Contents

Chapter 1

Introduction

This part gives the definition for portable COBOL applications across X/OPEN systems. It identifies the common set of language facilities that will be supported by COBOL compilers on the X/OPEN systems of the member companies.

The international standard for COBOL is that defined in the American National Standards document "ANSI X3.23 - 1974", to which most current COBOL compilers substantially conform.

The ANSI standard is incomplete in the area of facilities for interaction with the on-line user. To overcome this deficiency, most COBOL compilers provide extensions to the *ACCEPT* and *DISPLAY* verbs, but unfortunately they do this in incompatible ways. It is necessary therefore to specify the form of *ACCEPT* and *DISPLAY* to be included in the Common Applications Environment.

In order to have a definition that is achievable on X/OPEN systems within a short timeframe, and one that would immediately have wide acceptance, it has been based on the definition of COBOL embodied in a popular product, Micro Focus LEVEL II COBOL which itself conforms to the ANSI Standard.

The Micro Focus LEVEL II language specification includes enhancements to the ANSI standard in addition to the extensions to *ACCEPT* and *DISPLAY*. None of these is currently included in the X/OPEN definition, although they may be supported on specific member systems.

The X/OPEN definition also applies a few restrictions to the ANSI based parts of the LEVEL II definition.

Whilst the definition is based on the specification embodied in a particular product, the means of implementation across X/OPEN systems may vary.

The X/OPEN COBOL definition is given in Chapter 2. It is derived from the Syntax Summary (Appendix F) of the LEVEL II COBOL Reference Manual, with the elements that have been included in the X/OPEN definition clearly printed in bold type. The semantics of the language are those of the ANSI 74 standard as documented in the Micro Focus LEVEL II COBOL language specification. The LEVEL II facilities that are additional to the ANSI standard are indicated by shading. With the exception of the extensions to *ACCEPT* and *DISPLAY* these are all excluded from the X/OPEN definition.

Chapter 3 summarises the functions in the ANSI 74 standard, and the LEVEL II specification, which are excluded from the X/OPEN definition. These are described in relation to the ANSI defined *modules*. This information is included to allow those familiar with the ANSI standard to obtain a quick appreciation of the X/OPEN definition, and it is useful in assessing whether a particular compiler is likely to meet the specification.

Chapter 2

COBOL Definition

The definition is derived from the Syntax Summary sheets from the Micro Focus LEVEL II COBOL Reference Manual (Appendix F) with the following specific notation:

● Shaded areas	indicate Micro Focus extensions to the ANSI standard or features that are documentary only. These are distinguished to the right of the shading by an E or an F for the extensions and a D for documentary only.
● Bold type	Items included in the X/OPEN definition are printed in bold type. For portability across X/OPEN systems, only these elements should be used in application programs.

The tables use standard COBOL notation to define the language syntax:

● Upper case	is used for COBOL language keywords. Those underlined must be present if the clause is present; those not underlined are "noise words", which may be included to improve readability but are otherwise not processed by the compiler.
● Lower case	strings represent substitutable arguments, for example data names and literal values.
● Square brackets	are used to enclose optional clauses (according to the context); any clause not so enclosed is mandatory.
● Braces	are used to enclose alternatives. One of the alternatives enclosed within the braces must be used. Note that braces and square brackets may be used together to indicate alternative constructs within an optional clause.
● Elipses ...	are used to denote that the preceeding clause may be repeated a number of times. There must be at least one occurence, unless the clause is optional (enclosed in square brackets).

The specification of Micro Focus LEVEL II COBOL includes a further category of clause, those which are optional unless the *ANSI parameter* is supplied to the compiler. Since the X/OPEN definition is the ANSI standard as embodied in Micro Focus LEVEL II COBOL, these clauses are shown as mandatory.

GENERAL FORMAT FOR IDENTIFICATION DIVISION

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

GENERAL FORMAT FOR ENVIRONMENT DIVISION

```
ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER.
    source-computer-entry   [WITH DEBUGGING MODE].

OBJECT-COMPUTER.
    object-computer-entry

    [                               {WORDS     } ]
    [ , MEMORY SIZE integer         {CHARACTERS} ]
    [                               {MODULES   } ]

    [, PROGRAM COLLATING SEQUENCE IS alphabet-name]

    [, SEGMENT-LIMIT IS segment number].
```

```
[SPECIAL-NAMES.

 ┌   ┌SYSIN  ┐                   ┐
 │ , │       │ IS mnemonic-name-1 │
 └   └SYSOUT ┘                   ┘

 [ , ┌TAB      ┐ IS mnemonic-name-2]
 [ , └FORMFEED ┘ IS mnemonic-name-3]

 ┌        ┌Ø┐
 │ SWITCH │:│ [IS mnemonic-name] ON STATUS IS condition-name-1
 │        │:│
 └        └7┘                                                      ┐
                               [OFF STATUS IS condition-name-2     │
                                                                   ┘

 ┌ , alphabet-name IS                                                          ┐ . . .
 │ ┌STANDARD-1                ┌┌┌THROUGH┐                          ┐┐  ┐      │
 │ │NATIVE          literal-1 │││THRU   ┘ literal-2                ││  │      │
 │ │                          └└ALSO literal-3 [, ALSO literal-4] . . .┘┘  │      │
 │ │                ┌          ┌┌┌THROUGH┐                   ┐┐          │      │
 │ │                │literal-5 │││THRU   ┘ literal-6         ││          │      │
 └ └                └          └└ALSO literal-7 [, ALSO literal-8]┘┘ . . . ┘      ┘

 [,CURRENCY SIGN IS literal-9]
 [,DECIMAL-POINT IS COMMA]
 [,CURSOR IS data-name-1]                                                  E
 [,CONSOLE IS CRT]          ]                                              E
 [,CRT STATUS IS data-name-2] .
```

```
[INPUT-OUTPUT SECTION.

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

 [ I-O-CONTROL.
   [ ; RERUN [ON {file-name-1         }]
   [             {implementor-name    }]
   [                                                                         ]
   [                 { {[END OF] {REEL} }                   }]
   [                 { {         {UNIT} }  OF file-name-2   }]
   [           EVERY { {integer-1 RECORDS}                  }] . . .          D
   [                 { integer-2 CLOCK-UNITS                }]
   [                 { condition-name                       }]

   [        [RECORD     ]  AREA FOR                                ]
   [ ; SAME [SORT       ]                                          ]
   [        [SORT-MERGE ]        file-name-3 [, file-name-4] . . . ] . . .

   [ ; MULTIPLE FILE TAPE CONTAINS                                               D
   [          file-name-5  [POSITION integer-3]                      ]
   [                  [, file-name-6  [POSITION integer-4]   ] . . . ] . . . .  ]]
```

GENERAL FORMAT FOR FILE-CONTROL ENTRY

SEQUENTIAL SELECT:

```
SELECT file-name [ OPTIONAL     ] file-name-1                              E
                 [ NOT OPTIONAL ]

           [ { LINE ADVANCING          }      ]                            E
ASSIGN TO  [ {                {REEL}   } FILE ]
           [ { MULTIPLE       {UNIT}   }      ]

   {external-file-name-literal} [ , {external-file-name-literal} ] ...     D
   {file-identifier           } [   {file-identifier           } ]

   [ ; RESERVE integer-1 {AREA } ]                                         D
   [                     {AREAS} ]

   ; ORGANIZATION IS [ { SEQUENTIAL      } ]                               E
                     [ { LINE SEQUENTIAL } ]

   [; ACCESS MODE IS SEQUENTIAL]

   [                { EXCLUSIVE } ]                                        F
   [ ; LOCK MODE IS { AUTOMATIC } ]
   [                { MANUAL    } ]

   [; FILE STATUS IS data-name].
```

RELATIVE SELECT:

```
SELECT file-name  [NOT OPTIONAL]                                        E

ASSIGN TO {external-file-name-literal} [, {external-file-name-literal}]
          {file-identifier           } [  {file-identifier           }]

  [; RESERVE integer-1 {AREA }]                                         D
  [                    {AREAS}]

  ORGANIZATION IS RELATIVE

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

  [LOCK MODE IS                                                 ]       F
  [   {{MANUAL   }[WITH LOCK ON [MULTIPLE] {RECORD }]}          ]
  [   {{AUTOMATIC}[                        {RECORDS}]}          ]
  [   {EXCLUSIVE                                     }          ]

  [; FILE STATUS IS data-name].
```

INDEXED SELECT:

```
SELECT file-name [NOT OPTIONAL]

ASSIGN TO {external-file-name-literal} [, {external-file-name-literal}]
          {file-identifier           } [  {file-identifier           }]...

   [; RESERVE integer-1 {AREA }]                                      D
   [                    {AREAS}]

   ; ORGANIZATION IS INDEXED

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

   [LOCK MODE IS                                                  ]   F
   [    {{MANUAL   }[WITH LOCK ON [MULTIPLE] {RECORD }]}]
   [    {{AUTOMATIC}[                        {RECORDS}]}]
   [    {EXCLUSIVE                                     }]

   ; RECORD KEY IS data-name-1

   [; ALTERNATE RECORD KEY IS data-name-2 [WITH DUPLICATES] ] ...

   [; FILE STATUS IS data-name-3].
```

SORT OR MERGE SELECT:

$$\underline{\text{SELECT}}\ \text{file-name}$$

$$\underline{\text{ASSIGN}}\ \text{TO} \left\{\begin{array}{l}\text{external-file-name-literal}\\ \text{file-identifier}\end{array}\right\} \ldots .$$

GENERAL FORMAT FOR THE DATA DIVISION

```
 DATA DIVISION.
 FILE SECTION.
[[FD file-name                                                            ]    ]
 [  [; BLOCK CONTAINS [integer-1 TO] integer 2 {RECORDS   }]              ]     D
 [                                              {CHARACTERS}]             ]
 [                                                                        ]
 [  [; RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS]               ]     D
 [                                                                        ]
 [  ; LABEL {RECORD IS   } {STANDARD}                                     ]     D
 [          {RECORDS ARE } {OMITTED }                                     ]
 [                                                                        ]
 [  [; VALUE OF data-name-1 IS {data-name-2}                              ]     D
 [                             {literal-1  }                              ]
 [                                                                        ]
 [            [, data-name-3 IS {data-name-4}] ...]                       ]     D
 [                              {literal-2  }                             ]
 [                                                                        ]
 [  [; DATA {RECORD IS  } data-name-3  [, data-name-4] ...]               ]     D
 [          {RECORDS ARE}                                                 ]
 [                                                                        ]
 [  [; LINAGE IS {data-name-5} LINES                              ]       ]
 [  [            {integer-5  }                                    ]       ]
 [  [                                                             ]       ]
 [  [          [, WITH FOOTING AT {data-name-6}]                  ]       ]
 [  [                             {integer-6  }                   ]       ]
 [  [                                                             ]       ]
 [  [ [, LINES AT TOP {data-name-7}]                              ]       ]
 [  [                 {integer-7  }                               ]       ]
 [  [                                                             ]       ]
 [  [          [, LINES AT BOTTOM {data-name-8}]                  ]       ]
 [  [                             {integer-8  }                   ]       ]
 [                                                                        ]
 [  [; CODE-SET IS alphabet-name] .                                       ]     D
 [                                                                        ]
 [  [record-description-entry] ...                                        ] ...
 [                                                                        ]
 [[SD file-name                                                   ]            D
 [ [; RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS]        ]
 [                                                                ]
 [ [; DATA {RECORD IS  } data-name-1 [, data-name-2] ...] .       ]
 [         {RECORDS ARE}                                          ]
 [ record-description-entry ...  ...                              ]    ]
```

```
[WORKING-STORAGE SECTION                        ]
[                                               ]
[    [77-level-description-entry]               ]
[    [record-description-entry  ] ...           ]

[LINKAGE SECTION                                ]
[                                               ]
[    [77-level-description-entry]               ]
[    [record-description-entry  ] ...           ]

[COMMUNICATION SECTION                          ]
[                                               ]
[    [communication-description-entry]          ]
[    [record-description-entry ...] ...         ]
```

GENERAL FORMAT FOR DATA DESCRIPTION ENTRY

FORMAT 1:

$$\text{level-number}\ \left\{\begin{array}{l}\text{data-name-1}\\ \underline{\text{FILLER}}\end{array}\right\}$$

$$[;\ \underline{\text{REDEFINES}}\ \text{data-name-2}]$$

$$\left[;\ \left\{\begin{array}{l}\underline{\text{PICTURE}}\\ \underline{\text{PIC}}\end{array}\right\}\ \text{IS picture-string}\right]$$

$$\left[;\ [\underline{\text{USAGE}}\ \text{IS}]\ \left\{\begin{array}{l}\underline{\text{COMPUTATIONAL}}\\ \underline{\text{COMP}}\\ \underline{\text{COMPUTATIONAL-3}}\\ \underline{\text{COMP-3}}\\ \underline{\text{DISPLAY}}\\ \underline{\text{INDEX}}\end{array}\right\}\right]$$

$$\left[;\ \underline{\text{SIGN}}\ \text{IS}\ \left\{\begin{array}{l}\underline{\text{LEADING}}\\ \underline{\text{TRAILING}}\end{array}\right\}\ [\underline{\text{SEPARATE}}\ \text{CHARACTER}]\right]$$

$$\left[\begin{array}{l};\ \underline{\text{OCCURS}}\\ \quad\left\{\begin{array}{l}\text{integer -1}\ \underline{\text{TO}}\ \text{integer-2 TIMES}\ \underline{\text{DEPENDING}}\ \text{ON data-name-3}\\ \text{integer-2 TIMES}\end{array}\right\}\\ \quad\left[\left\{\begin{array}{l}\underline{\text{ASCENDING}}\\ \underline{\text{DESCENDING}}\end{array}\right\}\ \text{KEY IS data-name-4}\ \ [,\ \text{data-name-5}]\ \ldots\right]\ldots\\ \qquad[\underline{\text{INDEXED}}\ \text{BY index-name-1}\ [,\ \text{index-name-2}]\ \ldots]\end{array}\right]$$

$$\left[;\ \left\{\begin{array}{l}\underline{\text{SYNCHRONIZED}}\\ \underline{\text{SYNC}}\end{array}\right\}\ \left\{\begin{array}{l}\underline{\text{LEFT}}\\ \underline{\text{RIGHT}}\end{array}\right\}\right]$$ D

$$\left[;\ \left\{\begin{array}{l}\underline{\text{JUSTIFIED}}\\ \underline{\text{JUST}}\end{array}\right\}\ \text{RIGHT}\right]$$

$$[;\ \underline{\text{BLANK}}\ \text{WHEN}\ \underline{\text{ZERO}}]$$

$$[;\ \underline{\text{VALUE}}\ \text{IS literal}].$$

FORMAT 2:

$$\textbf{66 data-name-1; }\underline{\textbf{RENAMES}}\textbf{ data-name-2 }\left[\left\{\begin{array}{l}\underline{\textbf{THROUGH}}\\ \underline{\textbf{THRU}}\end{array}\right\}\textbf{ data-name-3}\right].$$

FORMAT 3:

$$\textbf{88 condition-name ; }\left\{\begin{array}{l}\underline{\textbf{VALUE}}\textbf{ IS}\\ \underline{\textbf{VALUES}}\textbf{ ARE}\end{array}\right\}\textbf{ literal-1 }\left[\left\{\begin{array}{l}\underline{\textbf{THROUGH}}\\ \underline{\textbf{THRU}}\end{array}\right\}\textbf{ literal-2}\right]$$

$$\left[\textbf{, literal-3 }\left[\left\{\begin{array}{l}\underline{\textbf{THROUGH}}\\ \underline{\textbf{THRU}}\end{array}\right\}\textbf{ literal-4}\right]\right]\ldots\ .$$

GENERAL FORMAT FOR COMMUNICATION DESCRIPTION ENTRY

FORMAT 1:

```
CD cd-name;

                        [ [ [; SYMBOLIC QUEUE IS data-name-1]                 ]
                        [   [; SYMBOLIC SUB-QUEUE-1 IS data-name-2]           ]
                        [   [; SYMBOLIC SUB-QUEUE-2 IS data-name-3]           ]
                        [   [; SYMBOLIC SUB-QUEUE-3 IS data-name-4]           ]
                        [   [; MESSAGE DATE IS data-name-5]                   ]
                        [   [; MESSAGE TIME IS data-name-6]                   ]
FOR [INITIAL] INPUT     [   [; SYMBOLIC SOURCE IS data-name-7]                ]
                        [   [; TEXT LENGTH IS data-name-8]                    ]
                        [   [; END KEY IS data-name-9]                        ]
                        [   [; STATUS KEY IS data-name-10]                    ]
                        [   [; MESSAGE COUNT IS data-name-11] ]               ]
                        [ [data-name-1, data-name-2, . . . , data-name-11]    ] .
```

FORMAT 2:

```
CD cd-name; FOR OUTPUT
  [ ; DESTINATION COUNT IS data-name-1]
  [ ; TEXT LENGTH IS data-name-2]
  [ ; STATUS KEY IS data-name-3]

  [ ; DESTINATION TABLE OCCURS integer-2 TIMES
      [ ; INDEXED BY index-name-1 [, index-name-2] ... ] ]
  [ ; ERROR KEY IS data-name-4]
  [ ; SYMBOLIC DESTINATION IS data-name-4].
```

GENERAL FORMAT FOR PROCEDURE DIVISION

DECLARATIVE FORMAT:

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

    ┌DECLARATIVES.                                               ┐
    │⎧ section-name SECTION [segment-number].  ⎫                 │
    │⎨     declarative-sentence                ⎬                 │
    │⎩ [paragraph-name.  ]                     ⎭                 │
    │  [  [sentence] . . .] . . .                  . . .         │
    └END DECLARATIVES.                                           ┘

    ⎧ section-name SECTION [segment-number].  ⎫
    ⎨ [paragraph-name.  ]                     ⎬
    ⎩ [  [sentence] . . .] . . .              ⎭ . . .
```

NON-DECLARATIVE FORMAT:

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

    ⎧paragraph-name.      ⎫
    ⎩    [sentence] . . . ⎭ . . .
```

GENERAL FORMAT FOR ACCEPT STATEMENT

```
                        { CONSOLE       }
ACCEPT identifier FROM  { mnemonic-name }
```

```
                    { [    { data-name-2 } ]            }
                    { [ AT { literal-1   } ] FROM CRT   }
ACCEPT data-name-1  {                                   }      E
                    {   AT { data-name-2 }              }
                    {      { literal-1   }              }
```

```
                        { DATE }
ACCEPT identifier FROM  { DAY  }
                        { TIME }
```

```
ACCEPT cd-name MESSAGE COUNT
```

GENERAL FORMAT FOR ADD STATEMENT

$$\underline{\textbf{ADD}}\ \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\} \left[\begin{matrix}\text{, identifier-2}\\ \text{, literal-2}\end{matrix}\right] \ldots$$

$$\underline{\textbf{TO}}\ \text{identifier-m}\ [\underline{\textbf{ROUNDED}}]$$

$$\left[\text{, identifier-n}\ [\underline{\textbf{ROUNDED}}]\right] \ldots$$

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

$$\underline{\textbf{ADD}}\ \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\}, \left\{\begin{matrix}\text{identifier-2}\\ \text{literal-2}\end{matrix}\right\} \left[\begin{matrix}\text{, identifier-3}\\ \text{, literal-3}\end{matrix}\right] \ldots$$

$$\underline{\textbf{GIVING}}\ \text{identifier-m}\ [\underline{\textbf{ROUNDED}}] \left[\text{, identifier-n}\ [\underline{\textbf{ROUNDED}}]\right] \ldots$$

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

$$\underline{\textbf{ADD}}\ \left\{\begin{matrix}\underline{\textbf{CORRESPONDING}}\\ \underline{\textbf{CORR}}\end{matrix}\right\} \text{identifier-1}\ \underline{\textbf{TO}}\ \text{identifier-2}\ [\underline{\textbf{ROUNDED}}]$$

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

GENERAL FORMAT FOR ALTER STATEMENT

$$\underline{\textbf{ALTER}}\ \{\text{procedure-name-1}\ \underline{\textbf{TO}}\ [\underline{\textbf{PROCEED}}\ \underline{\textbf{TO}}]\ \text{procedure-name-2}\} \ldots$$

GENERAL FORMAT FOR CALL STATEMENT

```
CALL {identifier-1} [USING data-name-1 [, data-name-2] . . . ]
     {literal-1   }

                    [; ON OVERFLOW imperative-statement]
```

GENERAL FORMAT FOR CANCEL STATEMENT

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

GENERAL FORMAT FOR CLOSE STATEMENT

```
                        [{REEL} [WITH NO REWIND]]
                        [{UNIT} [FOR REMOVAL   ]]
CLOSE file-name-1       [                        ]          D
                        [WITH   {NO REWIND}      ]
                        [       {LOCK     }      ]

       [                [{REEL} [WITH NO REWIND]]]
       [                [{UNIT} [FOR REMOVAL   ]]]
       [, file-name-2   [                        ]]
       [                [WITH   {NO REWIND}      ]]
       [                [       {LOCK     }      ]] . . .
```

GENERAL FORMAT FOR COMMIT STATEMENT

```
COMMIT
```

GENERAL FORMAT FOR COMPUTE STATEMENT

$$\begin{array}{l}
\underline{\textbf{COMPUTE}}\ \textbf{identifier-1}\ [\underline{\textbf{ROUNDED}}]\ \left[,\ \textbf{identifier-2}\ [\underline{\textbf{ROUNDED}}]\right]\ \ldots \\
\qquad\qquad = \textit{arithmetic-expression} \\
\qquad\qquad [;\ \textbf{ON}\ \underline{\textbf{SIZE}}\ \underline{\textbf{ERROR}}\ \textit{imperative statement}]
\end{array}$$

GENERAL FORMAT FOR COPY STATEMENT

$$\underline{\textbf{COPY}} \left\{\begin{array}{l}\textbf{text-name}\\ \text{external-file-name-literal}\end{array}\right\} \left[\left\{\begin{array}{l}\underline{\textbf{OF}}\\ \underline{\textbf{IN}}\end{array}\right\} \left\{\begin{array}{l}\textbf{library-name}\\ \text{library-name-literal}\end{array}\right\}\right]$$

E

$$\left[\underline{\textbf{REPLACING}} \left\{ , \left\{\begin{array}{l}\textit{==pseudo-text-1==}\\ \textbf{identifier-1}\\ \textbf{literal-1}\\ \textbf{word-1}\end{array}\right\} \underline{\textbf{BY}} \left\{\begin{array}{l}\textit{==pseudo-text-2==}\\ \textbf{identifier-2}\\ \textbf{literal-2}\\ \textbf{word-2}\end{array}\right\}\right\} \ldots\right]$$

GENERAL FORMAT FOR DELETE STATEMENT

$$\underline{\textbf{DELETE}}\ \textbf{file-name RECORD}\ [;\ \underline{\textbf{INVALID}}\ \textbf{KEY}\ \textit{imperative-statement}]$$

GENERAL FORMAT FOR DISABLE STATEMENT

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

GENERAL FORMAT FOR DISPLAY STATEMENT

$$\underline{\text{DISPLAY}} \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\} \left[\begin{matrix}\text{, identifier-2}\\ \text{, literal-2}\end{matrix}\right] \ldots \ [\underline{\text{UPON}}\ \underline{\text{CONSOLE}}]$$

$$\underline{\text{DISPLAY}} \left\{\begin{matrix}\text{data-name-1}\\ \text{literal-3}\end{matrix}\right\} \left[\underline{\text{AT}} \left\{\begin{matrix}\text{data-name-2}\\ \text{literal-4}\end{matrix}\right\}\right] \underline{\text{UPON}} \left\{\begin{matrix}\underline{\text{CRT}}\\ \underline{\text{CRT-UNDER}}\end{matrix}\right\}$$

E

GENERAL FORMAT FOR DIVIDE STATEMENT

$$\underline{\text{DIVIDE}} \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\}$$

$$\underline{\text{INTO}}\ \text{identifier-2}\ \ [\underline{\text{ROUNDED}}]$$

$$[\text{, identifier-3}\ [\underline{\text{ROUNDED}}]] \ldots$$

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

$$\underline{\text{DIVIDE}} \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\} \left\{\begin{matrix}\underline{\text{INTO}}\\ \underline{\text{BY}}\end{matrix}\right\} \left\{\begin{matrix}\text{identifier-2}\\ \text{literal-2}\end{matrix}\right\}$$

$$\underline{\text{GIVING}}\ \text{identifier-3}\ [\underline{\text{ROUNDED}}]$$

$$[\text{, identifier-4}\ [\underline{\text{ROUNDED}}]] \ldots$$

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

$$\underline{\text{DIVIDE}} \left\{\begin{matrix}\text{identifier-1}\\ \text{literal-1}\end{matrix}\right\} \left\{\begin{matrix}\underline{\text{INTO}}\\ \underline{\text{BY}}\end{matrix}\right\} \left\{\begin{matrix}\text{identifier-2}\\ \text{literal-2}\end{matrix}\right\}$$

$$\underline{\text{GIVING}}\ \text{identifier-3}\ [\underline{\text{ROUNDED}}]$$

$$\underline{\text{REMAINDER}}\ \text{identifier-4}$$

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

GENERAL FORMAT FOR ENABLE STATEMENT

$\underline{\text{ENABLE}} \begin{Bmatrix} \underline{\text{INPUT}} \ [\underline{\text{TERMINAL}}] \\ \underline{\text{OUTPUT}} \end{Bmatrix}$ cd-name WITH $\underline{\text{KEY}} \begin{Bmatrix} \text{identifier-1} \\ \text{literal-1} \end{Bmatrix}$

GENERAL FORMAT FOR ENTER STATEMENT

ENTER language-name [routine-name]. D

GENERAL FORMAT FOR EXIT STATEMENT

EXIT [PROGRAM].

GENERAL FORMAT FOR GO TO STATEMENT

GO TO [procedure-name].

GO TO procedure-name-1 {, procedure-name-2} . . .

DEPENDING ON identifier

GENERAL FORMAT FOR IF STATEMENT

$\underline{\textbf{IF}}$ **condition;** $\underline{\text{THEN}} \begin{Bmatrix} \textit{statement-1} \\ \underline{\textbf{NEXT}}\ \underline{\textbf{SENTENCE}} \end{Bmatrix}$ E

$\left[\underline{;\ \text{ELSE}} \begin{Bmatrix} \textit{statement-2} \\ \underline{\textbf{NEXT}}\ \underline{\textbf{SENTENCE}} \end{Bmatrix} \right]$

GENERAL FORMAT FOR INSPECT STATEMENT

INSPECT identifier-1 **TALLYING** *tally-clause*

INSPECT identifier-1 **REPLACING** *replacing-clause*

INSPECT identifier **TALLYING** *tally-clause* **REPLACING** *replacing-clause*

TALLY CLAUSE

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

REPLACING CLAUSE

```
{ CHARACTERS BY {identifier-6} [{BEFORE} INITIAL {identifier-7}]                 }
{               {literal-4   } [{AFTER }         {literal-5   }]                 }
{   {   {ALL    } {   {identifier-5}  BY {identifier-6}                          }
{   { , {LEADING} { , {literal-3   }     {literal-4   }                          }
{   {   {FIRST  } {                                                              }
{   [{BEFORE} INITIAL {identifier-7}]  }                                          }
{   [{AFTER }         {literal-5   }]  }...}...                                  }
```

GENERAL FORMAT FOR MERGE STATEMENT

```
MERGE file-name-1

    ON {ASCENDING } KEY data-name-1 [, data-name-2] ...
       {DESCENDING}

   [ON {ASCENDING } KEY data-name-3 [, data-name-4] ...] ...
       {DESCENDING}

   [COLLATING SEQUENCE IS alphabet-name]

   USING file-name-2, file-name-3 [, file-name-4] ...

   { OUTPUT PROCEDURE IS section-name-1 [{THROUGH} section-name-2] }
   {                                    [{THRU   }               ] }
   { GIVING file-name-5                                            }
```

GENERAL FORMAT FOR MOVE STATEMENT

$$\underline{\textbf{MOVE}}\ \left\{\begin{matrix}\textbf{identifier-1}\\ \textbf{literal-1}\end{matrix}\right\}\ \underline{\textbf{TO}}\ \textbf{identifier-2}\ [\textbf{, identifier-3}]\ \ldots$$

$$\underline{\textbf{MOVE}}\ \left\{\begin{matrix}\underline{\textbf{CORRESPONDING}}\\ \underline{\textbf{CORR}}\end{matrix}\right\}\ \textbf{identifier-1}\ \underline{\textbf{TO}}\ \textbf{identifier-2}$$

GENERAL FORMAT FOR MULTIPLY STATEMENT

$$\underline{\textbf{MULTIPLY}}\ \left\{\begin{matrix}\textbf{identifier-1}\\ \textbf{literal-1}\end{matrix}\right\}$$

$$\underline{\textbf{BY}}\ \textbf{identifier-2}\ [\underline{\textbf{ROUNDED}}]$$

$$\left[\textbf{, identifier-3}\ [\underline{\textbf{ROUNDED}}]\right]\ \ldots$$

$$[\textbf{; ON}\ \underline{\textbf{SIZE}}\ \underline{\textbf{ERROR}}\ \textbf{\textit{imperative-statement}}]$$

$$\underline{\textbf{MULTIPLY}}\ \left\{\begin{matrix}\textbf{identifier-1}\\ \textbf{literal-1}\end{matrix}\right\}\ \underline{\textbf{BY}}\ \left\{\begin{matrix}\underline{\textbf{identifier-2}}\\ \underline{\textbf{literal-2}}\end{matrix}\right\}$$

$$\underline{\textbf{GIVING}}\ \textbf{identifier-3}\ [\underline{\textbf{ROUNDED}}]\ \left[\textbf{, identifier-4}\ [\underline{\textbf{ROUNDED}}]\right]\ \ldots$$

$$[\textbf{; ON}\ \underline{\textbf{SIZE}}\ \underline{\textbf{ERROR}}\ \textbf{\textit{imperative-statement}}]$$

GENERAL FORMAT FOR OPEN STATEMENT

```
       ⎧ INPUT file-name-1 ⎡REVERSED       ⎤                        ⎫
       ⎪                   ⎣WITH NO REWIND ⎦                        ⎪
       ⎪                                                            ⎪
       ⎪     ⎡, file-name-2 ⎡REVERSED       ⎤⎤ ...                  ⎪
       ⎪     ⎣              ⎣WITH NO REWIND ⎦⎦                      ⎪
OPEN   ⎨                                                            ⎬ ...
       ⎪ OUTPUT file-name-3  [WITH NO REWIND]                       ⎪
       ⎪      [, file-name-4  [WITH NO REWIND] ] ...                ⎪
       ⎪                                                            ⎪
       ⎪ I-O file-name-5   [, file-name-6] ...                      ⎪
       ⎩ EXTEND file-name-7 [, file-name-8] ...                     ⎭
```

GENERAL FORMAT FOR PERFORM STATEMENT

```
PERFORM procedure-name-1

     [ { THROUGH }                   ]
     [ { THRU    }  procedure-name-2 ]


PERFORM procedure-name-1

     [ { THROUGH }                   ]  { identifier-1 }
     [ { THRU    }  procedure-name-2 ]  { integer-1    }  TIMES


PERFORM procedure-name-1

     [ { THROUGH }                   ]
     [ { THRU    }  procedure-name-2 ]  UNTIL condition-1


PERFORM procedure-name-1

     [ { THROUGH }                   ]
     [ { THRU    }  procedure-name-2 ]

   VARYING  { identifier-2 }        { identifier-3 }
            { index-name-1 }  FROM  { index-name-2 }
                                    { literal-1    }

      BY { identifier-4 }  UNTIL condition-1
         { literal-2    }

  [ AFTER  { identifier-5 }        { identifier-6 }                 ]
  [        { index-name-3 }  FROM  { index-name-4 }                 ]
  [                                { literal-3    }                 ]
  [                                                                 ]
  [    BY { identifier-7 }  UNTIL condition-2                       ]
  [       { literal-4    }                                          ]
  [                                                                 ]
  [   [ AFTER  { identifier-8 }        { identifier-9 }  ]          ]
  [   [        { index-name-5 }  FROM  { index-name-6 }  ]          ]
  [   [                                { literal-5    }  ]          ]
  [   [                                                  ]          ]
  [   [ BY { identifier }  UNTIL condition-3             ]          ]
  [   [    { literal-6  }                                ]          ]
```

GENERAL FORMAT FOR READ STATEMENT

READ file-name [NEXT] RECORD [INTO identifier]

[WITH [KEPT] LOCK] F

[; **AT END** ***imperative-statement***]

READ file-name RECORD [INTO identifier]

[WITH [KEPT] LOCK] F

[; **KEY IS data-name**]

[; **INVALID KEY** ***imperative-statement***]

GENERAL FORMAT FOR RECEIVE STATEMENT

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

[; NO DATA *imperative-statement*]

GENERAL FORMAT FOR RELEASE STATEMENT

RELEASE record name [FROM identifier]

GENERAL FORMAT FOR RETURN STATEMENT

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

GENERAL FORMAT FOR REWRITE STATEMENT

$$\underline{\text{REWRITE}}\ \text{record-name}\ [\underline{\text{FROM}}\ \text{identifier}]$$

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

GENERAL FORMAT FOR SEARCH STATEMENT

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

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

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

$$\left[;\ \underline{\text{WHEN}}\ \textit{condition-2}\left\{\begin{matrix}\textit{imperative-statement-3}\\ \underline{\text{NEXT}}\ \underline{\text{SENTENCE}}\end{matrix}\right\}\right]\ \ldots$$

$$\underline{\text{SEARCH}}\ \underline{\text{ALL}}\ \text{identifier-1}\ [;\ \text{AT}\ \underline{\text{END}}\ \textit{imperative-statement-1}]$$

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

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

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

GENERAL FORMAT FOR SEND STATEMENT

```
SEND cd-name FROM identifier-1

                                    { WITH identifier-2 }
SEND cd-name  [FROM identifier-1]   { WITH ESI          }
                                    { WITH EMI          }
                                    { WITH EGI          }

    [                         { { identifier-3 } [ LINE  ] } ]
    [ { BEFORE }  ADVANCING   { { integer      } [ LINES ] } ]
    [ { AFTER  }              { { mnemonic-name }          } ]
    [                         { { PAGE          }          } ]
```

GENERAL FORMAT FOR SET STATEMENT

```
                                                       { identifier-3 }
SET { identifier-1    [identifier-2]   } . . . TO      { index-name-3 }
    { index-name-1    [index-name-2]   }               { integer-1    }

                                                       { identifier-4 }
SET { index-name-4   [, index-name-5]  } . . . { UP BY   } { integer-2    }
    { identifier-5   [, identifier-6]  }       { DOWN BY } { index-name-6 }
```

GENERAL FORMAT FOR SORT STATEMENT

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

                  [ON {ASCENDING  } KEY data-name-3  [, data-name-4] . . .] . .
                  [   {DESCENDING }                                       ]

   [COLLATING SEQUENCE IS alphabet-name]

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

   {OUTPUT PROCEDURE IS section-name-3 [{THROUGH} section-name-4]}
   {                                   [{THRU   }               ]}
   {GIVING file-name-4                                           }
```

GENERAL FORMAT FOR START STATEMENT

```
                      [     {IS EQUAL            }           ]
                      [     {IS =                }           ]
START file-name       [ KEY {IS GREATER THAN     } data-name ]
                      [     {IS >                }           ]
                      [     {IS NOT LESS THAN    }           ]
                      [     {IS NOT <            }           ]

   [; INVALID KEY imperative-statement]
```

GENERAL FORMAT FOR STOP STATEMENT

```
STOP {RUN    }
     {literal}
```

GENERAL FORMAT FOR STRING STATEMENT

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

$$\left[, \left\{\begin{array}{l}\text{identifier-4}\\ \text{literal-4}\end{array}\right\} \left[, \left\{\begin{array}{l}\text{identifier-5}\\ \text{literal-5}\end{array}\right\}\right] \ldots \underline{\text{DELIMITED}}\ \text{BY} \left\{\begin{array}{l}\text{identifier-6}\\ \text{literal-6}\\ \underline{\text{SIZE}}\end{array}\right\}\right] \ldots$$

$$\underline{\text{INTO}}\ \text{identifier-7}\ [\text{WITH}\ \underline{\text{POINTER}}\ \text{identifier-8}]$$

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

GENERAL FORMAT FOR SUBTRACT STATEMENT

$$\underline{\text{SUBTRACT}} \left\{\begin{array}{l}\text{identifier-1}\\ \text{literal-1}\end{array}\right\} \left[, \left\{\begin{array}{l}\text{identifier-2}\\ \text{literal-2}\end{array}\right\}\right] \ldots$$

$$\underline{\text{FROM}}\ \text{identifier-m}\ [\underline{\text{ROUNDED}}]\ \left[, \text{identifier-n}\ [\underline{\text{ROUNDED}}]\ \right] \ldots$$

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

$$\underline{\text{SUBTRACT}} \left\{\begin{array}{l}\text{identifier-1}\\ \text{literal-1}\end{array}\right\} \left[\begin{array}{l}, \text{identifier-2}\\ , \text{literal-2}\end{array}\right] \ldots$$

$$\underline{\text{FROM}} \left\{\begin{array}{l}\text{identifier-m}\\ \text{literal-m}\end{array}\right\}$$

$$\underline{\text{GIVING}}\ \text{identifier-n}\ [\underline{\text{ROUNDED}}]\ \left[, \text{identifier-o}\ [\underline{\text{ROUNDED}}]\ \right] \ldots$$

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

$$\underline{\text{SUBTRACT}} \left\{\begin{array}{l}\underline{\text{CORRESPONDING}}\\ \underline{\text{CORR}}\end{array}\right\} \text{identifier-1}$$

$$\underline{\text{FROM}}\ \text{identifier-2}\ [\underline{\text{ROUNDED}}]$$

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

GENERAL FORMAT FOR UNSTRING STATEMENT

```
UNSTRING identifier-1

  [                          { identifier-2 } [          { identifier-3 } ]     ]
  [ DELIMITED BY [ALL]       { literal-1    } [ , OR [ALL] { literal-2  } ] ... ]
  [                                                                            ]

  INTO identifier-4 [, DELIMITER IN identifier-5] [, COUNT IN identifier-6]

     [, identifier-7 [, DELIMITER IN identifier-8] [, COUNT IN identifier-9] ] ...

  [WITH POINTER identifier-10] [TALLYING IN identifier-11]

  [; ON OVERFLOW imperative-statement]
```

GENERAL FORMAT FOR USE STATEMENT

```
USE AFTER STANDARD { EXCEPTION } PROCEDURE
                   { ERROR     }

     { file-name-1 [ , file-name-2] ... }
     { INPUT                            }
  ON { OUTPUT                           }
     { I-O                              }
     { EXTEND                           }

                            { cd-name-1                             }
                            { [ALL REFERENCES OF] identifier-1      }
USE FOR DEBUGGING ON        { file-name-1                           }
                            { procedure-name-1                      }
                            { ALL PROCEDURES                        }

                            [   cd-name-2                           ]
                            [   [ALL REFERENCES OF] identifier-2    ]
                            [ ; file-name-2                         ] ... .
                            [   procedure-name-2                    ]
                            [   ALL PROCEDURES                      ]
```

GENERAL FORMAT FOR WRITE STATEMENT

<u>WRITE</u> record-name [<u>FROM</u> identifier-1]

$$\left[\begin{Bmatrix}\underline{\textbf{BEFORE}}\\ \underline{\textbf{AFTER}}\end{Bmatrix} \textbf{ ADVANCING } \left\{\begin{matrix}\begin{Bmatrix}\textbf{integer}\\ \textbf{identifier-2}\end{Bmatrix}\begin{Bmatrix}\textbf{LINE}\\ \textbf{LINES}\end{Bmatrix}\\ \begin{Bmatrix}\underline{\text{PAGE}}\\ \underline{\text{TAB}}\end{Bmatrix}\end{matrix}\right\}\right]$$

E

$$\left[\textbf{; AT }\begin{Bmatrix}\underline{\textbf{END-OF-PAGE}}\\ \underline{\textbf{EOP}}\end{Bmatrix}\textbf{\textit{imperative statement}}\right]$$

<u>WRITE</u> record-name [<u>FROM</u> identifier]

[; <u>INVALID</u> KEY *imperative-statement*]

GENERAL FORMAT FOR CONDITION STATEMENTS

Relation condition:

$$\left\{\begin{array}{l}\textbf{identifier-1}\\ \textbf{literal-1}\\ \textbf{arithmetic-expression-1}\\ \textbf{index-name-1}\end{array}\right\}\left\{\begin{array}{l}\textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{\textbf{GREATER}}\textbf{ THAN}\\ \textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{\textbf{LESS}}\textbf{ THAN}\\ \textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{\textbf{EQUAL}}\textbf{ to}\\ \textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{\geq}\\ \textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{\leq}\\ \textbf{IS [}\underline{\textbf{NOT}}\textbf{] }\underline{=}\end{array}\right\}\left\{\begin{array}{l}\textbf{identifier-2}\\ \textbf{literal-2}\\ \textbf{arithmetic-expression-2}\\ \textbf{index-name-2}\end{array}\right\}$$

Class condition:

$$\textbf{identifier IS [}\underline{\textbf{NOT}}\left\{\begin{array}{l}\underline{\textbf{NUMERIC}}\\ \underline{\textbf{ALPHABETIC}}\end{array}\right\}$$

Sign condition:

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

Condition-name condition:

condition-name

Switch-status condition:

condition-name

Negated simple condition:

<u>NOT</u> simple-condition

Combined condition:

$$\textbf{condition}\left\{\left\{\begin{array}{l}\underline{\textbf{AND}}\\ \underline{\textbf{OR}}\end{array}\right\}\textbf{condition}\right\}\ \ldots$$

Abreviated combined relation Condition:

$$\textbf{relation-condition}\left\{\left\{\begin{array}{l}\underline{\textbf{AND}}\\ \underline{\textbf{OR}}\end{array}\right\}\textbf{[}\underline{\textbf{NOT}}\textbf{] [relational-operator] object}\right\}\ \ldots$$

MISCELLANEOUS FORMATS

QUALIFICATION:

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

$$\textbf{paragraph-name}\left[\left\{\begin{array}{l}\underline{\textbf{OF}}\\ \underline{\textbf{IN}}\end{array}\right\}\textbf{ section-name}\right]$$

$$\textbf{text-name}\left[\left\{\begin{array}{l}\underline{\textbf{OF}}\\ \underline{\textbf{IN}}\end{array}\right\}\textbf{ library-name}\right]$$

SUBSCRIPTING:

$$\left\{\begin{array}{l}\textbf{data-name}\\ \textbf{condition-name}\end{array}\right\}\left(\textbf{subscript-1 }\left[\textbf{, subscript-2 }[\textbf{, subscript-3}]\right]\right)$$

INDEXING:

$$\left\{\begin{matrix}\textbf{data-name}\\ \textbf{condition-name}\end{matrix}\right\}\left(\left\{\begin{matrix}\textbf{index-name-1}\\ \textbf{literal-1}\end{matrix}\ [\{\pm\}\textbf{literal-2}]\right\}\right.$$

$$\left[,\left\{\begin{matrix}\textbf{index-name-2}\\ \textbf{literal-3}\end{matrix}\ [\{\pm\}\textbf{literal-4}]\right\}\right.$$

$$\left.\left[,\left\{\begin{matrix}\textbf{index-name-3}\\ \textbf{literal-5}\end{matrix}\ [\{\pm\}\textbf{literal-6}]\right\}\right]\right]\Bigg)$$

IDENTIFIER: FORMAT 1

$$\textbf{data-name-1}\left[\left\{\begin{matrix}\underline{\textbf{OF}}\\ \underline{\textbf{IN}}\end{matrix}\right\}\textbf{data-name-2}\right]\ldots$$

$$\left[\left(\textbf{subscript-1}\ \left[,\textbf{subscript-2}\ [,\textbf{subscript-3}]\ \right]\right)\right]$$

IDENTIFIER: FORMAT 2

$$\textbf{data-name-1}\left[\left\{\begin{matrix}\underline{\textbf{OF}}\\ \underline{\textbf{IN}}\end{matrix}\right\}\textbf{data-name-2}\right]\ldots$$

$$\left[\left(\left\{\begin{matrix}\textbf{index-name-1}\\ \textbf{literal-1}\end{matrix}\ [\{\pm\}\textbf{literal-2}]\right\}\right.\right.$$

$$\left[,\left\{\begin{matrix}\textbf{index-name-2}\\ \textbf{literal-3}\end{matrix}\ [\{\pm\}\textbf{literal-4}]\right\}\right.$$

$$\left.\left.\left[,\left\{\begin{matrix}\textbf{index-name-3}\\ \textbf{literal-5}\end{matrix}\ [\{\pm\}\textbf{literal-6}]\right\}\right]\right]\Bigg)\right]$$

Chapter 3

Summary of Exclusions

This Chapter summarises the exclusions from both the ANSI Standard and the Micro Focus LEVEL II definition in a convenient form for those familiar with the ANSI document. It describes the exclusions in relation to the ANSI *modules*. It is also useful for assessing whether a particular compiler meets the X/OPEN definition.

- Modules included in the ANSI standard but excluded by both LEVEL II and X/OPEN definitions:

  Report Writer

- Complete modules that are in the ANSI standard and LEVEL II but are excluded from the X/OPEN definition:

  Communication
  Sort-Merge

- Modules included by X/OPEN at a lower level (as defined in ANSI standard) than that defined in LEVEL II. The following list identifies the items excluded:

  Debug
  - *DEBUG-ITEM*
  - *USE FOR DEBUGGING* statement

  Inter-Program Communication
  - *CALL* statement *identifier-1* option
  - *ON OVERFLOW* phrase of *CALL* statement
  - *CANCEL* statement in entirety

- Individual elements included in both ANSI and LEVEL II but excluded from the X/OPEN definition:

  ENTER statement (Nucleus module)

  CLOSE statement phrases:
  - *REEL*
  - *UNIT*
  - *WITH NO REWIND*
  - *FOR REMOVAL*

  OPEN statement phrases:
  - *WITH NO REWIND*
  - *REVERSED*

- Individual elements in ANSI Standard, but defined as "documentary only" in LEVEL II COBOL, and excluded from X/OPEN definition:

RERUN	clause in	Sequential I/O, Relative I/O and Indexed I/O modules
MULTIPLE FILE TAPE	clause in	Sequential I/O module
BLOCK CONTAINS	clause in	Sequential I/O, Relative I/O and Indexed I/O modules
CODE-SET	clause in	Sequential I/O module
LABEL	clause in	Sequential I/O, Relative I/O and Indexed I/O modules

Summary of Exclusions

VALUE OF	clause in	Sequential I/O, Relative I/O and Indexed I/O modules
DATA	clause in	Sequential I/O, Relative I/O and Indexed I/O modules
RECORD CONTAINS	clause in	Sequential I/O, Relative I/O and Indexed I/O modules