                      [ AREAS ]

[ ORGANIZATION IS ] INDEXED

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

[ RECORD KEY IS rec-key ]

[ ALTERNATE RECORD KEY IS alt-key [ WITH DUPLICATES ] ] ...

[ FILE STATUS IS file-stat ] .
```

Format 4 – Sort or Merge File

```
SELECT file-name ASSIGN TO file-spec .

I-O-CONTROL . [
    [ APPLY { DEFERRED-WRITE
              EXTENSION extend-amt
              FILL-SIZE
              MASS-INSERT
            [ CONTIGUOUS ] PREALLOCATION preall-amt
              PRINT-CONTROL
              WINDOW window-ptrs } ON { file-name } ... ] ...
```


RERUN [ON file-name] EVERY
integer CLOCK–UNITS
[END OF]
REEL
UNIT
integer RECORDS
OF file-name
...
SAME
RECORD
SORT
SORT–MERGE
AREA FOR { same-area-file } { same-area-file } ...
... .

Data Division Format

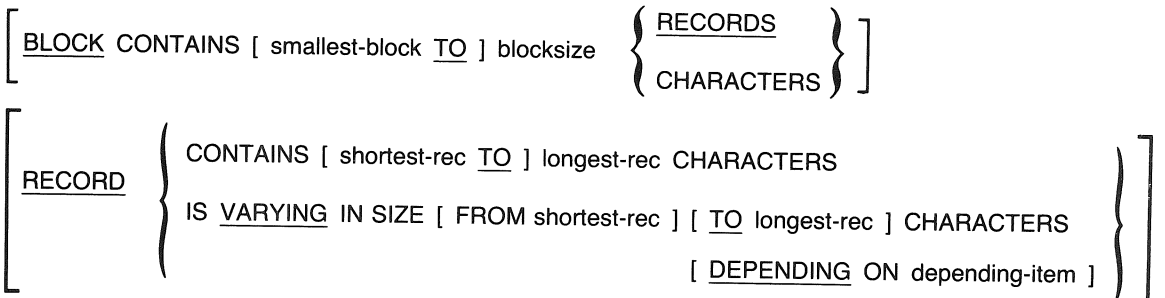
```
[ DATA DIVISION.

[ FILE SECTION.

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

[ WORKING-STORAGE SECTION.

      [ record-description-entry ] ... ]

[ LINKAGE SECTION.

      [ record-description-entry ] ... ] ]
```

Data Division Entries

File description entries:

Format 1 – Sequential File

FD file-name



(continued on next page)

Format 1 – Sequential File (Cont.)

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

[ VALUE OF ID IS file-spec ]

[ DATA { RECORDS ARE } { rec-name } ... ]
       { RECORD IS   }

[ LINAGE IS { page-size } LINES [ WITH FOOTING AT footing-line ]
                                [ LINES AT TOP top-lines ] [ LINES AT BOTTOM bottom-lines ] ]

[ CODE–SET IS alphabet-name ] .
```

Format 2 – Relative File

```
FD file-name
   [ BLOCK CONTAINS [ smallest-block TO ] blocksize { RECORDS      } ]
                                                    { CHARACTERS   }

   [ RECORD { CONTAINS [ shortest-rec TO ] longest-rec CHARACTERS                                          } ]
            { IS VARYING IN SIZE [ FROM shortest-rec ] [ TO longest-rec ] CHARACTERS                       }
            {                                          [ DEPENDING ON depending-item ]                 }

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

(continued on next page)

Format 2 – Relative File (Cont.)

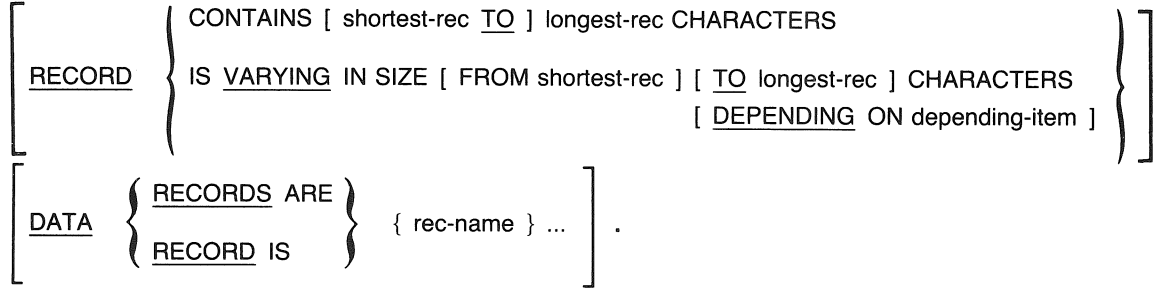
```
[ VALUE OF ID IS file-spec ]

[ DATA { RECORDS ARE  } { rec-name } ... ] .
       { RECORD IS    }
```

Format 3 – Indexed File

```
FD  file-name
    [ BLOCK CONTAINS [ smallest-block TO ] blocksize { RECORDS    } ]
                                                     { CHARACTERS }
```

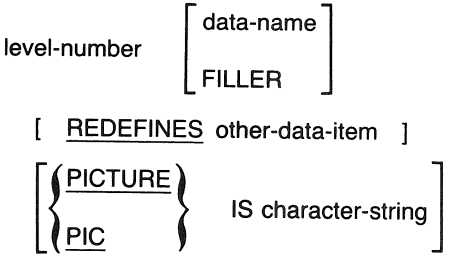
```
[ RECORD { CONTAINS [ shortest-rec TO ] longest-rec CHARACTERS                                   } ]
         { IS VARYING IN SIZE [ FROM shortest-rec ] [ TO longest-rec ] CHARACTERS               }
                                                    [ DEPENDING ON depending-item ]

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

[ VALUE OF ID IS file-spec ]

[ DATA { RECORDS ARE } { rec-name } ... ] .
       { RECORD IS   }
```

Sort-merge file description entry:



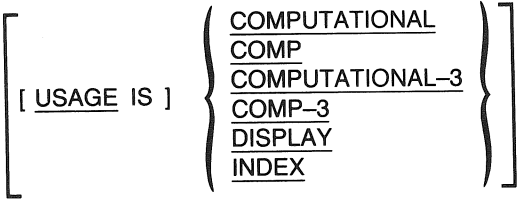
Data description entries:

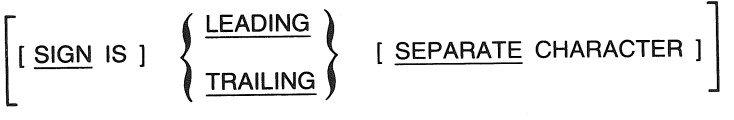
Format 1

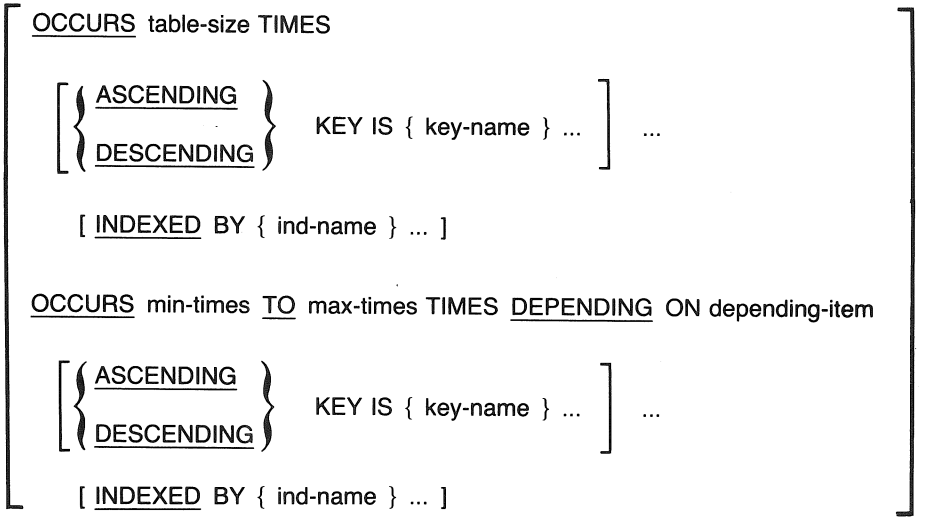


(continued on next page)

Data description entries: – Format 1 (Cont.)





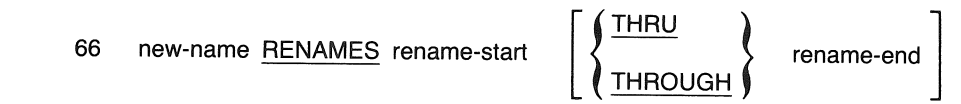


(continued on next page)

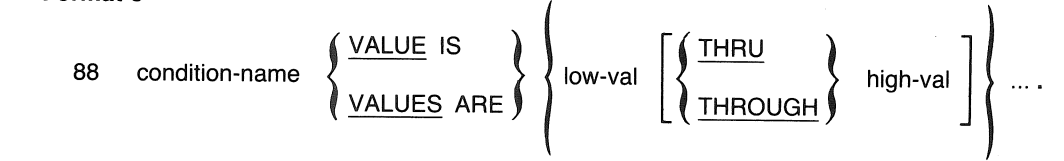
Data description entries: – Format 1 (Cont.)

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

[ { JUSTIFIED } RIGHT ]
[ { JUST      }       ]

[ BLANK WHEN ZERO ]

[ VALUE IS lit ] .
```

Format 2

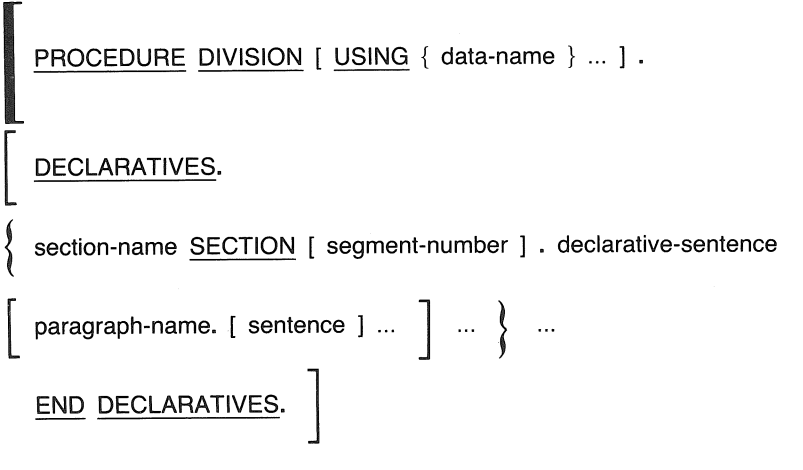


Format 3



Procedure Division Format

Format 1



```
{ section-name SECTION [ segment-number ] .
[ paragraph-name. [ sentence ] ... ] ... } ...
```

Format 2

```
[ PROCEDURE DIVISION [ USING { data-name } ... ] .
[ paragraph-name. [ sentence ] ... ] ... ]
```

Procedure Division Statements

ACCEPT dest-item [FROM input-source]

ACCEPT dest-item FROM { DATE | DAY | TIME }

ACCEPT dest-item

FROM LINE NUMBER { line-num | line-id [PLUS [plus-num]] | PLUS [plus-num] }

FROM COLUMN NUMBER { column-num | column-id [PLUS [plus-num]] | PLUS [plus-num] }

ERASE [TO END OF] { SCREEN / LINE }

WITH BELL

UNDERLINED

BOLD

WITH BLINKING

PROTECTED [SIZE protect-length]

WITH CONVERSION

REVERSED

WITH NO ECHO

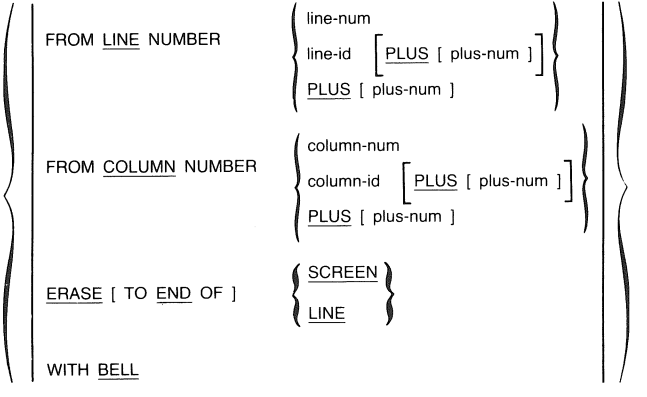
DEFAULT IS { def-src-lit / def-src-item }

CONTROL KEY IN key-dest-item

[ON EXCEPTION stment]

(continued on next page)

ACCEPT CONTROL KEY IN key-dest-item



[ON EXCEPTION stment]

ADD { num } ... TO { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

ADD { num } { num } ... GIVING { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

ADD { CORRESPONDING / CORR } grp–1 TO grp–2 [ROUNDED] [ON SIZE ERROR stment]

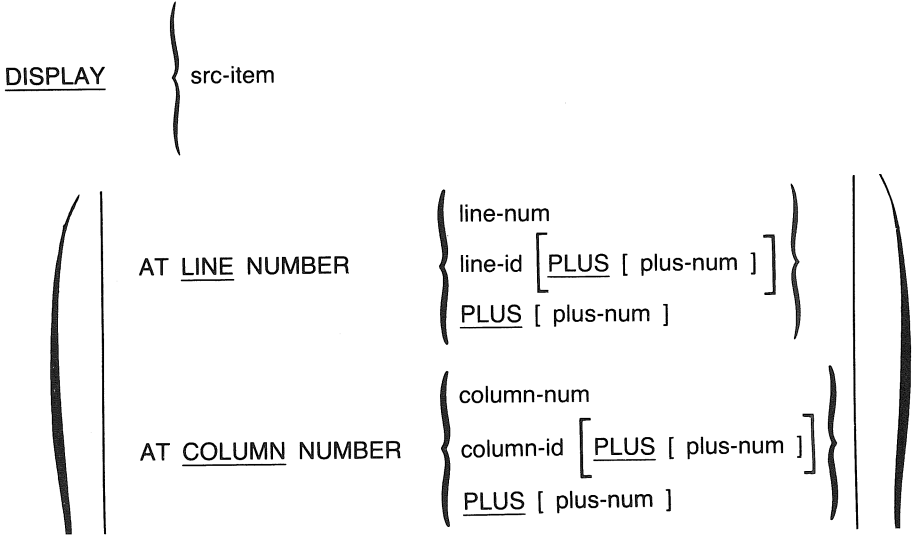
```
CALL prog-name

  [                 ⎧ ⎧ [ BY REFERENCE ] ⎫               ⎫   [ ⎧ BY REFERENCE  ⎫               ]      ]
  [ USING           ⎨ ⎨                  ⎬ { arg } ...   ⎬   [ ⎨               ⎬ { arg } ...   ] ...  ]
  [                 ⎩ ⎩   BY DESCRIPTOR  ⎭               ⎭   [ ⎩ BY DESCRIPTOR ⎭               ]      ]
```

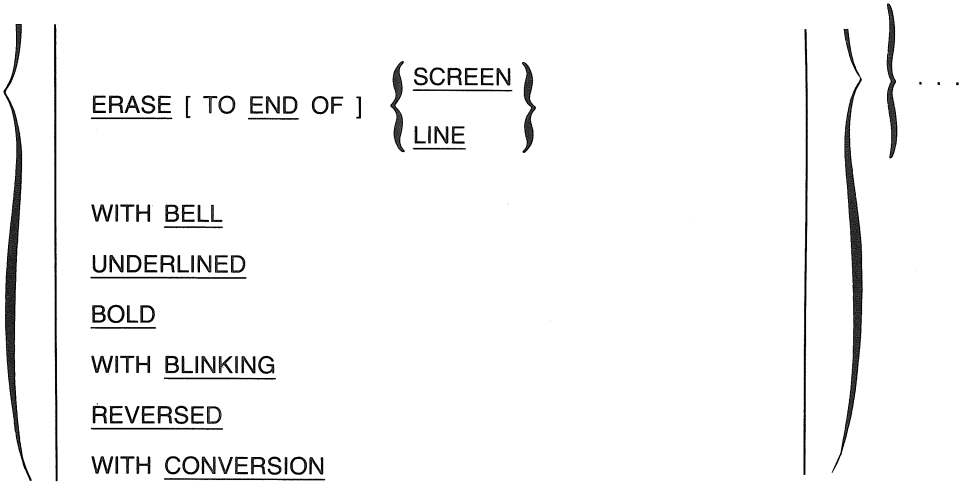
```
                           [ ⎧ REEL ⎫                       ] ⎫
                           [ ⎨      ⎬ [ FOR REMOVAL ]        ] ⎪
          ⎧                [ ⎩ UNIT ⎭                        ] ⎪
CLOSE     ⎨  file-name     [                                 ] ⎬ ...
          ⎩                [          ⎧ NO REWIND ⎫          ] ⎪
                           [   WITH   ⎨           ⎬          ] ⎪
                           [          ⎩ LOCK      ⎭          ] ⎭
```

COMPUTE { rsult [ROUNDED] } ... = arithmetic-expression [ON SIZE ERROR stment]

DELETE file-name RECORD [INVALID KEY stment]

DISPLAY { src-item } ... [UPON output-dest] [WITH NO ADVANCING]


DISPLAY
src-item
AT LINE NUMBER
line-num
line-id [PLUS [plus-num]]
PLUS [plus-num]
AT COLUMN NUMBER
column-num
column-id [PLUS [plus-num]]
PLUS [plus-num]



[WITH NO ADVANCING]

DIVIDE srcnum INTO { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

DIVIDE srcnum INTO srcnum GIVING { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

DIVIDE srcnum BY srcnum GIVING { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

DIVIDE srcnum INTO srcnum GIVING rsult [ROUNDED] REMAINDER remaind
[ON SIZE ERROR stment]

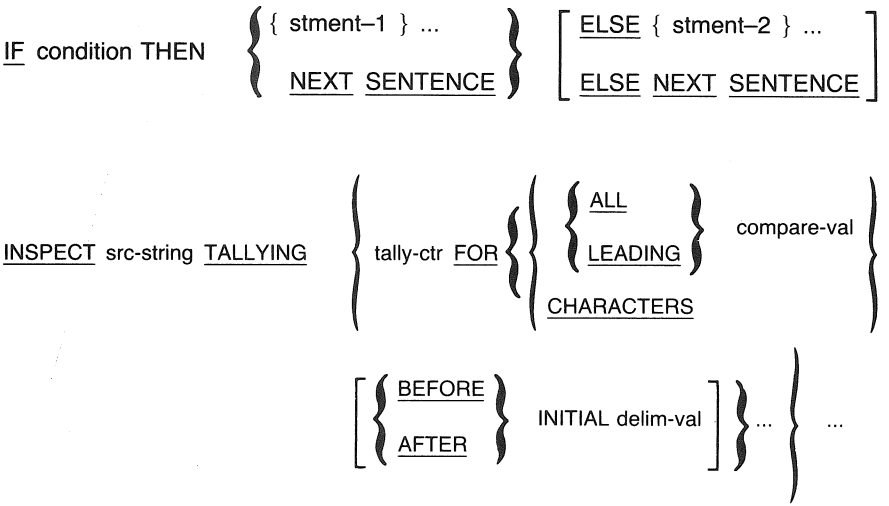
DIVIDE srcnum BY srcnum GIVING rsult [ROUNDED] REMAINDER remaind
[ON SIZE ERROR stment]

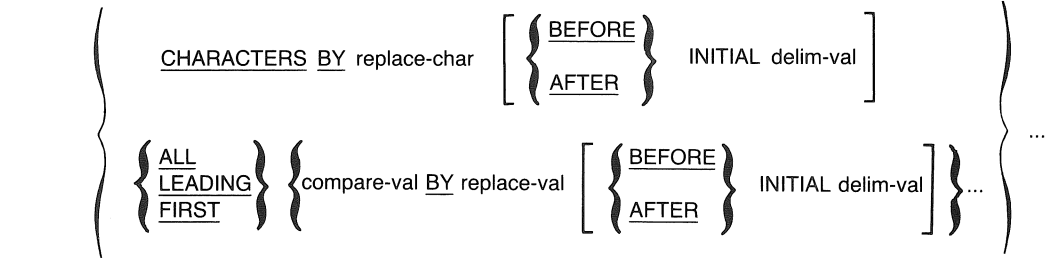
EXIT .

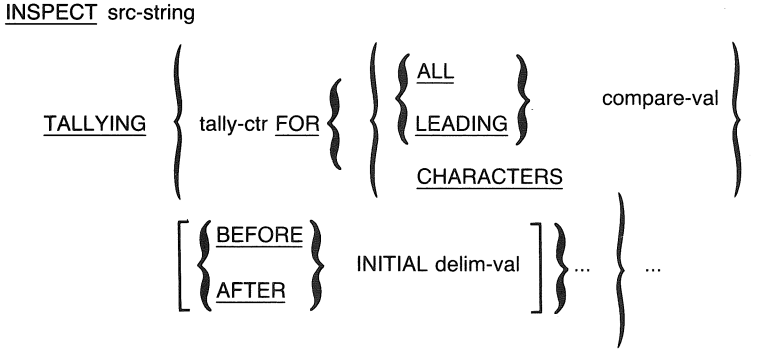
EXIT PROGRAM

GO TO proc-name

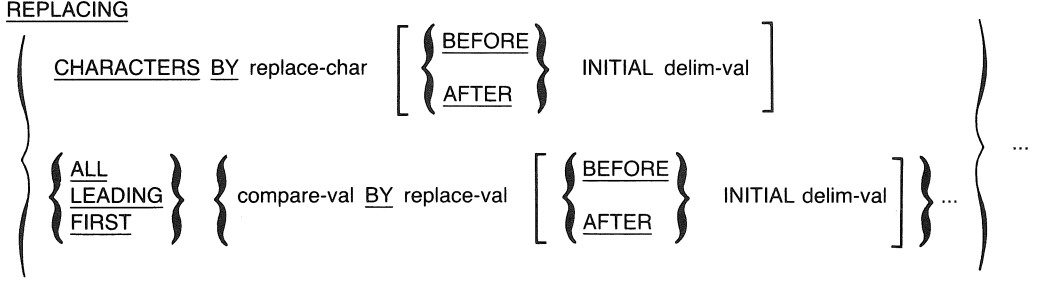
GO TO proc-name { proc-name } ... DEPENDING ON num

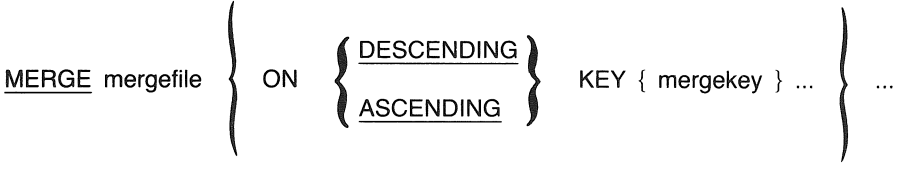

IF condition THEN
{ stment–1 } ...
NEXT SENTENCE
ELSE { stment–2 } ...
ELSE NEXT SENTENCE
INSPECT src-string TALLYING
tally-ctr FOR
ALL
LEADING
CHARACTERS
compare-val
BEFORE
AFTER
INITIAL delim-val
...
...


INSPECT src-string REPLACING
CHARACTERS BY replace-char
BEFORE
AFTER
INITIAL delim-val
ALL
LEADING
FIRST
compare-val BY replace-val
BEFORE
AFTER
INITIAL delim-val
...
...



(continued on next page)

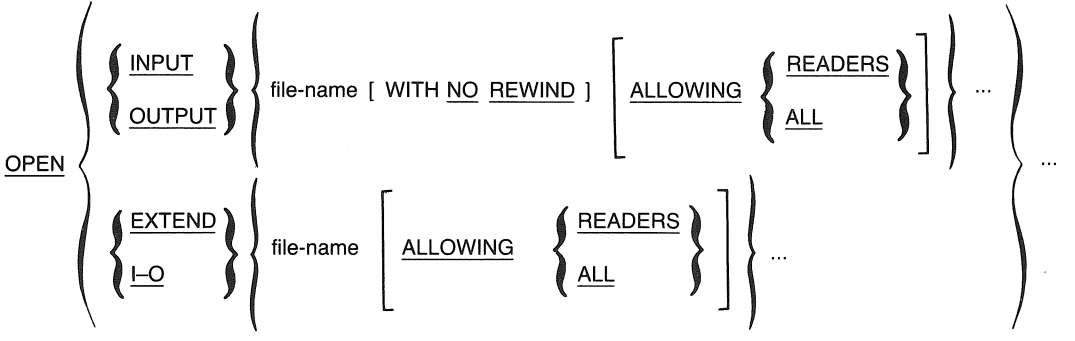

REPLACING
CHARACTERS BY replace-char
BEFORE
AFTER
INITIAL delim-val
ALL
LEADING
FIRST
compare-val BY replace-val
BEFORE
AFTER
INITIAL delim-val
...
...

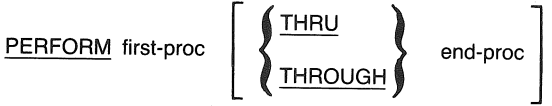

MERGE mergefile
ON
DESCENDING
ASCENDING
KEY { mergekey } ...
...

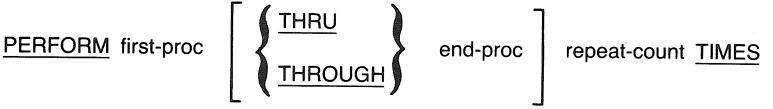
```
[ COLLATING SEQUENCE IS alpha ]

  USING  infile  { infile } ...

 ( OUTPUT PROCEDURE IS first-proc  [ { THRU    }  end-proc ] )
 (                                   { THROUGH }            )
 ( GIVING { outfile } ...                                   )


MOVE  { src-item }  TO   { dest-item } ...
      { lit      }


MOVE  { CORRESPONDING }  src-item TO dest-item
      { CORR          }
```

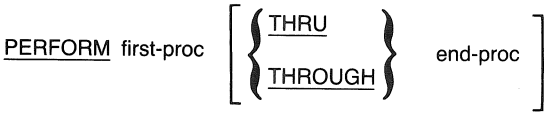
MULTIPLY srcnum BY { rsult [ROUNDED] } ... [ON SIZE ERROR stment]

MULTIPLY srcnum BY srcnum GIVING { rsult [ROUNDED] } ... [ON SIZE ERROR stment]









(continued on next page)

PERFORM first-proc [{ THRU / THROUGH } end-proc] VARYING var FROM init BY increm UNTIL cond

[AFTER var FROM init BY increm UNTIL cond] ...

READ file-name [NEXT] RECORD [INTO dest-item] [AT END stment]

READ file-name RECORD [INTO dest-item] [KEY IS key-name] [INVALID KEY stment]

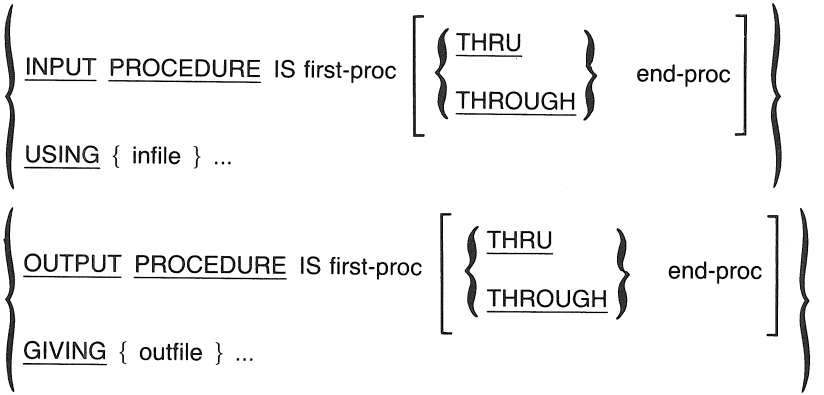
RELEASE rec [FROM src-area]

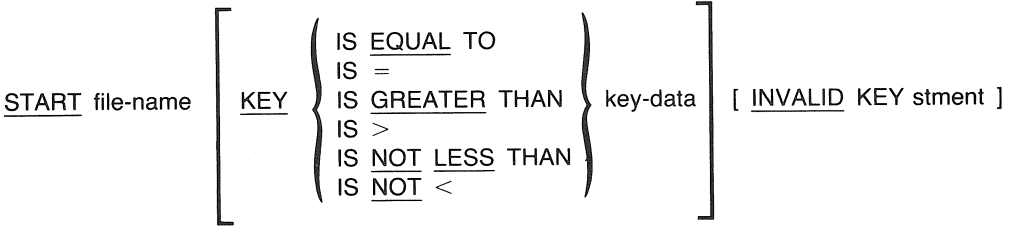
RETURN smrg-file RECORD [INTO dest-area] AT END stment

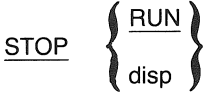
REWRITE rec-name [FROM src-item] [INVALID KEY stment]

```
SEARCH src-table [ VARYING pointr ] [ AT END stment ]
        { WHEN cond { stment          } } ...
                    { NEXT SENTENCE   }

SEARCH ALL src-table [ AT END stment ] WHEN { elemnt { IS EQUAL TO } arg }
                                           { IS =   }
                                           { cond-name              }

        [ AND { elemnt { IS EQUAL TO } arg } ] ... { stment          }
              { IS =   }                           { NEXT SENTENCE }
              { cond-name              }
```

```
SET { rsult } ... TO val

SET { indx } ... { UP BY      } increm
                 { DOWN BY    }

SORT sortfile { ON { DESCENDING } KEY { sortkey } ... } ...
              {    { ASCENDING  }                     }

    [ WITH DUPLICATES IN ORDER ]

    [ COLLATING SEQUENCE IS alpha ]
```


INPUT PROCEDURE IS first-proc
THRU
THROUGH
end-proc
USING { infile } ...
OUTPUT PROCEDURE IS first-proc
THRU
THROUGH
end-proc
GIVING { outfile } ...

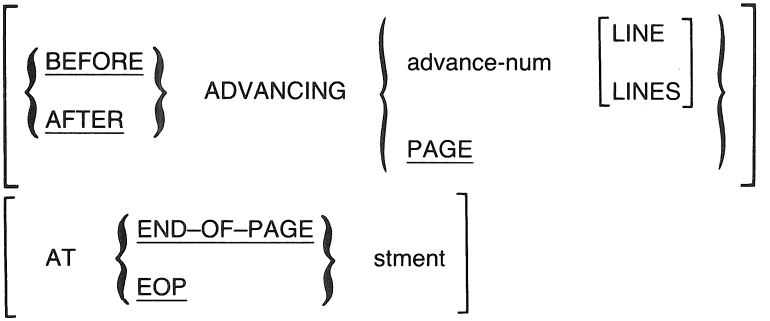

START file-name
KEY
IS EQUAL TO
IS =
IS GREATER THAN
IS >
IS NOT LESS THAN
IS NOT <
key-data
[INVALID KEY stment]


STOP
RUN
disp

```
STRING    { { src-string }...  DELIMITED BY  { delim } } ...
                               { SIZE  }
          INTO dest-string [ WITH POINTER pointr ] [ ON OVERFLOW stment ]

SUBTRACT { num } ... FROM { rsult [ ROUNDED ] } ...  [ ON SIZE ERROR stment ]

SUBTRACT { num } ... FROM num GIVING { rsult [ ROUNDED ] } ...  [ ON SIZE ERROR stment ]

SUBTRACT { CORRESPONDING } grp–1 FROM grp–2 [ ROUNDED ]  [ ON SIZE ERROR stment ]
         { CORR          }
```

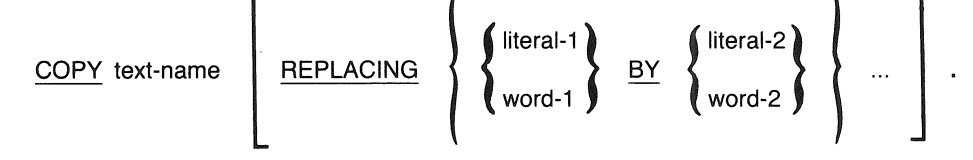
```
UNSTRING src-string [ DELIMITED BY [ ALL ] delim [ OR [ ALL ] delim ] ... ]
         INTO { dest-string [ DELIMITER IN delim-dest ] [ COUNT IN countr ] } ...
         [ WITH POINTER pointr ]
         [ TALLYING IN tally-ctr ]

         [ ON OVERFLOW stment ]

USE AFTER STANDARD { EXCEPTION } PROCEDURE ON { { file-name } ... }
                   { ERROR     }              { INPUT          }
                                              { OUTPUT         }
                                              { I-O            }
                                              { EXTEND         }
```

WRITE rec-name [FROM src-item]



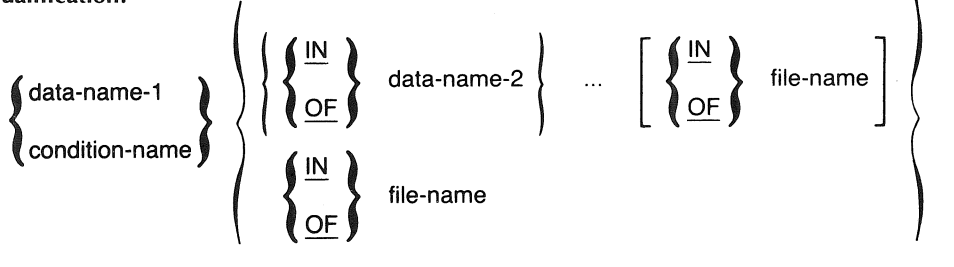
WRITE rec-name [FROM src-item] [INVALID KEY stment]

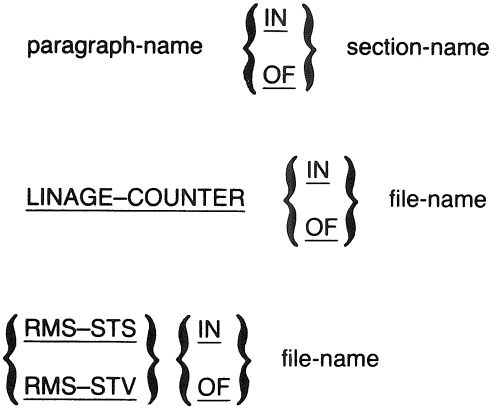
COPY Statement



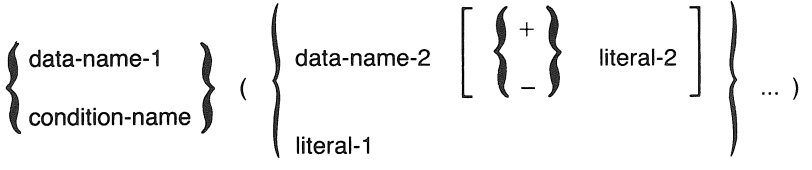
Miscellaneous Formats

Qualification:

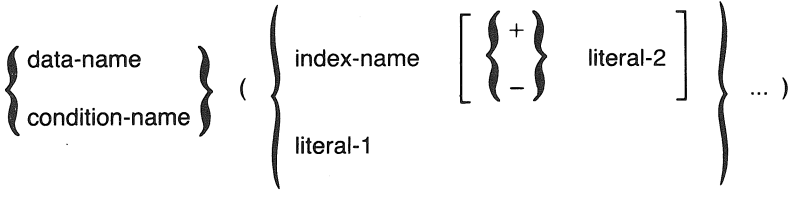



paragraph-name
IN
OF
section-name
LINAGE–COUNTER
IN
OF
file-name
RMS–STS
RMS–STV
IN
OF
file-name

Subscripting:



Indexing:

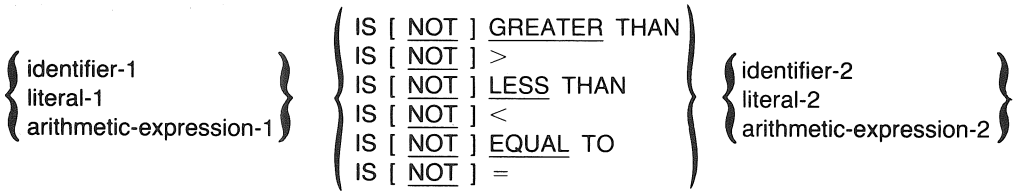


Identifier:

data-name [qualification] [subscripting]

data-name [qualification] [indexing]

Relation condition:



Class condition:

identifier IS [NOT] { NUMERIC / ALPHABETIC }

Switch-status and condition-name condition:

condition-name

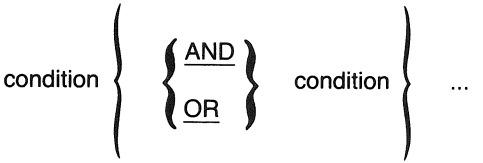
Sign condition:

arithmetic-expression IS [NOT] { POSITIVE / NEGATIVE / ZERO }

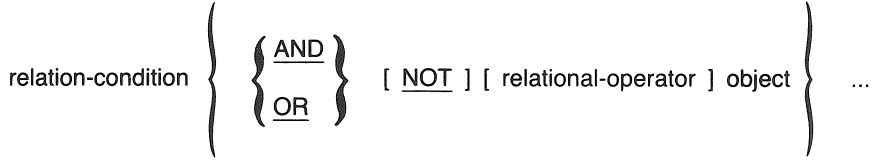
Negated simple condition:

NOT simple-condition

Combined condition:



Abbreviated combined relation condition:



FROM option:

record-name FROM identifier

INTO option:

file-name INTO identifier

Segmentation:

section-name SECTION [segment-number] .

AT END option:

AT END stment

INVALID KEY option:

INVALID KEY stment

PICTURE Clause Characters

Data Characters

A Alphabetic character

X Alphanumeric character

9 Numeric character

Operation Symbols

S Sign

V Assumed decimal point location

P Assumed decimal point scaling position

Replacement Characters

Z Leading zeros replaced by spaces

***** 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 paragraph)
B	Space character
0	Zero
/	Slash character
,	Comma character
.	Period character
+	Plus sign when item is positive, minus when negative; floating when more than one
–	Minus sign when item is negative, blank when positive; 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

Figurative Constants

ZERO, ZEROS, ZEROES	Represents the value zero, or one or more of the character "0", depending on context.
SPACE, SPACES	Represents one or more of the space character.
HIGH-VALUE, HIGH-VALUES	Represents one or more of the character with the highest ordinal position in the program collating sequence.
LOW-VALUE, LOW-VALUES	Represents one or more of the character with the lowest ordinal position in the program collating sequence.
QUOTE, QUOTES	Represents one or more of the quotation mark character (").
ALL literal	Represents one or more occurrences of the string of characters comprising the literal.

FILE STATUS Values

	FILE STATUS	Input-Output Statements	File Organization	Access Method	Meaning
	00	All	All	All	Successful
	02	REWRITE WRITE	Ind	All	Created duplicate alternate key
	05	OPEN	Seq	Seq	Optional file not present
*	13	READ	All	Seq	No next logical record (at end)
*	15	READ	Seq	Seq	Optional file not present (at end)
*	16	READ	All	Seq	No valid next record (at end)
	21	REWRITE	Ind	Seq	Primary key changed after READ
	21	WRITE	Ind	Seq	Attempted nonascending key value (invalid key)

(continued on next page)

FILE STATUS Values (Cont.)

FILE STATUS	Input-Output Statements	File Organization	Access Method	Meaning
22	REWRITE	Ind	All	Duplicate alternate key (invalid key)
22	WRITE	Ind, Rel	All	Duplicate key (invalid key)
23	DELETE READ REWRITE START	Ind, Rel	Ran	Record not in file (invalid key)
24	WRITE	Ind, Rel	All	Boundary violation (invalid key)
30	All	All	All	All other permanent errors
34	WRITE	Seq	Seq	Boundary violation
90	READ	All	All	Record locked by another user; record is available in record area

91	OPEN	All	All	File locked by another program; record is not available
92	DELETE READ REWRITE START WRITE	All	All	Record locked by another program
93	DELETE REWRITE	All	Seq	No previous READ
94	CLOSE	All	All	File never opened or already closed
94	OPEN	All	All	File already open, or closed with lock
94	DELETE READ REWRITE START WRITE	All	All	File not open, or incompatible open mode

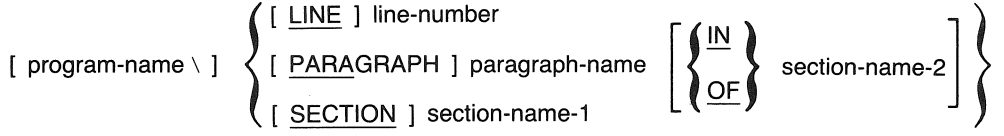
(continued on next page)

FILE STATUS Values (Cont.)

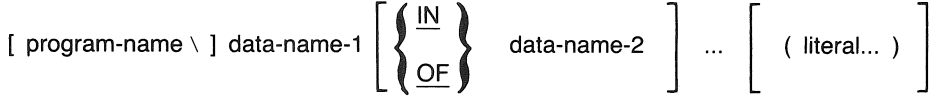
FILE STATUS	Input-Output Statements	File Organization	Access Method	Meaning
95	OPEN	All	All	No file space on device
96	OPEN	All	All	Same area busy
97	OPEN	All	All	File not found
98	CLOSE	All	All	Any other CLOSE error

* The value 10 replaces 13, 15, and 16 if the /FIPS:74 switch is used during compilation.

Using the COBOL-81 Symbolic Debugger (Command Formats)



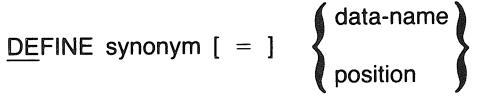
Specifies *position* in Debugger commands



Specifies *data-name* in Debugger commands

Using the COBOL-81 Symbolic Debugger (Command Formats) (Cont.)

Removes one or all of the current breakpoints



Specifies a synonym for a data-name or position

Displays the contents of a data item

HELP [topic-word]

Supplies information about a Debugger command or topic

MOVE literal [[TO] data-name]

Changes the value of a COBOL data item

PROCEED [integer]

Starts program or continues execution after a breakpoint

SET BREAKPOINT position [DISPLAY data-name] [PROCEED { integer / ALWAYS }]

Inserts a breakpoint at the indicated position

Using the COBOL-81 Symbolic Debugger (Command Formats) (Cont.)

SHOW { BREAKPOINTS / SYNONYMS }	Lists current synonyms or breakpoints, along with their actual names
STOP [RUN]	Stops program execution and ends the debugging session
UNDEFINE synonym	Deletes a synonym

Special Registers

Name	Size	Used With	Description
LINAGE-COUNTER	S9(4) COMP	Linage Files	Names a line counter when a file description entry contains a LINAGE clause. Its value is the number of the current line within the page body.
RMS-STS	S9(4) COMP	RMS-11	Contains the primary return-status value of an I/O operation (RMS-STV is the secondary return-status value.).
RMS-STV	S9(4) COMP	RMS-11	Contains the secondary return-status value of an I/O operation (RMS-STS is the primary return-status value.).

ACCEPT
ACCESS
ADD
ADVANCING
AFTER
ALL
ALLOWING
ALPHABET
ALPHABETIC
ALPHABETIC-LOWER
ALPHABETIC-UPPER
ALPHANUMERIC
ALPHANUMERIC-EDITED
ALSO
ALTER
ALTERNATE
AND
ANY
APPLY
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUTHOR

BATCH
BEFORE
BEGINNING
BELL
BIT
BITS
BLANK
BLINKING
BLOCK
BOLD
BOOLEAN
BOTTOM
BY

CALL
CANCEL
CD
CF
CH
CHARACTER
CHARACTERS
CLOCK-UNITS
CLOSE
COBOL
CODE
CODE-SET
COLLATING
COLUMN
COMMA
COMMIT
COMMON
COMMUNICATION
COMP
COMP-1
COMP-2
COMP-3
COMP-4

COMP-5
COMP-6
COMPUTATIONAL
COMPUTATIONAL-1
COMPUTATIONAL-2
COMPUTATIONAL-3
COMPUTATIONAL-4
COMPUTATIONAL-5
COMPUTATIONAL-6
COMPUTE
CONCURRENT
CONFIGURATION
CONNECT
CONTAINS
CONTENT
CONTINUE
CONTROL
CONTROLS
CONVERSION
CONVERTING
COPY
CORR
CORRESPONDING
COUNT
CURRENCY
CURRENT

DATA
DATE
DATE-COMPILED
DATE-WRITTEN
DAY
DAY-OF-WEEK
DB
DB-ACCESS-CONTROL-KEY
DB-CONDITION
DB-CURRENT-RECORD-ID
DB-CURRENT-RECORD-NAME
DB-EXCEPTION
DB-RECORD-NAME
DB-SET-NAME
DB-STATUS
DE
DEBUG-CONTENTS
DEBUG-ITEM
DEBUG-LENGTH
DEBUG-LINE
DEBUG-NAME
DEBUG-NUMERIC-CONTENTS
DEBUG-SIZE
DEBUG-START
DEBUG-SUB
DEBUG-SUB-1
DEBUG-SUB-2
DEBUG-SUB-3
DEBUG-SUB-ITEM
DEBUG-SUB-N
DEBUG-SUB-NUM
DEBUGGING
DECIMAL-POINT
DECLARATIVES
DEFAULT
DELETE
DELIMITED

Reserved Words (Cont.)

DELIMITER
DEPENDING
DESCENDING
DESCRIPTOR
DESTINATION
DETAIL
DICTIONARY
DISABLE
DISCONNECT
DISPLAY
DISPLAY-6
DISPLAY-7
DISPLAY-9
DIVIDE
DIVISION
DOWN
DUPLICATE
DUPLICATES
DYNAMIC

ECHO
EGI
ELSE
EMI
EMPTY
ENABLE
END
END-ACCEPT
END-ADD
END-CALL
END-COMMIT
END-COMPUTE
END-CONNECT
END-DELETE
END-DISCONNECT
END-DIVIDE
END-ERASE
END-EVALUATE
END-FETCH
END-FIND
END-FINISH
END-FREE
END-GET
END-IF
END-KEEP
END-MODIFY
END-MULTIPLY
END-OF-PAGE
END-PERFORM
END-READ
END-READY
END-RECEIVE
END-RECONNECT
END-RETURN
END-REWRITE
END-ROLLBACK
END-SEARCH
END-START
END-STORE
END-STRING
END-SUBTRACT
END-UNSTRING
END-WRITE
ENDING

ENTER
ENVIRONMENT
EOP
EQUAL
EQUALS
ERASE
ERROR
ESI
EVALUATE
EVERY
EXCEEDS
EXCEPTION
EXCLUSIVE
EXIT
EXOR
EXTEND
EXTERNAL

FAILURE
FALSE
FD
FETCH
FILE
FILE-CONTROL
FILLER
FINAL
FIND
FINISH
FIRST
FOOTING
FOR
FREE
FROM

GENERATE
GET
GIVING
GLOBAL
GO
GREATER
GROUP

HEADING
HIGH-VALUE
HIGH-VALUES

I-O
I-O-CONTROL
IDENTIFICATION
IF
IN
INCLUDING
INDEX
INDEXED
INDICATE
INITIAL
INITIALIZE
INITIATE
INPUT
INPUT-OUTPUT
INSPECT
INSTALLATION
INTO
INVALID
IS

Reserved Words (Cont.)

JUST
JUSTIFIED

KEEP
KEY

LABEL
LAST
LD
LEADING
LEFT
LENGTH
LESS
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
LINKAGE
LOCALLY
LOCK
LOW-VALUES

MATCHES
MEMBER
MEMBERSHIP
MEMORY
MERGE
MESSAGE
MODE
MODIFY
MODULES
MOVE
MULTIPLE
MULTIPLY

NATIVE
NEGATIVE
NEXT
NO
NON-NULL
NOT
NULL
NUMBER
NUMERIC
NUMERIC-EDITED

OBJECT-COMPUTER
OCCURS
OF
OFF
OFFSET
OMITTED
ON
ONLY
OPEN
OPTIONAL
OR
ORDER
ORGANIZATION
OTHER
OTHERS

OUTPUT
OVERFLOW
OWNER

PADDING
PAGE
PAGE-COUNTER
PERFORM
PF
PH
PIC
PICTURE
PLUS
POINTER
POSITION
POSITIVE
PRINTING
PRIOR
PROCEDURE
PROCEDURES
PROCEED
PROGRAM
PROGRAM-ID
PROTECTED
PURGE

QUEUE
QUOTE
QUOTES

RANDOM
RD
READ
READERS
READY
REALM
REALMS
RECEIVE
RECONNECT
RECORD
RECORD-NAME
RECORDS
REDEFINES
REEL
REFERENCE
REFERENCE-MODIFIER
REFERENCES
REGARDLESS
RELATIVE
RELEASE
REMAINDER
REMOVAL
RENAMES
REPLACE
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RESERVE
RESET
RETAINING
RETRIEVAL
RETURN
REVERSED

Reserved Words (Cont.)

REWIND
REWRITE
RF
RH
RIGHT
RMS-FILENAME
RMS-STS
RMS-STV
ROLLBACK
ROUNDED
RUN

SAME
SCREEN
SD
SEARCH
SECTION
SECURITY
SEGMENT
SEGMENT-LIMIT
SELECT
SEND
SENTENCE
SEPARATE
SEQUENCE
SEQUENCE-NUMBER
SEQUENTIAL
SET
SETS
SIGN
SIZE
SORT
SORT-MERGE
SOURCE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL-NAMES
STANDARD
STANDARD-1
STANDARD-2
START
STATUS
STOP
STORE
STRING
SUB-QUEUE-1
SUB-QUEUE-2
SUB-QUEUE-3
SUB-SCHEMA
SUBTRACT
SUCCESS
SUM
SUPPRESS
SYMBOLIC
SYNC
SYNCHRONIZED

TABLE
TALLYING
TAPE
TENANT
TERMINAL
TERMINATE

TEST
TEXT
THAN
THEN
THROUGH
THRU
TIME
TIMES
TO
TOP
TRAILING
TRUE
TYPE

UNDERLINED
UNEQUAL
UNIT
UNLOCK
UNSTRING
UNTIL
UP
UPDATE
UPDATERS
UPON
USAGE
USAGE-MODE
USE
USING

VALUE
VALUES
VARYING

WAIT
WHEN
WHERE
WITH
WITHIN
WORDS
WORKING-STORAGE
WRITE
WRITERS

ZERO
ZEROES
ZEROS

COBOL Data Types

The way a data item is represented in the Data Division of a COBOL program determines the size and format of the item in storage. The following tables: (a) match COBOL data description entries with their corresponding PDP-11 storage formats, and (b) show the allocated storage in bytes for the entry.

For example, a data item described as PIC S9(4) USAGE IS DISPLAY SIGN IS TRAILING would be stored in four bytes of storage as a right overpunch value.

Note

DISPLAY is the default USAGE for numeric, alphabetic, and alphanumeric data items. Thus, the specification USAGE IS DISPLAY is optional for display numeric, alphabetic, and alphanumeric data types.

Unscaled Data Items and Corresponding Storage Data Types

Unscaled Data Item			Storage Data Type
PICTURE Clause	USAGE Clause	Allocated Storage in Bytes	
PIC S9(n) [n <= 18]	USAGE IS DISPLAY	n	Right overpunch
PIC S9(n) [n <= 18]	USAGE IS DISPLAY SIGN IS TRAILING	n	Right overpunch
PIC S9(n) [n <= 18]	USAGE IS DISPLAY SIGN IS LEADING	n	Left overpunch
PIC S9(n) [n <= 18]	USAGE IS DISPLAY SIGN IS TRAILING SEPARATE	n+1	Right separate

(continued on next page)

Unscaled Data Items and Corresponding Storage Data Types (Cont.)

Unscaled Data Item			Storage Data Type
PICTURE Clause	USAGE Clause	Allocated Storage in Bytes	
PIC S9(n) [n <= 18]	USAGE IS DISPLAY SIGN IS LEADING SEPARATE	n+1	Left separate
PIC 9(n) [n <= 18]	USAGE IS DISPLAY	n	Unsigned numeric
PIC 9(n) [n <= 4]	USAGE IS COMP	2	Word integer*
PIC 9(n) [5 <= n <= 9]	USAGE IS COMP	4	Two word integer*

PIC 9(n) [10 <= n <= 18]	USAGE IS COMP	8	Four word integer*
PIC S9(n) [n <= 4]	USAGE IS COMP	2	Word integer
PIC S9(n) [5 <= n <= 9]	USAGE IS COMP	4	Two word integer
PIC S9(n) [10 <= n <= 18]	USAGE IS COMP	8	Four word integer
N/A	USAGE IS INDEX	2	One word integer
PIC S9(n) [n <= 18]	USAGE IS COMP-3	(n+1)/2 rounded up	Packed decimal
PIC 9(n) [n <= 18]	USAGE IS COMP-3	(n+1)/2 rounded up	Packed decimal*

(continued on next page)

Unscaled Data Items and Corresponding Storage Data Types (Cont.)

Unscaled Data Item			Storage Data Type
PICTURE Clause	USAGE Clause	Allocated Storage in Bytes	
PIC X(n) [n <= 65,535]	USAGE IS DISPLAY	n	ASCII Text
PIC A(n) [n <= 65,535]	USAGE IS DISPLAY	n	ASCII Text

Legend:

*The generated code treats this data type as a signed operand in all contexts except when it is a receiving-field operand. In this case, the compiler stores the absolute value of the data type.

N/A Not Applicable

Scaled Data Items and Corresponding Storage Data Types

Scaled Data Item			Storage Data Type
PICTURE Clause	USAGE Clause	Storage Allocated in Bytes	
PIC S9(n)V9(s) [(n+s) <= 18]	USAGE IS DISPLAY	n+s	Right (trailing) overpunch
PIC S9(n)V9(s) [(n+s) <= 18]	USAGE IS DISPLAY SIGN IS TRAILING	n+s	Right (trailing) overpunch
PIC S9(n)V9(s) [(n+s) <= 18]	USAGE IS DISPLAY SIGN IS LEADING	n+s	Left (leading) overpunch
PIC S9(n)V9(s) [(n+s) <= 18]	USAGE IS DISPLAY SIGN IS TRAILING SEPARATE	n+s+1	Right (trailing) separate

(continued on next page)

Scaled Data Items and Corresponding Storage Data Types (Cont.)

Scaled Data Item			Storage Data Type
PICTURE Clause	**USAGE Clause**	**Storage Allocated in Bytes**	
PIC S9(n)V9(s) [(n + s) <= 18]	USAGE IS DISPLAY SIGN IS LEADING SEPARATE	n + s + 1	Left (leading) separate
PIC 9(n)V9(s) [(n + s) <= 18]	USAGE IS DISPLAY	n + s	Unsigned numeric
PIC 9(n)V9(s) [(n + s) <= 4]	USAGE IS COMP	2	Word integer*
PIC 9(n)V9(s) [5 <= (n + s) <= 9]	USAGE IS COMP	4	Two word integer*

PIC 9(n)V9(s) [10 <= (n+s) <= 18]	USAGE IS COMP	8	Four word integer*
PIC S9(n)V9(s) [(n+s) <= 4]	USAGE IS COMP	2	Word integer
2PIC S9(n)V9(s) [5 <= (n+s) <= 9]	USAGE IS COMP	4	Two word integer
PIC S9(n)V9(s) [10 <= (n+s) <= 18]	USAGE IS COMP	8	Four word integer
PIC 9(n)V9(s) [(n+s) <= 18]	USAGE IS COMP-3	(n+s+1)/2 rounded up	Packed decimal*
PIC S9(n)V9(s) [(n+s) <= 18]	USAGE IS COMP-3	(n+s+1)/2 rounded up	Packed decimal

(continued on next page)

Scaled Data Items and Corresponding Storage Data Types (Cont.)

Legend:

* The generated code treats this data type as a signed operand in all contexts except when it is a receiving-field operand. In this case, the compiler stores the absolute value of the data type.

N/A Not Applicable

Character Sets

In the following table, characters belonging to set C constitute the COBOL character set. Set L contains those characters that can appear in nonnumeric literals. The characters in set X delimit lines of the source text.

Decimal	Octal	Character	Set
000	000	NUL	L
001	001	SOH	L
002	002	STX	L
003	003	ETX	L
004	004	EOT	L
005	005	ENQ	L
006	006	ACK	L
007	007	BEL	L
008	010	BS	L
009	011	HT	C

(continued on next page)

Character Sets (Cont.)

Decimal	Octal	Character	Set
010	012	LF	X
011	013	VT	X
012	014	FF	X
013	015	CR	X
014	016	SO	L
015	017	SI	L
016	020	DLE	L
017	021	DC1	L
018	022	DC2	L
019	023	DC3	L
020	024	DC4	L
021	025	NAK	L
022	026	SYN	L
023	027	ETB	L
024	030	CAN	L
025	031	EM	L

Decimal	Octal	Character	Set
026	032	SUB	L
027	033	ESC	L
028	034	FS	L
029	035	GS	L
030	036	RS	L
031	037	US	L
032	040	space	C, L
033	041	!	L
034	042	"	C, L
035	043	#	L
036	044	$	C, L
037	045	%	L
038	046	&	L
039	047	'	L
040	050	(	C, L

(continued on next page)

Character Sets (Cont.)

Decimal	Octal	Character	Set
041	051	)	C, L
042	052	*	C, L
043	053	+	C, L
044	054	,	C, L
045	055	–	C, L
046	056	.	C, L
047	057	/	C, L
048	060	0	C, L
049	061	1	C, L
050	062	2	C, L
051	063	3	C, L
052	064	4	C, L
053	065	5	C, L
054	066	6	C, L
055	067	7	C, L

Decimal	Octal	Character	Set
056	070	8	C, L
057	071	9	C, L
058	072	:	L
059	073	;	C, L
060	074	<	C, L
061	075	=	C, L
062	076	>	C, L
063	077	?	L
064	100	@	L
065	101	A	C, L
066	102	B	C, L
067	103	C	C, L
068	104	D	C, L
069	105	E	C, L
070	106	F	C, L
071	107	G	C, L

(continued on next page)

Character Sets (Cont.)

Decimal	Octal	Character	Set
072	110	H	C, L
073	111	I	C, L
074	112	J	C, L
075	113	K	C, L
076	114	L	C, L
077	115	M	C, L
078	116	N	C, L
079	117	O	C, L
080	120	P	C, L
081	121	Q	C, L
082	122	R	C, L
083	123	S	C, L
084	124	T	C, L
085	125	U	C, L
086	126	V	C, L
087	127	W	C, L

Decimal	Octal	Character	Set
088	130	X	C, L
089	131	Y	C, L
090	132	Z	C, L
091	133	[	L
092	134	\	L
093	135	]	L
094	136	^	L
095	137	_	L
096	140	‘	L
097	141	a	L
098	142	b	L
099	143	c	L
100	144	d	L
101	145	e	L
102	146	f	L
103	147	g	L

(continued on next page)

Character Sets (Cont.)

Decimal	Octal	Character	Set
104	150	h	L
105	151	i	L
106	152	j	L
107	153	k	L
108	154	l	L
109	155	m	L
110	156	n	L
111	157	o	L
112	160	p	L
113	161	q	L
114	162	r	L
115	163	s	L
116	164	t	L
117	165	u	L

Decimal	Octal	Character	Set
118	166	v	L
119	167	w	L
120	170	x	L
121	171	y	L
122	172	z	L
123	173	{	L
124	174	|	L
125	175	}	L
126	176	~	L
127	177	DEL	L

digital

Printed in U.S.A.