COMPOSE

This statement can only be used if Con-nect/Con-form is installed.

```
COMPOSE
  [RESETTING-clause]
  [MOVING-clause]
  [ASSIGNING-clause]
  [FORMATTING-clause]
  [EXTRACTING-clause]
```

If you specify more than one clause, they will be processed in the order shown above.

This chapter covers the following topics:

- Function
- Syntax Description
- Formatting Process
- Dialog Mode
- Non-Natural Programs
- Examples

For an explanation of the symbols used in the syntax diagrams, see *Syntax Symbols*.

Function

The `COMPOSE` statement may be used to initiate text formatting by Con-form (the text formatter within Con-nect) directly from a Natural program.

The text to be formatted can either be supplied using variables or it may be retrieved from a Con-nect text block (a document containing Con-form formatting commands).

The contents of Natural variables can be passed to Con-form as variables for dynamic inclusion in the formatted text.

The values contained in a Con-form variable can also be returned to the Natural program from the text formatter.

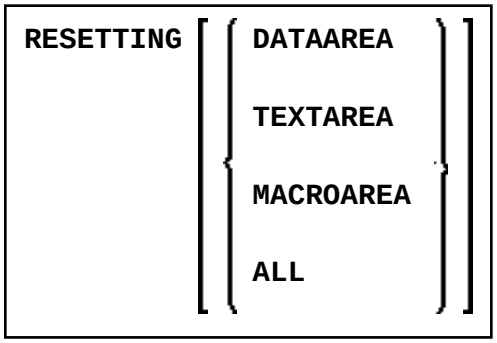
When the Con-form instructions are completed (resulting in a formatted document), the output is passed to one of the following places:

- a Natural report,
- a document in the Con-nect system file,
- variables in the Natural program that execute the `COMPOSE` statement,
- a non-Natural program.

Syntax Description

RESETTING-clause	This clause is used to delete information from the text format buffer area and to release memory from the `COMPOSE` buffer allocated by the `CSIZE` profile parameter in the Natural parameter module. See *RESETTING Clause*.
MOVING-clause	This clause is used to move text lines to the text formatter buffer area, or directly to the formatter, and to retrieve formatted text output from the work space of the formatter. See *MOVING Clause*.
ASSIGNING-clause	This clause is used to assign the values of Natural variables to text variables. See *ASSIGNING Clause*.
FORMATTING-clause	This clause is used to create text in final formatted form, that is, with correct line and page breaks, using input which can be a combination of text and Con-form statements. See *FORMATTING Clause*.
EXTRACTING-clause	This clause is used to assign the values of text variables to Natural variables. See *EXTRACTING Clause*.

RESETTING Clause



This clause may be used to delete the following from the text format buffer area:

- `DATAAREA` deletes all active text variables.
- `TEXTAREA` deletes all text input data.
- `MACROAREA` deletes all text macros.
- `ALL` deletes all of the above.

Note:
For compatibility reasons, the keyword TEXTAREA refers to the formatter's "Data Area" as used in the MOVING clause.

MOVING Clause

This clause may be used to move one or more text values to the text format buffer area (Syntax 1). This area may be used as a source of input for formatting operations. If the text formatter is currently waiting for input (see *Dialog Mode*), the text will be passed directly to it without being stored in Con-form's text area (Syntax 1 and Syntax 2). The source input is terminated with the LAST option. If the formatted text is currently waiting for output (see *Dialog Mode*), Syntax 3 of the MOVING clause is used to pass control back from the Natural program to the formatter. For description of the status variables, see the FORMATTING clause.

Depending on the status of the dialog mode, one of the following forms of the MOVING clause may be used:

Syntax 1

Syntax 1 of the MOVING clause is applicable when formatting has not begun or the formatter is in dialog mode for input and is waiting for input (TERM in the first status variable).

```
MOVING [operand1] ... 37  [TO DATAAREA] [LAST] [STATUS [TO] operand2 [operand3
[operand4 [operand5]]]]
```

Syntax 2

Syntax 2 of the MOVING clause is applicable when the formatter is in dialog mode for both input and output, and is waiting for further input (TERM in the first status variable). The formatter will not accept more than one line of input in this mode.

The execution context may change between succession of executed COMPOSE statements. Therefore it is necessary to re-specify the output variables even when the formatter is waiting for input.

```
MOVING { [operand1] [TO DATAAREA] } [OUTPUT] TO VARIABLES operand6 ... 20
       { LAST                     }
       [STATUS [TO] operand2 [operand3 [operand4 [operand5]]]]
```

Syntax 3

Syntax 3 of the MOVING clause is applicable when the formatter is in dialog mode for output (and possibly for input at the same time), and is passing output to the Natural program (STRG in the first status variable).

```
MOVING OUTPUT [TO VARIABLES] operand6 ... 20 [STATUS [TO] operand2 [operand3
[operand4 [operand5]]]]
```

Operand Definition Table:

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
operand1	C	S	A			A	N	P									yes	no
operand2		S				A											yes	yes
operand3		S								B							yes	yes
operand4		S								B							yes	yes
operand5		S								B							yes	yes
operand6		S	A			A											yes	no

Operand2 must be defined with format/length A4. *Operand3*, *operand4*, and *operand5* must be defined with format/length B4.

ASSIGNING Clause

This clause is used to assign values to Con-form text variables. These text variables may subsequently be referred to in formatting operations.

The text variable name(s) should be specified in upper case.

ASSIGNING [**TEXTVARIABLE**] {*operand1*=*operand2*}, **...** 19

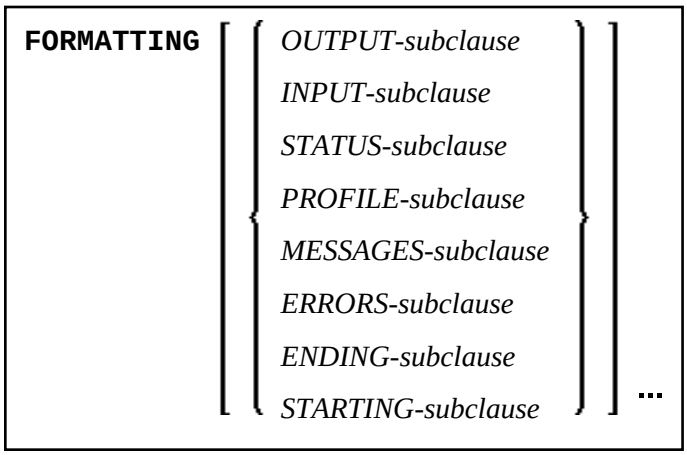
Operand Definition Table:

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
operand1	C	S				A											yes	no
operand2	C	S				A	N	P									yes	yes

FORMATTING Clause

This clause causes Con-form to produce formatted output.

The formatting options are specified in one or more subclauses. If subclauses are omitted, Con-form will apply default formatting options. The status variable is used in dialog mode.



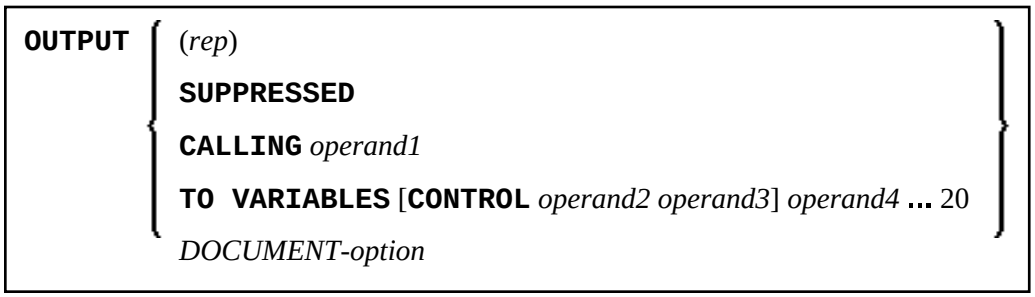
Syntax Element Description:

OUTPUT-subclause	The output medium. This can be a Natural report, a Con-nect cabinet, one or more Natural variables (or an array of Natural variables), or a non-Natural program. See *Output Subclause*.	
INPUT-subclause	The input medium. This can be a Con-nect document, the `COMPOSE` data area (see the `MOVING` clause), the environment of the Natural program(s) executing the `COMPOSE` statement(s) (see the `MOVING` clause), a non-Natural program, or a mixture of these four possibilities.	
STATUS-subclause	The status of the formatting operation. The formatting operation may involve multiple executions of a `COMPOSE` statement (in *Dialog Mode*). For example, the input is fed into the formatter's work space by a Natural program, and the output is passed from the formatter's work space into the environment of a Natural program (that is, one or more Natural variables). Therefore it is necessary to inform the Natural program of the formatting status. The following variables are passed to the Natural program during the formatting process:	
	State	`TERM` when the dialog mode is ready for input.
		`STRG` when the dialog mode is ready for output.
		`END` if the formatting process was completed successfully.
		`ENDX` if the formatting process was completed unsuccessfully.
	Position	Page and line number of the document that is being formatted. The page and line numbers are kept separately in two variables (page position and line position).
	Amount of Output Data	The number of lines of formatted output which are being passed to the Natural program. The formatter uses this number as the pointer to the next output variable to be filled. The value is incremented by 1 before the output line is issued. If the current value is out of range, the value is set to 1.
PROFILE-subclause	Text block to be processed before input is processed.	
MESSAGES-subclause	Control the output of warning messages and statistical information and error processing.	
ERRORS-subclauses		
ENDING-subclause	See *ENDING Subclause*.	
STARTING-subclause	See *STARTING Subclause*.	

OUTPUT Subclause

This subclause enables you to direct Con-form's formatted text output to a specific destination.

If this subclause is omitted, Natural's main printer will be used as the default output device.



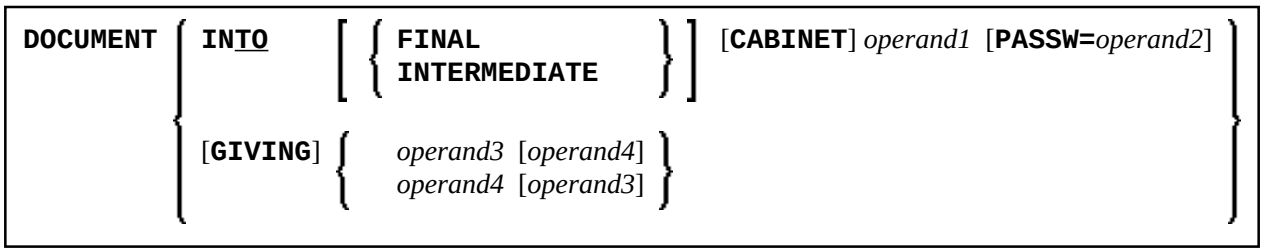
Operand Definition Table:

Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
operand1	C	S				A										yes	no
operand2	C	S				A										yes	no
operand3	C	S				A										yes	no
operand4		S	A			A										yes	no

Syntax Element Description:

OUTPUT (*rep*)	If the output is directed to a printer (that is, the report number is not 0 and a Con-nect printer profile has been loaded (by the Con-nect API function Z-DRIVER), the settings of that profile will be used to control the text highlighting options of the formatted output text. If a printer profile is active and the logical form feed controls were not specified, page ejects will be inserted by use of the appropriate internal Natural nucleus functions. Any other highlighting text option which is not reflected in the currently active Con-nect printer profile will be ignored. **Note:** Executions of the `COMPOSE RESETTING ALL` or `COMPOSE FORMATTING` statement with non-report output destination will unload a printer profile from the formatter's workspace. If output is directed to Report 0 or if a printer profile is not active, Con-nect will pass the responsibility of the output handling to the Natural nucleus routines. In this case, only the highlighting text options boldface, underline and italics will be recognized. **Note:** A report which is referred to in a `DEFINE PRINTER (`*n*`) OUTPUT 'CONNECT'` statement must not be specified as output destination in a `COMPOSE FORMATTING` statement.
OUTPUT SUPPRESSED	This option causes the output to be `SUPPRESSED`.
OUTPUT CALLING	See the section *Non-Natural Programs*.
OUTPUT TO VARIABLES	Generally, the formatted text will be passed in final format to an array of Natural variables. Each line fills one variable (if necessary, the line may be truncated to fit into the variables). Text highlighting options will be ignored, with the exception of the `CONTROL` variables specified, which will be used to emphasize sections of the text (that is, boldface or underscore). If the `CONTROL` variables `I` and `N` are specified, the formatted text will be produced in an intermediate format (that is, with interspersed logical control sequences). *Operand2* and *operand3* must be of format/length A1. For further information, see the section *Dialog Mode*.
OUTPUT DOCUMENT	See *DOCUMENT Option* below.

DOCUMENT Option



Operand Definition Table:

Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
operand1	C	S				A										yes	no
operand2		S				A										yes	no
operand3		S								B						yes	yes
operand4		S								B						yes	yes

OUTPUT DOCUMENT

Operand3 (format/length B10) is used by the formatter to pass a unique key from the document back to the Natural program. It is supported for compatibility reasons only.

Operand4 (format/length B4) is used by the formatter to pass an ISN which points to the formatted output document back to the Natural program. This ISN can be useful when referencing the document in successive calls to Con-nect APIs.

If *operand1* (which may be up to 8 characters long) is not specified, the document will be added to the current user's cabinet (that is, to the cabinet whose ID is identical to the currently active Natural user ID).

A password (up to 8 characters) must be specified if storing the document in a cabinet to which the currently assumed user ID has no access.

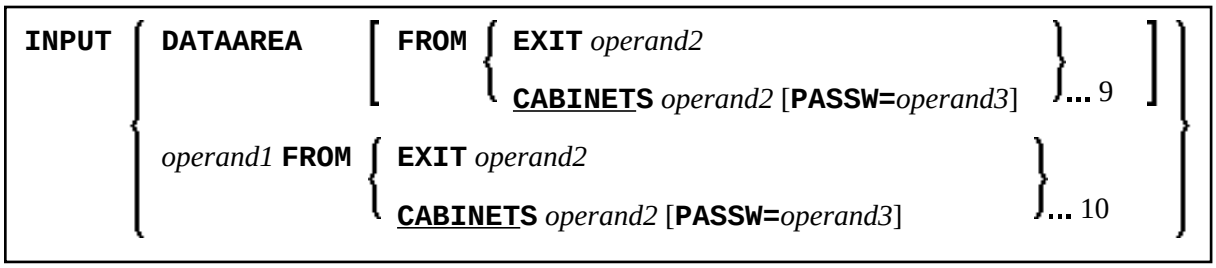
Con-form enforces adherence to Con-nect access restrictions and only accepts cabinet IDs which have been defined to Con-nect.

Note:
Cabinet IDs must be specified in upper case.

The document will be added to the folder COMPOSE without a document name. The subject line will be filled with the name of the program executing the `COMPOSE FORMATTING` statement along with the date and time of execution.

If the keyword `INTERMEDIATE` has been omitted, the document will be created in final form text. In this case, specific text highlighting options such as boldface or italics will be ignored.

INPUT Subclause



Operand Definition Table:

Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
operand1	C	S				A										yes	no
operand2	C	S				A										yes	no
operand3		S				A										yes	no

This subclause may be used to specify the sources which will supply input for the text formatter. The input may be taken from Con-form's data area (a mixture of text from the data area and from the dialog mode is also possible) which must be filled by one or more `MOVING` operations, or from a text block (specified by *operand1*). The text block may be contained in a Con-nect cabinet, or it may be supplied by a non-Natural program. It will be invoked using the same conventions which apply to the `CALL` statement. A hierarchy of Con-nect cabinets or non-Natural programs may be specified, each of which will be scanned in turn for the text block specified in *operand1*.

A password must be specified if the document is stored in a cabinet to which the currently assumed user ID has no access.

Con-form enforces adherence to Con-nect access restrictions and only accepts cabinet IDs which have been defined to Con-nect.

If this subclause is omitted, the Con-form data area will be processed.

Note:
Cabinet and text block IDs must be specified in upper case.

STATUS Subclause

[**STATUS** *operand1* [*operand2* [*operand3* [*operand4*]]]]

Operand Definition Table:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
Operand1	S	A	yes	no
Operand2	S	B	yes	no
Operand3	S	B	yes	no
Operand4	S	B	yes	no

Syntax Element Description:

operand 1	Contains the Status variable "State".
operand 2	Contains the Status variable "Position (page number)".
operand 3	Contains the Status variable "Position (line number)".
operand 4	Contains the Status variable "Amount of Output Data".

PROFILE Subclause

```
PROFILE operand1
```

Operand Definition Table:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
operand1	C S	A	yes	no

This subclause causes the content of the specified text block to be processed prior to any input which has been specified with the *INPUT-subclause* (by default, a text block will not be processed as a profile).

MESSAGES Subclause

```
MESSAGES { [LISTED] [ON] (rep) }
         { SUPPRESSED          }
```

Warning messages and statistical information are to be displayed upon completion of formatting. SUPPRESSED indicates that no messages are to be displayed and errors are to be ignored.

ERRORS Subclause

```
ERRORS { [LISTED] [ON] (rep) }
       { INTERCEPTED         }
```

This subclause may be used to specify the actions to be performed when a formatting error occurs. The error may be simply ignored, it may be processed by Natural's standard error-processing routine, or it may be listed on a specified Natural report (*rep*).

Note:
Errors and messages are mutually exclusive. Some errors may cause the standard Natural error-process routine to be invoked, even if a different option was specified. Errors or messages must not be directed to a report which is directed to the Con-nect system by a `DEFINE PRINTER (n) OUTPUT 'CONNECT'` statement.

ENDING Subclause

ENDING { [**AT**] [**PAGE**] *operand1* | **AFTER** *operand1* [**PAGES**] }

Operand Definition Table:

Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
operand1	C	S					N	P								yes	no

This subclause causes output of formatted text to be suppressed following a page with a specified number, or alternatively, it limits the amount of formatted output to a specified number of pages.

STARTING Subclause

STARTING [**FROM**] [**PAGE**] *operand1*

Operand Definition Table:

Operand	Possible Structure					Possible Formats										Referencing Permitted	Dynamic Definition
operand1	C	S					N	P								yes	no

This subclause causes output of formatted text to be suppressed until the page with the specified number (*operand1*) is reached.

EXTRACTING Clause

EXTRACTING [**TEXTVARIABLE**] {*operand* =*operand2*},**...** 19

Operand Definition Table:

Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
operand1		S				A	N	P									yes	yes
operand2	C	S				A											yes	no

This clause may be used to assign the values of text variables to Natural variables. The current text variable settings may be the result of previous formatting operations.

The text variable name(s) must be specified in upper case.

Formatting Process

The formatting process begins when the FORMATTING clause of the COMPOSE statement is executed (even if text input via a MOVING clause is intended, but no such input has been provided yet). While the formatting process is active, the text input resulting from the execution of the COMPOSE MOVING statement is fed directly into the formatter's work space (and cannot be re-used in a later formatting process). If the formatting process is inactive, the text input is stored intermediately in the COMPOSE buffer in the "DATAAREA". Thus the input can be re-used for multiple formatting processes.

Since the formatter's buffer is not cleared at the end of the Natural program, the respective COMPOSE statements need not be executed within one Natural program; they can be issued in several successively invoked programs.

The execution of a RESETTING or FORMATTING clause, or a serious formatting error, causes the termination of an ongoing formatting pass.

End-of-input is specified by the LAST subclause of the MOVING clause.

When a Con-nect document is specified as the source of input, end-of-input is assumed when the end of that document is reached.

Note:
It is recommended to use the STATUS subclause of the FORMATTING or MOVING clause to make sure that the formatting process is always in the appropriate status for a given processing step.

Dialog Mode

Dialog Mode Processing is the set of interactions which are performed between a user program and the formatter while formatting input and producing output.

Dialog mode allows a user program to supply raw text as input to the formatter at any level of the input hierarchy. It also accepts formatted output directly in the current program environment.

The dialog is achieved by subdividing the formatting process into a series of steps, each of which is separately invoked by a COMPOSE statement.

- Dialog Mode for Input

- Dialog Mode for Output
- Dialog Mode for Input and Output
- Execution of COMPOSE Statements in Dialog Mode

Dialog Mode for Input

Dialog mode for input is entered if the source of the input text is DATAAREA, or if the formatting control statement .TE ON is encountered, and Con-form's data area does not contain any more text to be processed. Dialog mode for input is signalled by the word TERM in the first STATUS variable.

The user program should respond by supplying the required input by invoking the MOVING function in a subsequently-processed COMPOSE statement. The user program can terminate terminal input by specifying the LAST option of the MOVING clause, or .TE OFF if terminal input was invoked by .TE ON, as text through the MOVING function. The formatter will signal the end of the formatting process with END, or ENDX in the case of an error in the first status variable.

Dialog Mode for Output

Dialog mode for output is entered if the destination of the output is TO VARIABLES. The formatter passes control back to the Natural program environment as soon as the supplied Natural variables are filled or a page break is reached (whichever occurs first). Dialog mode for output is signalled with STRG in the first STATUS variable. The user program should respond by taking the formatted output just placed into the Natural variables and designate another set of Natural variables as the output destination in a subsequently processed COMPOSE MOVING statement. The end of the formatting process is indicated with END, or ENDX in the case of an error.

Note:
When dialog mode is used (see the INPUT and OUTPUT subclauses), the formatting operation is usually spread across several executions of a COMPOSE statement.

Dialog Mode for Input and Output

Dialog mode can be entered for combined input and output processing. Therefore, when the formatter requests for further input (indicated by TERM) or when the formatter provides output (indicated by STRG), the Natural program must take the appropriate action.

When dialog mode is entered for combined input and output processing, only one line of input is accepted by the formatter at a time. In the case of input mode only, multiple lines are accepted at one time.

Execution of COMPOSE Statements in Dialog Mode

While it has been pointed out that dialog mode is entered via a COMPOSE FORMATTING statement which encompasses a series of COMPOSE MOVING executions, please note the following:

- COMPOSE ASSIGNING and COMPOSE EXTRACTING statements are valid while dialog mode is active.
- COMPOSE RESETTING and FORMATTING will force the immediate termination of all formatting.

Non-Natural Programs

Depending on the parameters specified with the FORMATTING clause, input and output may be processed by non-Natural programs. Such programs are invoked by the same mechanism that is used within the CALL statement.

COMPOSE exchanges parameters with these programs using the standard linkage conventions (dynamic loading is not possible in a CICS environment).

Note:
Input/output processing by non-Natural programs is only possible on mainframe computers; on other platforms, the appropriate parts of the COMPOSE statement are ignored.

Depending on the status of the formatting process, two or three parameters are passed between the formatter and the non-Natural programs:

Parameter 1 (format/length A1)	Function code is passed from the formatter to non-Natural programs. Possible values:	
	I	Initiate (input, output).
	O	Open document (input).
	R	Read one line of document (input).
	W	Write one line of output (output).
	C	Close document (input).
	T	Terminate (input, output).
Parameter 2 (format/length B1)	Response code is passed from non-Natural programs to the formatter. Possible values:	
	X'00'	Function successfully completed.
	X'01'	In response to function O: document could not be found.
		In response to function R: end of document was reached.
	X'FF'	Function not completed.
Parameter 3 (format A1/256)	In the case of the functions O and W, these parameters are passed from the formatter to non-Natural programs. However, the parameters from the function R are passed from non-Natural programs to the formatter.	
	Bytes 1 - 2	Signify the length n of this parameter.
	Bytes 3 - 4	Empty.
	Bytes 5 - n	Function O: Document name.
		Function R: Line read by the non-Natural program.
		Function W: Line of output from the formatter.
	Output is preceded by `N` if a form feed is required, otherwise by `1`. Specific options for highlighting text such as boldface and italics are ignored if the output is passed to a non-Natural program.	

Examples

- Example 1
- Example 2
- Example 3

- Example 4
- Example 5

Example 1

```
COMPOSE RESETTING ALL
        FORMATTING INPUT 'TEXT' FROM CABINET 'TLIB'
        OUTPUT (1)
        MESSAGES LISTED ON (0)
```

The above COMPOSE statement results in a formatted output of the text block TEXT within the Con-nect cabinet TLIB which is produced on Report 1. Errors and statistical messages are displayed on Report 0 (the default printer).

Example 2

```
COMPOSE RESETTING ALL
COMPOSE MOVING '.FI ON' 'This is an example'
COMPOSE MOVING 'for use of Con-form from'
               'within Natural applications' LAST
COMPOSE FORMATTING
```

The above COMPOSE statements result in a formatted output of text on Report 0 (default printer).

Example 3

```
COMPOSE ASSIGNING 'VAR1' = 'Text1', 'VAR2' = 540
```

The above COMPOSE statement results in the assignment of values to Con-form text variables &VAR1 and &VAR2 in a Con-nect procedure.

Example 4

Text Block XYZ in XYLIB:

```
.FI ON
Dear Mr &name.,
.IL
I am pleased to invite you to a presentation of our new product &prod..
```

Natural Program:

```
...
INPUT #NAME (A32) #PROD (A32)
COMPOSE ASSIGNING 'NAME' = #NAME, 'PROD' = #PROD
        FORMATTING INPUT 'XYZ' FROM CABINET 'XYLIB'
        OUTPUT (1) MESSAGES SUPPRESSED
...
```

Input Map Produced by Program:

```
  #NAME Davenport
  #PROD Natural 4.2
```

Resulting Output:

```
  Dear Mr Davenport,

  I am pleased to invite you to a presentation of our new product Natural 4.2.
```

Example 5

This is an example of formatting in dialog mode with combined input/output handling. The example program initiates the line-oriented formatting mode of Con-form, passes some commands/variables to Con-form, and performs a subroutine which displays status information and formatted output lines on the screen.

```
DEFINE DATA LOCAL
01 #LINES_PER_PERFORM(P5)   /* counts repeat-loops per PERFORM CNF_OUT
01 #TRACE(A1) INIT<'N'>     /* if 'Y' displays additional trace-infos
01 #OUT_FORM(A1) INIT<'F'>  /* output-format
01 #START_PAGE (P3) INIT<1> /* beginning of display
01 #CNTR (P5)               /* Loop-Counter
01 #STATI                   /* Status-Information
 02 #STATUS   (A4)          /* can be STRG TERM END or ENDX
 02 #PAGE     (B4)          /* actual page-number
 02 #LINE     (B4)          /* actual line-number on page (not .tt/.bt)
 02 #NO_LINES (B4)          /* number of lines returned
 02 REDEFINE #NO_LINES
  03 #NO_LINES_I (I4)
01 #OUTPUT(A30/4)           /* output of formatted line
01 #INDEX (P3)              /* index as pointer to out line
END-DEFINE
*
SET KEY ALL
SET CONTROL 'M9'
INPUT
   008/008 'Demonstration of formatted output to Variable'(I)
   /   08X 'Enter page to start display    :' #START_PAGE(AD=MIL)
   /   08X 'Display additional trace-data ?:' #TRACE(AD=MIT)
   /   08X 'Please select the output-format:' #OUT_FORM(AD=MIT)
           '(F=Final without BP/US-marks'
   /   44X 'M=Final with BP/US-marks "<>"'
   /   09X '(only, if CMF-Zap 2056 applied =>) I=Intermediate)'
   /   50X 'PF3=Exit'(I)
*
IF *PF-KEY EQ 'PF3'
   SET CONTROL 'MB'
   STOP
END-IF
*
IF NOT (#OUT_FORM EQ 'F' OR EQ 'M' OR EQ 'I')
   REINPUT ' Please enter valid code!' MARK *#OUT_FORM ALARM
END-IF
*
WRITE TITLE LEFT
     'Stat * Page     * Line     * No.Lines >> Formatted Output'(I)
   / '-'(79)(I)
*
SET CONTROL 'MB'
COMPOSE RESETTING ALL /* clear all con-form buffers
RESET #NO_LINES
*
* start line-oriented formatting-mode here
* starting from 0
```

```
DECIDE ON FIRST VALUE OF #OUT_FORM
   VALUE 'F'
      COMPOSE FORMATTING
           OUTPUT TO VARIABLES #OUTPUT (1:4)      /* to Output
           STATUS #STATUS #PAGE #LINE #NO_LINES   /* get Status
   VALUE 'M'
      COMPOSE FORMATTING
           OUTPUT TO VARIABLES CONTROL '<' '>'
                               #OUTPUT (1:4)      /* to output
           STATUS #STATUS #PAGE #LINE #NO_LINES   /* get Status
   VALUE 'I'
      COMPOSE FORMATTING
           OUTPUT TO VARIABLES CONTROL 'I' 'N'
                               #OUTPUT (1:4)      /* to output
           STATUS #STATUS #PAGE #LINE #NO_LINES   /* get Status
   NONE
      STOP
END-DECIDE
*
RESET #NO_LINES
*
* put some commands to con-form to see something
*
COMPOSE MOVING
     '.pl 16;.hs 2;.tt 1Formatierung in Variable//;.tt 2//' /* Cmd
     OUTPUT TO VARIABLES #OUTPUT (1:4)           /* to Output
     STATUS #STATUS #PAGE #LINE #NO_LINES        /* get Status
*
COMPOSE MOVING
     '.fs 1;.bt Ende Seite #//;.fi on;.tb *=15' /* Commands
     OUTPUT TO VARIABLES #OUTPUT (1:4)           /* to Output
     STATUS #STATUS #PAGE #LINE #NO_LINES        /* get Status
*
*
* loop 40-times, send commands to con-form and display output
*
COMPOSE ASSIGNING 'Wert' = '1-20' /* Assign value to variable &Wert
*
FOR #CNTR 1 40                       /* Loop some time
   IF #STATUS NE 'TERM' /* no wait-for-input => error!!!!
      IF #STATUS EQ 'STRG'
         IGNORE
      ELSE
         WRITE 'Unexpected Status-code' #STATUS(AD=OI) 'found!'
             / 'Execution has stopped....'
         STOP
      END-IF
   END-IF
*
   IF #CNTR EQ 21
      COMPOSE ASSIGNING 'Wert' = '21-40' /* Assign a variable-value
   END-IF
   COMPOSE ASSIGNING 'CNTR' = #CNTR /* Again assignment
   COMPOSE MOVING
        '.BP;&Wert *Durchlauf &CNTR;.BR'      /* Commands
        OUTPUT TO VARIABLES #OUTPUT (1:4)     /* to Output
        STATUS #STATUS #PAGE #LINE #NO_LINES  /* get Status
   PERFORM CNF-OUT                            /* show result
END-FOR
COMPOSE MOVING
     LAST                                  /* End-Of-Processing
     OUTPUT TO VARIABLES #OUTPUT (1:4)     /* to Output
```

```
     STATUS #STATUS #PAGE #LINE #NO_LINES  /* get Status
*
IF #TRACE EQ 'Y'
   WRITE 'End of processing...'(I)
END-IF
*
* Subroutines
*
PERFORM CNF-OUT
*
* Subroutine to display any waiting output from con-form
*
DEFINE SUBROUTINE CNF-OUT
   RESET #LINES_PER_PERFORM
   REPEAT UNTIL #STATUS EQ 'TERM' /* TERM = input waiting
      PERFORM BREAK               /* do some break-processing
      AT BREAK OF #PAGE
         IF #PAGE GT #START_PAGE
            WRITE '-'(79)(I)
         END-IF
         IF #TRACE EQ 'Y'
            WRITE 'End of this page...'(I)
         END-IF
         NEWPAGE
      END-BREAK
      IF #PAGE GE #START_PAGE     /* show line of output
         IF #NO_LINES_I GT 0
            FOR #INDEX 1 #NO_LINES_I
               ADD 1 TO #LINES_PER_PERFORM /* count loops
               WRITE NOTIT NOHDR #STATUS '*' #PAGE '*' #LINE
                                         '*' #NO_LINES
                                         '>>' #OUTPUT (#INDEX)
            END-FOR
         END-IF
      END-IF
      IF #STATUS NE 'STRG'         /* if no wait on out
         ESCAPE BOTTOM
      END-IF
      RESET #NO_LINES
      COMPOSE MOVING
           OUTPUT TO VARIABLES #OUTPUT (1:4)    /* get Output
           STATUS #STATUS #PAGE #LINE #NO_LINES /* Status
   END-REPEAT
*
   IF #TRACE EQ 'Y'
    WRITE 'Count of Lines per PERFORM was'(I) #LINES_PER_PERFORM(AD=OI)
   END-IF
*
 END-SUBROUTINE
 SET CONTROL 'MB'
END
```