LIANT

RM/COBOL®

Syntax Summary

Version 8.0

This document provides complete syntax for all RM/COBOL commands, divisions, entries, statements, and other general formats. Use this pamphlet in conjunction with the *RM/COBOL Language Reference Manual* and the *RM/COBOL User's Guide*.

The *RM/COBOL Syntax Summary* has been prepared for all implementations of RM/COBOL. Consult the *RM/COBOL User's Guide* for all appropriate operating system rules and conventions (such as command line invocation).

Document Number 401214-0303

Table of Contents

Compile Command

The format of the Compile Command is as follows:

```
   rmcobol filename [[(] [[~]option] ... [)comment]]
```

filename is the name of the source file to be compiled.

option specifies a compiler option, described below. A tilde (~) preceding the option character negates the option. Options may be specified in either uppercase or lowercase letters. If an option is repeated in a command, the last occurrence of the option is used. Each option may be preceded by a hyphen. If any option is preceded by a hyphen, then a leading hyphen must precede all options. When assigning a value to an option, the equal sign is optional if leading hyphens are used.

comment is used to annotate the command.

Option	Description
A	Direct the compiler to generate the allocation map in the listing.
B	Define as binary sequential those sequential files not explicitly declared to be line sequential in their file control entries.
C	Suppress the inclusion of copied text in the listing.
D	Direct RM/COBOL to compile all source programs as if the WITH DEBUGGING MODE clause appeared in each compiled program.
E	Suppress the inclusion of the source program component in the listing except for lines associated with diagnostic messages.
F={(*keyword-list*)|*keyword*}	Direct the compiler to flag occurrences of these language elements: COM1 INTERMEDIATE COM2 OBSOLETE EXTENSION SEG1 HIGH SEG2 If leading hyphens are used, the parentheses are optional.

Option	Description
G=*path*	Designate a file to be used as the compiler configuration.
H=*path*	Designate a file as a supplement to the compiler configuration.
K	Suppress the banner message and the terminal error listing.
L[=*path*]	Direct the compiler to produce a listing file and optionally specify the directory in which to place the listing file.
M	Direct the compiler to suppress automatic input conversion for Format 1 and 3 ACCEPT statements with numeric operands and to suppress right justification of justified operands. Direct the compiler to suppress automatic output conversion for numeric fields of Format 3 DISPLAY statements.
N	Suppress the generation of an object program.
O=*path*	Specify the directory pathname where the object file will be placed.
P	Direct the compiler to write a copy of the listing to the printer.
Q	Direct the compiler to eliminate debugging information from generated object programs.
R	Direct the compiler to generate a sequential number in the first six columns of source records as they appear on the listing.
S	Direct the compiler to assume a separate sign when the SIGN clause is not specified for a DISPLAY usage, signed numeric data item (that is, for a data item whose character-string within a PICTURE clause begins with S).
T	Direct the compiler to write a copy of the listing to the standard output device.

Option	Description
U[=**{B**|**D**|**P}**]	Direct the compiler to assume an alternative usage for data items described as COMP or COMPUTATIONAL. The U Option specified alone or as U=B directs the compiler to assume BINARY usage for data items described as COMP or COMPUTATIONAL. The U=D Option directs the compiler to assume DISPLAY usage for items described as COMP or COMPUTATIONAL. The U=P Option directs the compiler to assume PACKED-DECIMAL usage for items described as COMP or COMPUTATIONAL.
V	Define as line sequential those sequential files not explicitly declared to be binary sequential in their file control entries.
W=*n*	Specify the amount of memory (in kilobytes) that the compiler should use for its internal table storage. *n* can be a decimal number from 32 to 16384.
X	Direct the compiler to generate a cross reference map in the listing.
Y=*n*	Direct the compiler to output the symbol table and debug line table to the object program file. *n* can be 0 to 3.
Z=*version*	Indicate the version of the RM/COBOL runtime you want to use. *version* can be 7 through 11.
2	Direct the compiler to accept source programs created for the RM/COBOL 2.*n* compiler.
7	Specify the semantic rules under which the program is to be compiled as conforming to the American National Standard COBOL 1974.

Runtime Command

The format of the Runtime Command is as follows:

```
   runcobol filename [option] ...
```

filename is the name of the main program of the run unit.

option specifies a runtime system option, described below. Options may be specified in either uppercase or lowercase letters. Each option may be preceded by a hyphen. If any option is preceded by a hyphen, then a leading hyphen must precede all options. When assigning a value to an option, the equal sign is optional if leading hyphens are used.

Option	Description
A=[*delim*] [*string*] [*delim*]	Pass an argument to the main program. The delimiter characters are optional if *string* does not contain spaces.
B=*n*	Specify a maximum buffer size for use with the ACCEPT and DISPLAY statements.
C=*pathname*	Designate a file to be used as the runtime configuration file.
D	Invoke the RM/COBOL Interactive Debugger.
I	Collect RM/COBOL program instrumentation data.
K	Suppress the banner message and the STOP RUN message.
L=*pathname*	Designate RM/COBOL non-COBOL subprogram libraries.
M	Direct that level 2 ANSI semantics are to be used for Format 1 ACCEPT and DISPLAY statements.
S=*n . . . n*	Set (or reset) the initial value of switches in the RM/COBOL program.
T=*n*	Specify the amount of memory (*n* bytes) to be used for a sort operation.
V	Direct that a trace of support modules loaded by the RM/COBOL runtime system be displayed.
X=*pathname*	Designate a file as a supplement to the runtime configuration.

Debug Commands

The Debug commands are as follows.

Command	Description
A	Set breakpoints and resume program execution from the current location. **A** [*line* [**+** *intraline*] [**,** [*prog-name*] [**,** [*count*]]]]
B	Display all currently set breakpoints or set breakpoints at specific procedural statements. **B** [*line* [**+** *intraline*] [**,** [*prog-name*] [**,** [*count*]]]]
C	Clear any breakpoints that have been set with the A or B Command. **C** [*line* [**+** *intraline*] [**,** [*prog-name*]]]
D	Display on the screen the value of a specified data item. **Identifier Format** **D** *name-1* [{ **IN** | **OF** } *name-2*] … [*script*] [*refmod*] [**,** { *type* | { ***** | **&** } [*type*] }] [**#** *alias*] **Address-Size Format** **D** [*base* **:**] *address* [**+** *occur-size* ***** *occur-num*] …**,** *size* **,** [*type*] [**#** *alias*] **Alias Format** **D** **#** *alias*
E	End Debug; the currently executing program runs until completion. **E**
L	Specify a line on the monitor screen at which command input echoes and Debug responses are to be displayed. **L** [*line-display*]

Command	Description
M	Change the value of a specified data item. **Identifier Format** **M** *name-1* [{ **IN** | **OF** } *name-2*] … [*script*] [*refmod*] [**,** { *type* | { ***** | **&** } [*type*] }] [**#** *alias*] **,** *value* **Address-Size Format** **M** [*base* **:**] *address* [**+** *occur-size* ***** *occur-num*] …**,** *size* **,** [*type*] [**#** *alias*] **,** *value* **Alias Format** **M** # *alias* **,** *value*
Q	Stop program execution. **Q**
R	Specify that program execution resume at the current location, or at another location specified in the command. **R** [*statement-address*]
S	Specify that program execution occur one step at a time. **S** [**P** | **S**][*count*]
T	Monitor the value of a specified data item, and suspend execution whenever a change in that value occurs. That is, a data trap. **Identifier Format** **T** *name-1* [{ **IN** | **OF** } *name-2*] … [*script*] [*refmod*] [**,** { *type* | { ***** | **&** } [*type*] }] [**#** *alias*] **Address-Size Format** **T** [*base* **:**] *address* [**+** *occur-size* ***** *occur-num*] …**,** *size* **,** [*type*] [**#** *alias*] **Alias Format** **T** # *alias*
U	Clear some or all currently active data traps. **Identifier Format** **U** *name-1* [{ **IN** | **OF** } *name-2*] … [*script*] [*refmod*] [**,** { *type* | { ***** | **&** } [*type*] }] **Address-Size Format** **U** [*base* **:**] *address* [**+** *occur-size* ***** *occur-num*] …**,** *size* **,** [*type*] **Alias Format** **U** # *alias*

Note In the Address-Size formats for the D, M, T, and U commands, *base* is one of the following:

- **U** *arg-num*, for a formal argument and *arg-num* is the formal argument number.
- **B** *item-num*, for a based linkage item and *item-num* is the based linkage item number.
- **G** for the GIVING formal argument.
- **X** *ext-num*, for an external data item and *ext-num* is the external item number.

Source Program General Format

```
 identification-division
[ environment-division ]
[ data-division ]
[ procedure-division ]
[ nested-source-program ]...
[ end-program-header ]
```

Identification Division General Format

```
{ IDENTIFICATION }  DIVISION.
{ ID             }

PROGRAM-ID. { program-name-1 }  [ IS {| COMMON  |} PROGRAM ] .
            { literal-1      }  [    {| INITIAL |}         ]

[ AUTHOR. [ comment-entry-1 ]... ]
[ INSTALLATION. [ comment-entry-2 ]... ]
[ DATE-WRITTEN. [ comment-entry-3 ]... ]
[ DATE-COMPILED. [ comment-entry-4 ]... ]
[ SECURITY. [ comment-entry-5 ]... ]
[ REMARKS. [ comment-entry-6 ]... ]
```

Environment Division General Format

```
[ ENVIRONMENT DIVISION. ]

[ CONFIGURATION SECTION. ]

[ SOURCE-COMPUTER. [ computer-name-1
    [ WITH DEBUGGING MODE ]. ] ]

[ OBJECT-COMPUTER. [ computer-name-2
    [ MEMORY SIZE integer-1 { WORDS      } ]
    [                       { CHARACTERS } ]
    [                       { MODULES    } ]
    [ PROGRAM COLLATING SEQUENCE IS alphabet-name-1 ]
    [ SEGMENT-LIMIT IS segment-number-1 ] . ] ]

[ SPECIAL-NAMES. [
    [                 { IS mnemonic-name-1 [ {| ON STATUS IS condition-name-1  |} ] } ]
    [                 {                    [ {| OFF STATUS IS condition-name-2 |} ] } ]
    [ switch-name-1   {                                                          } ]
    [                 { {| ON STATUS IS condition-name-1  |}                     } ]
    [                 { {| OFF STATUS IS condition-name-2 |}                     } ] ...
    [                                                                              ]
    [ feature-name-1 IS mnemonic-name-2                                            ]
    [                                                                              ]
    [ low-volume-I-O-name-1 IS mnemonic-name-3                                     ]
```

(continued on next page)

(continued from previous page)

```
[ ALPHABET alphabet-name-1 IS

      { STANDARD-1                                                                        }
      { STANDARD-2                                                                        }
      { NATIVE                                                                            }
      { code-name-1                                                                       } ] ...
      { { literal-1 [ { THROUGH } literal-2 ]                                             }
      { {             { THRU    }                                                         }
      { {                                                                                 }
      { {                 [ ALSO literal-3 [ { THROUGH } literal-4 ] ] ... } ...          }
      {                                      { THRU    }                                  }

[ SYMBOLIC [ CHARACTER  ] { { symbolic-character-1 } ... [ IS  ]
           [ CHARACTERS ]                                [ ARE ]

        { integer-1 } ...  } ...  [ IN alphabet-name-2 ] ] ...

[ CLASS class-name-1 IS { literal-5 [ { THROUGH } literal-6 ] } ... ] ...
                                      { THRU    }

[ CURRENCY SIGN IS literal-7 ]

[ DECIMAL-POINT IS COMMA ]

[ NUMERIC SIGN IS { LEADING  } [ SEPARATE CHARACTER ] ] .   ]   ]   ]
                  { TRAILING }
```

Underlined (required) keywords: STANDARD-1, STANDARD-2, NATIVE, THROUGH, THRU, ALSO, SYMBOLIC, IN, CLASS, CURRENCY, DECIMAL-POINT, COMMA, NUMERIC, SIGN, LEADING, TRAILING, SEPARATE.

(continued from previous page)

```
[ INPUT-OUTPUT SECTION. ]

  FILE-CONTROL.
      { file-control-entry-1 }...

[ I-O-CONTROL. [

    [ RERUN [ ON { file-name-1    } ]
            [    { rerun-name-1   } ]

                  { { [END OF] { REEL } } OF file-name-2 }
                  { {          { UNIT } }                }
            EVERY { { integer-1 RECORDS }                } ]...
                  {                                      }
                  { integer-2 CLOCK-UNITS                }
                  { condition-name-1                     }

    [ SAME [ RECORD     ] AREA FOR file-name-3 { file-name-4 }... ]...
           [ SORT       ]
           [ SORT-MERGE ]

    [ MULTIPLE FILE TAPE CONTAINS                          ]
    [                                                      ]... . ] ] ] ]
    [   { file-name-5 [POSITION IS integer-3] }...         ]
```

File Control Entry General Formats

File Control Entry

$$\underline{\text{SELECT}}\ \left[\left[\underline{\text{NOT}}\right]\ \underline{\text{OPTIONAL}}\right]\ \textit{file-name-1}$$

$$\underline{\text{ASSIGN}}\ \text{TO}\ \left\{\begin{array}{l} \left\{\begin{array}{l}\textit{data-name-1}\\ \textit{literal-1}\end{array}\right\} \\ \\ \left\{\begin{array}{l}\underline{\text{DISPLAY}}\\ \underline{\text{INPUT}}\\ \underline{\text{OUTPUT}}\\ \underline{\text{INPUT - OUTPUT}}\\ \underline{\text{RANDOM}}\\ \underline{\text{TAPE}}\\ \textit{device-name-1}\end{array}\right\}\ \left[\begin{array}{l}\textit{data-name-1}\\ \textit{literal-1}\end{array}\right]\end{array}\right\}$$

$$\left[\underline{\text{RESERVE}}\ \left\{\begin{array}{l}\textit{integer-1}\\ \underline{\text{NO}}\end{array}\right\}\ \left[\underline{\text{ALTERNATE}}\right]\ \left[\begin{array}{l}\text{AREA}\\ \text{AREAS}\end{array}\right]\right]$$

$$\left[\left[\underline{\text{ORGANIZATION}}\ \text{IS}\right]\ \left\{\begin{array}{l}\left[\begin{array}{l}\underline{\text{BINARY}}\\ \underline{\text{LINE}}\end{array}\right]\ \underline{\text{SEQUENTIAL}}\\ \\ \underline{\text{RELATIVE}}\\ \\ \underline{\text{INDEXED}}\end{array}\right\}\right]$$

$$\left[\underline{\text{PADDING}}\ \text{CHARACTER}\ \text{IS}\ \left\{\begin{array}{l}\textit{data-name-2}\\ \textit{literal-2}\end{array}\right\}\right]$$

$$\left[\underline{\text{RECORD}}\ \underline{\text{DELIMITER}}\ \text{IS}\ \left\{\begin{array}{l}\underline{\text{STANDARD - 1}}\\ \textit{delimiter-name-1}\end{array}\right\}\right]$$

$$\left[\underline{\text{ACCESS}}\ \text{MODE}\ \text{IS}\ \left\{\left\{\begin{array}{l}\underline{\text{SEQUENTIAL}}\\ \underline{\text{RANDOM}}\\ \underline{\text{DYNAMIC}}\end{array}\right\}\ \left[\underline{\text{RELATIVE}}\ \text{KEY}\ \text{IS}\ \textit{data-name-3}\right]\right\}\right]$$

(continued on next page)

File Control Entry *(continued from previous page)*

```
[ LOCK MODE IS

      { { MANUAL    } [ WITH LOCK ON [ MULTIPLE ] { RECORD  } ] } ]
      { { AUTOMATIC }                             { RECORDS }   } ]
      {                                                         } ]
      { EXCLUSIVE                                               }
[ CODE-SET IS alphabet-name-1 ]
[ COLLATING SEQUENCE IS alphabet-name-2 ]
[ RECORD KEY IS { data-name-4                          }
[               { split-key-name-1 = { data-name-5 }...}

        [ WITH DUPLICATES ] ]
[ ALTERNATE RECORD KEY IS { data-name-6                          }
[                         { split-key-name-2 = { data-name-7 }...}
        [ WITH DUPLICATES ] ]...
[ FILE STATUS IS data-name-8 ] .
```

Sort-Merge File Control Entry

```
SELECT file-name-1

                { { data-name-1 }                                 }
                { { literal-1   }                                 }
    ASSIGN TO   {                                                 } .
                { { SORT          }                               }
                { { SORT-MERGE    } [ data-name-1 ]               }
                { { MERGE         } [ literal-1   ]               }
                { { device-name-1 }                               }
```

Data Division General Format

```
[ DATA DIVISION. ]

[ FILE SECTION. ]

[ [ file-description-entry-1 { record-description-entry-1 }...            ]...  ]
[ [ sort-merge-file-description-entry-1 { record-description-entry-2 }... ]     ]

[ WORKING-STORAGE SECTION. ]

[ [ 77-level-description-entry-1 ]...  ]
[ [ record-description-entry-3   ]     ]

[ LINKAGE SECTION. ]

[ [ 77-level-description-entry-2 ]...  ]
[ [ record-description-entry-4   ]     ]

[ COMMUNICATION SECTION. ]

[ [ communication-description-entry-1 { record-description-entry-5 }... ]...  ]

[ SCREEN SECTION. ]

[ [ screen-description-entry-1 ]...  ]  ]
```

file-description-entry

```
FD file-name-1
    [IS EXTERNAL]
    [IS GLOBAL]
    [BLOCK CONTAINS [integer-1 TO] integer-2 {RECORDS   }]
                                             {CHARACTERS}
    [         {CONTAINS [integer-3 TO] integer-4 CHARACTERS                     }]
    [ RECORD  {IS VARYING IN SIZE                                               }]
    [         {     [[FROM integer-5] [TO integer-6] CHARACTERS]                }]
    [         {     [DEPENDING ON data-name-1]                                  }]
    [LABEL {RECORD IS   } {STANDARD}]
    [      {RECORDS ARE } {OMITTED }]
    [VALUE OF {{LABEL        } IS {data-name-2}}...]
    [         {{label-name-1 }    {literal-1  }}   ]
    [DATA {RECORD IS  } {data-name-3}...]
    [     {RECORDS ARE}                 ]
    [LINAGE IS {data-name-4} LINES [WITH FOOTING AT {data-name-5}]
    [          {integer-7  }       [                {integer-8  }]
    [      [LINES AT TOP {data-name-6}] [LINES AT BOTTOM {data-name-7 }]]
    [      [             {integer-9  }] [                {integer-10  }]]
    [CODE-SET IS alphabet-name-1] .
```

sort-merge-file-description-entry

```
SD file-name-1
    [         {CONTAINS [integer-3 TO] integer-4 CHARACTERS                     }]
    [ RECORD  {IS VARYING IN SIZE                                               }]
    [         {     [[FROM integer-5] [TO integer-6] CHARACTERS]                }]
    [         {     [DEPENDING ON data-name-1]                                  }]
    [DATA {RECORD IS  } {data-name-3}...] .
    [     {RECORDS ARE}                 ]
```

record-description-entry

```
{ data-description-entry-1 }...
```

77-level-description-entry

```
data-description-entry-2
```

data-description-entry

Format 1: Data-Name Full Declaration

```
                 [ data-name-1 ]
level-number-1   [ FILLER      ]

     [ REDEFINES data-name-2 ]
     [ IS EXTERNAL ]
     [ IS GLOBAL ]
     [ { PICTURE }                       ]
     [ { PIC     }  IS character-string-1 ]

     [                   { BINARY [ ( integer-3 ) ]              } ]
     [                   { COMPUTATIONAL                         } ]
     [                   { COMP                                  } ]
     [                   { COMPUTATIONAL-1                       } ]
     [                   { COMP-1                                } ]
     [                   { COMPUTATIONAL-3                       } ]
     [                   { COMP-3                                } ]
     [                   { COMPUTATIONAL-4 [ ( integer-3 ) ]     } ]
     [  [ USAGE IS ]     { COMP-4 [ ( integer-3 ) ]              } ]
     [                   { COMPUTATIONAL-5 [ ( integer-3 ) ]     } ]
     [                   { COMP-5 [ ( integer-3 ) ]              } ]
     [                   { COMPUTATIONAL-6                       } ]
     [                   { COMP-6                                } ]
     [                   { DISPLAY                               } ]
     [                   { INDEX                                 } ]
     [                   { PACKED-DECIMAL                        } ]
     [                   { POINTER                               } ]
```

(continued on next page)

Format 1: Data-Name Full Declaration *(continued from previous page)*

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

[ OCCURS { integer-2 TIMES                                                      } 
[        { [ integer-1 TO ] integer-2 TIMES DEPENDING ON data-name-3            }
[
[        [ { ASCENDING  } KEY IS { data-name-4 }... ]...
[        [ { DESCENDING }                           ]
[
[        [ INDEXED BY { index-name-1 }... ]                                     ]

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

[ { JUSTIFIED } RIGHT ]
[ { JUST      }       ]

[ BLANK WHEN ZERO ]
[ VALUE IS literal-1 ] .
```

Format 2: Data-Name Renames

```
66 data-name-1

    RENAMES data-name-2 [ { THROUGH } data-name-3 ] .
                        [ { THRU    }             ]
```

Format 3: Condition-Name Declaration

```
88 condition-name-1

    { VALUE IS   } { literal-1 [ { THROUGH } literal-2 ] }
    { VALUES ARE } {           [ { THRU    }           ] }...
                   {                                     }
                   { relational-operator literal-1       }

          [ WHEN SET TO FALSE IS literal-3 ] .
```

Format 4: Constant-Name Declaration

$$\text{78}\ \ \textit{constant-name-1}$$

$$\underline{\text{VALUE}}\ \ \text{IS}\ \left\{\begin{array}{l}\textit{literal-1}\\ \textit{constant-expression-1}\end{array}\right\}.$$

communication-description-entry

Format 1: Input CD

$$\underline{\text{CD}}\ \ \textit{cd-name-1}\ \ \text{FOR}\ [\underline{\text{INITIAL}}]\ \underline{\text{INPUT}}$$

$$\left[\begin{array}{l}
\left\{\left\|\begin{array}{l}
\text{SYMBOLIC}\ \underline{\text{QUEUE}}\ \text{IS}\ \textit{data-name-1}\\
\text{SYMBOLIC}\ \underline{\text{SUB-QUEUE-1}}\ \text{IS}\ \textit{data-name-2}\\
\text{SYMBOLIC}\ \underline{\text{SUB-QUEUE-2}}\ \text{IS}\ \textit{data-name-3}\\
\text{SYMBOLIC}\ \underline{\text{SUB-QUEUE-3}}\ \text{IS}\ \textit{data-name-4}\\
\underline{\text{MESSAGE}}\ \underline{\text{DATE}}\ \text{IS}\ \textit{data-name-5}\\
\underline{\text{MESSAGE}}\ \underline{\text{TIME}}\ \text{IS}\ \textit{data-name-6}\\
\text{SYMBOLIC}\ \underline{\text{SOURCE}}\ \text{IS}\ \textit{data-name-7}\\
\underline{\text{TEXT}}\ \underline{\text{LENGTH}}\ \text{IS}\ \textit{data-name-8}\\
\underline{\text{END}}\ \underline{\text{KEY}}\ \text{IS}\ \textit{data-name-9}\\
\underline{\text{STATUS}}\ \underline{\text{KEY}}\ \text{IS}\ \textit{data-name-10}\\
\text{MESSAGE}\ \underline{\text{COUNT}}\ \text{IS}\ \textit{data-name-11}
\end{array}\right\|\right\}\\
\\
\left\{\begin{array}{l}
\textit{data-name-1}\ \ \textit{data-name-2}\ \ \textit{data-name-3}\ \ \textit{data-name-4}\\
\quad\textit{data-name-5}\ \ \textit{data-name-6}\ \ \textit{data-name-7}\ \ \textit{data-name-8}\\
\quad\textit{data-name-9}\ \ \textit{data-name-10}\ \ \textit{data-name-11}
\end{array}\right\}
\end{array}\right].$$

Format 2: Output CD

$$\begin{array}{l}
\underline{\text{CD}}\ \ \textit{cd-name-1}\ \ \text{FOR}\ \underline{\text{OUTPUT}}\\
\quad[\underline{\text{DESTINATION}}\ \underline{\text{COUNT}}\ \text{IS}\ \textit{data-name-1}]\\
\quad[\underline{\text{TEXT}}\ \underline{\text{LENGTH}}\ \text{IS}\ \textit{data-name-2}]\\
\quad[\underline{\text{STATUS}}\ \underline{\text{KEY}}\ \text{IS}\ \textit{data-name-3}]\\
\quad\left[\begin{array}{l}\underline{\text{DESTINATION}}\ \underline{\text{TABLE}}\ \underline{\text{OCCURS}}\ \textit{integer-1}\ \text{TIMES}\\ \qquad[\underline{\text{INDEXED}}\ \text{BY}\ \{\textit{index-name-1}\}\cdots]\end{array}\right]\\
\quad[\underline{\text{ERROR}}\ \underline{\text{KEY}}\ \text{IS}\ \textit{data-name-4}]\\
\quad[\text{SYMBOLIC}\ \underline{\text{DESTINATION}}\ \text{IS}\ \textit{data-name-5}].
\end{array}$$

Format 3: Input-Output CD

```
CD  cd-name-1  FOR [ INITIAL ] I-O
   ⎡                                                          ⎤
   ⎢   ⎧ MESSAGE DATE IS data-name-1       ⎫                  ⎥
   ⎢   ⎪ MESSAGE TIME IS data-name-2       ⎪                  ⎥
   ⎢   ⎪ SYMBOLIC TERMINAL IS data-name-3  ⎪                  ⎥
   ⎢   ⎨ TEXT LENGTH IS data-name-4        ⎬                  ⎥ .
   ⎢   ⎪ END KEY IS data-name-5            ⎪                  ⎥
   ⎢   ⎩ STATUS KEY IS data-name-6         ⎭                  ⎥
   ⎢                                                          ⎥
   ⎢   ⎧ data-name-1  data-name-2  data-name-3  data-name-4 ⎫ ⎥
   ⎣   ⎩      data-name-5  data-name-6                      ⎭ ⎦
```

screen-description-entry

Format 1: Screen Group

```
level-number-1 ⎡ screen-name-1 ⎤
               ⎣ FILLER        ⎦
    ⎡ BACKGROUND IS color-name-1     ⎤
    ⎣ BACKGROUND-COLOR IS integer-1  ⎦
    ⎡ FOREGROUND IS color-name-2     ⎤
    ⎣ FOREGROUND-COLOR IS integer-2  ⎦
    [ [ USAGE IS ] DISPLAY ]
    ⎡ [ SIGN IS ] ⎧ LEADING  ⎫  [ SEPARATE CHARACTER ] ⎤
    ⎣             ⎩ TRAILING ⎭                         ⎦
    [ AUTO ]
    [ SECURE ]
    [ REQUIRED ]
    [ FULL ] .
    { screen-description-entry-1 } ...
```

Format 2: Screen Literal

```
                 ⎡ screen-name-1 ⎤
level-number-1   ⎣ FILLER        ⎦

      ⎡ BELL ⎤
      ⎣ BEEP ⎦

      ⎡          ⎧ SCREEN    ⎫ ⎤
      ⎢ BLANK    ⎨ LINE      ⎬ ⎥
      ⎣          ⎩ REMAINDER ⎭ ⎦

      [ BLINK ]

      ⎡          ⎧ EOS    ⎫ ⎤
      ⎢ ERASE    ⎨ EOL    ⎬ ⎥
      ⎣          ⎩ SCREEN ⎭ ⎦

      ⎡ [NO] HIGHLIGHT ⎤
      ⎣ LOWLIGHT       ⎦

      ⎡ REVERSE       ⎤
      ⎢ REVERSED      ⎥
      ⎣ REVERSE-VIDEO ⎦

      [ UNDERLINE ]

      ⎡ BACKGROUND IS color-name-1       ⎤
      ⎣ BACKGROUND-COLOR IS integer-1    ⎦

      ⎡ FOREGROUND IS color-name-2       ⎤
      ⎣ FOREGROUND-COLOR IS integer-2    ⎦

      ⎡        ⎡                ⎧ ⎡ PLUS ⎤ integer-3 ⎫ ⎤ ⎤
      ⎢ LINE   ⎢ NUMBER  IS     ⎨ ⎣ +    ⎦           ⎬ ⎥ ⎥
      ⎣        ⎣                ⎩ identifier-1       ⎭ ⎦ ⎦

      ⎡ ⎧ COLUMN ⎫ ⎡             ⎧ ⎡ PLUS ⎤ integer-4 ⎫ ⎤ ⎤
      ⎢ ⎩ COL    ⎭ ⎢ NUMBER  IS  ⎨ ⎣ +    ⎦           ⎬ ⎥ ⎥
      ⎣            ⎣             ⎩ identifier-2       ⎭ ⎦ ⎦

      [ [ VALUE IS ] literal-1 ] .
```

Format 3: Screen Field

$$\textit{level-number-1} \left[\begin{array}{l} \textit{screen-name-1} \\ \text{FILLER} \end{array} \right]$$

$$\left[\begin{array}{l} \underline{\text{BELL}} \\ \underline{\text{BEEP}} \end{array} \right]$$

$$\left[\underline{\text{BLANK}} \left\{ \begin{array}{l} \underline{\text{SCREEN}} \\ \underline{\text{LINE}} \\ \underline{\text{REMAINDER}} \end{array} \right\} \right]$$

$$\left[\underline{\text{BLINK}} \right]$$

$$\left[\underline{\text{ERASE}} \left\{ \begin{array}{l} \underline{\text{EOS}} \\ \underline{\text{EOL}} \\ \text{SCREEN} \end{array} \right\} \right]$$

$$\left[\begin{array}{l} [\underline{\text{NO}}]\ \underline{\text{HIGHLIGHT}} \\ \underline{\text{LOWLIGHT}} \end{array} \right]$$

$$\left[\begin{array}{l} \underline{\text{REVERSE}} \\ \underline{\text{REVERSED}} \\ \underline{\text{REVERSE-VIDEO}} \end{array} \right]$$

$$\left[\underline{\text{UNDERLINE}} \right]$$

$$\left[\begin{array}{l} \underline{\text{BACKGROUND}}\ \text{IS}\ \textit{color-name-1} \\ \underline{\text{BACKGROUND-COLOR}}\ \text{IS}\ \textit{integer-1} \end{array} \right]$$

$$\left[\begin{array}{l} \underline{\text{FOREGROUND}}\ \text{IS}\ \textit{color-name-2} \\ \underline{\text{FOREGROUND-COLOR}}\ \text{IS}\ \textit{integer-2} \end{array} \right]$$

$$\left[\underline{\text{LINE}} \left[\text{NUMBER IS} \left\{ \begin{array}{l} \left[\begin{array}{l} \underline{\text{PLUS}} \\ + \end{array} \right] \textit{integer-3} \\ \textit{identifier-1} \end{array} \right\} \right] \right]$$

$$\left[\left\{ \begin{array}{l} \underline{\text{COLUMN}} \\ \underline{\text{COL}} \end{array} \right\} \left[\text{NUMBER IS} \left\{ \begin{array}{l} \left[\begin{array}{l} \underline{\text{PLUS}} \\ + \end{array} \right] \textit{integer-4} \\ \textit{identifier-2} \end{array} \right\} \right] \right]$$

(continued on next page)

Format 3: Screen Field *(continued from previous page)*

```
                                           { { | FROM { identifier-7 } | } }
{ PICTURE }                                { { |      { literal-1    } | } }
{ PIC     } IS character-string-1          { { | TO identifier-8       | } }
                                           { USING identifier-9           }
[ [ USAGE IS ] DISPLAY ]
[ BLANK WHEN ZERO ]
[ { JUSTIFIED }       ]
[ { JUST      } RIGHT ]
[ [ SIGN IS ] { LEADING  } [ SEPARATE CHARACTER ] ]
[             { TRAILING }                        ]
[ AUTO ]
[ SECURE ]
[ REQUIRED ]
[ FULL ] .
```

Procedure Division General Formats

Format 1: Declaratives or Sections

```
                                  ┌ ┌ USING { data-name-1 }...                    ┐ ┐
[ PROCEDURE DIVISION               │ {                                             } │ .
                                  └ └ { GIVING    } data-name-2                   ┘ ┘
                                        { RETURNING }

[ DECLARATIVES.
{ section-name-1 SECTION [ segment-number-1 ].
      USE-statement-1.
[ paragraph-name-1.

      [ sentence-1 ]... ]... }...
  END DECLARATIVES. ]
{ section-name-2 SECTION [ segment-number-2 ].
[ paragraph-name-2.
       [ sentence-2 ]... ]... }...  ]
```

Format 2: Paragraphs

```
                                  ┌ ┌ USING { data-name-1 }...                    ┐ ┐
[ PROCEDURE DIVISION               │ {                                             } │ .
                                  └ └ { GIVING    } data-name-2                   ┘ ┘
                                        { RETURNING }
{ paragraph-name-3.
       [ sentence-3 ]... }...  ]
```

General Formats for COBOL Statements

The following sections describe the formats for COBOL statements.

ACCEPT Statement

Format 1: Accept From System-Name

$$\underline{\text{ACCEPT}}\ \ \textit{identifier-1}\ \ \left[\ \underline{\text{FROM}}\ \left\{\begin{array}{l}\textit{mnemonic-name-3}\\ \textit{low-volume-I-O-name-1}\end{array}\right\}\right]\ \ \left[\ \underline{\text{END - ACCEPT}}\ \right]$$

Format 2: Accept From Implicit Definition

$$\underline{\text{ACCEPT}}\ \ \textit{identifier-2}\ \ \underline{\text{FROM}}\ \left\{\begin{array}{l}\underline{\text{CENTURY - DATE}}\\ \underline{\text{CENTURY - DAY}}\\ \underline{\text{DATE}}\ \left[\ \underline{\text{YYYYMMDD}}\ \right]\\ \underline{\text{DATE - AND - TIME}}\\ \underline{\text{DATE - COMPILED}}\\ \underline{\text{DAY}}\ \left[\ \underline{\text{YYYYDDD}}\ \right]\\ \underline{\text{DAY - AND - TIME}}\\ \underline{\text{DAY - OF - WEEK}}\\ \underline{\text{ESCAPE}}\ \ \text{KEY}\\ \underline{\text{EXCEPTION}}\ \ \text{STATUS}\\ \underline{\text{TIME}}\end{array}\right\}\ \ \left[\ \underline{\text{END - ACCEPT}}\ \right]$$

Format 3: Accept Terminal I-O

```
                                                          ‖ NO { BEEP }                              ‖
                                                          ‖    { BELL }                              ‖
                                                          ‖ BLINK                                    ‖
                                                          ‖ { COLUMN   } { identifier-3 }            ‖
                                                          ‖ { COL      } { literal-2    }            ‖
                                                          ‖ { POSITION }                             ‖
                                                          ‖ CONTROL { identifier-4 }                 ‖
                                                          ‖         { literal-3    }                 ‖
                                                          ‖ CONVERT                                  ‖
                                                          ‖ CURSOR { identifier-5 }                  ‖
                                                          ‖        { literal-4    }                  ‖
                                                          ‖ ECHO                                     ‖
                                                          ‖ ERASE [ EOL ]                            ‖
                                                          ‖       [ EOS ]                            ‖
                                                          ‖ { HIGH      }                            ‖
ACCEPT { identifier-1 [ UNIT { identifier-2 } ]   [       ‖ { HIGHLIGHT }                            ‖ ]  } . . .
       {                     { literal-1    }             ‖ { LOW       }                            ‖
                                                          ‖ { LOWLIGHT  }                            ‖
                                                          ‖ { SECURE    }                            ‖
                                                          ‖ { OFF       }                            ‖
                                                          ‖ LINE { identifier-6 }                    ‖
                                                          ‖      { literal-5    }                    ‖
                                                          ‖ PROMPT [ literal-6 ]                     ‖
                                                          ‖ { REVERSE         }                      ‖
                                                          ‖ { REVERSED        }                      ‖
                                                          ‖ { REVERSE - VIDEO }                      ‖
                                                          ‖ SIZE { identifier-7 }                    ‖
                                                          ‖      { literal-7    }                    ‖
                                                          ‖ TAB                                      ‖
                                                          ‖ BEFORE TIME { identifier-8 }             ‖
                                                          ‖             { literal-8    }             ‖
                                                          ‖ UPDATE                                   ‖

   [ ON { EXCEPTION } [ identifier-9 ] { imperative-statement-1 }
   [    { ESCAPE    }                  { NEXT SENTENCE          }

        [ NOT ON { EXCEPTION } imperative-statement-2 ] ]      [ END - ACCEPT ]
        [        { ESCAPE    }                        ] ]
```

Format 4: Accept Input CD Message Count

$$\underline{\text{ACCEPT}}\ \textit{cd-name-1}\ \ \text{MESSAGE}\ \underline{\text{COUNT}}\ \ \left[\underline{\text{END-ACCEPT}}\right]$$

Format 5: Accept Screen-Name

$$\underline{\text{ACCEPT}}\ \textit{screen-name-1}\ \left[\text{AT}\left\{\left\|\begin{array}{l}\underline{\text{LINE}}\ \text{NUMBER}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{integer-1}\end{array}\right\}\\ \\ \left\{\begin{array}{l}\underline{\text{COLUMN}}\\ \underline{\text{COL}}\end{array}\right\}\text{NUMBER}\left\{\begin{array}{l}\textit{identifier-2}\\ \textit{integer-2}\end{array}\right\}\end{array}\right\|\right\}\right]$$

$$\left[\text{ON}\left\{\begin{array}{l}\underline{\text{EXCEPTION}}\\ \underline{\text{ESCAPE}}\end{array}\right\}\textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \text{ON}\left\{\begin{array}{l}\underline{\text{EXCEPTION}}\\ \underline{\text{ESCAPE}}\end{array}\right\}\textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-ACCEPT}}\right]$$

ADD Statement

Format 1: Add…To

$$\underline{\text{ADD}} \left\{\begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array}\right\}\cdots \quad \underline{\text{TO}} \quad \left\{\textit{identifier-2} \; \left[\underline{\text{ROUNDED}}\right]\right\}\cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END - ADD}}\right]$$

Format 2: Add…Giving

$$\underline{\text{ADD}} \left\{\begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array}\right\}\cdots \quad \text{TO} \quad \left\{\begin{array}{l} \textit{identifier-2} \\ \textit{literal-2} \end{array}\right\}\cdots$$

$$\underline{\text{GIVING}} \; \left\{\textit{identifier-3} \; \left[\underline{\text{ROUNDED}}\right]\right\}\cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END - ADD}}\right]$$

Format 3: Add Corresponding

$$\underline{\text{ADD}} \left\{\begin{array}{l} \underline{\text{CORRESPONDING}} \\ \underline{\text{CORR}} \end{array}\right\} \; \textit{identifier-1} \quad \underline{\text{TO}} \quad \textit{identifier-2} \; \left[\underline{\text{ROUNDED}}\right]$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END - ADD}}\right]$$

ALTER Statement

$$\underline{\text{ALTER}} \; \left\{\textit{procedure-name-1} \quad \underline{\text{TO}} \quad \left[\underline{\text{PROCEED}} \; \underline{\text{TO}}\right] \; \textit{procedure-name-2}\right\}\cdots$$

CALL Statement

Format 1: Call…On Overflow

```
                                        { [ BY REFERENCE ] { identifier-2 } ...        }
                                        {                  { OMITTED      }            }
                                        {                                              }
                                        {                  { identifier-2 }            }
                            || USING    {  BY CONTENT      { literal-2    } ...        } ...  ||
CALL { identifier-1 }    [  ||          {                  { OMITTED      }            }      ||  ]
     { literal-1    }    [  ||          {  { identifier-2 }                            }      ||  ]
                            ||          {  { literal-2    } ...                        }      ||
                            ||          {  { OMITTED      }                            }      ||
                            ||                                                                ||
                            ||   { GIVING    }                                                ||
                            ||   { RETURNING } identifier-3                                   ||

     [ ON OVERFLOW imperative-statement-1 ]

     [ END-CALL ]
```

Format 2: Call…On Exception

```
                                        { [ BY REFERENCE ] { identifier-2 } ...        }
                                        {                  { OMITTED      }            }
                                        {                                              }
                                        {                  { identifier-2 }            }
                            || USING    {  BY CONTENT      { literal-2    } ...        } ...  ||
CALL { identifier-1 }    [  ||          {                  { OMITTED      }            }      ||  ]
     { literal-1    }    [  ||          {  { identifier-2 }                            }      ||  ]
                            ||          {  { literal-2    } ...                        }      ||
                            ||          {  { OMITTED      }                            }      ||
                            ||                                                                ||
                            ||   { GIVING    }                                                ||
                            ||   { RETURNING } identifier-3                                   ||

     [ ON EXCEPTION imperative-statement-1 ]

     [ NOT ON EXCEPTION imperative-statement-2 ]

     [ END-CALL ]
```

CALL PROGRAM Statement

$$\underline{\text{CALL}}\ \ \underline{\text{PROGRAM}} \left\{\begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array}\right\} \left[\ \underline{\text{USING}} \left\{\begin{array}{l} \textit{identifier-2} \\ \textit{literal-2} \\ \underline{\text{OMITTED}} \end{array}\right\}\cdots\right]$$

$$\left[\text{ON}\ \ \underline{\text{EXCEPTION}}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{END-CALL}}\right]$$

CANCEL Statement

$$\underline{\text{CANCEL}} \left\{\begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array}\right\}\cdots$$

CLOSE Statement

$$\underline{\text{CLOSE}} \left\{ \textit{file-name-1} \left[\begin{array}{l} \left\{\begin{array}{l} \underline{\text{REEL}} \\ \underline{\text{UNIT}} \end{array}\right\} \left[\begin{array}{l} \text{WITH}\ \ \underline{\text{NO}}\ \ \underline{\text{REWIND}} \\ \text{FOR}\ \ \underline{\text{REMOVAL}} \end{array}\right] \\ \\ \text{WITH} \left\{\begin{array}{l} \underline{\text{NO}}\ \ \underline{\text{REWIND}} \\ \underline{\text{LOCK}} \end{array}\right\} \end{array}\right] \right\}\cdots$$

COMPUTE Statement

$$\underline{\text{COMPUTE}} \left\{\textit{identifier-1}\ \left[\underline{\text{ROUNDED}}\right]\right\}\cdots\ =\ \textit{arithmetic-expression-1}$$

$$\left[\text{ON}\ \ \underline{\text{SIZE}}\ \ \underline{\text{ERROR}}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \text{ON}\ \ \underline{\text{SIZE}}\ \ \underline{\text{ERROR}}\ \ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-COMPUTE}}\right]$$

CONTINUE Statement

$$\underline{\text{CONTINUE}}$$

DELETE Statement

$$\underline{\text{DELETE}}\ \ \textit{file-name-1}\ \ \text{RECORD}$$

$$\left[\underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \ \underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-DELETE}}\right]$$

DELETE FILE Statement

$$\underline{\text{DELETE}}\ \ \underline{\text{FILE}}\ \left\{\textit{file-name-2}\right\}\cdots\quad\left[\underline{\text{END-DELETE}}\right]$$

DISABLE Statement

$$\underline{\text{DISABLE}}\ \left[\begin{array}{l}\underline{\text{INPUT}}\ \left[\underline{\text{TERMINAL}}\right]\\ \underline{\text{I-O}}\ \ \underline{\text{TERMINAL}}\\ \underline{\text{OUTPUT}}\\ \underline{\text{TERMINAL}}\end{array}\right]\ \textit{cd-name-1}\ \left[\text{WITH}\ \ \underline{\text{KEY}}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\right]$$

DISPLAY Statement

Format 1: Display Upon System-Name

$$\underline{\text{DISPLAY}}\ \left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\cdots\ \left[\underline{\text{UPON}}\left\{\begin{array}{l}\textit{mnemonic-name-3}\\ \textit{low-volume-I-O-name-1}\end{array}\right\}\right]$$

$$\left[\text{WITH}\ \ \underline{\text{NO}}\ \ \underline{\text{ADVANCING}}\right]$$

Format 2: Display Terminal I-O

```
                                                    ⎧ ⎧ BEEP ⎫                                  ⎫
                                                    ⎪ ⎩ BELL ⎭                                  ⎪
                                                    ⎪ BLINK                                     ⎪
                                                    ⎪ ⎧ COLUMN   ⎫ ⎧identifier-3⎫              ⎪
                                                    ⎪ ⎨ COL      ⎬ ⎩literal-3   ⎭              ⎪
                                                    ⎪ ⎩ POSITION ⎭                              ⎪
           ⎧                                      ⎡ ⎪ CONTROL ⎧identifier-4⎫                    ⎪ ⎤ ⎫
           ⎪                                      ⎢ ⎪         ⎩literal-4   ⎭                    ⎪ ⎥ ⎪
           ⎪                                      ⎢ ⎪ CONVERT                                   ⎪ ⎥ ⎪
           ⎪                                      ⎢ ⎪ ERASE ⎡ EOL ⎤                             ⎪ ⎥ ⎪
DISPLAY    ⎨ ⎧identifier-1⎫ ⎡UNIT ⎧identifier-2⎫⎤ ⎢ ⎨       ⎣ EOS ⎦                             ⎬ ⎥ ⎬ . . .
           ⎪ ⎩literal-1   ⎭ ⎣     ⎩literal-2   ⎭⎦ ⎢ ⎪ ⎧ HIGH      ⎫                             ⎪ ⎥ ⎪
           ⎪                                      ⎢ ⎪ ⎪ HIGHLIGHT ⎪                             ⎪ ⎥ ⎪
           ⎪                                      ⎢ ⎪ ⎨ LOW       ⎬                             ⎪ ⎥ ⎪
           ⎪                                      ⎣ ⎪ ⎩ LOWLIGHT  ⎭                             ⎪ ⎦ ⎪
           ⎩                                        ⎪ LINE ⎧identifier-5⎫                       ⎪   ⎭
                                                    ⎪      ⎩literal-5   ⎭                       ⎪
                                                    ⎪ ⎧ REVERSE       ⎫                         ⎪
                                                    ⎪ ⎨ REVERSED      ⎬                         ⎪
                                                    ⎪ ⎩ REVERSE-VIDEO ⎭                         ⎪
                                                    ⎪ SIZE ⎧identifier-6⎫                       ⎪
                                                    ⎩      ⎩literal-6   ⎭                       ⎭
```

Format 3: Display Screen-Name

```
          ⎧               ⎡    ⎧ LINE NUMBER ⎧ identifier-1 ⎫                          ⎫ ⎤ ⎫
          ⎪               ⎢    ⎪             ⎩ integer-1    ⎭                          ⎪ ⎥ ⎪
DISPLAY   ⎨ screen-name-1 ⎢ AT ⎨                                                       ⎬ ⎥ ⎬ . . .
          ⎪               ⎢    ⎪ ⎧ COLUMN ⎫ NUMBER ⎧ identifier-2 ⎫                    ⎪ ⎥ ⎪
          ⎩               ⎣    ⎩ ⎩ COL    ⎭        ⎩ integer-2    ⎭                    ⎭ ⎦ ⎭
```

DIVIDE Statement

Format 1: Divide…Into

$$\underline{\text{DIVIDE}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\} \; \underline{\text{INTO}} \; \left\{ \textit{identifier-2} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-DIVIDE}} \right]$$

Format 2: Divide…Into…Giving

$$\underline{\text{DIVIDE}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\} \; \underline{\text{INTO}} \; \left\{ \begin{array}{l} \textit{identifier-2} \\ \textit{literal-2} \end{array} \right\}$$

$$\underline{\text{GIVING}} \; \left\{ \textit{identifier-3} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-DIVIDE}} \right]$$

Format 3: Divide…By…Giving

$$\underline{\text{DIVIDE}} \left\{ \begin{array}{l} \textit{identifier-2} \\ \textit{literal-2} \end{array} \right\} \; \underline{\text{BY}} \; \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\}$$

$$\underline{\text{GIVING}} \; \left\{ \textit{identifier-3} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \; \underline{\text{ERROR}} \; \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-DIVIDE}} \right]$$

Format 4: Divide…Into…Giving…Remainder

```
DIVIDE { identifier-1 }  INTO { identifier-2 }
       { literal-1    }       { literal-2    }

    GIVING identifier-3 [ROUNDED]  REMAINDER identifier-4

    [ON SIZE ERROR imperative-statement-1]

    [NOT ON SIZE ERROR imperative-statement-2]

    [END-DIVIDE]
```

Format 5: Divide…By…Giving…Remainder

```
DIVIDE { identifier-2 }  BY { identifier-1 }
       { literal-2    }     { literal-1    }

    GIVING identifier-3 [ROUNDED]  REMAINDER identifier-4

    [ON SIZE ERROR imperative-statement-1]

    [NOT ON SIZE ERROR imperative-statement-2]

    [END-DIVIDE]
```

ENABLE Statement

```
        [ INPUT [TERMINAL] ]
ENABLE  [ I-O TERMINAL     ]  cd-name-1  [ WITH KEY { identifier-1 } ]
        [ OUTPUT           ]             [          { literal-1    } ]
        [ TERMINAL         ]
```

ENTER Statement

```
ENTER language-name-1  [routine-name-1]
```

EVALUATE Statement

```
            { identifier-1  }   [      { identifier-2 } ]
            { literal-1     }   [      { literal-2    } ]
EVALUATE    { expression-1  }   [ ALSO { expression-2 } ] ...
            { TRUE          }   [      { TRUE         } ]
            { FALSE         }   [      { FALSE        } ]

  {  {        { ANY                                                                                                  }
  {  {        { condition-1                                                                                          }
  {  { WHEN   { TRUE                                                                                                 }
  {  {        { FALSE                                                                                                }
  {           { [ NOT ] { { identifier-3              } [ { THROUGH } { identifier-4              } ] }              }
  {           {         { { literal-3                 } [ { THRU    } { literal-4                 } ] }              }
  {           {         { { arithmetic-expression-1   } [             { arithmetic-expression-2   } ] }              }

     [        { ANY                                                                                                  } ]     }
     [        { condition-2                                                                                          } ]     }
     [ ALSO   { TRUE                                                                                                 } ] ... } ...
     [        { FALSE                                                                                                } ]     }
              { [ NOT ] { { identifier-5              } [ { THROUGH } { identifier-6              } ] }              }
              {         { { literal-5                 } [ { THRU    } { literal-6                 } ] }              }
              {         { { arithmetic-expression-3   } [             { arithmetic-expression-4   } ] }              }

            imperative-statement-1 } ...

  [ WHEN OTHER imperative-statement-2 ]

  [ END-EVALUATE ]
```

EXIT Statement

Format 1: Exit Paragraph

$\underline{\text{EXIT}}$

Format 2: Exit Program

$\underline{\text{EXIT}}\ \ \underline{\text{PROGRAM}}$

Format 3: Exit In-Line Perform

$\underline{\text{EXIT}}\ \ \underline{\text{PERFORM}}\ \left[\ \underline{\text{CYCLE}}\ \right]$

Format 4: Exit Paragraph or Section

$$\underline{\text{EXIT}}\ \left\{ \begin{array}{l} \underline{\text{PARAGRAPH}} \\ \underline{\text{SECTION}} \end{array} \right\}$$

GOBACK Statement

$\underline{\text{GOBACK}}$

GO TO Statement

Format 1: Go To (Alterable)

$\underline{\text{GO}}\ \ \text{TO}\ \left[\ \textit{procedure-name-1}\ \right]$

Format 2: Go To (Non-Alterable)

$\underline{\text{GO}}\ \ \text{TO}\ \ \textit{procedure-name-1}$

Format 3: Go To…Depending On

$\underline{\text{GO}}\ \ \text{TO}\ \left\{ \textit{procedure-name-1} \right\}\cdots\ \ \underline{\text{DEPENDING}}\ \ \text{ON}\ \ \textit{identifier-1}$

IF Statement

```
                        { statement-1    }
IF condition-1   THEN   { NEXT SENTENCE  }

      [        { statement-2    } ]
      [ ELSE   { NEXT SENTENCE  } ]

      [ END-IF ]
```

INITIALIZE Statement

```
INITIALIZE  {identifier-1}...  [ WITH FILLER ]

      [ { ALL           }              ]
      [ { category-name }  TO  VALUE   ]

      [                  {                     { identifier-2 } }...  ]
      [ THEN  REPLACING  { category-name DATA BY { literal-1    } }     ]

      [ THEN  TO  DEFAULT ]
```

where *category name* is:

```
{ ALPHABETIC            }
{ ALPHANUMERIC          }
{ ALPHANUMERIC-EDITED   }
{ DATA-POINTER          }
{ NUMERIC               }
{ NUMERIC-EDITED        }
```

INSPECT Statement

Format 1: Inspect…Tallying

$$\underline{\text{INSPECT}}\ \ \textit{identifier-1}\ \ \ \underline{\text{TALLYING}}$$

$$\left\{ \textit{identifier-2}\ \ \underline{\text{FOR}} \left\{ \begin{array}{l} \underline{\text{CHARACTERS}} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \\ \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{literal-1} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \right\} \ldots \end{array} \right\} \ldots \right\} \ldots$$

Format 2: Inspect…Replacing

$$\underline{\text{INSPECT}}\ \ \textit{identifier-1}\ \ \ \underline{\text{REPLACING}}$$

$$\left\{ \begin{array}{l} \underline{\text{CHARACTERS}}\ \ \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{literal-3} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \\ \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \\ \underline{\text{FIRST}} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{literal-1} \end{array} \right\} \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{literal-3} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \right\} \ldots \end{array} \right\} \ldots$$

Format 3: Inspect…Tallying…Replacing

$$\underline{\text{INSPECT}}\ \ \textit{identifier-1}\ \ \ \underline{\text{TALLYING}}$$

$$\left\{ \textit{identifier-2}\ \ \underline{\text{FOR}} \left\{ \begin{array}{l} \underline{\text{CHARACTERS}} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \\ \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{literal-1} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \right\} \ldots \end{array} \right\} \ldots \right\} \ldots$$

$$\underline{\text{REPLACING}}$$

$$\left\{ \begin{array}{l} \underline{\text{CHARACTERS}}\ \ \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{literal-3} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \\ \left\{ \begin{array}{l} \underline{\text{ALL}} \\ \underline{\text{LEADING}} \\ \underline{\text{FIRST}} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{literal-1} \end{array} \right\} \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{literal-3} \end{array} \right\} \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\} \right] \ldots \right\} \ldots \end{array} \right\} \ldots$$

Format 4: Inspect…Converting

$$\underline{\text{INSPECT}}\ \ \textit{identifier-1}\ \ \ \underline{\text{CONVERTING}}$$

$$\left\{\begin{array}{l}\textit{identifier-6}\\ \textit{literal-4}\end{array}\right\}\ \ \underline{\text{TO}}\ \ \left\{\begin{array}{l}\textit{identifier-7}\\ \textit{literal-5}\end{array}\right\}\ \left[\left\{\begin{array}{l}\underline{\text{BEFORE}}\\ \underline{\text{AFTER}}\end{array}\right\}\text{ INITIAL }\left\{\begin{array}{l}\textit{identifier-4}\\ \textit{literal-2}\end{array}\right\}\right]\cdots$$

MERGE Statement

$$\underline{\text{MERGE}}\ \ \textit{file-name-1}\ \left\{\text{ON }\left\{\begin{array}{l}\underline{\text{ASCENDING}}\\ \underline{\text{DESCENDING}}\end{array}\right\}\text{ KEY }\left\{\textit{data-name-1}\right\}\cdots\right\}\cdots$$

$$\left[\text{COLLATING }\underline{\text{SEQUENCE}}\text{ IS }\ \textit{alphabet-name-1}\right]$$

$$\underline{\text{USING}}\ \ \textit{file-name-2}\ \left\{\textit{file-name-3}\right\}\cdots$$

$$\left\{\begin{array}{l}\underline{\text{OUTPUT}}\ \underline{\text{PROCEDURE}}\text{ IS }\textit{procedure-name-1}\ \left[\left\{\begin{array}{l}\underline{\text{THROUGH}}\\ \underline{\text{THRU}}\end{array}\right\}\textit{procedure-name-2}\right]\\ \underline{\text{GIVING}}\ \left\{\textit{file-name-4}\right\}\cdots\end{array}\right\}$$

MOVE Statement

Format 1: Move…To

$$\underline{\text{MOVE}}\ \left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\ \ \underline{\text{TO}}\ \ \left\{\textit{identifier-2}\right\}\cdots$$

Format 2: Move Corresponding

$$\underline{\text{MOVE}}\ \left\{\begin{array}{l}\underline{\text{CORRESPONDING}}\\ \underline{\text{CORR}}\end{array}\right\}\ \textit{identifier-1}\ \ \underline{\text{TO}}\ \ \left\{\textit{identifier-2}\right\}\cdots$$

MULTIPLY Statement

Format 1: Multiply…By

$$\underline{\text{MULTIPLY}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\} \ \underline{\text{BY}} \ \left\{ \textit{identifier-2} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-MULTIPLY}} \right]$$

Format 2: Multiply…Giving

$$\underline{\text{MULTIPLY}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\} \ \underline{\text{BY}} \ \left\{ \begin{array}{l} \textit{identifier-2} \\ \textit{literal-2} \end{array} \right\}$$

$$\underline{\text{GIVING}} \ \left\{ \textit{identifier-3} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON } \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \text{ ON } \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-MULTIPLY}} \right]$$

OPEN Statement

$$\underline{\text{OPEN}} \left[\underline{\text{EXCLUSIVE}} \right]$$

$$\left\{ \begin{array}{l} \underline{\text{INPUT}} \left\{ \textit{file-name-1} \left[\text{WITH } \underline{\text{LOCK}} \right] \left[\begin{array}{l} \underline{\text{REVERSED}} \\ \text{WITH } \underline{\text{NO}} \ \underline{\text{REWIND}} \end{array} \right] \right\} \cdots \\ \underline{\text{OUTPUT}} \left\{ \textit{file-name-2} \left[\text{WITH } \underline{\text{LOCK}} \right] \left[\text{WITH } \underline{\text{NO}} \ \underline{\text{REWIND}} \right] \right\} \cdots \\ \underline{\text{I-O}} \left\{ \textit{file-name-3} \left[\text{WITH } \underline{\text{LOCK}} \right] \right\} \cdots \\ \underline{\text{EXTEND}} \left\{ \textit{file-name-4} \left[\text{WITH } \underline{\text{LOCK}} \right] \right\} \cdots \end{array} \right\} \cdots$$

PERFORM Statement

Format 1: Perform (Once)

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

$$\left[\textit{imperative-statement-1} \quad \underline{\text{END-PERFORM}} \right]$$

Format 2: Perform…Times

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

$$\left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{integer-1} \end{array} \right\} \underline{\text{TIMES}}$$

$$\left[\textit{imperative-statement-1} \quad \underline{\text{END-PERFORM}} \right]$$

Format 3: Perform…Until

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

$$\left[\text{WITH} \ \underline{\text{TEST}} \left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \right] \underline{\text{UNTIL}} \ \textit{condition-1}$$

$$\left[\textit{imperative-statement-1} \quad \underline{\text{END-PERFORM}} \right]$$

Format 4: Perform…Varying

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

$$\left[\text{WITH } \underline{\text{TEST}} \left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \right]$$

$$\underline{\text{VARYING}} \left\{ \begin{array}{l} \textit{identifier-2} \\ \textit{index-name-1} \end{array} \right\} \ \underline{\text{FROM}} \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{index-name-2} \\ \textit{literal-1} \end{array} \right\} \ \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-4} \\ \textit{literal-2} \end{array} \right\}$$

$$\underline{\text{UNTIL}} \ \textit{condition-1}$$

$$\left[\begin{array}{l} \underline{\text{AFTER}} \left\{ \begin{array}{l} \textit{identifier-5} \\ \textit{index-name-3} \end{array} \right\} \ \underline{\text{FROM}} \left\{ \begin{array}{l} \textit{identifier-6} \\ \textit{index-name-4} \\ \textit{literal-3} \end{array} \right\} \ \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-7} \\ \textit{literal-4} \end{array} \right\} \\ \\ \qquad \underline{\text{UNTIL}} \ \textit{condition-2} \end{array} \right] \cdots$$

$$\left[\textit{imperative-statement-1} \quad \underline{\text{END-PERFORM}} \right]$$

PURGE Statement

$$\underline{\text{PURGE}} \ \textit{cd-name-1}$$

READ Statement

Format 1: Read Sequential Access

$$\underline{\text{READ}}\ \textit{file-name-1}\ \left[\begin{array}{l}\underline{\text{NEXT}}\\ \underline{\text{PREVIOUS}}\end{array}\right]\ \text{RECORD}\ \left[\left\{\left\|\begin{array}{l}\text{WITH}\ [\underline{\text{NO}}]\ \underline{\text{LOCK}}\\ \underline{\text{INTO}}\ \textit{identifier-1}\end{array}\right\|\right\}\right]$$

$$[\text{AT}\ \underline{\text{END}}\ \textit{imperative-statement-1}]$$

$$[\underline{\text{NOT}}\ \text{AT}\ \underline{\text{END}}\ \textit{imperative-statement-2}]$$

$$[\underline{\text{END-READ}}]$$

Format 2: Read Random Access

$$\underline{\text{READ}}\ \textit{file-name-1}\ \text{RECORD}\ \left[\left\{\left\|\begin{array}{l}\text{WITH}\ [\underline{\text{NO}}]\ \underline{\text{LOCK}}\\ \underline{\text{INTO}}\ \textit{identifier-1}\end{array}\right\|\right\}\right]$$

$$\left[\underline{\text{KEY}}\ \text{IS}\ \left\{\begin{array}{l}\textit{data-name-1}\\ \textit{split-key-name-1}\end{array}\right\}\right]$$

$$[\underline{\text{INVALID}}\ \text{KEY}\ \textit{imperative-statement-1}]$$

$$[\underline{\text{NOT}}\ \underline{\text{INVALID}}\ \text{KEY}\ \textit{imperative-statement-2}]$$

$$[\underline{\text{END-READ}}]$$

RECEIVE Statement

$$\underline{\text{RECEIVE}}\ \textit{cd-name-1}\ \left\{\begin{array}{l}\underline{\text{MESSAGE}}\\ \underline{\text{SEGMENT}}\end{array}\right\}\ \underline{\text{INTO}}\ \textit{identifier-1}$$

$$[\underline{\text{NO}}\ \underline{\text{DATA}}\ \textit{imperative-statement-1}]$$

$$[\text{WITH}\ \underline{\text{DATA}}\ \textit{imperative-statement-2}]$$

$$[\underline{\text{END-RECEIVE}}]$$

RELEASE Statement

$$\underline{\text{RELEASE}}\ \ \textit{record-name-1}\ \ \left[\ \underline{\text{FROM}}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\right]$$

RETURN Statement

$$\underline{\text{RETURN}}\ \ \textit{file-name-1}\ \ \text{RECORD}\ \ \left[\underline{\text{INTO}}\ \ \textit{identifier-1}\right]$$

$$\left[\text{AT}\ \ \underline{\text{END}}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \text{AT}\ \ \underline{\text{END}}\ \ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-RETURN}}\right]$$

REWRITE Statement

$$\underline{\text{REWRITE}}\ \ \textit{record-name-1}\ \ \left[\ \underline{\text{FROM}}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\right]$$

$$\left[\underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \ \underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-REWRITE}}\right]$$

SEARCH Statement

Format 1: Search (Serial)

$$\underline{\text{SEARCH}}\ \textit{identifier-1}\ \left[\ \underline{\text{VARYING}}\left\{\begin{array}{l}\textit{identifier-2}\\ \textit{index-name-1}\end{array}\right\}\right]$$

$$\left[\text{AT}\ \underline{\text{END}}\ \textit{imperative-statement-1}\right]$$

$$\left\{\underline{\text{WHEN}}\ \textit{condition-1}\left\{\begin{array}{l}\textit{imperative-statement-2}\\ \underline{\text{NEXT}}\ \underline{\text{SENTENCE}}\end{array}\right\}\right\}\ldots$$

$$\left[\underline{\text{END-SEARCH}}\right]$$

Format 2: Search All (Binary)

$$\underline{\text{SEARCH}}\ \underline{\text{ALL}}\ \textit{identifier-1}$$

$$\left[\text{AT}\ \underline{\text{END}}\ \textit{imperative-statement-1}\right]$$

$$\underline{\text{WHEN}}\left\{\begin{array}{l}\textit{data-name-1}\left\{\begin{array}{l}\text{IS}\ \underline{\text{EQUAL}}\ \text{TO}\\ \text{IS}\ =\end{array}\right\}\left\{\begin{array}{l}\textit{identifier-3}\\ \textit{literal-1}\\ \textit{arithmetic-expression-1}\end{array}\right\}\\ \textit{condition-name-1}\end{array}\right\}$$

$$\left[\underline{\text{AND}}\left\{\begin{array}{l}\textit{data-name-2}\left\{\begin{array}{l}\text{IS}\ \underline{\text{EQUAL}}\ \text{TO}\\ \text{IS}\ =\end{array}\right\}\left\{\begin{array}{l}\textit{identifier-4}\\ \textit{literal-2}\\ \textit{arithmetic-expression-2}\end{array}\right\}\\ \textit{condition-name-2}\end{array}\right\}\right]\ldots$$

$$\left\{\begin{array}{l}\textit{imperative-statement-2}\\ \underline{\text{NEXT}}\ \underline{\text{SENTENCE}}\end{array}\right\}$$

$$\left[\underline{\text{END-SEARCH}}\right]$$

SEND Statement

Format 1: Send (Simple)

$$\underline{\text{SEND}}\ \textit{cd-name-1}\quad \underline{\text{FROM}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\}$$

Format 2: Send (Advancing/Replacing)

$$\underline{\text{SEND}}\ \textit{cd-name-1}\quad \left[\underline{\text{FROM}} \left\{ \begin{array}{l} \textit{identifier-1} \\ \textit{literal-1} \end{array} \right\} \right]\quad \text{WITH} \left\{ \begin{array}{l} \textit{identifier-2} \\ \underline{\text{ESI}} \\ \underline{\text{EMI}} \\ \underline{\text{EGI}} \end{array} \right\}$$

$$\left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{ADVANCING} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{integer-1} \end{array} \right\} \left[\begin{array}{l} \text{LINE} \\ \text{LINES} \end{array} \right] \\ \\ \left\{ \begin{array}{l} \textit{mnemonic-name-2} \\ \underline{\text{PAGE}} \end{array} \right\} \end{array} \right\} \right]$$

$$\left[\underline{\text{REPLACING}}\ \text{LINE} \right]$$

SET Statement

Format 1: Set Index

$$\underline{\text{SET}} \left\{ \left\{ \begin{array}{l} \textit{index-name-1} \\ \textit{identifier-1} \end{array} \right\} \cdots\quad \underline{\text{TO}} \left\{ \begin{array}{l} \textit{index-name-2} \\ \textit{identifier-2} \\ \textit{integer-1} \end{array} \right\} \right\} \cdots$$

Format 2: Set Index Up/Down

$$\underline{\text{SET}} \left\{ \left\{ \textit{index-name-3} \right\} \cdots\quad \left\{ \begin{array}{l} \underline{\text{UP}} \\ \underline{\text{DOWN}} \end{array} \right\} \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-3} \\ \textit{integer-2} \end{array} \right\} \right\} \cdots$$

Format 3: Set Switch On/Off

$$\underline{\text{SET}} \left\{ \left\{ \textit{mnemonic-name-1} \right\} \cdots\quad \underline{\text{TO}} \left\{ \begin{array}{l} \underline{\text{ON}} \\ \underline{\text{OFF}} \end{array} \right\} \right\} \cdots$$

Format 4: Set Condition-Name True/False

$$\underline{\text{SET}} \left\{ \left\{ \textit{condition-name-1} \right\} \cdots \quad \underline{\text{TO}} \left\{ \begin{array}{l} \underline{\text{TRUE}} \\ \underline{\text{FALSE}} \end{array} \right\} \right\} \cdots$$

Format 5: Set Pointer

$$\underline{\text{SET}} \left\{ \left\{ \begin{array}{l} \underline{\text{ADDRESS}} \left[\begin{array}{l} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right] \textit{data-name-1} \\ \textit{identifier-4} \end{array} \right\} \cdots \quad \underline{\text{TO}} \left\{ \begin{array}{l} \underline{\text{ADDRESS}} \left[\begin{array}{l} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right] \textit{identifier-5} \\ \textit{identifier-6} \\ \underline{\text{NULL}} \\ \underline{\text{NULLS}} \end{array} \right\} \right\} \cdots$$

Format 6: Set Pointer Up/Down

$$\underline{\text{SET}} \left\{ \left\{ \begin{array}{l} \underline{\text{ADDRESS}} \left[\begin{array}{l} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right] \textit{data-name-1} \\ \textit{identifier-4} \end{array} \right\} \cdots \left\{ \begin{array}{l} \underline{\text{UP}} \\ \underline{\text{DOWN}} \end{array} \right\} \underline{\text{BY}} \left\{ \begin{array}{l} \textit{identifier-7} \\ \textit{integer-3} \\ \underline{\text{LENGTH}} \left[\begin{array}{l} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right] \textit{identifier-8} \end{array} \right\} \right\} \cdots$$

SORT Statement

$$\underline{\text{SORT}} \ \textit{file-name-1} \left\{ \text{ON} \left\{ \begin{array}{l} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right\} \text{KEY} \left\{ \textit{data-name-1} \right\} \cdots \right\} \cdots$$

$$\left[\text{WITH } \underline{\text{DUPLICATES}} \text{ IN ORDER} \right]$$

$$\left[\text{COLLATING } \underline{\text{SEQUENCE}} \text{ IS } \textit{alphabet-name-1} \right]$$

$$\left\{ \begin{array}{l} \underline{\text{INPUT}} \ \underline{\text{PROCEDURE}} \text{ IS } \textit{procedure-name-1} \left[\left\{ \begin{array}{l} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\} \textit{procedure-name-2} \right] \\ \underline{\text{USING}} \left\{ \textit{file-name-2} \right\} \cdots \end{array} \right\}$$

$$\left\{ \begin{array}{l} \underline{\text{OUTPUT}} \ \underline{\text{PROCEDURE}} \text{ IS } \textit{procedure-name-3} \left[\left\{ \begin{array}{l} \underline{\text{THROUGH}} \\ \underline{\text{THRU}} \end{array} \right\} \textit{procedure-name-4} \right] \\ \underline{\text{GIVING}} \left\{ \textit{file-name-3} \right\} \cdots \end{array} \right\}$$

START Statement

```
                         ┌         ┌ IS [NOT] LESS THAN                   ┐                          ┐
                         │         │ IS [NOT] <                           │                          │
                         │         │ IS EQUAL TO                          │                          │
                         │         │ IS =                                 │                          │
                         │         │ IS [NOT] GREATER THAN                │                          │
                         │         │ IS [NOT] >                           │ ┌ data-name-1      ┐       │
START  file-name-1       │  KEY   <  IS GREATER THAN OR EQUAL TO          >  <                  >      │
                         │         │ IS >=                                │ └ split-key-name-1 ┘       │
                         │         │ IS LESS THAN  OR  EQUAL TO           │                          │
                         │         │ IS <=                                │                          │
                         │         │ IS FIRST                             │                          │
                         └         └ IS LAST                              ┘                          ┘

      ┌                ┌ identifier-1 ┐ ┐
      │ WITH  SIZE    <                > │
      └                └ integer-1    ┘ ┘

      [INVALID  KEY  imperative-statement-1]

      [NOT  INVALID  KEY  imperative-statement-2]

      [END-START]
```

STOP Statement

```
       ⎧ RUN ⎡ identifier-1 ⎤ ⎫
       ⎪     ⎣ integer-1    ⎦ ⎪
STOP   ⎨                      ⎬
       ⎪ ⎧ identifier-2 ⎫     ⎪
       ⎩ ⎩ literal-1    ⎭     ⎭
```

STRING Statement

```
         ⎧ ⎧ identifier-1 ⎫                 ⎧ identifier-2 ⎫ ⎫
STRING   ⎨ ⎩ literal-1    ⎭ ...  DELIMITED BY ⎨ literal-2    ⎬ ⎬ ...
         ⎩                                  ⎩ SIZE         ⎭ ⎭

     INTO identifier-3

     [ WITH POINTER identifier-4 ]

     [ ON OVERFLOW imperative-statement-1 ]

     [ NOT ON OVERFLOW imperative-statement-2 ]

     [ END-STRING ]
```

SUBTRACT Statement

Format 1: Subtract…From

$$\underline{\text{SUBTRACT}} \left\{ \begin{matrix} \textit{identifier-1} \\ \textit{literal-1} \end{matrix} \right\} \cdots \quad \underline{\text{FROM}} \quad \left\{ \textit{identifier-3} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \ \text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-SUBTRACT}} \right]$$

Format 2: Subtract…Giving

$$\underline{\text{SUBTRACT}} \left\{ \begin{matrix} \textit{identifier-1} \\ \textit{literal-1} \end{matrix} \right\} \cdots \quad \underline{\text{FROM}} \quad \left\{ \begin{matrix} \textit{identifier-2} \\ \textit{literal-2} \end{matrix} \right\}$$

$$\underline{\text{GIVING}} \left\{ \textit{identifier-3} \left[\underline{\text{ROUNDED}} \right] \right\} \cdots$$

$$\left[\text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \ \text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-SUBTRACT}} \right]$$

Format 3: Subtract Corresponding

$$\underline{\text{SUBTRACT}} \left\{ \begin{matrix} \underline{\text{CORRESPONDING}} \\ \underline{\text{CORR}} \end{matrix} \right\} \textit{identifier-1} \quad \underline{\text{FROM}} \quad \textit{identifier-2} \left[\underline{\text{ROUNDED}} \right]$$

$$\left[\text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-1} \right]$$

$$\left[\underline{\text{NOT}} \ \text{ON} \ \underline{\text{SIZE}} \ \underline{\text{ERROR}} \ \textit{imperative-statement-2} \right]$$

$$\left[\underline{\text{END-SUBTRACT}} \right]$$

UNLOCK Statement

$$\underline{\text{UNLOCK}}\ \ \textit{file-name-1}\ \begin{bmatrix}\text{RECORD}\\ \text{RECORDS}\end{bmatrix}$$

UNSTRING Statement

$$\underline{\text{UNSTRING}}\ \ \textit{identifier-1}$$

$$\left[\ \underline{\text{DELIMITED}}\ \text{BY}\ [\,\underline{\text{ALL}}\,]\begin{Bmatrix}\textit{identifier-2}\\ \textit{literal-1}\end{Bmatrix}\ \left[\ \underline{\text{OR}}\ [\,\underline{\text{ALL}}\,]\begin{Bmatrix}\textit{identifier-3}\\ \textit{literal-2}\end{Bmatrix}\right]\cdots\ \right]$$

$$\underline{\text{INTO}}\ \{\,\textit{identifier-4}\ \ [\,\underline{\text{DELIMITER}}\ \text{IN}\ \textit{identifier-5}\,]\ \ [\,\underline{\text{COUNT}}\ \text{IN}\ \textit{identifier-6}\,]\,\}\cdots$$

$$[\,\text{WITH}\ \ \underline{\text{POINTER}}\ \ \textit{identifier-7}\,]$$

$$[\,\underline{\text{TALLYING}}\ \ \text{IN}\ \ \textit{identifier-8}\,]$$

$$[\,\text{ON}\ \ \underline{\text{OVERFLOW}}\ \ \textit{imperative-statement-1}\,]$$

$$[\,\underline{\text{NOT}}\ \text{ON}\ \ \underline{\text{OVERFLOW}}\ \ \textit{imperative-statement-2}\,]$$

$$[\,\underline{\text{END-UNSTRING}}\,]$$

USE Statement

$$\underline{\text{USE}}\ [\,\underline{\text{GLOBAL}}\,]\ \ \underline{\text{AFTER}}\ \ \text{STANDARD}\ \begin{Bmatrix}\underline{\text{EXCEPTION}}\\ \underline{\text{ERROR}}\end{Bmatrix}$$

$$\underline{\text{PROCEDURE}}\ \ \text{ON}\ \left\{\begin{Vmatrix}\{\,\textit{file-name-1}\,\}\cdots\\ \underline{\text{INPUT}}\\ \underline{\text{OUTPUT}}\\ \underline{\text{I-O}}\\ \underline{\text{EXTEND}}\end{Vmatrix}\right\}$$

WRITE Statement

Format 1: Write Sequential File

$$\underline{\text{WRITE}}\ \ \textit{record-name-1}\ \ \left[\ \underline{\text{FROM}}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\right]$$

$$\left[\left\{\begin{array}{l}\underline{\text{BEFORE}}\\ \underline{\text{AFTER}}\end{array}\right\}\ \text{ADVANCING}\left\{\begin{array}{l}\left\{\begin{array}{l}\textit{identifier-2}\\ \textit{integer-1}\end{array}\right\}\left[\begin{array}{l}\text{LINE}\\ \text{LINES}\end{array}\right]\\ \\ \underline{\text{TO}}\ \ \text{LINE}\left\{\begin{array}{l}\textit{identifier-3}\\ \textit{integer-2}\end{array}\right\}\left[\text{ON}\ \ \underline{\text{NEXT}}\ \ \underline{\text{PAGE}}\right]\\ \\ \left\{\begin{array}{l}\textit{mnemonic-name-2}\\ \underline{\text{PAGE}}\end{array}\right\}\end{array}\right\}\right]$$

$$\left[\ \text{AT}\left\{\begin{array}{l}\underline{\text{END-OF-PAGE}}\\ \underline{\text{EOP}}\end{array}\right\}\ \textit{imperative-statement-1}\right]$$

$$\left[\ \underline{\text{NOT}}\ \text{AT}\left\{\begin{array}{l}\underline{\text{END-OF-PAGE}}\\ \underline{\text{EOP}}\end{array}\right\}\ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-WRITE}}\right]$$

Format 2: Write Relative and Indexed File

$$\underline{\text{WRITE}}\ \ \textit{record-name-1}\ \ \left[\ \underline{\text{FROM}}\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\}\right]$$

$$\left[\underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-1}\right]$$

$$\left[\underline{\text{NOT}}\ \ \underline{\text{INVALID}}\ \ \text{KEY}\ \ \textit{imperative-statement-2}\right]$$

$$\left[\underline{\text{END-WRITE}}\right]$$

General Format for END PROGRAM Header

$$\underline{\text{END}}\ \underline{\text{PROGRAM}} \left[\begin{matrix} \textit{program-name-1} \\ \textit{literal-1} \end{matrix}\right].$$

General Formats for COPY and REPLACE Statements

$$\underline{\text{COPY}} \left\{\begin{matrix} \textit{text-name-1} \\ \textit{literal-1} \end{matrix}\right\} \left[\left\{\begin{matrix} \underline{\text{IN}} \\ \underline{\text{OF}} \end{matrix}\right\} \left\{\begin{matrix} \textit{library-name-1} \\ \textit{literal-2} \end{matrix}\right\} \right]$$

$$\left[\underline{\text{REPLACING}} \left\{ \left\{\begin{matrix} \textit{== pseudo-text-1 ==} \\ \textit{identifier-1} \\ \textit{literal-3} \\ \textit{word-1} \end{matrix}\right\} \underline{\text{BY}} \left\{\begin{matrix} \textit{== pseudo-text-2 ==} \\ \textit{identifier-2} \\ \textit{literal-4} \\ \textit{word-2} \end{matrix}\right\} \right\} \cdots \right]$$

Format 1: Begin or Change Replacement

$$\underline{\text{REPLACE}} \left\{ \textit{== pseudo-text-1 ==}\ \ \underline{\text{BY}}\ \ \textit{== pseudo-text-2 ==} \right\} \cdots$$

Format 2: End Replacement

$$\underline{\text{REPLACE}}\ \underline{\text{OFF}}$$

General Formats for Conditions

Relation Condition

$$\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\\ \textit{arithmetic-expression-1}\\ \textit{index-name-1}\end{array}\right\} \; \textit{relational-operator} \; \left\{\begin{array}{l}\textit{identifier-2}\\ \textit{literal-2}\\ \textit{arithmetic-expression-2}\\ \textit{index-name-2}\end{array}\right\}$$

where the general format for the *relational-operator* is:

$$\left\{\begin{array}{l}
\text{IS } \left[\underline{\text{NOT}}\right] \underline{\text{GREATER}} \text{ THAN}\\
\text{IS } \left[\underline{\text{NOT}}\right] >\\
\text{IS } \left[\underline{\text{NOT}}\right] \underline{\text{LESS}} \text{ THAN}\\
\text{IS } \left[\underline{\text{NOT}}\right] <\\
\text{IS } \left[\underline{\text{NOT}}\right] \underline{\text{EQUAL}} \text{ TO}\\
\text{IS } \left[\underline{\text{NOT}}\right] =\\
\text{IS } \underline{\text{GREATER}} \text{ THAN } \underline{\text{OR}} \; \underline{\text{EQUAL}} \text{ TO}\\
\text{IS } >=\\
\text{IS } \underline{\text{LESS}} \text{ THAN } \underline{\text{OR}} \; \underline{\text{EQUAL}} \text{ TO}\\
\text{IS } <=\\
\text{IS } \left[\underline{\text{NOT}}\right] \underline{\text{LIKE}} \left[\left\{\left|\begin{array}{l}\left\{\underline{\text{TRIMMED}} \left[\begin{array}{l}\underline{\text{RIGHT}}\\ \underline{\text{LEFT}}\end{array}\right]\right\}\\ \left\{\begin{array}{l}\underline{\text{CASE-INSENSITIVE}}\\ \text{CASE-SENSITIVE}\end{array}\right\}\end{array}\right|\right\}\right]
\end{array}\right\}$$

LIKE Condition (Special Case of a Relation Condition)

$$\left\{\begin{array}{l}\textit{identifier-1}\\ \textit{literal-1}\end{array}\right\} \text{ IS } \left[\underline{\text{NOT}}\right] \underline{\text{LIKE}} \left[\left\{\left|\begin{array}{l}\left\{\underline{\text{TRIMMED}} \left[\begin{array}{l}\underline{\text{RIGHT}}\\ \underline{\text{LEFT}}\end{array}\right]\right\}\\ \left\{\begin{array}{l}\underline{\text{CASE-INSENSITIVE}}\\ \text{CASE-SENSITIVE}\end{array}\right\}\end{array}\right|\right\}\right] \left\{\begin{array}{l}\textit{identifier-2}\\ \textit{literal-2}\end{array}\right\}$$

Class Condition

$$\textit{identifier-1}\ \text{IS}\ \left[\underline{\text{NOT}}\right]\ \left\{\begin{array}{l}\underline{\text{NUMERIC}}\\ \underline{\text{ALPHABETIC}}\\ \underline{\text{ALPHABETIC-LOWER}}\\ \underline{\text{ALPHABETIC-UPPER}}\\ \textit{class-name-1}\end{array}\right\}$$

Sign Condition

$$\textit{arithmetic-expression-1}\ \text{IS}\ \left[\underline{\text{NOT}}\right]\ \left\{\begin{array}{l}\underline{\text{POSITIVE}}\\ \underline{\text{NEGATIVE}}\\ \underline{\text{ZERO}}\end{array}\right\}$$

Condition-Name Condition

condition-name-1

Switch-Status Condition

condition-name-2

Negated Condition

$\underline{\text{NOT}}$ *condition-1*

Combined Condition

$$\textit{condition-2}\ \left\{\left\{\begin{array}{l}\underline{\text{AND}}\\ \underline{\text{OR}}\end{array}\right\}\ \textit{condition-3}\right\}\cdots$$

Abbreviated Combined Relation Condition

$$\textit{relation-condition-1}\ \left\{\left\{\begin{array}{l}\underline{\text{AND}}\\ \underline{\text{OR}}\end{array}\right\}\ \left[\underline{\text{NOT}}\right]\ \left[\ \textit{relational-operator}\ \right]\ \textit{object-1}\right\}\cdots$$

General Formats for Qualification

Format 1: Qualification for Data-Names and Condition-Names

$$\left\{\begin{array}{l}\textit{data-name-1}\\\textit{condition-name-1}\end{array}\right\} \left\{\begin{array}{l} \left\{\left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{data-name-2}\right\}\ldots \left[\left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\left\{\begin{array}{l}\textit{file-name-1}\\\textit{cd-name-1}\end{array}\right\}\right] \\ \left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\left\{\begin{array}{l}\textit{file-name-1}\\\textit{cd-name-1}\end{array}\right\}\end{array}\right\}$$

Format 2: Qualification for LINAGE-COUNTER

$$\underline{\text{LINAGE-COUNTER}}\ \left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{file-name-2}$$

Format 3: Qualification for Screen-Names

$$\textit{screen-name-1}\ \left\{\left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{screen-name-2}\right\}\ldots$$

Format 4: Qualification for Split-Key-Names

$$\textit{split-key-name-1}\ \left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{file-name-3}$$

Format 5: Qualification for Paragraph Names

$$\textit{paragraph-name-1}\ \left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{section-name-1}$$

Format 6: Qualification for Text-Names (COPY Statement)

$$\textit{text-name-1}\ \left\{\begin{array}{l}\underline{\text{IN}}\\\underline{\text{OF}}\end{array}\right\}\ \textit{library-name-1}$$

Miscellaneous Formats

Sentence

statement-sequence-1 **.**

Statement Sequence

$$\left\{ \textit{imperative-statement-1} \ \ \text{THEN} \right\} \cdots \ \left\{ \begin{array}{l} \textit{imperative-statement-2} \\ \textit{conditional-statement-1} \end{array} \right\}$$

Subscripting

$$\left\{ \begin{array}{l} \textit{data-name-1} \\ \textit{condition-name-1} \end{array} \right\} \ \textbf{(} \ \left\{ \begin{array}{l} \textit{integer-1} \\ \left\{ \begin{array}{l} \textit{data-name-2} \\ \textit{index-name-1} \end{array} \right\} \left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} \ \textit{integer-2} \right] \end{array} \right\} \cdots \ \textbf{)}$$

Reference Modification

$$\textit{data-name-1} \ \textbf{(} \ \textit{leftmost-character-position-1} \ \textbf{:} \ \left[\textit{length-1} \right] \ \textbf{)}$$

Identifier

$$\textit{data-name-1} \left[\left\{ \begin{array}{c} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right\} \ \textit{data-name-2} \right] \cdots \left[\left\{ \begin{array}{c} \underline{\text{IN}} \\ \underline{\text{OF}} \end{array} \right\} \left\{ \begin{array}{l} \textit{file-name-1} \\ \textit{cd-name-1} \end{array} \right\} \right]$$

$$\left[\textbf{(} \ \left\{ \textit{subscript-1} \right\} \cdots \ \textbf{)} \right] \ \left[\textbf{(} \ \textit{leftmost-character-position-1} \ \textbf{:} \ \left[\textit{length-1} \right] \ \textbf{)} \right]$$

Special Registers

$$\underline{\text{ADDRESS}} \left[\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right] \textit{identifier-1}$$

$$\underline{\text{COUNT}} \left[\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right] \textit{data-name-1}$$

$$\underline{\text{COUNT-MAX}} \left[\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right] \textit{data-name-1}$$

$$\underline{\text{COUNT-MIN}} \left[\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right] \textit{data-name-1}$$

$$\underline{\text{LENGTH}} \left[\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right] \left\{\begin{matrix}\textit{identifier-1}\\ \textit{literal-1}\end{matrix}\right\}$$

$$\underline{\text{LINAGE-COUNTER}} \left[\left\{\begin{matrix}\underline{\text{IN}}\\ \underline{\text{OF}}\end{matrix}\right\} \textit{file-name-1}\right]$$

$$\underline{\text{PROGRAM-ID}}$$

$$\underline{\text{RETURN-CODE}}$$

Figurative-Constants

$$[\underline{\text{ALL}}]\ \underline{\text{HIGH-VALUE}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{HIGH-VALUES}}$$

$$[\underline{\text{ALL}}]\ \underline{\text{LOW-VALUE}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{LOW-VALUES}}$$

$$[\underline{\text{ALL}}]\ \underline{\text{NULL}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{NULLS}}$$

$$[\underline{\text{ALL}}]\ \underline{\text{QUOTE}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{QUOTES}}$$

$$[\underline{\text{ALL}}]\ \underline{\text{SPACE}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{SPACES}}$$

$$[\underline{\text{ALL}}]\ \underline{\text{ZERO}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{ZEROES}}$$
$$[\underline{\text{ALL}}]\ \underline{\text{ZEROS}}$$

$$\underline{\text{ALL}}\ \textit{literal-1}$$

$$[\underline{\text{ALL}}]\ \textit{symbolic-character-1}$$

Concatenation Expression

literal-1 & *literal-2*

Constant-Expression

$$[\underline{\text{NOT}}]\left\{\begin{array}{l} \textit{integer-1} \\ \underline{\text{NEXT}} \\ \left\{\begin{array}{l}\underline{\text{LENGTH}}\\ \underline{\text{SIZE}}\end{array}\right\} \text{OF} \left\{\begin{array}{l}\textit{data-name-4}\\ \textit{literal-4}\end{array}\right\} \\ \underline{\text{START}}\ \text{OF}\ \textit{data-name-6} \\ \underline{\text{DATE - COMPILED}} \\ (\textit{constant-expression-2}) \end{array}\right\} \left[\left\{\begin{array}{l} + \\ - \\ * \\ / \\ ** \\ \underline{\text{AND}} \\ \underline{\text{OR}} \\ \underline{\text{EXCLUSIVE}}\ \text{OR}\end{array}\right\} [\underline{\text{NOT}}] \left\{\begin{array}{l} \textit{integer-2} \\ \underline{\text{NEXT}} \\ \left\{\begin{array}{l}\underline{\text{LENGTH}}\\ \underline{\text{SIZE}}\end{array}\right\} \text{OF} \left\{\begin{array}{l}\textit{data-name-5}\\ \textit{literal-5}\end{array}\right\} \\ \underline{\text{START}}\ \text{OF}\ \textit{data-name-7} \\ \underline{\text{DATE - COMPILED}} \\ (\textit{constant-expression-3}) \end{array}\right\} \right] \ldots$$

PICTURE Character-String

The five categories of data that can be described with a PICTURE clause[1] are defined as follows:

1. **Alphabetic.** Its PICTURE character-string can contain only the symbol **A**. The contents of an alphabetic data item when represented in standard data format must be one or more alphabetic characters (“a” through “z”, “A” through “Z”, and space).

2. **Alphanumeric.** Its PICTURE character-string is restricted to certain combinations of the symbols **A**, **X** and **9**, and the item is treated as if the character-string contained all symbols **X**. The PICTURE character-string must contain at least one symbol **X** or a combination of the symbols **A** and **9**. A PICTURE character-string that contains all symbols **A** or all symbols **9** does not define an alphanumeric data item, since such character-strings define an alphabetic or numeric data item, respectively. The contents of an alphanumeric data item when represented in standard data format must be one or more characters in the character set of the computer.

3. **Alphanumeric edited.** Its PICTURE character-string is restricted to certain combinations of the following symbols: **A**, **X**, **9**, **B**, **0**, and slash (/). The PICTURE character-string must contain at least one symbol **A** or **X** and at least one symbol **B**, **0**, or slash (/). The contents of an alphanumeric edited data item when represented in standard data format must be two or more characters in the character set of the computer.

4. **Numeric.** Its PICTURE character-string can contain only the symbols **9**, **P**, **S**, and **V**. Its PICTURE character-string must contain at least one symbol **9** and not more than thirty symbols **9**. Each symbol **9** specifies a digit position. If unsigned, the contents of a numeric data item when represented in standard data format must be one or more numeric characters. If signed, a numeric data item may also contain a "+", "–", or other representation of an operational sign. The actual in-memory contents of a numeric data item are not standard data format when the usage is other than DISPLAY as specified by a USAGE clause applicable to the data description entry or when the data item is signed and the SEPARATE CHARACTER phrase is not specified in a SIGN clause applicable to the data description entry.

5. **Numeric edited.** Its PICTURE character-string is restricted to certain combinations of the following symbols: **B**, slash (**/**), **P**, **V**, **Z**, **0**, **9**, comma (**,**), period (**.**), asterisk (*****), minus (**–**), plus (**+**), **CR**, **DB**, and the currency symbol (the symbol **$** or the symbol specified in the CURRENCY SIGN clause of the SPECIAL-NAMES paragraph). The allowable combinations are determined from the order of precedence of symbols (see Table 1 on page 62) and the editing rules. The number of digit positions that can be represented in the PICTURE character-string must range from one to thirty, inclusive. The character-string must contain at least one symbol **0**, **B**, slash, **Z**, asterisk, plus, minus, comma, period, **CR**, **DB**, or the currency symbol. The contents of each of the character positions in a numeric edited data item must be consistent with the corresponding PICTURE symbol.

PICTURE Symbols

The functions of the symbols used in a PICTURE character-string to describe an elementary data item are as follows:

A Each symbol **A** in the character-string represents a character position that can contain only an alphabetic character ("a" through "z", "A" through "Z", and space). Each symbol **A** is counted in the size of the data item described by the PICTURE character-string.

B Each symbol **B** in the character-string represents a character position into which the character space will be inserted when the data item is the receiving item of an elementary MOVE statement. Each symbol **B** is counted in the size of the data item described by the PICTURE character-string.

P Each symbol **P** in the character-string indicates an assumed decimal scaling position and is used to specify the location of an assumed decimal point when the point is not within the number that appears in the data item. The scaling position symbol **P** is

[1] The additional data categories, index data and data pointer, also exist but do not use a PICTURE clause in their data description entry. An index data item is described with the USAGE IS INDEX clause. A data pointer data item is described with the USAGE IS POINTER clause.

not counted in the size of the data item described by the PICTURE character-string, but each symbol **P** is counted in determining the maximum number (30) of digit positions in numeric and numeric edited data items. The symbol **P** may appear only as a contiguous string in the leftmost or rightmost digit positions within a PICTURE character-string. Since the scaling position symbol **P** implies an assumed decimal point (to the left of the symbols **P** if they are the leftmost digit positions and to the right of the symbols **P** if they are the rightmost digit positions), the assumed decimal point symbol **V** is redundant either to the left or right of the symbols **P**, respectively, within such a PICTURE character-string. The symbol **P** and the insertion symbol period (**.**) cannot both occur in the same PICTURE character-string.

S The symbol **S** is used in the character-string to indicate the presence, but neither the representation nor, necessarily, the position of an operational sign. The symbol **S** must be written as the leftmost character in the PICTURE character-string. The symbol **S** is not counted in determining the size (in terms of standard data format characters) of the data item described by the PICTURE character-string unless the entry contains or is subject to a SIGN clause that specifies the SEPARATE CHARACTER phrase. The symbol **S** in the PICTURE character-string and the BLANK WHEN ZERO clause may not occur in the same data description entry.

V The symbol **V** is used in a character-string to indicate the location of the assumed decimal point and may appear only once in any single PICTURE character-string. The symbol **V** does not represent a character position and, therefore, is not counted in the size of the data item described by the PICTURE character-string. When the assumed decimal point is to the right of the rightmost symbol in the string representing a digit position or scaling position, or is to the left of scaling positions that represent the leftmost digit positions, the symbol **V** is redundant. The symbol **V** and the insertion symbol period (**.**) cannot both occur in the same PICTURE character-string.

X Each symbol **X** in the character-string is used to represent a character position that contains any allowable character from the character set of the computer. Each symbol **X** is counted in the size of the data item described by the PICTURE character-string.

Z Each symbol **Z** in a character-string may only be used to represent the leftmost leading numeric character positions that will be replaced by space characters when the contents of those character positions are leading zeroes and the data item is the receiving item of an elementary MOVE statement. Each symbol **Z** is counted in the size of the item described by the PICTURE character-string and in determining the maximum number (30) of digit positions allowed in a numeric edited data item. If the symbol **Z** is used to the right of the decimal point in a character-string, then all digit positions in that character-string must be described with the symbol **Z**. If the symbol **Z** represents all the digit-positions in the character-string, then the described data item is blank when zero, even if the BLANK WHEN ZERO clause is not specified.

9 Each symbol **9** in the character-string represents a character position that contains a numeric character. Each symbol **9** is counted in the size of the item described by the PICTURE character-string and in determining the maximum number (30) of digit positions in a numeric or numeric edited data item.

0 Each symbol **0** in the character-string represents a character position into which the character zero (“0”) will be inserted when the data item is the receiving item of an elementary MOVE statement and removed when a numeric edited data item is the sending item in an elementary MOVE statement with a numeric or numeric edited receiving data item. Each symbol **0** is counted in the size of the data item described by the PICTURE character-string. The symbol **0** does not represent a digit position in a numeric edited data item.

/ Each symbol slash (/) in the character-string represents a character position into which a character slash (“/”) will be inserted when the data item is the receiving item of an elementary MOVE statement. Each symbol slash (/) is counted in the size of the data item described by the PICTURE character-string.

, Each symbol comma (**,**) in the character-string represents a character position into which a character comma (“,”) will be inserted when the data item is the receiving item of an elementary MOVE statement. Each symbol comma (**,**) is counted in the size of the data item described by the PICTURE character-string.

. When the symbol period (**.**) appears in the character-string, it is an editing symbol that represents the decimal point for alignment purposes and, in addition, represents a character position into which the character period (“.”) will be inserted. The symbol period is counted in the size of the data item described by the PICTURE character-string. The symbols **P** and **V** cannot occur with a symbol period (**.**) in the same PICTURE character-string.

Note For a given program the functions of the period and comma are exchanged if the DECIMAL-POINT IS COMMA clause is stated in the SPECIAL-NAMES paragraph. In this exchange, the rules for the period apply to the comma and the rules for the comma apply to the period wherever they appear in a PICTURE character-string.

+, –, CR, DB
These symbols are used as editing sign control symbols. When used, they represent the character position into which the editing sign control symbol will be placed. The symbols are mutually exclusive in any one PICTURE character-string and each character used in the symbol is counted in determining the size of the data item described by the PICTURE character-string. If the symbols plus or minus occur

more than once (a floating sign control symbol), then one less than the total number of these symbols is counted in determining the maximum number (30) of digit positions allowed in a numeric edited data item. If a floating symbol plus or minus is used to the right of the decimal point in a character-string, then all digit positions in that character-string must be described with the symbol plus or minus, respectively. If a floating plus or minus symbol string represents all the digit-positions in the character-string, then the described data item is blank when zero, even if the BLANK WHEN ZERO clause is not specified.

***** Each symbol asterisk (*****) in the character-string represents a leading numeric character position into which a character asterisk ("*") will be placed when that position contains a leading zero and the data item is the receiving item of an elementary MOVE statement. Each symbol asterisk (*****) is counted in the size of the data item described by the PICTURE character-string and in determining the maximum number (30) of digit positions allowed in a numeric edited data item. If the symbol asterisk (*****) is used to the right of the decimal point in a character-string, then all digit positions in that character-string must be described with the symbol asterisk (*****). The symbol asterisk in the PICTURE character-string and the BLANK WHEN ZERO clause may not occur in the same data description entry. If the symbol asterisk represents all the digit-positions in the character-string, then, when zero, the described data item is all asterisks (ALL "*"), except that, if the character-string contains the symbol period (**.**), a character period (".") will occur at the specified location in the data item.

cs The currency symbol in a character-string is represented either by the currency sign (the symbol **$**) or by the single character specified in the CURRENCY SIGN clause in the SPECIAL-NAMES paragraph. The currency symbol in the character-string represents a character position into which a currency symbol is to be placed when the data item is the receiving item of an elementary MOVE statement. Each currency symbol is counted in the size of the data item described by the PICTURE character-string. If the currency symbol occurs more than once (a floating currency symbol), then one less than the total number of currency symbols is counted in determining the maximum number (30) of digit positions allowed in a numeric edited data item. If the currency symbol is used to the right of the decimal point in a character-string, then all digit positions in that character-string must be described with the currency symbol. If a floating currency symbol string represents all the digit-positions in the character-string, then the described data item is blank when zero, even if the BLANK WHEN ZERO clause is not specified.

Table 1: PICTURE Symbol Precedence

Second Symbol \ First Symbol		Non-floating Insertion Symbols									Floating Insertion Symbols						Other Symbols					
		B	**0**	**/**	**,**	**.**	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	$\left\{\begin{matrix}CR\\DB\end{matrix}\right\}$	**CS**	$\left\{\begin{matrix}Z\\*\end{matrix}\right\}$	$\left\{\begin{matrix}Z\\*\end{matrix}\right\}$	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	**CS**	**CS**	**9**	**A X**	**S**	**V**	**P**	**P**
Non-floating Insertion Symbols	**B**	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X
	0	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X
	/	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X
	,	X	X	X	X	X	X			X	X	X	X	X	X	X	X			X		X
	.	X	X	X	X		X			X	X		X		X		X					
	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$																					
	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	X	X	X	X	X				X	X	X			X	X	X			X	X	X
	$\left\{\begin{matrix}CR\\DB\end{matrix}\right\}$	X	X	X	X	X				X	X	X			X	X	X			X	X	X
	CS						X															
Floating Insertion Symbols	$\left\{\begin{matrix}Z\\*\end{matrix}\right\}$	X	X	X	X		X			X	X											
	$\left\{\begin{matrix}Z\\*\end{matrix}\right\}$	X	X	X	X	X	X			X	X	X								X		X
	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	X	X	X	X					X			X									
	$\left\{\begin{matrix}+\\-\end{matrix}\right\}$	X	X	X	X	X				X			X	X						X		
	CS	X	X	X	X		X								X							
	CS	X	X	X	X	X	X								X	X				X		
Other Symbols	**9**	X	X	X	X	X	X			X	X		X		X		X	X	X	X		X
	A X	X	X	X													X	X				
	S																					
	V	X	X	X	X		X			X	X		X		X		X		X		X	
	P	X	X	X	X		X			X	X		X		X		X		X		X	
	P						X			X									X	X		X

General Format for Nested Source Programs

$$\left\{\begin{array}{l}\underline{\text{IDENTIFICATION}}\\ \underline{\text{ID}}\end{array}\right\}\ \underline{\text{DIVISION}}\,.$$

$$\underline{\text{PROGRAM-ID}}\,.\ \left\{\begin{array}{l}\textit{program-name-1}\\ \textit{literal-1}\end{array}\right\}\ \left[\text{IS}\ \ \underline{\text{INITIAL}}\ \ \text{PROGRAM}\right].$$

$$\left[\underline{\text{ENVIRONMENT}}\ \ \underline{\text{DIVISION}}\,.\ \textit{environment-division-content-1}\right]$$

$$\left[\underline{\text{DATA}}\ \ \underline{\text{DIVISION}}\,.\ \textit{data-division-content-1}\right]$$

$$\left[\underline{\text{PROCEDURE}}\ \ \underline{\text{DIVISION}}\,.\ \textit{procedure-division-content-1}\right]$$

$$\left[\begin{array}{l}\left[\textit{nested-source-program-1}\right]\cdots\\ \underline{\text{END}}\ \ \underline{\text{PROGRAM}}\ \left[\begin{array}{l}\textit{program-name-1}\\ \textit{literal-1}\end{array}\right].\end{array}\right]$$

General Format for *nested-source-program*

$$\left\{\begin{array}{l}\underline{\text{IDENTIFICATION}}\\ \underline{\text{ID}}\end{array}\right\}\ \underline{\text{DIVISION}}\,.$$

$$\underline{\text{PROGRAM-ID}}\,.\ \left\{\begin{array}{l}\textit{program-name-2}\\ \textit{literal-2}\end{array}\right\}\ \left[\text{IS}\ \left\{\left\|\begin{array}{l}\underline{\text{COMMON}}\\ \underline{\text{INITIAL}}\end{array}\right\|\right\}\ \text{PROGRAM}\right].$$

$$\left[\underline{\text{ENVIRONMENT}}\ \ \underline{\text{DIVISION}}\,.\ \textit{environment-division-content-2}\right]$$

$$\left[\underline{\text{DATA}}\ \ \underline{\text{DIVISION}}\,.\ \textit{data-division-content-2}\right]$$

$$\left[\underline{\text{PROCEDURE}}\ \ \underline{\text{DIVISION}}\,.\ \textit{procedure-division-content-2}\right]$$

$$\left[\textit{nested-source-program-2}\right]\cdots$$

$$\underline{\text{END}}\ \ \underline{\text{PROGRAM}}\ \left[\begin{array}{l}\textit{program-name-2}\\ \textit{literal-2}\end{array}\right].$$

General Format for a Sequence of Source Programs

```
{ { IDENTIFICATION } DIVISION.
{ { ID             }
{
{    PROGRAM-ID. { program-name-3 } [IS INITIAL PROGRAM].
{                { literal-3      }
{
{   [ENVIRONMENT DIVISION. environment-division-content-3 ]
{
{   [DATA DIVISION. data-division-content-3 ]
{
{   [PROCEDURE DIVISION. procedure-division-content-3 ]
{
{   [nested-source-program-3 ]...
{
{   END PROGRAM { program-name-3 }.  }...
{               { literal-3      }   }

  { IDENTIFICATION } DIVISION.
  { ID             }

     PROGRAM-ID. { program-name-4 } [IS INITIAL PROGRAM].
                 { literal-4      }

    [ENVIRONMENT DIVISION. environment-division-content-4 ]

    [DATA DIVISION. data-division-content-4 ]

    [PROCEDURE DIVISION. procedure-division-content-4 ]

[  [nested-source-program-4 ]...
[
[   [ END PROGRAM [ program-name-4 ] ]. ]
[   [             [ literal-4      ] ]  ]
```

Reserved Words

The DERESERVE keyword of the COMPILER-OPTIONS configuration record, which is described in the “Configuration” chapter of the *RM/COBOL User’s Guide*, can be used to make a reserved word a user-defined word whenever it occurs in the source program, but then the language feature provided by the construct in which the word appears is not available for programs compiled with that particular configuration setting.

ACCEPT
ACCESS
ADD
ADDRESS[2]
ADVANCING
AFTER
ALL
ALPHABET[2]
ALPHABETIC
ALPHABETIC-LOWER[2]
ALPHABETIC-UPPER[2]
ALPHANUMERIC[2]
ALPHANUMERIC-EDITED[2]
ALSO[2]
ALTER
ALTERNATE
AND
ANY[2]
ARE
AREA
AREAS
ASCENDING[2]
ASSIGN
AT
AUTHOR

BEEP
BEFORE
BELL[2]
BINARY
BLANK
BLINK
BLOCK
BOTTOM[2]
BY

CALL
CANCEL
CD[2]
CENTURY-DATE[2]
CENTURY-DAY[2]
CF[2]
CH[2]
CHARACTER
CHARACTERS
CLASS[2]
CLOCK-UNITS[2]
CLOSE
COBOL[2]
CODE[2]
CODE-SET
COL[2]
COLLATING
COLUMN[2]
COMMA
COMMON[2]
COMMUNICATION[2]
COMP
COMP-1
COMP-3

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

COMP-4[2]
COMP-5[2]COMP-6
COMPUTATIONAL
COMPUTATIONAL-1
COMPUTATIONAL-3
COMPUTATIONAL-4[2]
COMPUTATIONAL-5
[2]COMPUTATIONAL-6
COMPUTE
CONFIGURATION
CONTAINS
CONTENT[2]
CONTINUE[2]
CONTROL[2]
CONTROLS[2]
CONVERT
CONVERTING[2]
COPY
CORR
CORRESPONDING
COUNT[2]
COUNT-MAX[2]
COUNT-MIN[2]
CURRENCY
CURSOR[2]

DATA
DATA-POINTER[2]
DATE
DATE-AND-TIME[2]
DATE-COMPILED[2]
DATE-WRITTEN
DAY
DAY-AND-TIME[2]
DAY-OF-WEEK[2]
DE[2]
DEBUG-CONTENTS[2]
DEBUG-ITEM[2]
DEBUG-LINE[2]
DEBUG-NAME[2]
DEBUG-SUB-1[2]
DEBUG-SUB-2[2]
DEBUG-SUB-3[2]
DEBUGGING[2]
DECIMAL-POINT
DECLARATIVES
DEFAULT[2]
DELETE
DELIMITED[2]
DELIMITER[2]
DEPENDING
DESCENDING[2]
DESTINATION[2]
DETAIL[2]
DISABLE[2]
DISPLAY
DIVIDE
DIVISION
DOWN
DUPLICATES
DYNAMIC

ECHO
EGI[2]
ELSE
EMI[2]
ENABLE[2]
END
END-ACCEPT[2]
END-ADD[2]
END-CALL[2]
END-COMPUTE[2]
END-DELETE[2]
END-DIVIDE[2]
END-EVALUATE[2]
END-IF[2]
END-MULTIPLY[2]
END-OF-PAGE[2]
END-PERFORM[2]
END-READ[2]

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

END-RECEIVE[2]
END-RETURN[2]
END-REWRITE[2]
END-SEARCH[2]
END-START[2]
END-STRING[2]
END-SUBTRACT[2]
END-UNSTRING[2]
END-WRITE[2]
ENTER[2]
ENVIRONMENT
EOP[2]
EQUAL
ERASE
ERROR
ESCAPE[2]
ESI[2]
EVALUATE[2]
EVERY[2]
EXCEPTION
EXCLUSIVE[2]
EXIT
EXTEND
EXTERNAL[2]

FALSE[2]
FD
FILE
FILE-CONTROL
FILLER
FINAL[2]
FIRST
FIXED[2]
FOOTING[2]
FOR
FROM
FUNCTION[2]

GENERATE[2]
GIVING
GLOBAL[2]
GO
GOBACK[2]
GREATER
GROUP[2]

HEADING[2]
HIGH
HIGH-VALUE
HIGH-VALUES
HIGHLIGHT

I-O
I-O-CONTROL
ID[2]
IDENTIFICATION
IF
IN
INDEX
INDEXED
INDICATE[2]
INITIAL
INITIALIZE[2]
INITIATE[2]
INPUT
INPUT-OUTPUT
INSPECT
INSTALLATION
INTO
INVALID
IS

JUST
JUSTIFIED

KEY

LABEL
LAST[2]
LEADING

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

LEFT
LENGTH[2]
LESS
LIKE[2]
LIMIT[2]
LIMITS[2]
LINAGE[2]
LINAGE-COUNTER[2]
LINE
LINE-COUNTER[2]
LINES
LINKAGE
LOCK
LOW
LOWLIGHT[2]
LOW-VALUE
LOW-VALUES

MEMORY
MERGE[2]
MESSAGE[2]
MODE
MODULES
MOVE
MULTIPLY

NATIVE
NEGATIVE[2]
NEXT
NO
NOT
NULL[2]
NULLS[2]
NUMBER[2]
NUMERIC
NUMERIC-EDITED[2]

OBJECT-COMPUTER
OCCURS
OF
OFF
OMITTED
ON
OPEN
OPTIONAL[2]
OR
ORDER[2]
ORGANIZATION
OTHER[2]
OUTPUT
OVERFLOW

PACKED-DECIMAL[2]
PADDING[2]
PAGE
PAGE-COUNTER[2]
PERFORM
PF[2]
PH[2]
PIC
PICTURE
PLUS[2]
POINTER[2]
POSITION
POSITIVE[2]
PRINTING[2]
PROCEDURE
PROCEDURES[2]
PROCEED
PROGRAM
PROGRAM-ID
PROMPT
PURGE[2]

QUEUE[2]
QUOTE
QUOTES

RANDOM
RD[2]

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

READ
RECEIVE [2]
RECORD
RECORDING [2]
RECORDS
REDEFINES
REEL
REFERENCE [2]
REFERENCES [2]
RELATIVE
RELEASE [2]
REMAINDER
REMARKS [2]
REMOVAL [2]
RENAMES
REPLACE [2]
REPLACING
REPORT [2]
REPORTING [2]
REPORTS [2]
RERUN [2]
RESERVE
RESET [2]
RETURN [2]
RETURN-CODE [2]
RETURNING [2]
REVERSE
REVERSE-VIDEO [2]
REVERSED [2]
REWIND
REWRITE
RF [2]
RH [2]
RIGHT
ROUNDED
RUN

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

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

TAB
TABLE [2]
TALLYING
TAPE [2]
TERMINAL [2]
TERMINATE [2]
TEST [2]
TEXT [2]
THAN
THEN [2]
THROUGH
THRU
TIME
TIMES
TO
TOP [2]
TRAILING
TRUE [2]
TYPE [2]

UNIT
UNLOCK
UNSTRING [2]
UNTIL
UP
UPDATE
UPON [2]
USAGE
USE
USING

VALUE
VALUES
VARIABLE [2]
VARYING

WHEN
WITH
WORDS
WORKING-STORAGE
WRITE

ZERO
ZEROES
ZEROS

[2] This word is not considered reserved if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the *RM/COBOL User's Guide* for details on this option). In such cases, this word is treated as a user-defined word whenever it occurs in the source program.

Context-Sensitive Words

The words listed in Table 2 are context-sensitive words and are reserved in the specified language construct or context. If a context-sensitive word is used where the context-sensitive word is permitted in the general format, the word is treated as a keyword; otherwise it is treated as a user-defined word.

Table 2: Context-Sensitive Words

Context-Sensitive Word	**Language Construct or Context**
AUTO [3]	screen description entry
AUTOMATIC [3]	LOCK MODE clause
BACKGROUND [3]	screen description entry
BACKGROUND-COLOR [3]	screen description entry
CARD-PUNCH	ASSIGN clause in file control entry (device-name)
CARD-READER	ASSIGN clause in file control entry (device-name)
CASE-INSENSITIVE [3]	LIKE relational-operator
CASE-SENSITIVE [3]	LIKE relational-operator
CASSETTE	ASSIGN clause in file control entry (device-name)
CONSOLE	ASSIGN clause in file control entry (device-name)
CYCLE [3]	EXIT statement (Format 3)
DISC	ASSIGN clause in file control entry (device-name)
DISK	ASSIGN clause in file control entry (device-name)
EOL	ERASE clause in screen description entry and ERASE phrase in ACCEPT and DISPLAY statements
EOS	ERASE clause in screen description entry and ERASE phrase in ACCEPT and DISPLAY statements

[3] *This word in is not considered to be a context-sensitive word if the RM/COBOL (74) 2.0 compatibility option is present in the Compile Command (see the* RM/COBOL User's Guide *for details on this option). When that option is present, this word is treated as a user-defined word whenever it occurs in the source program.*

Table 2: Context-Sensitive Words *(Cont.)*

Context-Sensitive Word	Language Construct or Context
FOREGROUND [3]	screen description entry
FOREGROUND-COLOR [3]	screen description entry
FULL [3]	screen description entry
KEYBOARD	ASSIGN clause in file control entry (device-name)
LISTING	ASSIGN clause in file control entry (device-name)
MAGNETIC-TAPE	ASSIGN clause in file control entry (device-name)
MANUAL [3]	LOCK MODE clause
MULTIPLE [3]	LOCK MODE clause and I-O-CONTROL paragraph
PARAGRAPH [3]	EXIT statement (Format 4)
PREVIOUS [3]	READ statement (Format 1)
PRINT	ASSIGN clause in file control entry (device-name)
PRINTER	ASSIGN clause in file control entry (device-name)
PRINTER-1	ASSIGN clause in file control entry (device-name)
REQUIRED [3]	screen description entry
SORT-WORK	ASSIGN clause in file control entry (device-name)
TRIMMED [3]	LIKE relational-operator
UNDERLINE [3]	screen description entry
YYYYDDD [3]	FROM DAY phrase of ACCEPT statement (Format 2)
YYYYMMDD [3]	FROM DATE phrase of ACCEPT statement (Format 2)

The DERESERVE keyword of the COMPILER-OPTIONS configuration record, which is described in the "Configuration" chapter of the *RM/COBOL User's Guide*, can be used to make a context-sensitive word a user-defined word whenever it occurs in the source program, but then the language feature provided by the construct in which the word appears is not available for programs compiled with that particular configuration setting.

Nonreserved System-Names

Code-Name

EBCDIC

(Color-Integer) Color-Names

(0) BLACK
(1) BLUE
(2) GREEN
(3) CYAN
(4) RED
(5) MAGENTA
(6) BROWN
(7) WHITE

Computer-Names

user-defined-word-1

Delimiter-Names

BINARY-SEQUENTIAL
LINE-SEQUENTIAL

Device-Names

CARD-PUNCH
CARD-READER
CASSETTE
CONSOLE
DISC
DISK
KEYBOARD
LISTING
MAGNETIC-TAPE
PRINT
PRINTER
PRINTER-1
SORT-WORK

Feature-Names

C01
C02
C03
C04
C05
C06
C07
C08
C09
C10
C11
C12

Label-Names

FILE-ID
user-defined-word-2

Language-Names

user-defined-word-3

Low-Volume-I-O-Names

CONSOLE
SYSIN
SYSOUT

Rerun-Names

user-defined-word-4

Switch-Names

SWITCH-1	UPSI-0
SWITCH-2	UPSI-1
SWITCH-3	UPSI-2
SWITCH-4	UPSI-3
SWITCH-5	UPSI-4
SWITCH-6	UPSI-5
SWITCH-7	UPSI-6
SWITCH-8	UPSI-7