

# Natural in Batch Mode

This chapter contains special considerations that apply when running Natural in batch mode. It covers the following topics:

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

## What is Batch Mode?

Natural distinguishes between two processing modes:

- interactive mode (with Natural Studio)
- batch mode

The main difference between these two modes is that in interactive mode, the commands and data are input by the user by means of the 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 - without user interaction.

When Natural is run as a batch job, no interaction between Natural and the person who submitted the batch job is necessary. The batch job consists of programs that are executed sequentially and that receive sequential input data.

Batch mode is useful for mass data processing on a regular basis.

## Starting a Natural Session in Batch Mode

Batch mode is activated with the parameter `BATCHMODE`.

### To start a Natural session in batch mode

1. Start Natural with the dynamic parameter `BATCHMODE` as shown below:

```
natural BATCHMODE
```

The above call (where only the BATCHMODE parameter is specified) assumes that the required input and output channels have already been defined in the Configuration Utility. For information on the input and output channels, see *Using Natural in Batch Mode* later in this section). For information on the batch-mode-relevant profile parameters in the parameter file, see *Batch Mode* in the *Configuration Utility* documentation.

It is also possible to add the required input and output channels as dynamic parameters to the above call. This is illustrated in *Sample Session for Batch Mode* later in this section. Any input and output channels that are specified as dynamic parameters with the above call override the channel definitions in the parameter file.

2. Check the file which has been defined as the output channel. At its end, this file should contain the message that your session has terminated normally.

## Terminating a Natural Session in Batch Mode

A Natural session in batch mode is terminated when one of the following is encountered during the session:

- the system command FIN in the batch input file, or
- a TERMINATE statement in a Natural program which is being executed.

### Note:

When an end-of-input condition occurs in the batch input file, the batch session is also terminated. In this case, the file which has been defined as the output channel contains a message which indicates an unexpected end.

## Using Natural in Batch Mode

To start a Natural session in batch mode you have to specify the dynamic parameter BATCHMODE. In addition, input and output channels have to be defined as described below.

### Important:

The input channels CMSYNIN and/or CMOBJIN and the output channel CMPRINT are always required for batch mode.

The following topics are covered below:

- Input and Output Channels
- Code Pages for the Input and Output Files

## Input and Output Channels

The following parameters are available for batch mode:

| Parameter  | Description  |
|------------|--|
| CMSYNIN    | Defines the batch input file which contains the Natural commands and (optionally) data to be read by INPUT statements during execution of Natural programs.  |
| CMOBJIN    | Defines the batch input file which contains the data to be read by INPUT statements. This data can alternatively be placed in the file defined with the parameter CMSYNIN, immediately following the relevant RUN or EXECUTE command.                          |
| CMPRINT    | Defines the batch output file for the output resulting from DISPLAY, PRINT and WRITE statements in a Natural program.  |
| CMPT $nn$  | Defines an output file for additional reports referenced by any Natural program executed during the session. $nn$ is a two-digit decimal number in the range from 01 to 31 which corresponds to the report number used in a DISPLAY, PRINT or WRITE statement. |
| CMWRK $nn$ | Defines a work file referenced by any Natural program executed during the session. $nn$ is a two-digit decimal number in the range from 01 to 32 which corresponds to the number used in a READ WORK FILE or WRITE WORK FILE statement.                        |
| NATLOG     | Used to log messages that could not be written to the batch output file defined with the parameter CMPRINT. It is recommended to enable NATLOG in batch mode.  |

## Code Pages for the Input and Output Files

The following parameters are used to specify the code pages in which the input files are encoded and in which the output file shall be encoded.

| Parameter | Description   |
|-----------|---|
| CPSYNIN   | Specifies the code page in which the batch input file for commands is encoded. This file is defined with the parameter CMSYNIN. |
| CPOBJIN   | Specifies the code page in which the batch input file for data is encoded. This file is defined with the parameter CMOBJIN.     |
| CPPRINT   | Specifies the code page in which the batch output file shall be encoded. This file is defined with the parameter CMPRINT.       |

Encoding for CMSYNIN and CMOBJIN:

- If a code page is specified for one of the input files CMSYNIN or CMOBJIN, it is assumed that the data in the input file is encoded using this code page.
- If no code page is specified for one of the input files CMSYNIN or CMOBJIN, it is assumed that the data in the input file is encoded using the default code page specified in the Natural parameter CP.
- If no code page is specified in the Natural parameter CP, it is assumed that the data in the input file is encoded using the current system code page.

Encoding for CMPRINT:

- If a code page is specified for the output file CMPRINT, the output data will be encoded using this code page.

- If no code page is specified for the output file `CMPRINT`, the output data will be encoded using the default code page specified in the Natural parameter `CP`.
- If no code page is specified in the Natural parameter `CP`, the output data will be encoded using the current system code page.

If the encoding/decoding fails (for instance if a character is written to `CMPRINT` that is not contained in the code page used to encode the file), the batch job terminates with a startup error 42 (batch mode driver error) that specifies the file on which the encoding/decoding error occurred.

Note that it is possible in particular to specify UTF-8 as code page in each of these parameters. This allows for reading and writing Unicode data encoded in UTF-8.

## Sample Session for Batch Mode

This example demonstrates how to start Natural in batch mode. A simple Natural program is executed and data items are taken from the batch input file. After the items are processed with the `INPUT` statement, a `DISPLAY` statement follows, which writes the data to the batch output file. Then, Natural terminates.

This example uses the program `RECCONT` which is stored in the library `SYSEXBAT`.

### Note:

See the text `A-README` in the library `SYSEXBAT` for information on the objects that are stored in this library.

The sample session is invoked with the following call:

```
natural BATCHMODE CMSYNIN=cmd.txt CMOBJIN=data.txt CMPRINT=out.txt NATLOG=ALL
```

### Note:

This call assumes that all files can be found in the current directory and that the output is written to this directory. If the files are located in different directories or if the output is to be written to a different directory, you have to specify the path.

The parameters in the above call are described below:

### **BATCHMODE**

The parameter `BATCHMODE` enables batch mode and sets the value of the system variable `*DEVICE` to "BATCH".

### **CMSYNIN=cmd.txt**

The batch input file `cmd.txt` is a text file which is stored in your file system. The content of this file is shown below. It contains Natural system commands for logging on to the library `SYSEXBAT`, executing the Natural program `RECCONT`, and terminating the Natural session.

```
LOGON SYSEXBAT  
EXECUTE RECCONT  
FIN
```

The Natural program RECCONT has the following content:

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

#### **CMOBJIN=data.txt**

The INPUT statement in the program RECCONT uses the data which is defined in the batch input file *data.txt*. This is a text file which is stored in your file system. The content of this file is shown below.

```
Ben %
Smith
```

#### **Note:**

The character "%" indicates that the information continues in the next line.

#### **CMPRINT=out.txt**

The DISPLAY statement in the program RECCONT writes the data to the batch output file *out.txt* which is created in your file system. The content of this file is shown below:

```
NEXT LOGON SYSEXBAT
Logon accepted to library SYSEXBAT.
NEXT EXECUTE RECCONT
```

```
DATA Ben %
DATA Smith
Page      1                                25.04.05  13:39:09
```

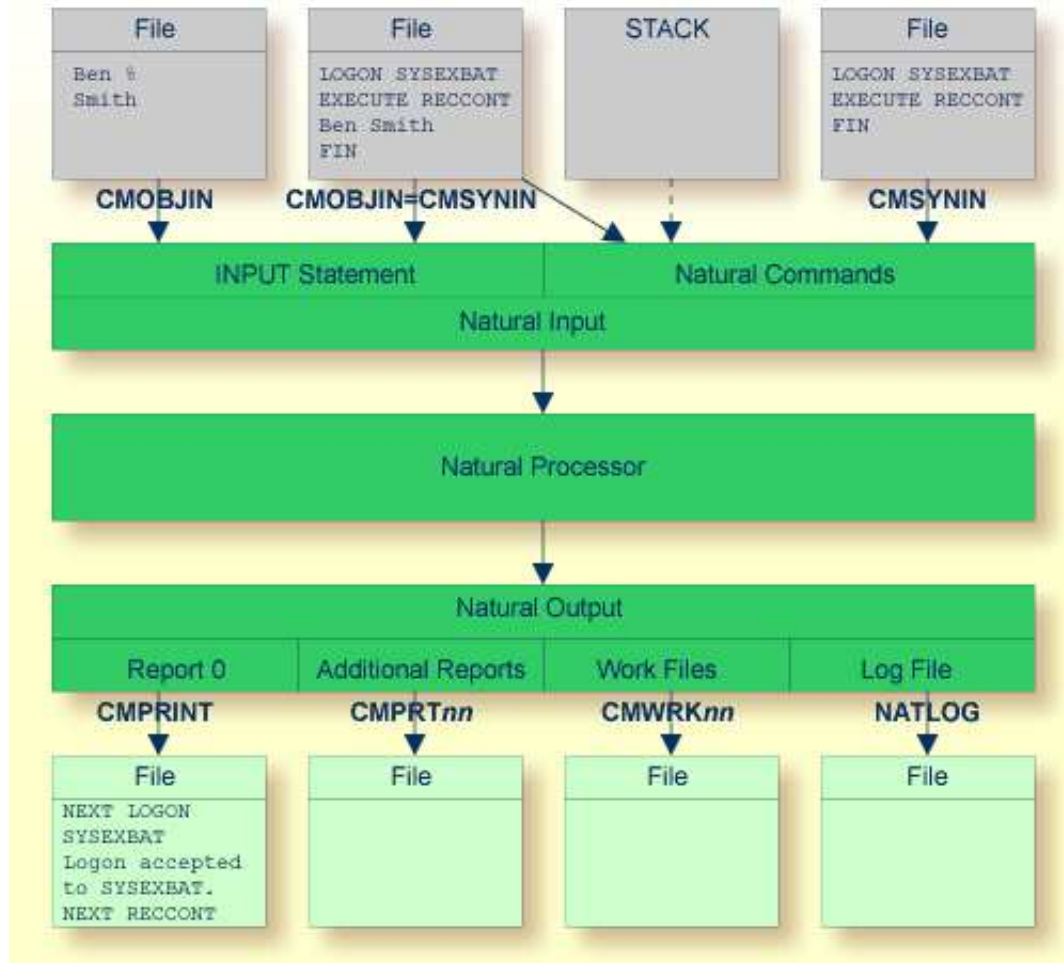
```
#FIRSTNAME #LASTNAME
-----
```

```
Ben      Smith
NEXT FIN
NAT9995 Natural session terminated normally.
```

#### **NATLOG=ALL**

When you invoke the sample session with the above call, a log file is created with contains all types of messages (which also includes the names of the batch input and outfile files). The log file is normally created in Natural's temporary directory which is defined in the local configuration file. See also the description of the NATLOG parameter.

The image below illustrates the different ways in which Natural reads input and writes output in batch mode.



As shown in the above graphic, you can proceed in one of the following ways:

- **CMOBJIN and CMSYNIN**

Different files are used for batch input. One file contains the Natural commands and the other file contains the data:

```
natural BATCHMODE CMSYNIN=cmd.txt CMOBJIN=data.txt CMPRINT=out.txt
```

- **CMSYNIN**

One file is used for batch input. It contains both the Natural commands and data:

```
natural BATCHMODE CMSYNIN=data.txt CMOBJIN=data.txt CMPRINT=out.txt
```

**Note:**

Even though only one batch input file is used, both parameters CMSYNIN and CMOBJIN have to be specified. Both parameters must refer to the same file.

- **CMOBJIN and STACK**

One file is used for batch input. It contains the data. The Natural commands are specified with the profile parameter `STACK`:

```
natural BATCHMODE CMOBJIN=data.txt STACK="(LOGON SYSEXBAT; RECCONT;FIN)"
```

## Batch Mode Detection

The system variable `*DEVICE` indicates whether Natural is running in batch mode or interactive mode.

| Mode             | Description  |
|------------------|--|
| Batch mode       | <code>*DEVICE</code> contains the value "BATCH". This value is set by the parameter <code>BATCHMODE</code> . |
| Interactive mode | <code>*DEVICE</code> contains a value other than "BATCH". In most cases, it contains the value "VIDEO".      |

Example:

```
IF *DEVICE = "BATCH" THEN
  WRITE 'This is the background task'
ELSE
  WRITE 'This is the interactive session'
END-IF
```

## Batch Mode Restrictions

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

- Interactive input or output is not possible.
- 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 *Hints for Using Natural Maps and Dialogs in Batch Mode*).
- The output appearance is not GUI-like (it is character-oriented output).
- No colors and video attributes are written to the batch output file defined by `CMPRINT`.
- Filler characters are not displayed within an `INPUT` statement.
- Certain Natural system commands are not executable in batch mode, and are ignored. In the *System Commands* documentation, a corresponding note is provided for each system command to which this restriction applies.

## 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 the Natural object Dialog).

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 for navigating and controlling 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` must be set to "FALSE".
- 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 dialog attribute `VISIBLE` has to be set to "TRUE" to make the dialog visible.

Example for the `AFTER-OPEN` event:

```
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
```

- If there is a `CLOSE` event, ensure that the appropriate code does not contain any GUI actions in batch mode.

Example for the `CLOSE` event:

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