A publication of

Philips Data Systems
SSS Training & Documentation
Apeldoorn, The Netherlands

Order Number 5122 993 11632

Manual Number C2A

CONTENTS Page

IDENTIFICATION DIVISION Format

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

ENVIRONMENT DIVISION Format

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
    [SOURCE-COMPUTER.   computer-name      [WITH DEBUGGING MODE].]
    OBJECT-COMPUTER.    computer-name
        [MEMORY SIZE integer { CHARACTERS }]
                             { MODULES    }
                             { WORDS      }
        [PROGRAM COLLATING SEQUENCE IS alphabet-name] .
    SPECIAL-NAMES.
        [system-name IS mnemonic-name] ...
        [ALPHABET alphabet-name IS { EBCDIC     }]  ...
                                   { NATIVE     }
                                   { STANDARD-1 }
        [SYMBOLIC CHARACTERS {symbolic-character-1 [, symbolic-character-2]...
            { IS  }
            { ARE }      integer-1 [, integer-2]...}... [IN alphabet-name]...
        [CURRENCY SIGN IS literal]
        [DECIMAL-POINT IS COMMA].
```

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

file-control-entry for SEQUENTIAL files

```
SELECT file-name
ASSIGN TO system-name-1 [, system-name-2] ...

[RESERVE integer [ AREA  ]]
                 [ AREAS ]

[FILE STATUS IS data-name]
[ORGANIZATION IS SEQUENTIAL]
[ACCESS MODE IS SEQUENTIAL] .
```

file-control-entry for RELATIVE files

```
SELECT file-name
ASSIGN TO system-name-1 [, system-name-2] . . .

[RESERVE integer [AREA / AREAS]]
[FILE STATUS IS data-name-1]
ORGANIZATION IS RELATIVE
[ACCESS MODE IS { SEQUENTIAL [, RELATIVE KEY is data-name-2] / { RANDOM / DYNAMIC }, RELATIVE KEY is data-name-2 }]
```

file-control-entry for INDEXED files

```
SELECT file-name
ASSIGN TO system-name-1 [, system-name-2] . . .

[RESERVE integer [AREA / AREAS]]
[FILE STATUS IS data-name-1]
ORGANIZATION IS INDEXED
[ACCESS MODE IS { SEQUENTIAL / RANDOM / DYNAMIC }]
RECORD KEY IS data-name-2 [, data-name-3] . . .
[ALTERNATE RECORD KEY IS data-name-4 [, data-name-5] . . .
    [WHEN data-name-6 IS [NOT] literal] [WITH DUPLICATES]] . . . .
```

input-output-control-entry

```
[RERUN [ON { file-name-1 / implementor-name }]
 EVERY { { [END OF] { REEL / UNIT } integer-1 RECORDS } OF file-name-2
       { integer-2 CLOCK-UNITS
       { condition-name                                } } . . .
SAME AREA FOR file-name-3 [, file-name-4] . . . . . .
```

Philips format:

```
[APPLY FOR MASTER-INDEX integer KEYS ON file-name-5 [, file-name-6] . . .
```

IBM format:

```
[APPLY CORE-INDEX TO data-name ON file-name-5 [, file-name-6] . . . ] . . .
```

DATA DIVISION Format

```
DATA DIVISION.
[ FILE SECTION.                                                    ]
[ [file-description-entry                                          ]
[       [record-description-entry]  . . .]  . . .                  ]
[ WORKING-STORAGE SECTION.                                         ]
[ [ noncontiguous-data-item-description-entry ]. . .               ]
[ [ record-description-entry                  ]                    ]
[ LINKAGE SECTION.                                                 ]
[ [ noncontiguous-data-item-description-entry ]. . .               ]
[ [ record-description-entry                  ]                    ]
[ COMMUNICATION SECTION.                                           ]
[ [communication-description-entry]                                ]
[       [record-desciption-entry]  . . .                           ]
[ REPORT SECTION.                                                  ]
[ [report-description-entry                                        ]
[       {report-group-description-entry}  . . .]  . . .            ]
```

File-Description-Entry

```
FD file-name
    [IS EXTERNAL]
    [BLOCK CONTAINS integer { RECORDS    } ]
    [                       { CHARACTERS } ]

    [RECORD CONTAINS integer CHARACTERS]
    [DATA { RECORD IS  } data-name-1 [, data-name-2] . . .]
          { RECORDS ARE }
    LABEL { RECORD IS  } { STANDARD }
          { RECORDS ARE } { OMITTED  }
[CODE-SET IS alphabet-name]
```

```
[                                                                                                  ]
[ VALUE OF { LABEL IS { literal-1 }                                                                ]
[          { [        { data-name-1 } ]                                                          ]
[          { [ DATASET-NAME IS { literal-2 } ]                                                    ]
[          { [                 { data-name-2 } ]                                                  ]
[          {       VOLUME-NAME IS { literal-3 }                                                   ]
[          {                      { data-name-3 }                                                 ]
[          {  [ , VOLUME-NAME IS { literal-4 } ] . . .                                            ]
[          {                     { data-name-4 }                                                  ]
[          {                                                                                      ]
[          {  implementor-name-1 IS { literal-5 }                                                 ]
[          {                        { data-name-5 }                                               ]
[          {    [ , implementor-name-2 IS { literal-6 } ] . . .                                   ]
[                                         { data-name-6 }                                          ]
```

```
LINAGE IS integer-1 LINES
    [WITH FOOTING AT integer-2]
    [LINES AT TOP integer-3]
    [LINES AT BOTTOM integer-4]
```

Communication-Description-Entry

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

CD cd-name [IS EXTERNAL] FOR OUTPUT [COMPRESSED]
  [TEXT LENGTH IS data-name-6]
  [STATUS KEY IS data-name-8]
  [SYMBOLIC DESTINATION IS data-name-10]  .
```

Common Formats and Clauses

```
Noncontiguous-data-item-description-entry
     77 elementary-item-description-entry
              [condition-description-entry] . . .

Record-description-entry
     {[level number   group-item-description-entry] . . .
              {level-number   elementary-item-description-entry
              [condition-description-entry] . . .} . . .} . . .

Group-item-description-entry
          [ data-name ]
          [ FILLER    ]
     [external-clause]
     [redefines-clause]
     [occurs-clause]
     [sign-clause]
     [usage-clause]
     [value-clause]
```

Elementary-item-description-entry

```
      [ data-name ]
      [ FILLER    ]
  [external-clause]
  [redefines-clause]
  [occurs-clause]
  [picture-clause]
  [usage-clause]
  [sign-clause]
  [synchronized-clause]
  [justified-clause]
  [blank-clause]
  [value-clause]
```

Condition-description-entry

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

Redefines-clause

```
REDEFINES [data-name]
```

Occurs-clause

```
OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-1 }
       { integer-2 TIMES                                       }

  [ { ASCENDING  } KEY IS data-name-2 [, data-name-3] . . . ] . . .
    { DESCENDING }
  [INDEXED BY index-name-1 [, index-name-2] . . .]
```

Usage-clause

```
[USAGE IS] ( DISPLAY          )
           ( COMP             )
           ( COMPUTATIONAL    )
           ( COMP-3           )
           ( COMPUTATIONAL-3  )
           ( COMP-4           )
           ( COMPUTATIONAL-4  )
           ( INDEX            )
```

Synchronized-clause

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

Sign-clause

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

Justified-clause

{ JUSTIFIED / JUST } RIGHT

Blank-clause

BLANK WHEN ZERO

Value-clause

[HEX] VALUE IS literal

Picture-clause

{ PICTURE / PIC } IS character-string

External-clause

IS EXTERNAL

Character-String

A character string is a valid combination of the following characters:

numeric	0 and 9
alphabetic	A B P S V X Z CR DB
special	+ − . , * ◊ $ / 'currency-sign'

The characters that can be used in a character-string depend on the type of item being described. The meanings of the allowable characters for the different types item are as follows:

Alphabetic item (A-from string)

A any alphabetic character
B blank insertion

Alphanumeric item (AN-form string)

X any character
A any alphabetic character
9 any numeric character

Alphanumeric edited (AE-form string)

X any character
A any alphabetic character
9 any numeric character
0 zero insertion
B blank insertion
/ slash insertion

Numeric item (N-form string)

9	any numeric character
P	assumed zero position
S	operational sign
V	assumed decimal point

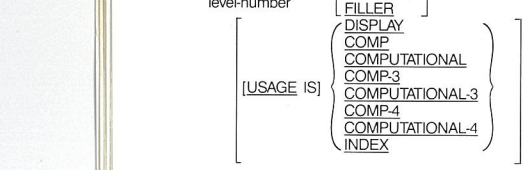
Numeric edited (NE-form string)

9	any numeric character
Z	numeric character or blank
*	* character or cheque protection
.	actual decimal point (1) - see below
,	character insertion (1) - see below
B	blank insertion
0	zero insertion
◊	◊ character or cheque protection
V	assumed decimal point
+	plus or minus sign (2) (3) (4) - see below
–	blank or minus sign (2) (3) (4) - see below
$	dollar sign insertion (3) - see below
CR	insert two blanks or CR (2) (5) - see below
DB	insert two blanks or DB (2) (5) - see below
P	assumed zero position
/	slash insertion

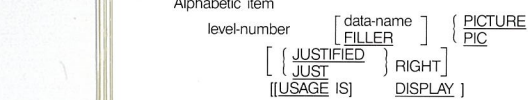
(1) Functions of . and , are reversed if DECIMAL-POINT IS COMMA is specified in Environment Division.
(2) First alternative if item positive; second alternative if item negative.
(3) Can be fixed or floating.
(4) Can be leading (fixed or floating) or trailing (fixed only).
(5) Must be trailing.

Summary of Item Descriptions

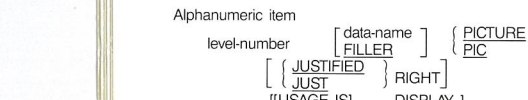
Group item



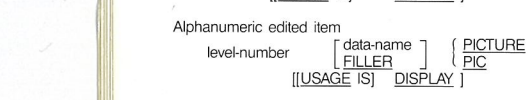
Alphabetic item



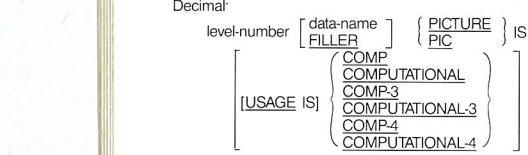
Alphanumeric item



Alphanumeric edited item

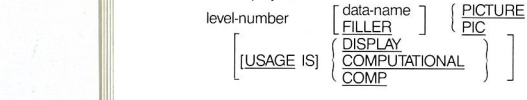


Decimal

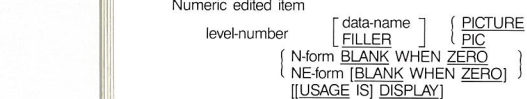


Index data item

Numeric display item



Numeric edited item



PROCEDURE DIVISION Format

Synopsis and Common Formats

Format 1:

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

[DECLARATIVES.

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

[paragraph-name.    [sentence] . . .] . . .} . . .

END DECLARATIVES.]

{[section-name SECTION]    [segment-number].

[paragraph-name.    [sentence] . . . ] . . .} . . .
```

Format 2:

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

{paragraph-name.    [sentence] . . .} . . .

sentence
{statement-1 [, statement-2] . . .} .
```

Statements in Alphabetical Order

```
ACCEPT    identifier  [ FROM { mnemonic-name }  ]
                       {  DATE          }
                       {  TIME          }
                       {  DAY           }

ACCEPT    cd-name  { MESSAGE COUNT          }
                   { NEXT SYMBOLIC SOURCE   }

ADD { identifier-1 } [ identifier-2 ] . . . TO identifier-m [rounded-option]
    { literal-1    } [ literal-2    ]
      [, identifier-n [rounded-option]] . . .
      [size-error-option]

ADD { identifier-1 } [ identifier-2 ] . . . TO { identifier-k }
    { literal-1    } [ literal-2    ]          { literal-3    }
      GIVING identifier-m [rounded-option]
           [, identifier-n  [rounded-option]] . . .
      [size-error-option]

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

```
CALL    literal-1
        [ [USING { identifier-1 } [ { identifier-2 } ] ... ] ]
                 { literal-2   }   { literal-3   }
                 { file-name-1 }   { file-name-2 }
                 { cd-name-1   }   { cd-name-2   }

CLOSE   file-name-1 [, file-name-2] ...

COMMIT  [RESERVE ACCESS TO file-name-1 [, file-name-2] ...]

COMPUTE identifier-1 [rounded-option] [, identifier-2 [rounded-option]] ...
            = arithmetic-expresion [size-error-option]

COPY    member-name [ { OF } library-name ]
                    { IN }

DELETE  file-name RECORD [invalid-key-option]

DISABLE [ INPUT  ] [TERMINAL] ] cd-name
        [ OUTPUT ]

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

DIVIDE  { identifier-1 }
        { literal      }   INTO identifier-2 [rounded-option]
              [, identifier-3 [rounded-option]] ...
            [size-error-option]

DIVIDE  { identifier-1 } { INTO } { identifier-2 }
        { literal-1    } { BY   } { literal-2    }
          GIVING identifier-3 [rounded-option]
                [, identifier-4] [rounded-option]] ...
          [size-error-option]

DIVIDE  { identifier-1 } { INTO } { identifier-2 }
        { literal-1    } { BY   } { literal-2    }
          GIVING identifier-3 [rounded-option]
          REMAINDER identifier-4
          [size-error-option]

ENABLE  [ INPUT  ] [TERMINAL] ] cd-name
        [ OUTPUT ]

EXIT    [PROGRAM]

GO TO   procedure-name

GO TO   procedure-name-1, procedure-name-2 [, procedure-name-n] ...
          DEPENDING ON identifier
```

```
IF condition THEN { statement-1 } [ ELSE statement-2 ]
                  { NEXT SENTENCE }  [ ELSE NEXT SENTENCE ]

INSPECT identifier-1 TALLYING

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

 [REPLACING
 { 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    }   }      }              }

INSPECT identifier-2 REPLACING
 { 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    }   }      }              }

MOVE { identifier-1 } TO identifier-2 [rounded-option]
     { literal      }

     [, identifier-3 [rounded-option]] ...

MULTIPLY { identifier-1 } BY identifier-2 [rounded-option]
         { literal-1    }
           [, identifier-3 [rounded-option]] ...
         [size-error-option]

MULTIPLY { identifier-1 } BY { identifier-2 }
         { literal-1    }    { literal-2    }
         GIVING identifier-3 [rounded-option]
                [, identifier-4 [rounded-option]] ...
         [size-error-option]

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

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

         [ { identifier-1 } TIMES ]
           { integer-1    }

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

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

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

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

         [ AFTER  { identifier-4 } FROM { identifier-6 }
                  { index-name-3 }      { index-name-4 }
                                        { literal-3    }

                  BY { identifier-7 } UNTIL condition-2
                     { literal-4    }

         [ AFTER  { identifier-8 } FROM { identifier-9 }
                  { index-name-5 }      { index-name-6 }
                                        { literal-5    }

                  BY { identifier-10 } UNTIL condition-3 ] ]
                     { literal-6     }

READ     file-name [NEXT] RECORD [INTO identifier]
         [at-end-option]

READ     file-name RECORD [INTO identifier]
         [KEY IS data-name-1 [, data-name-2] . . .]
         [invalid-key-option]

RECEIVE      cd-name
           { MESSAGE INTO identifier-1                    }
           { SEGMENT INTO identifier-2 [WITH identifier-3] }
             [no-data-option]

REWRITE      record-name [FROM identifier]
             [invalid-key-option]

ROLL-BACK
```

```
SEND     cd-name FROM identifier-1 [queue-full-option]

SEND     cd-name [FROM identifier-1]
         ( WITH ESI                          )
         ( WITH ESI WITH identifier-2        )
         ( WITH identifier-3                 )
         ( WITH EBI                          )
         ( WITH EMI                          )
         ( WITH EGI                          )
         [ ON { COLUMN } mnemonic-name       ]
              { COL    }
         [ { BEFORE } ADVANCING { { identifier-4 } [ LINE  ] } ]
           { AFTER  }           { { integer      } [ LINES ] }
                                { { PAGE         }           }
              [queue-full-option]

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

SET index-name-1 [, index-name-2] ...
    { UP BY   } { identifier-1 }
    { DOWN BY } { integer-1    }

START file-name [ KEY IS ( EQUAL TO                    ) data-name-1 [, data-name-2] ... ]
                         (  =                          )
                         ( GREATHER THAN                )
                         (  >                          )
                         ( NOT LESS THAN                )
                         ( NOT <                        )
    [invalid-key-option]

STOP { RUN }
     { literal }

STRING { identifier-1 } [ , identifier-2 ] ... DELIMITED BY { identifier-3 }
       { literal-1    } [ , literal-2    ]                  { literal-3    }
                                                            { SIZE         }

[ , { identifier-4 } [ , identifier-5 ] ... DELIMITED BY { identifier-6 } ] ...
    { literal-4    } [ , literal-5    ]                  { literal-6    }
                                                         { SIZE         }

    INTO identifier-7 [WITH POINTER identifier-8]
    [overflow-option]
```

```
SUBTRACT { identifier-1 } [ identifier-2 ] ...
         { literal-1   }   [ literal-2   ]

    FROM identifier-m [rounded-option]
      [, identifier-n [rounded-option]] ...
    [size-error-option]

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

    FROM { identifier-m } GIVING identifier-n [rounded-option]
         { literal-m   }

      [, identifier-0 [rounded-option]] ...
    [size-error-option]

UNSTRING identifier-1

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

USE AFTER STANDARD { ERROR     } PROCEDURE ON
                   { EXCEPTION }

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

WRITE record-name [FROM identifier-1]
      [ { BEFORE } ADVANCING { { integer    } [ LINE  ] } ]
        { AFTER  }           { { identifier-2 } [ LINES ] }
                             { { PAGE         }           }
        [ end-of-page-option ]
        [ invalid-key-option ]
```

Options

At-end-option

```
AT END    imperative statement
```

End-of-page-option

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

Invalid-key-option

INVALID KEY imperative statement

Rounded-option

ROUNDED

Size-error-option

ON SIZE ERROR imperative statement

Queue-full-option

QUEUE FULL imperative statement

No-data-option

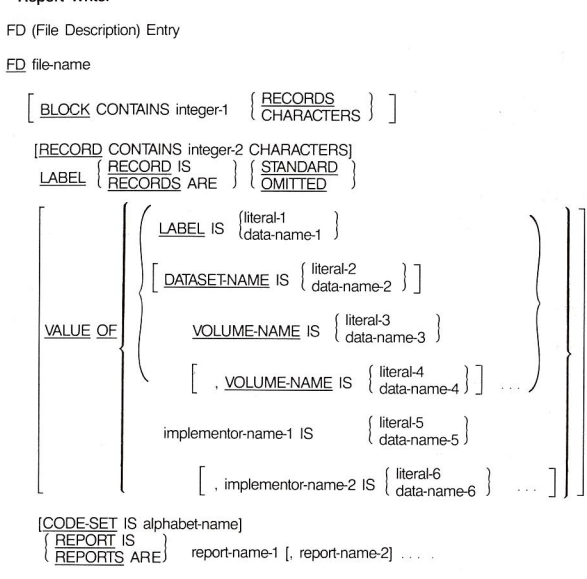
NO DATA imperative statement

Overflow-option

ON OVERFLOW imperative statement

Report Writer

FD (File Description) Entry



RD (Report Description) Entry

```
RD  report-name
    [CODE literal]
  [ { CONTROL IS  } { data-name-1 [, data-name-2] ...                  } ]
  [ { CONTROLS ARE} { FINAL [data-name-1 [, data-name-2] ...]          } ]

  [ PAGE [ LIMIT IS  ] integer-1 [ LINE  ]
         [ LIMITS ARE]           [ LINES ]

           [HEADING integer-2]
           [FIRST DETAIL integer-3]
           [LAST DETAIL integer-4]
           [FOOTING integer-5]                                         ] .
```

Report Group Description Entry

Report Group Entry

```
01      [data-name-1]
  [ LINE NUMBER IS { integer-1 [ON NEXT PAGE] } ]
                   { PLUS integer-2           }
  [ NEXT GROUP IS { integer-3    }              ]
                  { PLUS integer-4 }
                  { NEXT PAGE      }
                  { REPORT HEADING }
                  { RH             }
                  { PAGE HEADING   }
                  { PH             }
                  { CONTROL HEADING } { FINAL      }
                  { CH              } { data-name-2 }
    TYPE IS       { DETAIL         }
                  { DE             }
                  { CONTROL FOOTING } { FINAL      }
                  { CF              } { data-name-3 }
                  { PAGE FOOTING   }
                  { PF             }
                  { REPORT FOOTING }
                  { RF             }
  [[USAGE IS]     DISPLAY
```

Intermediate Entry

level-number [data-name]

```
[ LINE NUMBER IS { integer-1 [ON NEXT PAGE] } ]
                 { PLUS integer-2           }
  [[USAGE IS] DISPLAY] .
```

Elementary Entry

```
level-number [data-name-1]
( SOURCE IS identifier-1                                              )
( [SUM identifier-2 [, identifier-3] ...                              )
(     [UPON data-name-2 [, data-name-3] ...]] ...                     )
(   [ RESET ON { data-name-4 } ]                                      )
(              { FINAL       }                                        )
(   VALUE IS literal                                                  )
  [BLANK WHEN ZERO]
[ { COLUMN } NUMBER IS integer ]
[ { COL    }
    [GROUP INDICATIE]
[ { JUSTIFIED } RIGHT ]
[ { JUST      }
[ LINE NUMBER IS { integer-1 [ON NEXT PAGE ] } ]
                 { PLUS integer-2             }
  { PICTURE } IS character string
  { PIC     }
[[USAGE IS] DISPLAY]
```

Procedure Division

```
USE BEFORE REPORTING identifier.
```

Statements

```
GENERATE { data-name   }
         { report-name }

INITIATE report-name-1 [, report-name-2] ...

LIST FROM file-name-1 TO file-name-2 [USING literal]

SUPPRESS PRINTING

TERMINATE report-name-1 [, report-name-2] ...
```

Miscellaneous

Procedure-name identifier

```
( section-name                                                  )
( paragraph-name [ { IN } section-name ] )
                 { OF }
```

Identifier

```
data-name [qualification] [subscripting] [reference modification]
```

Condition-name identifier

condition-name [qualification] [subscripting]

Qualification

```
{ OF }   { data-name   }
{ IN }   { fd-name     }
         { report-name }
```

Subscripting

(subscript-1 [, subscript-2] . . .)

Subscript

```
{ integer-1                                          }
{ { data-name } [ { + } integer-2 ]                  }
{ { index-name }  { - }                              }
```

Reference modification

```
( { data-namre-1 }  :  [ data-name-2 ]  )
  { integer-1    }     [ integer-2   ]
```

Condition

1. Simple condition:

   class condition
   condition-name-condition
   relation condition
   sign condition

2. Complex condition

   negated simple condition
   combined condition
   negated combined condition

3. Abbreviated combined relation condition

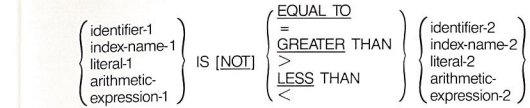
Class condition

identifier IS [NOT] { ALPHABETIC }
                    { NUMERIC   }

Condition-name condition

condition-name identifier

Relation condition



Sign condition

Negated simple condition

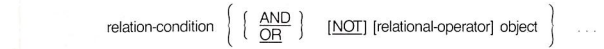
NOT simple-condition

Combined condition

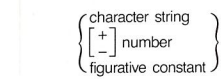
Negated combined condition

NOT (combined-condition)

Abbreviated combined relation condition



Literal



Figurative constant

HIGH-VALUE(S)
LOW-VALUE(S)
QUOTE(S)
SPACE(S)
ZERO(E)(S)
ALL literal
ALL symbolic-character

Layout of COBOL Record

Columns 1-6

sequence number of line

Column 7

* comment
/ skip to new page
– line continuation
D debugging line

Columns 8-11

Area A

coloumns 12-72

Area B

Columns 73-80

pogram identification (comment)

NOTE: The column numbers conform to the COBOL coding sheet.

All ANSCOBOL reserved words are included in the compiler, while the words marked with | are Philips extensions.

ACCEPT
ACCESS
ADD
ADVANCING
AFTER
ALL
| ALPHABET
ALPHABETIC
ALSO
ALTER
ALTERNATE
AND
| APPLY
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUTHOR

BEFORE
BLANK
BLOCK
BOTTON
BY

CALL
CANCEL
CD
CF
CH
CHARACTER
CHARACTERS
CLOCK-UNITS
CLOSE
COBOL
CODE
CODE-SET
| COL
COLLATING
COLUMN
COMMA
| COMMIT
COMMUNICATION
COMP
COMP-3
COMP-4
| COMPRESSED
COMPUTATIONAL
| COMPUTATIONAL-3
| COMPUTATIONAL-4
COMPUTE
CONFIGURATION
CONTAINS
CONTROL
CONTROLS
COPY
| CORE-INDEX
CORR
CORRESPONDING
COUNT
CURRENCY

DATA
DATE
DATE-COMPILED
DATE-WRITTEN
DAY
DE
DEBUG-CONTENTS
DEBUG-ITEM
DEBUG-LINE
DEBUG-NAME
DEBUG-SUBI
DEBUG-DUB-2
DEBUG-SUB-3
DEUGGING
DECIMAL-POINT
DECLARATIVES
| DECOMPRESSED
DELETE
DELIMITED
DELIMITER
DEPENDING
DESCENDING
DESTINATION
DETAIL
DISABLE
DISPLAY
DIVIDE
DIVISION
DOWN
DUPLICATES
DYNAMIC

| EBI
EGI
ELSE
EMI
ENABLE
END
END-OF-PAGE
ENTER
ENVIRONMENT
EOP
EQUAL
ERROR
ESI
EVERY
EXCEPTION
EXIT
EXTEND
| EXTERNAL

FD
FILE
FILE-CONTROL
FILLER
FINAL
FIRST
FOOTING
FOR
FROM
| FULL

GENERATE
GIVING
GO
GREATER
GROUP

HEADING
| HEX
HIGH-VALUE
HIGH-VALUES

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

JUST
JUSTIFIED

KEY
KEYS
KIND
LABEL
LAST
LEADING
LEFT
LENGTH
LESS
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
LINKAGE
LIST
LOCK
LOW-VALUE
LOW-VALUES

MASTER-INDEX
MEMORY
MERGE
MESSAGE
MODE
MODULES
MOVE
MULTIPLE
MULTIPLY

NATIVE
NEGATIVE
NEXT
NO
NOT
NUMBER
NUMERIC

OBJECT-COMPUTER
OCCURS
OF
OFF
OMITTED
ON
OPEN
OPTIONAL
OR
ORGANIZATION
OUTPUT
OVERFLOW

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

QUEUE
QUOTE
QUOTES

RANDOM
RD
READ
RECEIVE
RECORD
RECORDS
REDEFINES
REEL
REFERENCES
RELATIVE
RELEASE
REMAINDER
REMOVAL
RENAMES
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RESERVE
RESET
RETURN
RESERVED
REWIND
REWRITE
RF
RH
RIGHT
ROLL-BACK
ROUNDED
RUN

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

TABLE
TALLYING
TAPE
TERMINAL
TERMINATE
TEXT
THAN
THEN
THROUGH
THRU
TIME
TIMES
TO
TOP
TRAILING
TYPE

UNIT
UNSTRING
UNTIL
UP
UPON
USAGE
USE
USING

VALUE
VALUES
VARYING

WHEN
WITH
WORDS

WORKING-STORAGE
WRITE

ZERO
ZEROES
ZEROS

+
−
*
/
**
<
>
=

FILE STATUS codes

Value	Meaning
00	Successful completion.
10	The READ statement was unsuccessfully executed as a result of t to read a record when no next record exists in the dataset.
21	Sequence error for an indexed file with access mode is sequentia
22	Duplicate key; tried to add a duplicate key to an alternate index fil while the DUPLICATE clause was not specified for that alternate k For primary indices duplicate keys are never allowed.
23	No record found; tried to get access to a record identified by a key and that record does not exist in the file.
24	Boundary violation; tried to get access to a file beyond its limits (o for indexed and relative files).
30	Permanent error unsuccessful execution of an I/O statement becau hardware error.
34	Boundary violation; tried to get access to a sequential file beyond limits.
90	Program error; for return values see Operator Reference card.
91	DM : index of an indexed file is full; reorganize index. DBF: deadlock situation detected.
92	DM : Requested block currently under exclusive access by anoth program. DBF: Transaction rolled back on user's initiative via KICK commar workstation.
93	DM : Program error; requested block already under exclusive acc for this program via another FD.
94	Member could not be found on the specified library.
95	Specified file does not conform to the standards of SRCIN, or inc specification for a multi-volume file.
96	Specified device type must not be used for the specified functiona device.

STATUS KEY codes

Value	RECEIVE	SEND	ACCEPT	ENABLE	DISABLE	Meaning
00	X	X	X	X	X	No errors detected.
10	X	X	X	X	X	Destination disabled or control block disturbed.
20	X	X	X	X	X	Queue, subqueue, source or destination unknown.
50	X	X				Wrong length
90	X	X	X	X	X	Permanent I/O error on source or destination, or SEND (COBOL) SEGMENT while not allowed for this workstation.
91	X	X				Statement sequence or message sequence incorrect.
92	X	X				Temporary disconnected.
93		X				Station interrupted, request for no more messages, SEND accepted.
94				X		Device already attached.
95				X		Device not available, only given if DYN=NO.
96				X		Device not in configuration.
97	X	X		X	X	Device is down.
98				X		Usage type not allowed.
99	X	X				RECEIVE or SEND (Cobol) SEGMENT is done while previous sent or received segments are still in core.
9A	X	X				Destination or source of Cobol segment differs from the destination or source of the segment(s) in the segmentation buffer.
9B		X				A SEND is done without the 'FROM identifier' phrase and the end indicator value is not 0, 1, 2, 3, 4, 7, 8, 9, A, B, C, D, E or F.

Value	RECEIVE	SEND	ACCEPT	ENABLE	DISABLE	Meaning
9C	X	X				No transfer wanted (ENABLE) or end-o condition (RECEIVE end-of-interjob-communication (RECEIVE).
9D		X				Request for receive terminal, SEND not accepted.
9E	X	X		X		Wrong identification received (ENABLE).
9F	X	X				See value 91.
9G	X	X				No line activity.
9I	X	X				Block aborted.
9K	X					Unexpected type of
9L				X		Parameter value of CALM SETP is inc
9M	X	X		X		A physical connecti not possible.
9N	X					Request to switch t voice mode.
9P		X				Input available, SE accepted.
FF	X	X	X	X	X	Unknown logical st received from the monitor.

) or

by

data.

rrect.

on is

D

tus

P4000 Series

COBOL reference booklet

C2A user library

Data Systems

PHILIPS