Natural

Terminal Commands

Version 9.2.3

February 2025

ADABAS & NATURAL

This document applies to Natural Version 9.2.3 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1979-2025 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Document ID: NATMF-NNATTCOM-923-20250213

Table of Contents

Preface vii
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 Introduction to Terminal Commands 5
Purpose of Terminal Commands 6
Changing the Terminal Command Control Character 6
Issuing Terminal Commands 6
Using Terminal Commands in Programs 7
Terms Used in the Terminal Command Descriptions 7
%? - Help for Terminal Commands 7
3 Terminal Commands Grouped by Function 9
Case Translation 10
Copy/Clear 10
Language, Messages, Error Processing 10
Screens, Terminal, and Window Processing 11
Colors, Outlining 11
INPUT Statement, Stack 11
Session Recording and Screen Capturing 12
Statistics and Trace 12
Miscellaneous 13
Key Assignments 13
4 Terminal Command Key Assignments 15
Assigning Terminal Commands to Function Keys 16
CLEAR Key - Interrupt Current Operation 16
CTRL+D Keys - Interrupt Current Operation 16
RESET+ENTER Keys - Attention Interrupt 16
5 % - Continuation Indicator for INPUT in Batch 17
6 %% and %. - Interrupt Current Operation 19
%% in Online Mode 20
%% in Batch Mode 20
%. in Online Mode 21
%. in Batch Mode 21
7 %* - Inhibit Character Display 23
8 %.P - Delete First Entry from Stack 25
9 %.S - Read Stack Entry without Deleting it 27
10 %/ - End-of-File 29
11 %+ and %- - Enable/Disable Use of Natural Connection 31
12 %<RDC+ and %<RDC- - Activate/Deactivate Trace Recording 33
13 %<TECH - Display Technical Information 35
14 %<TEST - Invoke the Debugger 37
15 %= - Assigning Colors to Fields 39

16 %A - Play Back a Recording 41
17 %B - Activate/Deactivate a Recording 43
18 %B= - Specify Library for a Recording 45
19 %C - Copying Contents of Page Buffer 47
20 %CS and %CC - Copying Data to Stack or *COM 49
21 %D - Activate Keyword/Delimiter Mode 51
22 %D= - Control of Outlining 53
23 %DUE - Dump for Specific Error 55
24 %E - Display Screen Captures 57
25 %E= - Activate/Deactivate Error Processing 59
26 %F - Activate Forms/Screen Mode 61
27 %F= - Frame Characters for Window 63
28 %FM - Numeric Edit Mask Free Mode 65
29 %G - Set Playback Mode for a Recording 67
30 %H - Hardcopy Output 69
31 %I - Capture Current Screen 71
32 %J - Invoke Helproutine 73
33 %K and %KP - Simulate PF- and PA-Key 75
34 %L - Disable Lower- to Upper-Case Translation 77
35 %L= - Set Language Code 79
36 %M - Control of Message Line 81
Message Line Positioning 82
Message Line Protection 83
Message Line Color 83
37 %MSGSF - Display System Error Messages in Full 85
38 %N - Activate Non-Conversational Mode 87
39 %O - Deactivate Screen Capturing 89
40 %P - Capture Screen Sequence 91
41 %P= - CALL Options 93
%P=C - Pass Parameter Values instead of Address 94
%P=CC - Pass Parameter Values in a CICS Container 96
%P=I - Pass PCB Addresses to Callee 96
%P=L - Call of LE Dynamic Main Program 97
%P=LS - Call of LE Subroutine Program 97
%P=S - Standard Linkage for Call 98
%P=SC - Standard Linkage Call Simulating EXEC CICS LINK 98
%P=SQ - Standard Linkage for Call of Quasi-Reentrant Subprograms 98
%P=U - Separate Unit of Work for Called Program 99
%P=UT - Separate Unit of Work for Called Program, Tolerate Transaction Backout 99
%P=V - Roll-Out for Call 99
42 %Q - Suppress Next Input 101
43 %QO - Suppress Pseudo-Conversational Output 103
44 %QS - Simultaneous Output of Multiple Screens 105
45 %R - Repeat INPUT Statement 107

46 %RM - Protection of Light Pen Sensitive Fields 109
47 %RN - Suppress Compression of Screen Data 111
48 %RO - Activate/Deactivate Screen Optimization 113
49 %S - Resume Screen Capturing 115
50 %T - Position Cursor to Top of Active Window 117
51 %Tll/cc - Position Cursor to Line ll, Column cc 119
52 %T* - Position Cursor Outside Window 121
53 %T= - Activate Converter Routine for Device Type 123
54 %TRE - Activate/Deactivate External Trace 125
55 %TRI - Activate/Deactivate Internal Trace 127
56 %U - Translate Lower to Upper Case 129
57 %V - Control of Print Mode 131
58 %W - Window Processing 133
Window Size and Position on the Physical Screen 134
Window Position on a Logical Page 136
%WA and %WZ - Save Screen Image before Window 139
59 %X - Control of Infoline 141
Infoline 143
Statistics Line 143
60 %Y - Control of PF-Key Lines 145
Display Format of Function-Key Lines 146
Single- and Double-Line Display 147
Positioning of Function-Key Lines 147
Range of Displayed Function Keys 147
Intensified or Reverse Video Display of Function-Key Lines 148
Coloring of Function-Key Lines 148
Cursor Sensitivity 148
61 %Z - Clear Source Area 151

Preface

This documentation describes the Natural terminal commands. It is organized under the following headings:

Introduction to Terminal Commands	What are terminal commands? How to issue terminal commands and how to use them in a program.
Terminal Commands Grouped by Function	Provides an overview of the terminal commands ordered by functional groups.
Terminal Command Key Assignments	How to assign a frequently used terminal command to a function key. Information on CLEAR key, CTRL+D keys and RESET+ENTER keys.
Terminal Commands in Alphabetical Order	Descriptions of the terminal commands in alphabetical order.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://containers.softwareag.com/products** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software GmbH products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction to Terminal Commands

- Purpose of Terminal Commands 6
- Changing the Terminal Command Control Character 6
- Issuing Terminal Commands 6
- Using Terminal Commands in Programs 7
- Terms Used in the Terminal Command Descriptions 7
- %? - Help for Terminal Commands 7

Purpose of Terminal Commands

A complete functional overview of Natural terminal commands is given in the section *Terminal Commands Grouped by Function*.

Changing the Terminal Command Control Character

You can define another special character as control character; this is done with the session parameter `CF`.

When the control character is changed, all terminal commands which have been assigned to function keys will be adjusted accordingly.

Issuing Terminal Commands

Terminal commands can be used in a Natural runtime environment. The following rules apply:

- You can enter a terminal command in any unprotected field on the screen (including the message line if it is unprotected).
- A terminal command must be entered in one field, except the control character which may be entered in a preceding field.
- It is recommended to enter a blank after a terminal command, or to delete the remaining content of the field in which you enter the command. Otherwise, Natural might misinterpret the field content as part of the terminal command.
- If you enter multiple terminal commands on the same screen, only the first command encountered will be executed, and all others will be ignored.
- Terminal commands which have been entered incorrectly are ignored, but you will not receive a corresponding error message.
- If you enter a terminal command and at the same time data in other fields on the screen, only the terminal command will be executed; the data will not be processed.

Using Terminal Commands in Programs

Terminal commands may also be issued from within a program by using the `SET CONTROL` statement. When a terminal command is specified with a `SET CONTROL` statement, the control character is omitted.

Terms Used in the Terminal Command Descriptions

In the descriptions of several terminal commands, the terms “screen” and “window” are used with the following meanings:

Term	Meaning
Screen	Depending on the operating system under which Natural is running, “screen” refers either to the entire terminal screen as such, or to the operating-system window in which the Natural session is running, or to the Natural main output window. However, for the sake of convenience, the term “screen” is used in all these instances.
Window	Always refers to the Natural window (as explained with the terminal command `%W`).

%? - Help for Terminal Commands

```
%?
```

This terminal command displays help information on the Natural terminal commands.

3 Terminal Commands Grouped by Function

- Case Translation 10
- Copy/Clear 10
- Language, Messages, Error Processing 10
- Screens, Terminal, and Window Processing 11
- Colors, Outlining 11
- INPUT Statement, Stack 11
- Session Recording and Screen Capturing 12
- Statistics and Trace 12
- Miscellaneous 13
- Key Assignments 13

The following tables provide an overview of the terminal commands grouped by functions.

Case Translation

Terminal Command	Function
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.

Copy/Clear

Terminal Command	Function
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.

Language, Messages, Error Processing

Terminal Command	Function
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.

Screens, Terminal, and Window Processing

Terminal Command	Function
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, %KP*n*	Simulate PF- and PA-keys.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and %T*ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.

Colors, Outlining

Terminal Command	Function
%D=	Control of outlining.
%=	Assign colors to fields.

INPUT Statement, Stack

Terminal Command	Function
%	Continuation indicator for INPUT statements in batch programs.
%D	Activate keyword/delimiter mode.
%F	Activate forms/screen mode.
%FM	Activate/deactivate edit mask free mode.
%R	Repeat INPUT statement.
%/	Force end-of-file for INPUT in batch mode.

Terminal Command	Function
%.P	Delete first entry from Natural stack.
%.S	Read stack entry without deleting it.

Session Recording and Screen Capturing

Terminal Command	Function
%A	Play back a recording.
%B	Activate/deactivate a recording.
%B=	Specify library for a recording.
%E	Display screen captures.
%G	Set playback mode for a recording.
%I	Capture current screen.
%O	Deactivate screen capturing.
%P	Capture screen sequence.
%S	Resume screen capturing.

Statistics and Trace

Terminal Command	Function
%TRE	Activate/deactivate external trace.
%TRI	Activate/deactivate internal trace.
%X	Control of statistics line/infoline.
%<TECH	Display technical information.
%<TEST	Invoke the debugger.

Miscellaneous

Terminal Command	Function
%H	Produce hardcopy output.
%J	Invoke helproutine.
%P=	CALL options.
%Q	Suppress next input.
%QO	Suppress pseudo-conversational output.
%T=	Activate converter routine for specific device type.
%V	Control of print mode.
%?	Invoke help information for terminal commands.
%% and %.	Interrupt current Natural operation.
%+ and %-	Enable/disable use of Natural Connection.

Key Assignments

Key(s)	Function
CLEAR	Interrupt current Natural operation; interrupt recording.
CTRL+D	Interrupt current Natural operation.
RESET+ENTER	Terminate current processing loop.

4 Terminal Command Key Assignments

- Assigning Terminal Commands to Function Keys 16
- CLEAR Key - Interrupt Current Operation 16
- CTRL+D Keys - Interrupt Current Operation 16
- RESET+ENTER Keys - Attention Interrupt 16

Assigning Terminal Commands to Function Keys

For enhanced operating convenience, you can assign a frequently used terminal command to a function key. The following methods exist:

- Within a program, you can assign terminal commands to function keys by using the statement `SET KEY`.
- Function-key assignments can also be made by the Natural administrator via the profile parameter `KEY`.

CLEAR Key - Interrupt Current Operation

Pressing the CLEAR key has the same effect as the terminal command `%%`.

You can also use the CLEAR key to interrupt a recording that is being played back in step mode.

CTRL+D Keys - Interrupt Current Operation

Pressing the keys CTRL+D has the same effect as the terminal command `%%`.

RESET+ENTER Keys - Attention Interrupt

This function is only available under Com-plete on terminals connected via a local controller.

Pressing RESET and then ENTER terminates the current processing loop. The loop must contain a database access statement.

5 % - Continuation Indicator for INPUT in Batch

```
%
```

When a percent sign (`%`) is used as the last non-blank character of a data record in batch mode, it causes the next record to be treated as a continuation record.

For further information, see *Use of Terminal Commands in Batch Mode* in the description of the `INPUT` statement in the *Statements* documentation.

6 %% and %. - Interrupt Current Operation

- %% in Online Mode 20
- %% in Batch Mode 20
- %. in Online Mode 21
- %. in Batch Mode 21

```
{ %% }
{ %. }
```

These terminal commands can be used to interrupt the current operation.

Note: The terminal commands `%%` and `%.` will be ignored if the profile parameter `ESCAPE` is set to `OFF`.

%% in Online Mode

If you enter `%%` in any field on the screen, the currently active Natural program will be terminated immediately and Natural will return to command input mode. If you enter `%%` in command input mode, the Natural session will terminated.

`%%` has the following effects:

- The contents of the Natural stack will be deleted.
- Any logical database transaction currently being processed is backed out.
- The source program currently in the work area of the editor will not be affected.

Note: If you have specified a special character other than the percent sign (%) with the Natural profile and session parameter `CF`, you may also enter the specified character two times consecutively.

%% in Batch Mode

In batch mode, `%%` may be used to set restart points in the input files and thus ensure the synchronization of the input files in the case of an error.

Influence of Profile Parameter CC

Command	Function
`CC=ON`	If the Natural profile parameter `CC` is set and an error occurs during the compilation/execution of a Natural program in batch mode, the input data stream for the `SYNIN` and `OBJIN` input files will be flushed until a line containing `%%` in the first two positions is encountered (if no `%%` is encountered, it will be flushed until the end-of-file is reached). In addition, the contents of the Natural stack will be deleted. If more data are available in the input stream, Natural resumes processing with the line after `%%`.
`CC=OFF`	Any `%%` in the input data will be ignored.

%. in Online Mode

Online, `%.` is the same as `%%`, except that the Natural stack is not deleted.

%. in Batch Mode

In batch mode, `%.` causes reading of input values for the current `INPUT` statement to be terminated.

7 %* - Inhibit Character Display

`%*`

%* in Online Mode

This command may be used when entering sensitive data (for example, passwords). `%*` causes all fields on the current screen to be non-displayable.

When used with the `SET CONTROL` statement, `%*` causes all fields on the next screen to be non-displayable.

%* in Batch Mode

In batch mode, `%*` causes printing of the next input data record to be suppressed. The input line after the line which contains `%*` will not be printed. This is useful, for example, for suppressing the printing of passwords.

Example of `%*` in batch - logon to Natural Security:

```
//CMSYNIN DD *
%*
SYSSEC,DBA,DBA
...
```

The printing of all input data in batch may be controlled with the `ECHO` profile parameter (see the *Parameter Reference*).

8 %.P - Delete First Entry from Stack

```
%.P
```

This command deletes the topmost entry from the Natural stack.

For further information on the Natural stack, see the section *Stack* (in the *Programming Guide*).

9 %.S - Read Stack Entry without Deleting it

```
%.S
```

Normally, commands/data are deleted from the stack as soon as they have been read from the stack.

This command causes the next `INPUT` statement to read the topmost entry from the stack without deleting it. The entry is treated as input data, no matter whether it actually is data or a command.

This allows you check a stack entry and then, depending on its content, decide whether to process it or not.

To check the contents of the stack, you can also use the system variable `*DATA`.

For further information on the Natural stack, see the section *Stack Processing* (in the *Programming Guide*).

10 %/ - End-of-File

```
%/
```

When entered in the first two positions of an input record being read with an `INPUT` statement in batch mode, this command causes an end-of-file condition.

11 %+ and %- - Enable/Disable Use of Natural Connection

These commands only apply if Natural Connection is installed.

```
%  { +[N] }
   {  -   }
```

These terminal commands are used to activate/deactivate Natural Connection.

Command	Function
%+	This command causes the Natural system variable *DEVICE to be set to PC. This permits usage of the Natural Connection system.
%+N	Same as %+. In addition, this command causes no field names to be sent when uploading/downloading data.
%-	This command causes the Natural system variable *DEVICE to be set to the setting in effect prior to execution of a %+ command.

See the *Natural Connection* documentation for further information.

12 %<RDC+ and %<RDC- - Activate/Deactivate Trace Recording

```
% { <RDC+ }
  { <RDC- }
```

These commands are used in conjunction with the *Profiler Utility* to activate or deactivate the trace recording function of the Natural Data Collector.

For further information, see *Activating and Controlling the Natural Data Collector* in *SYSRDC Utility* in the *Utilities* documentation.

`%<RDC+`	Activates the trace recording function of the Natural Data Collector.
`%<RDC-`	Deactivates the trace recording function of the Natural Data Collector.

See also system command `RDC` in the *System Commands* documentation and profile parameter `RDC` in the *Parameter Reference* documentation.

13 %<TECH - Display Technical Information

%<TECH

14 %<TEST - Invoke the Debugger

```
%<TEST
```

With this terminal command, you invoke the debugger.

See the *Debugger* documentation for further information.

15 %= - Assigning Colors to Fields

```
%= [ blank
     F
     N
     [ { { O } attribute } = color ] ,...2
       { { M }            }
       { { T }            }
       { BG               }
       { WN               }
       { PT               }
       { FR               }
     ]
```

With this command, you can assign colors to field attributes for programs that were originally not written for color support. The command causes all fields/text defined with the specified attributes to be displayed in the specified color.

If predefined color assignments are not suitable for your terminal type, you can use this command to override the original assignments with new ones.

You can also use this command within Natural editors, for example, to define color assignments dynamically during map creation.

Command	Function
General settings:	
blank	Clear color translate table.
F	Newly defined colors are to override colors assigned by program.
N	Color attributes assigned by program are not to be modified.
Field types:	

Command	Function
O	Output fields (`AD=O`). For detailed information on the session parameter AD, see *Field Input/Output Characteristics* in the *Parameter Reference* documentation.
M	Input fields; that is, input-only fields (`AD=A`) and modifiable fields (`AD=M`).
T	Text constants.
Possible field attributes:	
B	Blinking
C	Italic
D	Default
I	Intensified
U	Underlined
V	Reverse video
A color can also be assigned to the following parts of a screen:	
BG	Background
WN	Foreground (that is, fields for which no color is defined)
PT	Default page title
FR	Frame of a window
Possible colors:	
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

Example:

```
%=TI=RE,OB=YE
```

This example assigns color red to intensified text fields and color yellow to all blinking output fields.

See also *Assigning Colors to Fields - Terminal Command %=* in the *Programming Guide*.

16 %A - Play Back a Recording

`%Aname`

With the terminal command `%Aname`, you cause the recording saved under that *name* to be executed again. This requires that your current library is the one in which the recording is stored.

If you issue the command `%Aname` while a session is being recorded, the recording specified with `%Aname` will not be executed but the command `%Aname` is included into the source that is being recorded. Thus, you can execute a recording from within another recording and concatenate a series of recordings to one another. However, you cannot have nested recordings; the execution of the recording that contains the `%Aname` command stops after that command and is not resumed when the execution of name finishes.

17 %B - Activate/Deactivate a Recording

`%Bname`

The recording process is activated with the terminal command `%Bname`, and deactivated with the terminal command `%B`.

Command	Function
`%Bname`	Activates the recording process. All subsequent actions will be recorded. The name you specify is the name under which the recorded data will be saved. Names of recordings must be unique within a library.
`%B`	Deactivates the current recording process. The recorded source is automatically saved, and you can then play it back as often as you wish. You may also use `%B` to insert additional actions into the recording: after you have interrupted the playback of a recording with the CLEAR key, enter the command `%B`, and all action you perform will be inserted into the source of the recording until you enter `%B` again. Then the execution of the recording will be resumed.

18 %B= - Specify Library for a Recording

```
%B=library-name
```

With the terminal command `%B=library-name`, you specify the library in which all subsequent recording actions are to be stored.

If you activate the recording process without having specified a library, the name of the library in which the recording is stored will be the same as the value of the system variable `*INIT-USER` at the time when the recording process is activated.

When you log on to another library during a session being recorded, the library in which the recording is being stored will remain the same (that is, either the one specified with `%B=` or the `*INIT-USER` library); this means that one recording may record actions across multiple applications.

19 %C - Copying Contents of Page Buffer

`%C`

This terminal command is used to copy the contents of the page buffer to the next available lines in the Natural source work area.

The page currently displayed by Natural will be copied into the Natural source work area. The page content will be written to the next free location in the source work area, where it can be modified using the Naturalsource editor.

To clear the source work area before copying the page, the `%Z` terminal command may be used.

Notes:

1. `%C` should not be used in an editor session. Modifications made to the source area outside of editor content are not recognized by the editor.
2. The page buffer (the logical output from Natural) is not necessarily the same as the screen buffer which is displayed on the screen.

Example:

```
DEFINE DATA LOCAL
1 I (I2)
END-DEFINE
FOR I = 1 TO 10
  WRITE I
  SET CONTROL 'C'
END-FOR
END
```

20 %CS and %CC - Copying Data to Stack or *COM

```
%C { C } { C } [ A ] [S]
   { S } { Lnn } [ I ]
                 [ color ]
                 [ 1 ]
                 [ W ]
```

`1` and `W` cannot be specified with `Lnn`.

With this terminal command, you can copy parts of a screen into the Natural stack (`%CS`) or into the system variable `*COM` (`%CC`). The protected data from a specific screen line are copied field by field (except with option `A`; see below).

The second letter in the command determines where the data are copied to:

- `%CC...`
  Copies the data into the system variable `*COM`.
- `%CS...`
  Copies the data into the Natural stack. The data are placed on top of the stack as input data (as with a `STACK TOP DATA` statement).

The third letter in the command determines the line from which the data are copied:

- `%CCC` **and** `%CSC`
  Copies all protected data from the line in which the cursor is positioned, beginning from the field in which the cursor is positioned.

- `%CCL`*nn* **and** `%CSL`*nn*
  Copies all protected data from line number *nn*.

Moreover, you have the following options:

- `%C...A`
  Copies all of a line, that is, not only the protected data, but also the modifiable fields; the line is not copied field by field, but as a whole (including field attributes).
- `%C...I`
  Copies only the intensified fields from a line.
- `%C...`*color*
  Copies only the fields of that color from a line.
- `%C...C1`
  Copies only one field, namely the field in which the cursor is positioned (regardless of its attributes). (`%C...L` *nn*`1` is not possible.)
- `%C...CW`
  Copies only the word (as delimited by blanks or special characters within a field) over which the cursor is positioned. (`%C...L`*nn*`W` is not possible.)
- `%C...S`
  Causes Natural to "stay" on the screen from which the data are copied, when the command is executed. This allows you to copy several different data from a screen, before you process the data.

When you enter the command directly as `%C...` (or assign it to a PF-key), it applies to the entire *physical screen*; this means that you can copy from the entire screen, no matter whether the data you wish to copy are inside or outside an active window.

When you issue the command via a `SET CONTROL` statement, it applies to the *logical page* built by Natural. In conjunction with the stay option, this allows you to copy all data you need from an entire logical page (which may be larger than the actual physical screen), before you process these data.

See also *Copying Data from a Screen* (in the *Programming Guide*).

21 %D - Activate Keyword/Delimiter Mode

```
%D
```

This command is used to activate keyword/delimiter mode. This mode is recommended for processing in batch mode.

See the `INPUT` statement in the *Statements* documentation for additional information.

22 %D= - Control of Outlining

The outlining feature is only available on certain types of terminals, usually those which also support the display of double-byte character sets.

```
                M
                I
%D=B   {  color-code  }  ...
                B
                R
                W
```

The terminal command `%D=B` is used to control outlining.

Outlining (boxing) is the capability to generate a line around certain fields when they are displayed on the terminal screen. Drawing such "boxes" around fields is another method of showing the user the lengths of fields and their positions on the screen.

The `%D=B` command provides the following options:

Command	Function
`%D=BM`	This command causes all physically modifiable fields (`AD=M` or `AD=A`) to be displayed with outlining. If this command is repeated, this feature will be disabled again: in other words, it will return to the default state.
`%D=BI`	This command causes all intensified fields (`AD=I`) to be displayed with outlining.
`%D=B`*`color-code`*	This command causes all fields of the specified color to be displayed with outlining. For valid color codes, see the session parameter `CD`.
`%D=BB`	This command causes "big boxes" to be generated: if several fields displayed underneath one another have identical display characteristics, one big box is drawn

Command	Function
	around all of them. Normally (if `%D=BB` is not specified) an individual box is drawn around each individual field.
`%D=BR`	This command causes all boxes to be reset; that is, no field will be displayed with outlining.
`%D=BW`	This command causes all windows which are to be displayed with frame (see the `DEFINE WINDOW` statement in the *Statements* documentation) to be displayed with outlining instead of frame.

When outlining is active, filler characters in modifiable fields are not displayed, because they serve the same purpose as outlining, namely showing the user the position and length of a field.

For further information, see *Outlining - Terminal Command %D=B* (in the *Programming Guide*).

23 %DUE - Dump for Specific Error

```
%DUE  { +nnnn }
      { -nnnn }
      { -ALL  }
```

The terminal command %DUE is used within a session to specify Natural error numbers for which a storage dump shall be taken. This may be helpful to get a dump for the analysis of a specific error situation by Software AG personnel. See also profile parameter DUE.

The %DUE command provides the following options:

Command	Function
%DUE+*nnnn*	Adds an error number for which a storage dump is to be taken. If the number is greater than 9999, a NAT1134 error message is issued.
%DUE-*nnnn*	Removes a previously specified error number from the table. If the number is greater than 9999, a NAT1134 error message is issued.
%DUE-ALL	Removes all previously specified error numbers from the table.

If an error occurs which has been specified by %DUE, a program check is forced. If profile parameter DU=OFF is set, it will be changed to DU=ON. For further processing, the DU profile parameter setting is honored.

Examples:

```
%DUE+82
%DUE+80
%DUE-82
%DUE-ALL
```

24 %E - Display Screen Captures

```
%E
```

With this command, you can display any screens that have been captured with the NATPAGE screen paging function.

This command suspends normal processing and enters the display mode of the screen paging function: A list of all screens captured with this function (by using the terminal commands %I and %P) is displayed; including screen number, time of capturing, and map name (if the screen is a map). From the list you can then select the screens you wish to have displayed. You cannot enter any data on the displayed screens.

When a screen is captured with %I or %P, two items of information are superimposed on the screen: the time when the screen was captured, and a screen number (the screens are numbered sequentially in the order in which they are captured).

In NATPAGE display mode, an input field (CMD) is provided in which you can enter the following scrolling commands:

Command	Function
TOP or T	Displays the first screen which was captured.
BOT or B	Displays the last screen which was captured.
nnn	Displays the screen with screen number *nnn*.
+*nnn*	Scrolls *nnn* screens forward.
-*nnn*	Scrolls *nnn* screens backward.
.	Terminates display mode.

If you attempt to scroll to a screen number which does not exist, either the first or the last screen is displayed, depending on the scrolling direction.

See also the terminal commands %P, %O, %S, and %I.

25 %E= - Activate/Deactivate Error Processing

```
%E= { ON  }
    { OFF }
```

With the terminal command `%E=OFF` any error transaction and `ON ERROR` processing is switched off, with `%E=ON` error transaction and `ON ERROR` processing is switched on again.

Command	Function
`%E=OFF`	Switches any error transaction (as identified by the system variable `*ERROR-TA` or defined in Natural Security) and `ON ERROR` processing off; any error that occurs will then be handled by normal Natural error processing. This may be used to locate an error in your applications' error processing if the structure of the application with various error-handling procedures on different levels makes it impossible for you to find out where exactly an error originally occurred.
`%E=ON`	Switches error transaction and `ON ERROR` processing on again.

For further information, see *Using an Error Transaction Program* in the *Programming Guide*

26 %F - Activate Forms/Screen Mode

```
%F
```

This terminal command activates the forms/screen mode.

See the `INPUT` statement (in the *Statements* documentation) for additional information on forms/screen mode.

27 %F= - Frame Characters for Window

```
%F=chv
```

With this terminal command, you can define the characters to be used in the frame of a window.

c	The first character will be used for the four corners of the window frame.
h	The second character will be used for the horizontal frame lines (top and bottom frame lines).
v	The third character will be used for the vertical frame lines (left and right frame lines).

Example: The command `%F=+-!` would cause the window frame to look like this:

```
+-----------------------+
!                       !
!                       !
!                       !
!                       !
+-----------------------+
```

For further information on window processing, see the `DEFINE WINDOW` statement (in the *Statements* documentation).

See also *Windows* in the section *Screen Design* of the *Programming Guide*.

28 %FM - Numeric Edit Mask Free Mode

```
%FM { + }
    { - }
```

This command is used to activate/deactivate edit mask free mode, a special capability to allow literals to be omitted during input into a field with a numeric edit mask.

Command	Function
`%FM-`	With this command, you switch edit mask free mode off.
`%FM+`	With this command, you switch edit mask free mode on.

The default setting at session startup is provided with profile parameter `EMFM`.

See the `INPUT` statement (in the *Statements* documentation) for additional information on the edit mask free mode.

29 %G - Set Playback Mode for a Recording

```
%G  { ON  }
    { OFF }
```

A recording can be played back in two modes:

- **Background Mode**
  In background mode, the entire recording is played back invisibly; that is, all the actions of the recording are carried out without anything being displayed to you on the terminal screen during the execution of the recording. Neither can you interrupt a recording that is played back in background mode (unless the recording contains the terminal command %R).
- **Step Mode**
  In step mode, a recording is played back step by step, and all actions are displayed to you. By pressing ENTER you proceed from one step to the next. In step mode, it is also possible for you to interrupt the recording by pressing the CLEAR key.

By default, a recording is played back in background mode.

Command	Function
%GON	With this command, you switch step mode on.
%GOFF	With this command, you switch back to background mode.
%G	With this command, you can toggle between the two modes.

30 %H - Hardcopy Output

```
%H [ destination
     ,
     .
     #destination
     -
     L ]
```

These terminal commands are used to produce hardcopy output from Natural reports on a printer or special destination, such as the source area.

Note: The commands are only valid if the hardcopy feature is implemented.

By default, a `%H` command is effective for the current logical output (that is, the current window without message line, function-key lines and statistics line/infoline). However, you can also print the current screen content (see `%H`, and `%H.` below).

You have the following options:

Command	Function
`%H`	This command activates the hardcopy output. The hardcopy output is deactivated when either an `INPUT` statement with modifiable fields is executed, or the end of the program is reached. All pages, which are processed when the hardcopy output is active, will be forwarded to the printing destination. These pages are not displayed on the screen. If `%H` is entered in an input field, only the current page will be printed. No selection window is displayed for the definition of the output destination; instead, the printer specified with the profile parameter `HCDEST` is used.
`%H`*`destination`*	Same as `%H`, but the printing destination is specified with destination. The *`destination`* can be 1 to 8 characters long.

Command	Function
`%H,`	All subsequent `%H` commands apply to the current screen.
`%H.`	All subsequent `%H` commands apply to the current logical page (applies by default).
`%H=`	This command activates/deactivates (toggle switch) the logging of all following pages. Logging means that all pages, which are displayed on the screen are forwarded to the printing destination as well. This feature may be used, for example, to log a sequence of output for administrative, debugging or educational purposes. For the hardcopy output, the printer specified with the profile parameter `HCDEST` is used.
`%H=`*`destination`*	Same as `%H=`, but the printing destination is specified with destination. The destination can be 1 to 8 characters long.
`%H#`*`destination`*	With `%H#`*`destination`*, you can have the hardcopy output routed to a special destination, as defined with a `DEFINE PRINTER` statement, for example, to the Natural source area, to Con-nect, or to the `INFOLINE`.
`%H-`	Deactivates the hardcopy output immediately. **Note:** When a `SET CONTROL 'H-'` statement is executed, data which were already written to the page buffer but have not been output yet, will not be routed to the printer. To also print these data, you have to code an `EJECT` statement before the `SET CONTROL 'H-'` statement.

Note: The `EJECT` statement does not affect the `%H` command. The `%H` command always causes Natural to advance pages.

31 %I - Capture Current Screen

```
%I
```

When you enter this command on a screen, the screen will be captured by the NATPAGE screen paging utility.

The maximum number of screens that can be captured is determined by the session parameter PD. If this number is exceeded, each additional screen will overwrite one that has already been captured, beginning with the one captured first.

To display screens that have been captured with the NATPAGE screen paging utility, use the terminal command %E.

See also the terminal commands %P, %O, %S, and %E.

32 %J - Invoke Helproutine

```
%Jhelproutine
```

This terminal command can be used to invoke an interactive helproutine.

When `%J` is used after interrupting the playback of a recording, the playback will be resumed after the helproutine has been executed.

33 %K and %KP - Simulate PF- and PA-Key

```
%K { nn }
   { Pn }
```

These terminal commands can be used to simulate the terminal function (PF, ENTER) and program attention (PA) keys. Before a key can be simulated, it must have been made sensitive using the SET KEY statement.

Command	Function
%K*nn*	Simulates the terminal function key numbered *nn* (PF1 to PF48). This permits PF-keys 13-24 to be assigned to PF-keys 1-12, or the activation of PF-keys not available on the keyboard used. This terminal command also makes function keys available in batch mode.
%K0	Simulates the ENTER key.
%KP*n*	Simulates the program attention key numbered *n* (PA1 to PA3) (see %K*nn*).

34 %L - Disable Lower- to Upper-Case Translation

```
%L
```

This command prevents that lower-case characters are translated to upper-case by Natural.

`%L` influences the interactive input which is entered, for example, with a Natural `INPUT` statement. It does not, however, influence the input from the stack.

You should also ensure that the TP monitor used does not perform any translation before data are passed to Natural.

See also the terminal command `%U`.

Application Programming Interface: `USR1005N`.

35 %L= - Set Language Code

```
%L=nn
```

With the terminal command `%L=nn`, you can set the language code *nn* to be used by Natural.

For a list of possible language codes, see the system variable `*LANGUAGE`.

To set the language code already at session start, specify the profile parameter `ULANG`.

36 %M - Control of Message Line

- Message Line Positioning 82
- Message Line Protection 83
- Message Line Color 83

```
%M [ T
     B
     [-]nn
     P
     BL
   = { GR } ]
     NE
     PI
     RE
     TU
     YE
```

With this terminal command, you can control the position, the protection mode and the color of the Natural message line.

This terminal command is ignored in batch mode.

This terminal command also applies to Natural's `NEXT` (or `MORE`) line.

Application Programming Interface: `USR1005N`.

Message Line Positioning

Command	Function
`%MT`	Causes the message line to be the top line of the screen.
`%MB`	Causes the message line to be the bottom line of the screen.
`%M`	Causes the current message line position to be switched from the top line to the bottom line of the screen (or vice versa), or from line *nn* to the bottom line.
`%M`*nn*	Causes the message line to be placed on line *nn* of the screen.
`%M-`*nn*	Causes the message line to be placed on the *nn*th line from the bottom of the screen. If the line number *nn* or -*nn* is not within the current screen, the message line will not be displayed.

Message Line Protection

Command	Function
`%MP`	Switches from protected to unprotected message line or vice versa.

Message Line Color

Command	Function
`%M=`*`color-code`*	When a color screen is used, this terminal command causes the message line to be displayed in the specified color (for an explanation of color codes, see the session parameter `CD`).

See also *Control of the Message Line - Terminal Command %M* in the *Programming Guide*.

37 %MSGSF - Display System Error Messages in Full

```
          ON
          +
          T
%MSGSF=   OFF
          -
          F
```

By default, a Natural system error message consists of: the name of the program and the number of the line that caused the error, followed by the actual text of the message. Depending on the size of the window in which the message is displayed, the actual text may be truncated. With the terminal command `%MSGSF`, you can avoid such truncation.

Command	Function
`%MSGSF=ON`	System error messages will be displayed in full; that is, program name, line number and actual message text will be displayed (this is the default).
`%MSGSF=+`	
`%MSGSF=T`	
`%MSGSF=OFF`	System error messages will be displayed in short form; that is, only the actual message text will be displayed (but not the program name and line number).
`%MSGSF=-`	
`%MSGSF=F`	

Notes:

1. Instead of `ON`, you can also specify "+" or "T" (true). Instead of `OFF`, you can also specify "-" or "F" (false).
2. The display format of system error messages can also be controlled with the profile parameter `MSGSF` (see the *Parameter Reference*).

38 %N - Activate Non-Conversational Mode

```
%N
```

This terminal command is used with a `SET CONTROL` statement and causes the next logical output screen to be displayed without requiring any user response for processing to continue; that is, after the screen has been displayed, processing will continue immediately without waiting for any user input.

This command may be used to send messages about the progress of program execution to the user.

Under IMS TM:

This terminal command does not apply under IMS TM. If it is used under IMS TM, it causes the next logical output screen to be suppressed.

39 %O - Deactivate Screen Capturing

```
%O
```

With this command, you deactivate the capturing of screens by the NATPAGE screen paging utility as activated by the terminal command %P.

The current screen is captured. All screens captured since the last %P command are retained.

See also the terminal commands %E, %I, %P and %S.

40 %P - Capture Screen Sequence

```
%P
```

With this command, you activate the NATPAGE screen paging utility to capture the current screen and all subsequent screens.

The maximum number of screens that can be captured is determined by the session parameter PD. If this number is exceeded, each additional screen will overwrite one that has already been captured, beginning with the one captured first.

All screens that have been captured with previous %P and %I commands will be deleted when you issue a %P command.

To display screens that have been captured, you use the terminal command %E.

See also the terminal commands %E, %I, %O and %S.

Application Programming Interface: USR0350N.

41 %P= - CALL Options

- %P=C - Pass Parameter Values instead of Address 94
- %P=CC - Pass Parameter Values in a CICS Container 96
- %P=I - Pass PCB Addresses to Callee 96
- %P=L - Call of LE Dynamic Main Program 97
- %P=LS - Call of LE Subroutine Program 97
- %P=S - Standard Linkage for Call 98
- %P=SC - Standard Linkage Call Simulating EXEC CICS LINK 98
- %P=SQ - Standard Linkage for Call of Quasi-Reentrant Subprograms 98
- %P=U - Separate Unit of Work for Called Program 99
- %P=UT - Separate Unit of Work for Called Program, Tolerate Transaction Backout 99
- %P=V - Roll-Out for Call 99

```
%P= { C  }
      { CC }
      { I  }
      { L  }
      { LS }
      { S  }
      { SC }
      { SQ }
      { U  }
      { UT }
      { V  }
```

Each %P= command applies only to the next call, and the call options are reset unconditionally on return from the call.

Therefore it is highly recommended to code the %P= directly before the relevant CALL statement.

Note: To set the CALL option permanently, use the profile parameter PGP or the corresponding parameter macro NTPGP.

When the Natural CICS Interface is used, the commands %P=S(C), %P=V, %P=C(C), %P=U and %P=UT may be used to set special options which apply when a Natural program calls a non-Natural program (via a CALL statement). For further information, see *Threadsafe Considerations* in the *TP Monitor Interfaces documentation*. In all other environments, these commands are ignored.

When the Natural IMS TM interface is used, the command %P=I may be used to set special options which apply when a Natural program calls a non-Natural program (via a CALL statement). In all other environments, this command is ignored.

For more information about how Natural supports IBM Language Environment (LE) subprograms, see *LE Subprograms* in the *Operations* documentation.

For details on calling non-Natural programs, see the CALL statement in the *Statements* documentation.

%P=C - Pass Parameter Values instead of Address

This command only applies if the Natural CICS Interface is used.

Normally, when a Natural program calls a non-Natural program under CICS, the address of the CALL statement parameter address list is passed in the COMMAREA. If you wish the parameter values themselves, rather than the address of their address list, to be passed in the COMMAREA, issue the terminal command %P=C before the call.

This makes it possible, for example, to use DPL for called CICS programs: A CICS program that resides in another CICS region can only be called with `%P=C`; as addresses within the "calling" region cannot be accessed by the "called" region, the parameter values have to be passed instead.

When `%P=C` is used, no parameters are passed in the TWA, but only parameter values in the CICS COMMAREA. All parameters of the `CALL` statement parameter list are copied adjacent to one another, regardless of their alignment. The resulting COMMAREA length is the sum of the individual parameter lengths (this has to be taken into consideration when determining the number of array occurrences to be passed). On return from the called program, the parameters are copied back.

When overlapping fields are passed or the same field is passed more than once, these fields should be made "read-only" for the called program; otherwise, unpredictable results may occur when parameter values are returned to the calling program.

For `%P=C`, the restriction applies that group arrays cannot be passed:

```
01 #GROUP (2)
   02 #FIELD1 (A1)
   02 #FIELD2 (P7)
```

Either pass them as individual arrays:

```
01 #GROUP
   02 #FIELD1 (A1/2)
   02 #FIELD2 (P7/2)
```

Or redefine them:

```
01 #GROUP
01 REDEFINE #GROUP
   02 #ARRAY (A1/10)
```

and specify the array name in the `CALL` statement.

Notes:

1. If both `%P=S` and `%P=C` are issued for a call, `%P=C` will be ignored.
2. CICS COMMAREAs are restricted to 32 KB.
3. If the parameter `CNTCALL` is set to `ON` in the macro `NTCICSP`, and the parameter value data length exceeds 32KB, a CICS container is passed instead of a COMMAREA.
4. Avoid storage violations in CICS by ensuring that a called program checks whether the length of the data area to be passed really matches the internal data structure. This is achieved by checking EIBCALEN against the COMMAREA length: `IF LENGTH OF DFHCOMMAREA GREATER THAN EIBCALEN` … (in COBOL) or `IF SIZE (DFHCOMMAREA) > EIBCALEN` … (in PL/1).

%P=CC - Pass Parameter Values in a CICS Container

This command only applies if the Natural CICS Interface is used.

This functionality requires CICS Transaction Server for z/OS version 3.1 or above.

The difference between `%P=C` and `%P=CC` is that the parameter values are passed in a CICS container right away. All restrictions except the length limitation of 32 KB still apply.

%P=I - Pass PCB Addresses to Callee

This command only applies if the Natural IMS TM interface is used.

`%P=I` indicates to Natural that PCB addresses are to passed to the 3GL program in the following `CALL` statement. As it is not possible to pass the address of a PCB from Natural to the 3GL program directly, a special convention is used to indicate to Natural that parameters passed to the 3GL program may contain either the position or the name of a PCB. For fields that satisfy this convention, Natural will pass the address of the PCB to the called 3GL program instead of the address of the specified parameter.

Only scalar fields of either format A8 or A12 are checked for a PCB specification. The fields that contain a PCB specification must comply with the following conventions:

- An A8 field must contain the string `'PCB=nnnn'` where *nnnn* is a four digit number that represents the position of the PCB in the PSB address list.
- An A12 field must contain the string `'PCB=name'` where *name* is the eight character name for a PCB that has been defined with the NTIMSPT macro when the Natural IMS TM interface was installed.

Example:

After `%P=I` has been executed, for the statement

```
CALL 'MY3GL' USING 'PCB=MYPCB ' 'PCB=0004' PARAM
```

the following parameters are passed to the 3GL program MY3GL:

- address of the PCB with the name "MYPCB" as specified with macro NTIMSPT,
- address of the fourth PCB in the PSB address list,
- address of the Natural field `PARAM`.

The following return codes are returned by the `RET` system function if the PCB specification is invalid:

Return Code	Explanation
-4	The specified value *nnnn* for format `'PCB=nnnn'` is not numeric.
-8	The specified PCB name has not been defined with the macro NTIMSPT.
-12	The specified PCB with position *nnnn* does not exist in the PSB address list.

%P=L - Call of LE Dynamic Main Program

This command only applies if the option for the support of the IBM Language Environment (LE) calling conventions has been set when Natural was installed.

This command causes control to be returned to Natural after a call to an LE dynamic main program.

By default, when an LE dynamic main program is invoked, control is not returned to Natural after the invoked program has been processed. For control to be returned after that program has been processed, you have to use `%P=L` before invoking the program.

For information on how Natural supports IBM Language Environment (LE) subprograms, see *LE Subprograms* (in the *Operations* documentation).

Note: Under CICS `%P=L` and `%P=LS` are equivalent, as LE main programs are not supported.

%P=LS - Call of LE Subroutine Program

This command only applies if the option for the support of the IBM Language Environment (LE) calling conventions has been set when Natural was installed.

This command causes that a dynamic or static non-Natural program is invoked using LE calling conventions, that is, via standard linkage conventions with Register 12 holding the address of the LE CAA.

For information on how Natural supports IBM Language Environment (LE) subprograms, see *LE Subprograms* (in the *Operations* documentation).

%P=S - Standard Linkage for Call

This command only applies if the Natural CICS Interface is used.

Normally, when a Natural program calls a non-Natural program under CICS, the call is accomplished by an `EXEC CICS LINK` request.

If standard linkage is to be used for the call instead, issue the terminal command `%P=S`. In this case, the called program must adhere to standard linkage conventions with standard register usage.

%P=SC - Standard Linkage Call Simulating EXEC CICS LINK

This command only applies if the Natural CICS Interface is used.

Normally, when a Natural program calls a non-Natural program under CICS, the call is accomplished by an `EXEC CICS LINK` request.

When this terminal command is used, the Natural CICS Interface invokes the non-Natural program using standard linkage conventions, but passing parameters as a `CICS LINK` does, that is, Register 1 points to a parameter list holding addresses of CICS EIB and a CICS COMMAREA.

Notes:

1. `%P=SC` can be combined with `%P=C`.
2. A CICS program invoked via standard linkage call must not return to the caller via `EXEC CICS RETURN`. Assembler programs must return via `DFHEIRET` macro call. COBOL programs must return via `GOBACK` statement.

%P=SQ - Standard Linkage for Call of Quasi-Reentrant Subprograms

This command only applies if Natural CICS Interface is used and if the Natural CICS Interface version installed at your site supports this command.

This command does not apply if a called non-Natural subprogram is defined to Natural with the `CSTATIC` or `RCA` profile parameter.

`%P=SQ` is functionally equivalent to the terminal command `%P=S`.

If Natural is defined with the CICS `CONCURRENCY(REQUIRED)` attribute and uses the CICS open transaction environment (OTE), all subprograms called using standard linkage conventions must be fully reentrant.

If a non-Natural subprogram is called that is quasi-reentrant but not threadsafe and fully reentrant while Natural runs on an open TCB, `%P=SQ` ensures that the called program is executed under the CICS QR TCB. Natural then issues the CICS command `EXEC CICS LINK PROGRAM('NCILINKQ')` to switch to the QR TCB and execute the `NCILINKQ` program which calls the subprogram using standard linkage conventions.

This command also applies to LE subprograms. For more information about how Natural supports IBM Language Environment (LE) subprograms, see *LE Subprograms* in the *Operations* documentation.

%P=U - Separate Unit of Work for Called Program

This command only applies if the Natural CICS Interface is used.

This command specifies that a CICS program called by Natural via DPL has its own unit of work, which is committed on return to Natural; that is, changes to recoverable resources made by the called program are committed or rolled-back independent of changes to recoverable resources made by Natural. In terms of CICS: This command causes an `EXEC CICS LINK` with the `SYNCONRETURN` option.

%P=UT - Separate Unit of Work for Called Program, Tolerate Transaction Backout

This command has almost the same meaning as `%P=U`, with one exception:

For `%P=U`, all CICS conditions on return from the called program other than `NORMAL` will result in a Natural NAT0920 error message with the reason code being the response code of the `EXEC CICS LINK` command.

For `%P=UT`, a CICS ROLLEDBACK response code does not result in an error, but is tolerated and treated as `NORMAL`.

%P=V - Roll-Out for Call

This command applies to Natural environments where several sessions share threads in the same region, such as the Natural CICS Interface and the Natural batch server environment.

Normally, when a Natural programs calls a non-Natural program and the called program issues a conversational terminal I/O, the Natural thread is blocked until the user has entered data.

`%P=V` can be used to prevent the Natural thread from being blocked: if this terminal command is specified, the parameter data passed from the Natural program to the called program are copied out of the Natural thread, and the thread is rolled out before the call. The thread is then available for another user. Upon return from the called program to the calling Natural program, the thread is rolled back in again, the (modified) data area is copied into the thread, and Natural processing continues.

Note: Only the parameters specified in the `CALL` statement are copied out of and back into the thread.

42 %Q - Suppress Next Input

```
%Q
```

In interactive processing, the `%Q` command is ignored. When issued in batch mode, the `%Q` command causes the output of maps, or screens produced by `INPUT` statements, to be suppressed.

`SET CONTROL 'Q'` causes the next `INPUT` statement *not* to be processed. This may be used, for example, if at the end of a helproutine the processing is to continue without the user having to press ENTER upon return from the help to the map.

43 %QO - Suppress Pseudo-Conversational Output

This command is only applicable under CICS.

```
%QO
```

In a Natural session running under CICS in pseudo-conversational mode, you can switch to another CICS transaction by issuing the statement `CALL 'CMTASK'`. To enable CICS to start the other transaction, a terminal I/O, i.e. an `INPUT` statement, is required after the `CALL` statement.

To suppress the screen output of that `INPUT` statement (which would be overwritten immediately by the started transaction anyway), you issue a `SET CONTROL 'QO'` statement before the `INPUT` statement.

44 %QS - Simultaneous Output of Multiple Screens

```
%QS
```

With this command, you can display multiple screens simultaneously.

`%QS` causes the next screen I/O not to be executed. The corresponding output screen is kept internally until the following I/O, when it is displayed together with the next screen. Therefore `%QS` only makes sense if the second output screen is a window, that is, if it does not entirely overlay the first screen that was suppressed with `%QS`.

Example: You can suppress the output of a screen A with `%QS`; the next screen B is a window which partially overlays screen A (perhaps a help window for one of the fields on screen A); with the next screen I/O, the window B and the "underlying" screen A are displayed simultaneously.

A `%QS` command only applies to the subsequent screen.

Note: As `%QS` reduces the number of screen I/O operations, it also improves performance.

45 %R - Repeat INPUT Statement

```
%R
```

This command causes `INPUT` statement repetition and the output screen to be rebuilt. All output data generated from the beginning of the `INPUT` statement will be reproduced.

%R in a Recording

By recording the terminal command `%R`, you can manipulate a single step in a recording when it is played back.

You may also use `%R` to overwrite input data in a recording that is being played back.

46 %RM - Protection of Light Pen Sensitive Fields

```
%RM
```

This command causes all light pen sensitive fields on the screen to be made write-protected, that is, the user can select them with a light pen, but not overwrite their contents. To switch the write-protection off again, issue the command again.

For a field to be light pen sensitive, it must be displayed intensified (session parameter `AD=I`) or blinking (`AD=B`), and the first character in the field must be one of the light pen designator characters ("?", ">" ,"&" or a blank); selecting a field with a light pen causes the designator character to be changed, therefore you can make the processing of fields selected with a light pen dependent on the values of the designator characters.

See also *Light Pen Support* (in the *Operations* documentation) and the `PEN` value of the system variable `*PF-KEY`.

47 %RN - Suppress Compression of Screen Data

```
%RN
```

For the next screen I/O, this command suppresses Natural's automatic compression of screen data and causes the entire screen to be sent instead.

At a screen I/O, Natural usually does not send the entire screen, but only the changed screen data. When a non-Natural program is invoked which causes a screen I/O, Natural in most TP environments recognizes the non-Natural I/O and sends the entire screen upon the next screen I/O. However, in some TP environments, the non-Natural I/O is not recognized by Natural; in such a case, `%RN` should be used.

`%RN` only applies to the next screen I/O.

48 %RO - Activate/Deactivate Screen Optimization

```
%RO { ON  }
    { +   }
    { T   }
    { OFF }
    { -   }
    { F   }
```

With this command you can switch Natural's automatic screen optimization off and back on again.

Normally, Natural's screen optimization causes screens to be sent as compressed as possible. If this should conflict with any TP monitor's screen optimization or hardware limitations, you can use this terminal command to switch Natural's screen optimization off; screens are then sent in non-compressed form.

If you use the session parameter `BX` setting `BX=L` or `BX=R`, you should switch off Natural's screen optimization.

Command	Function
%ROON	Activates screen optimization.
%RO+	
%ROT	
%ROOFF	Deactivates screen optimization.
%RO-	
%ROF	
%RO	Switches between the two modes.

Note: Instead of `ON`, you can also specify "+" or "T" (true). Instead of `OFF`, you can also specify "-" or "F" (false).

49 %S - Resume Screen Capturing

```
%S
```

This command is used to resume the capturing of screens by the NATPAGE screen paging utility; see also *NATPAGE Utility - Screen Capturing*.

The capturing of screens (which is activated with the terminal command %P) can be interrupted with the terminal command %O. After it has been interrupted, you can use the terminal command %S to resume the capturing of screens.

See also the terminal commands %E, %I, %O and %P.

50 %T - Position Cursor to Top of Active Window

```
%T
```

This command positions the cursor at the top left-hand corner of the active window on the next screen output.

51 %Tll/cc - Position Cursor to Line ll, Column cc

`%Tll/cc`

This command positions the cursor at line *ll*, column *cc* on the next screen output.

The line and column positions are counted beginning with 1 within the current logical page. For this reason, the message line, the function-key lines, and the statistics line/infoline (if active) are not taken into account for the cursor position; that is, `%T1/1` always positions to the beginning of the topmost data line of the page (if visible).

You have to execute the terminal command `%T+` before you can place the cursor at any desired position on the screen.

52 %T* - Position Cursor Outside Window

```
%T*
```

Normally, when a window is active and the window contains no input fields (`AD=A` or `AD=M`), the cursor is placed in the top left corner of the window.

This terminal command causes the cursor to be placed in a `*COM` system variable outside the window when the active window contains no input fields.

Command	Function
`%T*`	Switches between cursor placement in system variable `*COM` outside the window and standard cursor placement within the window.

`%T*` only applies to the next `INPUT` statement, and it must be issued *before* the `INPUT` statement.

See also *Positioning the Cursor to *COM - the %T* Terminal Command* (in the *Programming Guide*).

53 %T= - Activate Converter Routine for Device Type

```
%T= { 3270 }
      { 3277 }
      { 3278 }
      { 3279 }
      { TTY  }
      { TTYT }
      { TWX  }
      { WEB  }
      { RGUI }
      { BTCH }
      { BTC2 }
      { ASYN }
      { ASYL }
      { PRNT }
      { SCSP }
      { DBCS }
```

This command causes Natural to use the appropriate converter routine for the specified device type.

If the TP environment is not able to inform Natural of the physical terminal type it is working with (in which case the terminal type would be set automatically), this command may be used to specify the terminal type. This allows Natural to build the appropriate attribute sequence to operate the terminal.

The default terminal types used by Natural (if no information is passed from the TP environment) for z/OS environments are `%T=3270`.

Instead of the `%T=` command, you can also use the profile parameter `TTYPE`.

Note: The above syntax lists only the most commonly used terminal types. If your terminal is of another type, ask your Natural administrator to check the `NATCONFG` module to ascertain whether the terminal type can be specified with `%T=`.

54 %TRE - Activate/Deactivate External Trace

```
        { ON  }
        { +   }
        { T   }
%TRE    { OFF }
        { -   }
        { F   }
```

This command activates/deactivates the external trace function. Instead of `ON`, you can also specify "+" or "T" (true). Instead of `OFF`, you can also specify "-" or "F" (false).

The external trace function can also be activated/deactivated with the profile parameter `ETRACE` (see the *Parameter Reference*).

Important: Do not use this command without prior consultation of Software AG support. This function is intended primarily for Software AG internal use for debugging purposes. It writes trace data to an external trace data set depending on the TP environment in which Natural is running.

55 %TRI - Activate/Deactivate Internal Trace

```
        { ON  }
        { +   }
        { T   }
%TRI    { OFF }
        { -   }
        { F   }
```

This command activates/deactivates the internal trace function. Instead of `ON`, you can also specify "+" or "T" (true). Instead of `OFF`, you can also specify "-" or "F" (false).

The internal trace function can also be activated/deactivated with the profile parameter `ITRACE` (see the *Parameter Reference*).

Important: This function is intended primarily for Software AG internal use for debugging purposes. It passes trace data to the `SYSRDC` utility

56 %U - Translate Lower to Upper Case

```
%U
```

This command causes Natural to translate lower-case characters to upper-case for alphanumeric input data.

Upper-case translation is in effect by default.

`%U` influences the interactive input, which is entered, for example, with a Natural `INPUT` statement. It does not, however, influence the input from the stack.

See also the terminal command `%L`.

Application Programming Interface: `USR0350N`.

57 %V - Control of Print Mode

This terminal command only applies to terminals which support inverse (right-to-left) print mode. See also profile parameter `PM` in the *Parameter Reference* documentation, and *Bidirectional Language Support* in the *Unicode and Code Page Support* documentation.

```
%V  [ ON
      OFF
    N [ON | OFF]
    A [ON | OFF]
    K [ON | OFF] ]
```

Syntax Description:

Command	Function
`%VON`	Activates inverse (right-to-left) print mode (corresponds to `PM=I`). All non-alphanumeric fields, system variables and PF key lines are reversed. This ensures the correct display of these items on the terminal running in right-to-left mode.
`%VOFF`	Activates normal (left-to-right) print mode (corresponds to `PM=R`).
`%V`	Toggles between normal and inverted print mode.
`%VNON`	Activates inverse print mode for numeric fields if profile parameter `PM=I` is set or `%VON` was issued. This is the default setting.
`%VNOFF`	Activates normal print mode for numeric fields if profile parameter `PM=I` is set or `%VON` was issued. This may be used for sensitive terminals which do not invert numerics.
`%VN`	Toggles between normal print mode and inverse print mode for numeric fields. This may be used for sensitive terminals which do not invert numerics.

Command	Function
%VAON	If profile parameter PM=I is set, this causes alphanumeric input fields that are defined with field attribute PM=I to be sent to the terminal with the numeric field attribute. This ensures that these fields, which would otherwise be considered right-to-left cursor oriented in right-to-left mode, will be considered left-to-right. The corresponding hardware feature is required. Alternatively, field attribute (AD=Q) can be used individually instead of this terminal command.
%VAOFF	Alphanumeric input fields are sent to the terminal without the numeric field attribute. This is the default setting.
%VA	Toggles between %VAON and %VAOFF.
%VKON	Numeric fields for which a helproutine or edit mask is defined are internally converted by Natural to alphanumeric format (otherwise a user would not be able, for example, to enter a question mark in such a field to invoke the helproutine). With inverse print mode, however, this internal conversion would cause a numeric value in such a field to be interpreted from right-to-left. For numeric fields that are defined with PM=I and have a helproutine or edit mask, %VKON prevents the internal translation to alphanumeric format, thus ensuring the correct interpretation of values in these fields, while an edit mask will still be applied correctly and a question mark can nonetheless be entered in the field.
%VKOFF	Deactivates the feature described above. This is the default setting.
%VK	Toggles between %VKON and %VKOFF.

Application Programming Interface: USR0350N.

58 %W - Window Processing

- Window Size and Position on the Physical Screen 134
- Window Position on a Logical Page 136
- %WA and %WZ - Save Screen Image before Window 139

```
%W
```

Note: You are strongly recommended to use the `DEFINE WINDOW` statement instead of the `%W` command.

A Natural window is that segment of a logical page, built by a Natural program, which is displayed on the terminal screen.

The `%W` command controls the processing of this window.

The command must always be specified with parameters for the various functions as described hereafter. Multiple parameters may be specified with one `%W` command; they must be specified consecutively without any delimiter characters.

There is always a window present, although you may not be aware of its existence: unless specified differently (with a `%W` command or `DEFINE WINDOW` statement), the size of the window is identical to the physical size of your terminal screen.

See also the `DEFINE WINDOW` statement in the *Statements* documentation for information on window processing.

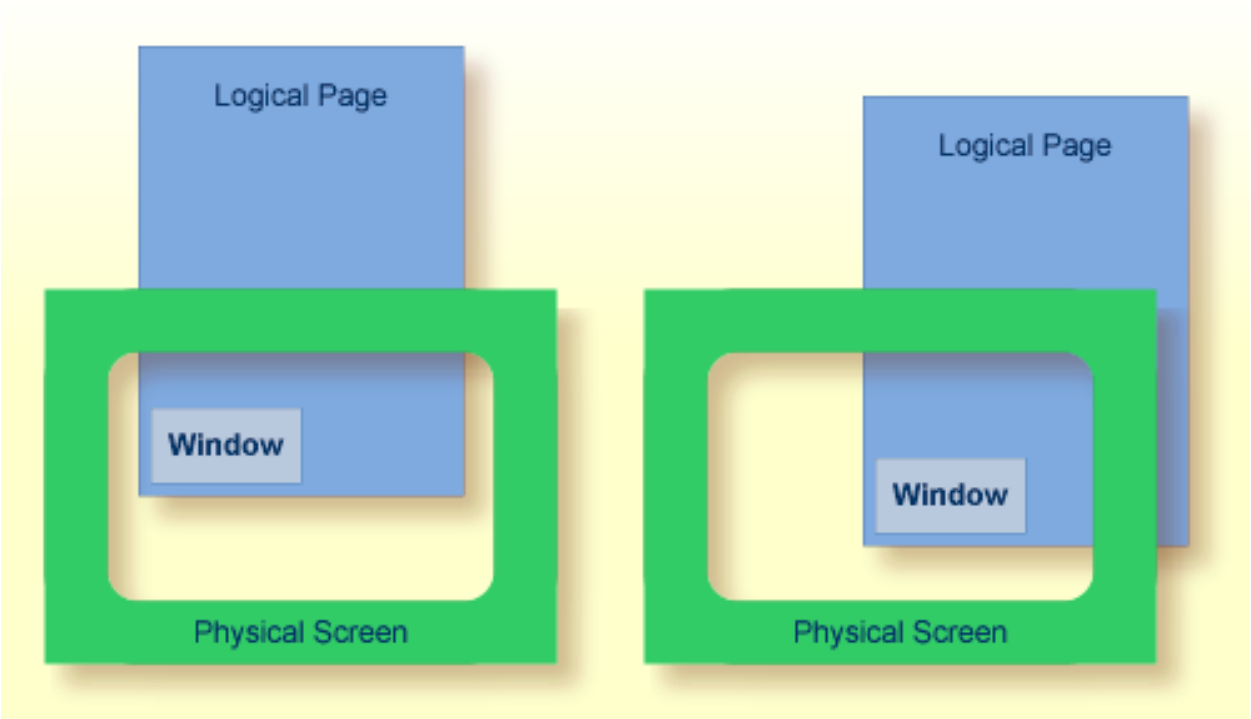
There are two types of window commands:

- commands to control the size and position of the window on the physical screen;
- commands to control the position of the window on the logical page created by the program.

Window Size and Position on the Physical Screen

The following window commands control the size and position of the window on the physical screen.

When you change the position of the window on the physical screen, the position of the window on the logical page will remain unchanged:



For information on possible window sizes, see the `DEFINE WINDOW` statement.

Command	Function
`%WB`	The window size (excluding frame) will be set to physical screen size. If a frame is defined, it will not be visible.
`%WB`*lll*`/`*ccc*	The top left corner of the window will be positioned to line number *lll*, column number *ccc* (lines and columns are counted on the physical screen). The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
`%WB0`	The window will be positioned to the top left corner of the screen. The window size will remain unchanged.
`%W#`	The top left corner of the window will be positioned to the cursor position. The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
`%W?`	The bottom right corner of the window will be set to the cursor position. The top left corner of the window will remain unchanged, and the size of the window will be adjusted accordingly.
`%WL`*nn*	The line size (horizontal extension) of the window (including frame, if specified) will be set to *nn*. If *nn* is omitted or specified larger than would fit on the screen, the line size will be set to the maximum possible (that is, to the right edge of the screen).

Command	Function
%WC*nn*	The column size (vertical extension) of the window (including frame, if specified) will be set to *nn*. If *nn* is omitted or specified larger than would fit on the screen, the column size will be set to the maximum possible (that is, to the bottom of the screen).

Column size and line size specifications refer to the overall physical size of the window (including frame, if specified), not to the size of what is logically visible inside the window.

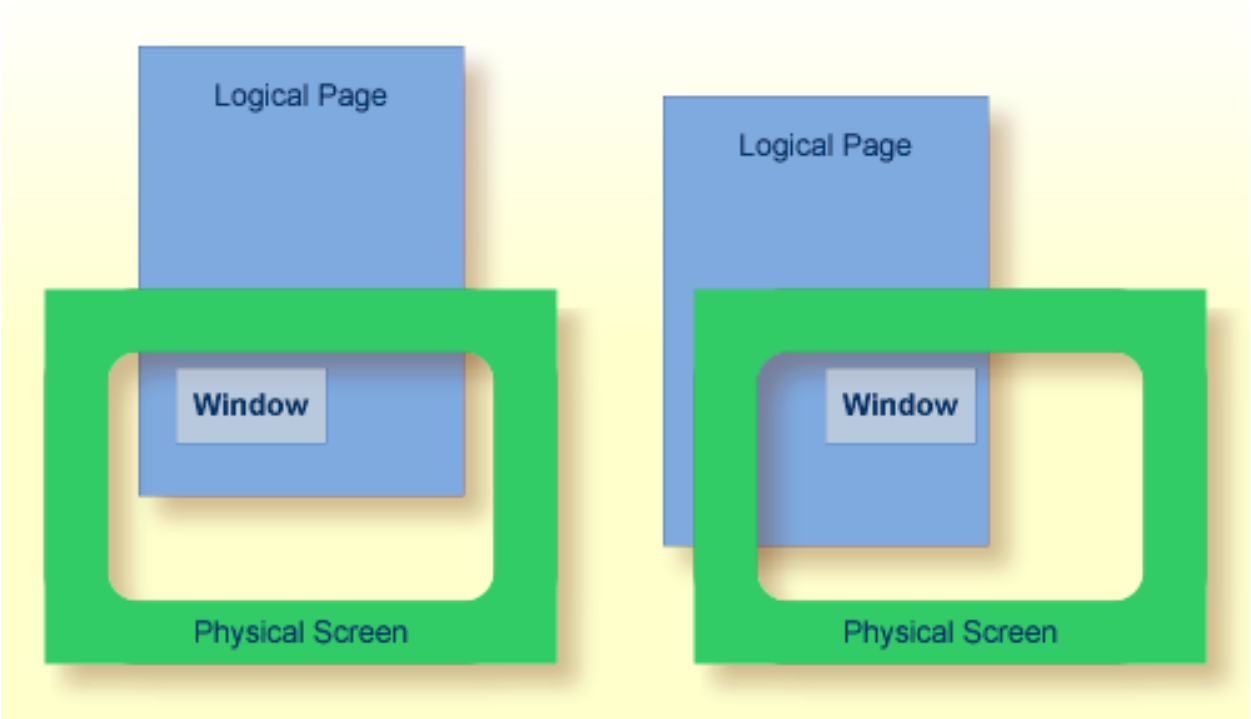
Any incorrect/impossible size or positioning commands will either be ignored or adjusted to the physical possibilities.

Command	Function
%WF	Switch on framing. The boundaries of the window will be indicated by a frame. If the window size is smaller than 4 lines by 13 columns, the frame will not be visible.
%WM	Switch off framing. The boundaries of the window will not be indicated by a frame. Switching off the frame does not change the size of the window (only the size of the page segment visible inside the window).
%WO	Suppress display of PF-key lines, message line and statistics line. This command only applies if the screen is a "real" window (that is, smaller than the physical screen). To cancel the effect of %WO, you issue %WO again (or %WD).
%WP	By default, the PF-key lines, the message line and the statistics line are displayed within a window. To display them on the screen outside the window, use %WP. To cancel the effect of %WP, you use %WD.
%WD	Cancels the effects of %WF, %WO and %WP (and of the TITLE option of the DEFINE WINDOW statement).
%WX	If there is a *COM field outside the window, this field is normally not write-protected. With %WX you can make it write-protected.
%WY	Cancels the effect of %WX.

Window Position on a Logical Page

The following window commands control the positioning of the window on the current logical page, that is, the current report/map produced by the Natural program for display. This logical page may be larger in size than the physical screen.

When you change the position of the window on the logical page, the size and position of the window on the physical screen will remain unchanged. In other words, the window is not moved over the page, but the page is moved "underneath" the window:



Unless specified differently by one of the following commands, the window will be placed at the top left corner of the logical page.

Command	Function
`%W*`	The position marked on the page by the cursor will be shifted to the top left corner of the window.
`%W`*`lll`*`,`*`ccc`	The position of the logical page determined by line number *`lll`*, column number *`ccc`* will be shifted to the top left corner of the window. Lines and columns are counted on the logical page.
`%W<`	Shift window left. The number of positions shifted is equal to the line size (horizontal extension) of the window.
`%W<<`	Shift window to leftmost position of page.
`%W<`*n*	Shift window left *n* positions (0 =< *n* =< logical line size).
`%W>`	Shift window right. The number of positions shifted is equal to the line size (horizontal extension) of the window.
`%W>>`	Shift window to rightmost position of page.
`%W>`*n*	Shift window right *n* positions (0 =< *n* =< logical line size).
`%W+`	Shift window down. The number of lines shifted is equal to the number of lines in the window. (*)
`%W++`	Shift window to bottom of page. (*)
`%W+`*n*	Shift window down *n* lines (0 =< *n* =< logical page size). (*)
`%W-`	Shift window up. The number of lines shifted is equal to the number of lines in the window.

Command	Function
%W--	Shift window to top of page.
%W-*n*	Shift window up *n* lines (0 =< *n* =< logical page size).
%WH	By default, the position of the window on the logical page is reset to *top left corner* after a screen I/O. %WH prevents the window position from being reset by the next I/O, that is, the set window position will be retained. %WH only applies to the next I/O.
%WS	Switch on the STAY option; that is, control will "stay" on the current page until the end of the page. If a page is not yet completely shown in vertical direction, a string of VVVV will appear in the message line. The window will be scrolled downward with every ENTER until the end of the logical page is reached. The next ENTER will cause control to be returned to the program. (This does not apply to pages created by an INPUT statement with input fields (session parameter AD=A or AD=M).)
%WN	Switch off the STAY option. When you press ENTER, control is returned to the program.

* The window can be shifted to the last non-blank line of the page at most.

Notes:

1. If you wish to use one of the above commands within a program to shift the window, assign the command to a function key (with a SET KEY statement).
2. If you wish to specify it with a SET CONTROL statement, this statement must be followed by a REINPUT statement (that is, it must be placed between the REINPUT statement and corresponding INPUT statement); otherwise Natural will not be able to uniquely identify the window to which the command is to be applied (and will ignore it).
3. As a rule, however, no SET CONTROL 'W' statement should be placed between an INPUT statement with WINDOW='*window-name*' option and the corresponding REINPUT statement.

Examples of Command Combinations

The various parameters to be specified with the %W command may also be combined with one another; for example:

%W<<--	Position window to top left corner of page.
%W>>++	Position window to bottom right corner of page.
%W++-	Display the next to last window segment of the page.
%W+3>6	Position window 3 lines down and 6 positions right on the page.
%W10+>	Position window to line 10 of the page, then 1 window down and 1 window right.

%WL40C10++-3	Define the window with a line size of 40 positions and a page size of 10 lines, move that window to the bottom of the page, then move it up 3 lines on the page.
%WL30C10B3/15--<<	Define the window with a line size of 30 positions, page size of 10 lines, position the window at line 3 column 15 on the physical screen and move this window to the top left corner of the page.
%WFS	Generate a frame around the window and set the STAY option on.

When you specify multiple parameters with the %W command, note that the maximum number of characters after the percent sign (%) is 24; any further characters will be ignored.

The parameters are evaluated in the sequence in which they are specified, so that different sequences of the same parameters may lead to different results.

%WA and %WZ - Save Screen Image before Window

Command	Function
%WA	With %WA, you activate a "save screen image before window" feature. When this feature is activated and a window is to be opened, all active screen data that will be overlaid by the window are saved. When the same window is moved, the saved screen image is reconstructed before the window is built up at the new location on the screen. In addition, it is possible to rebuild the saved images of multiple dependent windows whenever the calling window becomes active again. When the current INPUT statement uses a window, the screen image is stored before the window is output. Whenever the same INPUT statement is repeated, the current or all subsequent stored screen images are recovered and are written back to the screen. This feature makes it possible, for example, to use windows in a PC-like manner. For a given window, any number of dependent windows can be written to the screen. All these windows will disappear from the screen when the main input window is re-executed. The buffer contents (screen images) are deleted whenever Natural performs a full-screen I/O, when Natural returns to command mode (NEXT), after a LOGON command, or after the CLEAR key has been pressed.
%WZ	With %WZ, you deactivate a previously entered %WA command.

59 %X - Control of Infoline

- Infoline 143
- Statistics Line 143

```
%X [ [ I ] [ + ]
     [ S ] [ - ]
     B
     T
     nn
     -nn          ]
```

This terminal command controls the display of the Natural **statistics line** or **infoline**.

Application Programming Interface: `USR1005N`.

Command	Function
`%X+`	Switches the display of the statistics line/infoline on.
`%X-`	Switches the display of the statistics line/infoline off.
`%X`	Switches the display of the statistics line/infoline on and off (toggle switch).
`%XI+`	Displays the line in infoline mode.
`%XI-`	Displays the line in statistics display mode.
`%XI`	Switches between the two display modes.
`%XS+`	Displays additional statistics (see below).
`%XS-`	Returns to display of original statistics.
`%XS`	Switches between additional and original statistics.
`%XB`	Displays the statistics line/infoline in the bottom line of the screen.
`%XT`	Displays the statistics line/infoline in the top line of the screen.
`%X`*nn*	Displays the statistics line/infoline on line *nn* of the screen. If the line number *nn* is not within the current screen, the statistics line/infoline will not be displayed.
`%X-`*nn*	Displays the statistics line/infoline on the *nn*th line from the bottom of the screen. If the line number -*nn* is not within the current screen, the statistics line/infoline will not be displayed.

Infoline

Data can be written to the infoline by specifying the output destination `INFOLINE` with the `DEFINE PRINTER` statement. Only a single line can be written to the infoline. The infoline can be used to have status information displayed, for example, for debugging purposes; it can also be used as separator line (as defined by SAA standards).

Statistics Line

When the line is used in statistics display mode, the following statistical information is provided:

Command Option	Function
IO	Number of bytes transmitted to the screen during the previous screen operation.
AIO	Average number of bytes transmitted per terminal screen operation (since beginning of Natural session).
L	Logical line number of top line within the logical page shown in the current window.
C	Logical column number of the leftmost column within the logical page shown in the current window.
LS	Logical line size of current page (as defined by session parameters).
PS	Logical page size of current page (as defined by session parameters).
PLS	Physical line size of window.
PCS	Physical page size of window.
FLD	Number of fields generated in the last screen.
CLS	Number of program calls during the last terminal I/O.
ADA	Number of Adabas calls during the last terminal I/O.

Additional statistics displayed with `%XS+`:

Command Option	Function
MIN	Minimum duration (in seconds) of Natural activities between two terminal I/O operations in the current Natural session.
MAX	Maximum duration (in seconds) of Natural activities between two terminal I/O operations in the current Natural session.
AVR	Average duration (in seconds) of Natural activities between two terminal I/O operations in the current Natural session.
LST	Duration (in seconds) of Natural activities between the last but one and the last but two terminal I/O operations.

For further information, see *Statistics Line/Infoline - Terminal Command %X* (in the *Programming Guide*).

60 %Y - Control of PF-Key Lines

- Display Format of Function-Key Lines 146
- Single- and Double-Line Display 147
- Positioning of Function-Key Lines 147
- Range of Displayed Function Keys 147
- Intensified or Reverse Video Display of Function-Key Lines 148
- Coloring of Function-Key Lines 148
- Cursor Sensitivity 148

```
%Y [ { N } ]
     { S }
     { P }
     { W }
     { U }
     { A }
     { H }
     { B }
     { T }
     { [-]nn }
     { F }
     { L }
     { X }
     { I }
     { V }
     { C }
```

Alternatively, you may specify:

```
%Y=color-code [color-code] [color-code]
```

The terminal command `%Y` is used to control the display of the Natural PF-key lines.

Note: On graphical user interfaces, this command is ignored.

Application Programming Interface: `USR1005N`.

Display Format of Function-Key Lines

Command	Function
`%YN`	Displays the function-key lines in normal tabular Software AG format.
`%YS`	Displays the function-key lines in sequential format, and only showing those keys to which names have been assigned (`PF1=value`,`PF2=value`, etc.).
`%YP`	Displays the function-key lines in PC-like sequential format; corresponds to `%YS`, except that `Fn=` instead of `PFn=` is displayed before the names.
`%YW`	Corresponds to `%YP` - but applies only if the function-key lines are displayed within a window.
`%YU`	Cancels the effect of `%YW`.

Single- and Double-Line Display

Command	Function
%YA	All-line display. Displays both function-key lines.
%YH	Half display. Displays only one function-key line; for normal tabular display mode (%YN), this is the line with the function-key names; for the other two display modes (%YS and %YP), this is the upper line to be displayed.

Positioning of Function-Key Lines

Command	Function
%YB	Displays the function-key lines at the bottom of the screen.
%YT	Displays the function-key lines at the top of the screen.
%Y*nn*	Displays the function-key lines on line *nn* on the screen.
%Y-*nn*	Displays the function-key lines on the *nn*th line from the bottom of the screen. If the line number *nn* or -*nn* is not within the current screen, the function-key lines will not be displayed.

Range of Displayed Function Keys

Command	Function
%YF	Displays the first range of function keys (that is, usually 1 to 12).
%YL	Displays the last range of function keys (that is, usually 13 to 24).
%YX	Is used to toggle-switch between the two displays.

Intensified or Reverse Video Display of Function-Key Lines

Command	Function
%YI	Displays the function-key lines intensified. Enter %YI again to switch back from intensified to non-intensified display.
%YV	Displays the function-key lines in reverse video. Enter %YV again to switch back from reverse video display to normal display.

Coloring of Function-Key Lines

Command	Function
%Y=*color-code(s)*	Displays the function-key lines in the specified colors. The *color-codes* you can specify are the same as with the session parameter CD). You can specify up to three *color-codes*: the first *color-code* refers to the first function-key line (the one displaying the function-key numbers), the second *color-code* refers to the second function-key line (the one displaying the function-key names), and the third *color-code* refers to the background of both lines. For example, %Y=GRPIYE would cause the characters in the first line to be displayed in green, the characters in the second line in pink, and the background of both lines in yellow. **Note:** The third *color-code* will be ignored.

Cursor Sensitivity

Command	Function
%YC	This command makes the function-key lines cursor-sensitive. This means that they react like an action bar on a PC screen: the user just moves the cursor to the desired function-key number or name displayed and presses ENTER, and Natural will react as if the corresponding function key had been pressed. Enter %YC again to switch cursor-sensitivity off again (toggle switch). By using %YC in conjunction with tabular Software AG display format (%YN) and having only the function-key names displayed (%YH), you may equip your applications with very comfortable

Command	Function
	action bar processing: the user merely has to select a function name with the cursor and press ENTER, and the function will be executed.

To activate the display of the function-key lines in a program, you use the session parameter `KD=ON`.

For further information, see *Control of Function-Key Lines - Terminal Command %Y* (in the *Programming Guide*).

61 %Z - Clear Source Area

```
%Z
```

This command clears the contents of the Natural source area.

It can only be used with the `SET CONTROL` statement.