

# Natural in Batch Mode

This section contains special considerations that apply when running Natural in batch mode and covers the following topics:

- What is Batch Mode?
- Batch Mode Detection
- Batch Mode Restrictions
- Real Batch Mode
- Sample Session for Batch Mode
- Hints for Using Natural Maps and Dialogs in Batch Mode

For information on starting/terminating a Natural batch session, refer to Starting/Terminating Natural.

---

## What is Batch Mode?

Natural distinguishes two processing modes:

- Interactive Mode
- Batch Mode

The main difference between these two modes is that in interactive mode, the input of commands and data comes from a keyboard and the output is displayed on a screen. In batch mode, input is read from a file and output is written to a file.

This enables Natural to run as a background batch job, where no interaction between the computer and the person who submitted the batch job is necessary. The batch job consists of a set of programs such that each is completed before the next program is started. The programs are executed serially and receive sequential input data.

Batch mode is of interest for mass data processing and re-usable execution.

## Batch Mode Detection

The system variable \*DEVICE shows if Natural is running in batch or interactive mode.

in batch mode	*DEVICE is equal "BATCH"
in interactive mode	*DEVICE is not equal "BATCH" (in most of the cases *DEVICE is equal "VIDEO")

Example:

```

IF *DEVICE = "BATCH" THEN
  /* Batch Mode Process */
  INPUT USING MAP ...
ELSE
  /* Interactive Process */
  PROCESS GUI ...
END-IF

```

## Batch Mode Restrictions

When Natural is running in batch mode, some features are not available or are disabled:

- There is no mouse support;
- No different character fonts are available;
- Only data for an INPUT statement can be processed, dialog input is only conditionally supported (see section Hints for Using Natural Maps and Dialogs in Batch Mode);
- Output appearance is not GUI-like (it is character-oriented output);
- No colors and video attributes (such as blinking, underlined, reverse video) are written to the batch output file CMPRINT;
- Filler characters are not displayed within an INPUT statement;
- No interactive input or output is possible.

## Real Batch Mode

To run Natural in real batch mode you have to specify the dynamic parameter BATCHMODE. In addition, input and output channels have to be defined (as described below).

### Input and Output Channels

The following input and output channels are necessary in batch mode:

- CMSYNIN (Batch Input File for Natural Commands and INPUT Data)  
CMSYNIN is used for the input file that contains Natural commands, and (optionally) data to be read by INPUT statements during execution of Natural programs.
- CMOBJIN (Batch Input File for Natural INPUT Data)  
CMOBJIN is used for data intended to be read by Natural INPUT statements. These types of data can alternatively be placed in the CMSYNIN file immediately following the relevant RUN or EXECUTE command.
- CMPRINT (Batch Output File)  
CMPRINT is used for the output resulting from DISPLAY, PRINT and WRITE statements in a Natural program.
- CMPRTnn (Output File for Additional Reports)  
CMPRTnn is used for additional reports referenced by any Natural program executed during the session. "nn" is two digit decimal number in the range 01 to 31 corresponding to the report number used in a DISPLAY, PRINT or WRITE statement.
- CMWRKnn (Batch Output File for Natural Work Files)  
CMWRKnn is used for Natural work files referenced by any Natural program executed during the session. "nn" is a two digit decimal number in the range 01 to 32 corresponding to the number used in a READ WORK FILE or WRITE WORK FILE statement.
- NATLOG (Natural Log File)  
NATLOG is used to log messages that could not be written to the batch output file CMPRINT.  
It is recommended to enable NATLOG in batch mode.

### CMPRtnn Specifications in Batch Mode

In order to allow the user to specify variable print-file names, alpha-format system variables and numeric counter markers may be embedded in the filename specification for CMPRTnn. The supported alpha-format system variables are:

- \*APPLIC-ID
- \*APPLIC-NAME
- \*DEVICE

- \*ETID
- \*INIT-USER
- \*LIBRARY-ID
- \*NET-USER
- \*PID
- \*PROGRAM
- \*USER
- \*USER-NAME

If any of these strings (in upper-case only) are encountered within the printfile specification, they will be replaced at run-time with the contents of the appropriate system variable. Additionally, a counter marker (#) may be used. This will be replaced by a 2-digit counter which will automatically be incremented for each print file.

## Sample Session for Batch Mode

The following example demonstrate how to start Natural in batch mode. A simple Natural program is executed and two data items are taken from the batch input file. After the items are processed from the INPUT statement, a subsequent DISPLAY statement writes the data to the batch output file. Then, Natural terminates.

### Natural Commands - CMSYNIN=batch.cmd:

```
LOGON SYSEXBAT
EXECUTE RECCONT
FIN
```

### Natural Input Data - CMOBJIN=batch.inp:

```
Ben,%
Smith
```

### Natural Program RECCONT in Library SYSEXBAT:

```
DEFINE DATA LOCAL
  1 #firstname (A10)
  1 #lastname (A10)
END-DEFINE
INPUT (IP=OFF AD=M) #firstname #lastname
DISPLAY #firstname #lastname
END
```

### Natural Command Line:

```
NATURAL BATCHMODE CMSYNIN=batch.cmd CMOBJIN=batch.inp CMPRINT=batch.out BMSIM=MF NATLOG=ALL
```

When you invoke Natural using the command line described above, Natural produces the following output.

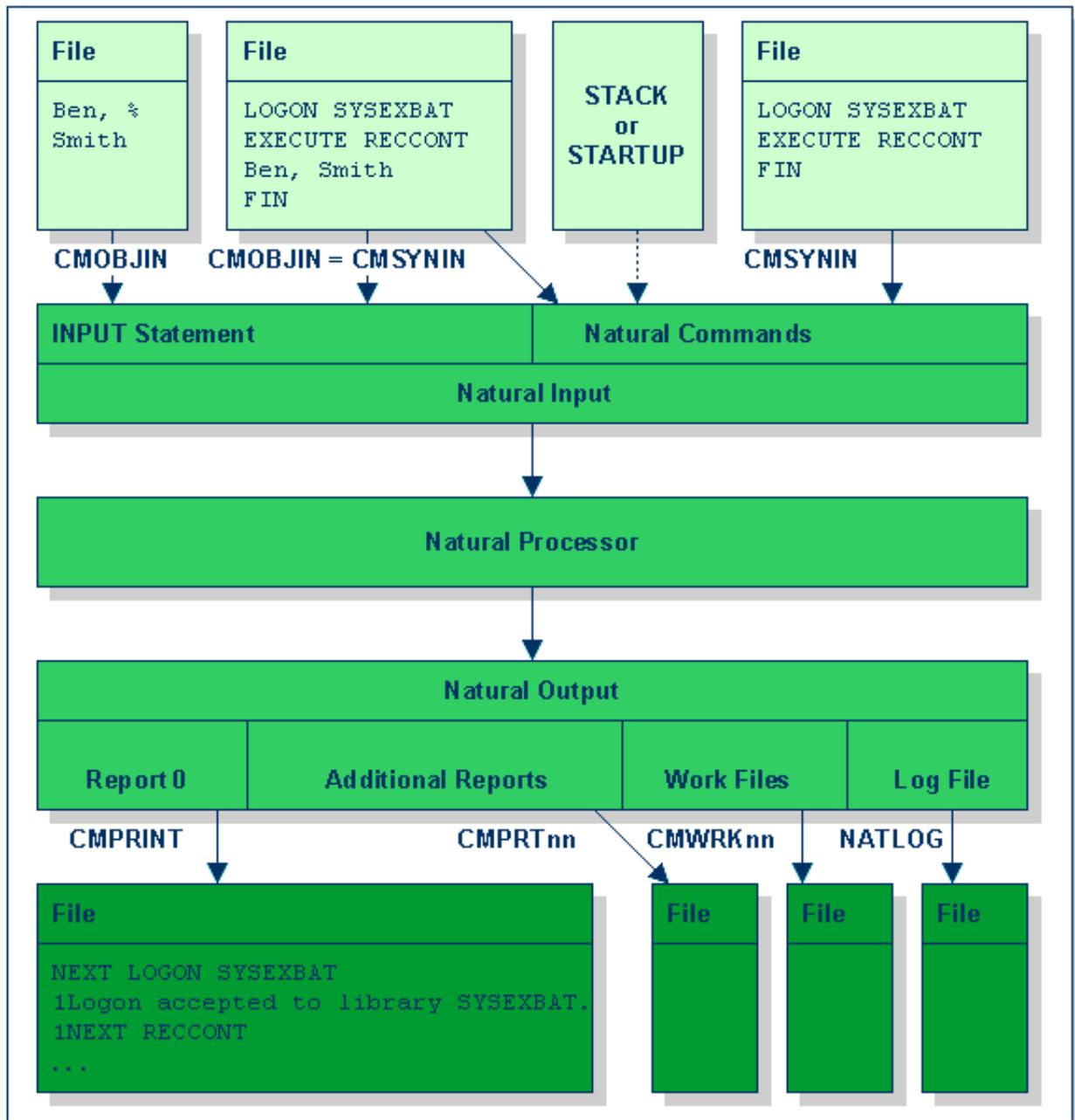
### Contents of Batch Output File - CMPRINT=batch.out - after Execution:

```
_NEXT LOGON SYSEXBAT
1Logon accepted to library SYSEXBAT.
1NEXT EXECUTE RECCONT
1#firstname          #lastname
_DATA Ben,%
_DATA Smith
1Page      1          99-06-18  10:17:11
0#FIRSTNAME #LASTNAME
```

```

-----
0Ben      Smith
1NEXT FIN
1NAT9995 NATURAL session terminated normally.
    
```

The following image illustrates how Natural reads input and writes output in batch mode.



## Hints for Using Natural Maps and Dialogs in Batch Mode

If an application is designed to run in batch mode as well as in interactive mode, the following considerations should be taken into account.

Within Natural, there are two ways to read input data:

- Using a Map (by using an INPUT statement or the Natural object Map)
- Using a Dialog (by using a Natural Dialog object)

In batch mode, data have to be processed using an INPUT statement, because a dialog does not allow data processing in batch mode. Terminal commands to navigate and control the data are also not supported by a dialog. Nevertheless, a dialog may be executed in batch mode. In this case, however, the dialog must be altered in the following way:

- The dialog attribute "visible state" must be turned off (state equals not visible).
- Within the event AFTER-OPEN, code should be inserted to read data during batch mode processing. If Natural runs in batch mode, an INPUT statement should be coded to get the input data. For interactive mode, the "visible state" has to be turned on to make the dialog visible.
- If there is a CLOSE event, ensure that the appropriate code does not contain any GUI actions in batch mode.

### Example for Event AFTER-OPEN:

```
IF *DEVICE EQ "BATCH" THEN
  /* Batch Mode Processing: Call a Map */
  INPUT USING MAP "BATCHINP" #p1 #p2 #p3

  /* ... further data processing ... */

  /* Close dialog immediately */
  CLOSE DIALOG *DIALOG-ID
ELSE
  /* Interactive Mode Processing: Make dialog visible */
  #DLG$WINDOW.VISIBLE = TRUE
END-IF
```

### Example for Event CLOSE:

```
IF *DEVICE NE "BATCH" THEN
  /* ... GUI actions ... */
END-IF
```