

Natural für Windows

Terminalkommandos

Version 6.3.8 für Windows

Februar 2010

Dieses Dokument gilt für Natural ab Version 6.3.8 für Windows.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuausgaben bekanntgegeben werden.

Copyright © 1992-2010 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, Vereinigte Staaten von Amerika, und/oder ihre Lizenzgeber..

Der Name Software AG, webMethods und alle Software AG Produktnamen sind entweder Warenzeichen oder eingetragene Warenzeichen der Software AG und/oder der Software AG USA, Inc und/oder ihrer Lizenzgeber. Andere hier erwähnte Unternehmens- und Produktnamen können Warenzeichen ihrer jeweiligen Eigentümer sein.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://documentation.softwareag.com/legal/> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Diese Software kann Teile von Drittanbieterprodukten enthalten. Die Hinweise zu den Urheberrechten und Lizenzbedingungen der Drittanbieter entnehmen Sie bitte den "License Texts, Copyright Notices and Disclaimers of Third Party Products". Dieses Dokument

ist Bestandteil der Produktdokumentation und befindet sich unter <http://documentation.softwareag.com/legal/> und/oder im Wurzelverzeichnis des lizenzierten Produkts.





Inhaltsverzeichnis

1	Terminalkommandos	1
2	Einführung in Terminalkommandos	3
	Wozu dienen Terminalkommandos?	4
	Ändern des Terminalkommando-Steuerzeichens	4
	Eingabe eines Terminalkommandos	4
	Verwendung von Terminalkommandos in Programmen	5
	Verwendete Begriffe	5
3	Terminalkommandos nach Funktionsgruppen	7
	Umsetzen von Klein- in Großbuchstaben	8
	Kopieren, Löschen	8
	Sprache, Meldungen, Fehlerbehandlung	8
	Bildschirm und Fenster-Verarbeitung	8
	Farben	9
	INPUT-Statement	9
	Statistikzeile	9
	Verschiedene	9
	Tastenbelegungen	10
4	Terminalkommando-Tastenbelegung	11
	Funktionstasten mit Terminalkommandos belegen	12
	CLEAR-Taste – Aktive Operation unterbrechen	12
5	%% und % – Aktive Operation unterbrechen	13
	%% im Online-Betrieb	14
	%% im Batch-Betrieb	14
	% im Online-Betrieb	15
	% im Batch-Betrieb	15
6	%* – Anzeige von Eingabezeichen unterdrücken	17
7	%<TECH – Technische Informationen anzeigen	19
8	%= – Zuordnen von Farben zu Feldern	21
9	%C – Seitenpuffer kopieren	23
10	%CS und %CC – Daten in den Stack bzw. in *COM kopieren	25
11	%E= – Fehlerbehandlung ein-/ausschalten	27
12	%FM – Freimodus für numerische Editiermasken	29
13	%H – Hardcopy-Ausgabe	31
14	%I= – Symbol für Ausgabefenster	33
15	%J – Helproutine aufrufen	35
16	%K und %KP – Simulieren von PF- und PA-Tasten	37
17	%L – Keine Umsetzung von Klein- in Großbuchstaben	39
18	%L= – Sprachcode	41
19	%M – Steuerung der Meldungszeile	43
	Positionierung der Meldungszeile	44
	Farbe der Meldungszeile	45
20	%N – Aktivieren des Non-Conversational-Modus	47
21	%Q – Map-Ausgabe im Batch-Betrieb unterdrücken	49

22 %QS – Gleichzeitige Ausgabe mehrerer Schirme	51
23 %R – INPUT-Statement wiederholen	53
24 %T – Cursor am Schirmanfang platzieren	55
25 %Tll/cc – Cursor in Zeile ll, Spalte cc platzieren	57
26 %T* – Cursor außerhalb des Fensters platzieren	59
27 %U – Umsetzen von Klein- in Großbuchstaben	61
28 %V – Steuerung des Print-Modus	63
29 %W – Window-Verarbeitung	65
Größe und Position des Fensters auf dem physischen Bildschirm	66
Position des Fensters auf der logischen Seite	69
%WA und %WZ – Bildschirm speichern vor Fenster	71
30 %X – Steuerung der Infoline	73
Infoline	74
31 %Z – Arbeitsbereich des Editors löschen	75

1 Terminalkommandos

Diese Dokumentation beschreibt die Natural-Terminalkommandos. Sie ist in die folgenden Abschnitte untergliedert:

	Einführung in Terminalkommandos	Wozu dienen Terminalkommandos? Ändern des Terminalkommando-Steuerzeichens, Eingabe eines Terminalkommandos, Verwendung von Terminalkommandos in Programmen, verwendete Begriffe und Hilfe-Informationen für Terminalkommandos.
	Terminalkommandos nach Funktionsgruppen	Liefert eine Übersicht über die nach Funktion eingeteilten Terminalkommandos.
	Terminalkommando-Tastenbelegungen	Belegung von Funktionstasten mit häufig benutzten Terminalkommandos, Informationen zur CLEAR-Taste.
	Terminalkommandos in alphabetischer Reihenfolge	Beschreibungen der Terminalkommandos in alphabetischer Reihenfolge.

2 Einführung in Terminalkommandos

- Wozu dienen Terminalkommandos? 4
- Ändern des Terminalkommando-Steuerzeichens 4
- Eingabe eines Terminalkommandos 4
- Verwendung von Terminalkommandos in Programmen 5
- Verwendete Begriffe 5

Wozu dienen Terminalkommandos?

Natural-Terminalkommandos werden benutzt, um z.B.:

- die Bildschirm-Anzeige und das Bildschirm-Format wie z.B. die Positionierung der PF-Tastenzeile und Meldungszeile und die Zuordnung der Farben einzurichten,
- Fehlersuch-Informationen zu der aktuellen Umgebung zu erhalten,
- eine aktuelle Natural-Operation zu unterbrechen.

Sie können ein Terminalkommando aufrufen, während gleichzeitig eine Anwendung ausgeführt wird. Außer bei den Natural-Eingabeaufforderungen können Terminalkommandos in einem alphanumerischen Eingabefeld eingegeben werden. Das erste Zeichen eines Terminalkommandos ist das Terminalkommando-Steuerzeichen, das durch Setzen eines Natural Session-Parameters angegeben werden kann. Das Standardsteuerzeichen ist das Prozentzeichen (%).

Eine vollständige Funktionsübersicht der Natural-Terminalkommandos erhalten Sie im Abschnitt [Terminalkommandos nach Funktionsgruppen](#).

Ändern des Terminalkommando-Steuerzeichens

Sie können ein anderes Sonderzeichen als Steuerzeichen definieren, und zwar mit dem Session-Parameter CF.

Wenn das Steuerzeichen mit diesem Parameter geändert wird, werden alle Terminalkommandos, die einer Funktionstaste zugeordnet sind, automatisch entsprechend angepasst.

Eingabe eines Terminalkommandos

Terminalkommandos können in einer Natural-Laufzeitumgebung benutzt werden. Für die Verwendung von Terminalkommandos gelten folgende Regeln:

- Sie können das Steuerzeichen in jedes ungeschützte Feld oder an jeder beliebigen Stelle eingeben, falls der Schirm nur geschützte Felder enthält.
- Wenn Sie nur das Steuerzeichen eingeben, wird ein Fenster angezeigt, in dem Sie ein Terminalkommando eingeben können.
- Falsch eingegebene Terminalkommandos werden ignoriert, allerdings erhalten Sie keine entsprechende Fehlermeldung.
- Wenn Sie Daten in ein nicht geschütztes Feld eingeben haben, bevor das Terminalkommando-fenster angezeigt wird, werden die Daten nicht verarbeitet.

Verwendung von Terminalkommandos in Programmen

Terminalkommandos können mit einem `SET CONTROL`-Statement auch innerhalb eines Programms eingegeben werden; jedoch wird dann das Steuerzeichen weggelassen.

Verwendete Begriffe

In den Beschreibungen mehrerer Terminalkommandos werden die Begriffe „Bildschirm“ und „Fenster“ verwendet, und zwar mit folgenden Bedeutungen:

Begriff	Bedeutung
Bildschirm	Je nach dem Betriebssystem, unter dem Natural läuft, ist mit „Bildschirm“ entweder der ganze Bildschirm als solcher oder das Betriebssystemfenster, in dem die Natural-Session läuft, bzw. das Natural-Hauptausgabefenster gemeint. Der Einfachheit halber wird jedoch in allen Fällen der Begriff „Bildschirm“ (oder „Schirm“) verwendet.
Fenster	Hiermit ist immer das Natural-Fenster (wie unter Terminalkommando <code>%W</code> erklärt) gemeint.

3

Terminalkommandos nach Funktionsgruppen

- Umsetzen von Klein- in Großbuchstaben 8
- Kopieren, Löschen 8
- Sprache, Meldungen, Fehlerbehandlung 8
- Bildschirm und Fenster-Verarbeitung 8
- Farben 9
- INPUT-Statement 9
- Statistikzeile 9
- Verschiedene 9
- Tastenbelegungen 10

Die folgenden Tabellen liefern eine Übersicht über die nach Funktion eingeteilten Terminalkommandos.

Umsetzen von Klein- in Großbuchstaben

Terminalkommando	Funktion
%L	Keine Umsetzung von Klein- in Großbuchstaben
%U	Umsetzen von Klein- in Großbuchstaben

Kopieren, Löschen

Terminalkommando	Funktion
%C	Seitenpuffer kopieren.
%CC	Daten in Systemvariable *COM kopieren.
%CS	Daten in den Stack kopieren.
%Z	Arbeitsbereich des Editors löschen.

Sprache, Meldungen, Fehlerbehandlung

Terminalkommando	Funktion
%E=	Fehlerbehandlung ein-/ausschalten
%L=	Sprachcode setzen
%M	Steuerung der Meldungszeile

Bildschirm und Fenster-Verarbeitung

Terminalkommando	Funktion
%K	Rahmen-Zeichen für Fenster.
%Knn, %KPn	Simulieren von PF- und PA-Tasten.
%N	Aktivieren des Non-Conversational-Modus.
%QS	Gleichzeitige Ausgabe mehrerer Schirme.
%T and %T11/cc	Cursor-Position setzen.

Terminalkommando	Funktion
%T*	Cursor außerhalb des Fensters platzieren.
%W	Natural-Window-Verarbeitung.
%*	Anzeige von Eingabezeichen unterdrücken. Im Batch-Betrieb das Ausdrucken der nachfolgenden Eingabedatenzeile unterdrücken

Farben

Terminalkommando	Funktion
%=	Zuordnung von Farben zu Feldern.

INPUT-Statement

Terminalkommando	Funktion
%FM	Aktivieren/Deaktivieren des Freimodus für Editiermasken.
%R	INPUT-Statement wiederholen.

Statistikzeile

Terminalkommando	Funktion
%X	Steuerung der Statistikzeile/Infoline.
%<TECH	Technische Informationen anzeigen.

Verschiedene

Terminalkommando	Funktion
%H	Hardcopy-Ausgabe.
%J	Helproutine aufrufen.
%Q	Map-Ausgabe im Batch-Betrieb unterdrücken.
%V	Steuerung des Print-Modus.
%% and %.	Unterbrechen der gerade aktiven Natural-Operation.

Tastenbelegungen

Taste	Funktion
CLEAR	Unterbrechen der gerade aktiven Natural-Operation.

4 Terminalkommando-Tastenbelegung

- Funktionstasten mit Terminalkommandos belegen 12
- CLEAR-Taste — Aktive Operation unterbrechen 12

Funktionstasten mit Terminalkommandos belegen

Zum Zwecke einer benutzerfreundlicheren Bedienung können Sie Funktionstasten mit häufig benutzten Terminalkommandos belegen.

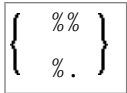
In einem Programm können Sie mit dem Statement `SET KEY` Funktionstasten mit Terminalkommandos belegen.

CLEAR-Taste — Aktive Operation unterbrechen

Das Drücken der Taste `CLEAR` (bzw. `LÖSCH`) hat dieselbe Wirkung wie das Terminalkommando `%%`. Es unterbricht die gerade aktive Natural-Operation.

5 %% und %. — Aktive Operation unterbrechen

- %% im Online-Betrieb 14
- %% im Batch-Betrieb 14
- %. im Online-Betrieb 15
- %. im Batch-Betrieb 15



Diese Terminalkommandos können verwendet werden, um die aktuelle Operation zu unterbrechen.



Anmerkung: Wenn der Profilparameter `ESCAPE` auf `OFF` gesetzt ist, werden die Terminalkommandos `%%` und `%.` ignoriert.

%% im Online-Betrieb

Wenn Sie `%%` in einem Feld auf dem Schirm eingeben, wird das Natural-Programm, das gerade ausgeführt wird, sofort abgebrochen, und Sie gelangen wieder in den Kommandoingabe-Modus.

Wenn Sie `%%` im Kommandoingabe-Modus eingeben, wird die Natural-Session beendet (entspricht dem Systemkommando `FIN`).

`%%` hat folgende Auswirkungen:

- Der Inhalt des Natural-Stacks wird gelöscht.
- Eine logische Datenbank-Transaktion, die gerade ausgeführt wird, wird abgebrochen (`BACKOUT`).
- Das im Editor befindliche Source-Programm wird nicht beeinflusst und bleibt erhalten.

%% im Batch-Betrieb

Im Batch-Betrieb können Sie mit `%%` Restart-Punkte in den Eingabedateien setzen und so die Synchronisation der Eingabedateien im Falle eines Fehlers sicherstellen.

Einfluss des Profilparameters `CC`

Kommando	Funktion
<code>CC=ON</code>	Wenn der Profilparameter <code>CC</code> gesetzt ist und bei der Kompilierung/Ausführung eines Natural-Programms im Batch-Betrieb ein Fehler auftritt, wird der Eingabedatenstrom für die Eingabedateien <code>SYNIN</code> und <code>OBJIN</code> bis zu der nächsten Zeile, die mit <code>%%</code> beginnt, gelöscht (wenn kein <code>%%</code> gefunden wird, wird er bis zum Dateiende gelöscht). Außerdem wird der Inhalt des Natural-Stacks gelöscht. Falls weitere Daten im Eingabestrom vorhanden sind, setzt Natural die Verarbeitung mit der Zeile nach <code>%%</code> fort.
<code>CC=OFF</code>	<code>%%</code> in den Eingabedaten wird ignoriert.

% . im Online-Betrieb

Im Online-Betrieb entspricht % . dem Kommando %%, außer dass der Inhalt des Natural-Stacks nicht gelöscht wird.

% . im Batch-Betrieb

Auf Großrechnern im Batch-Betrieb bewirkt % ., dass das Lesen der Eingabewerte für das gerade ausgeführte INPUT-Statement beendet wird.

6 %* — Anzeige von Eingabezeichen unterdrücken

`%*`

Die Verwendung dieses Terminalkommandos empfiehlt sich bei der Eingabe sensibler Daten (z.B. Passwort).

Dieses Terminalkommando bewirkt, dass sämtliche Daten, die auf dem aktuellen Schirm eingegeben werden, nicht angezeigt werden.

Wenn Sie `%*` mit einem `SET CONTROL`-Statement verwenden, werden die Eingaben sämtlicher Felder des nachfolgenden Schirms nicht angezeigt.

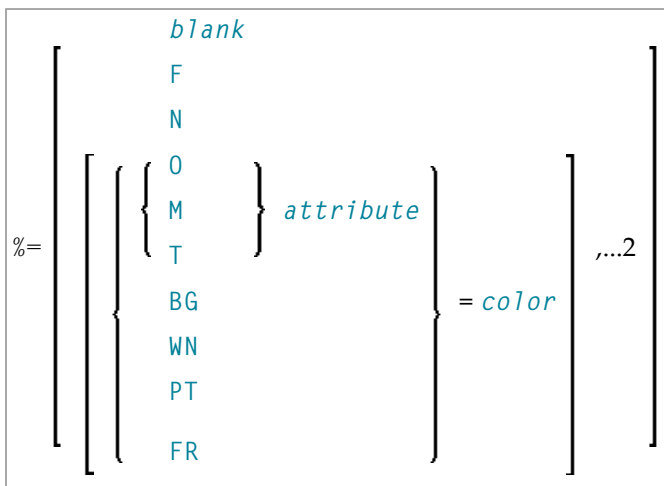
7 %<TECH — Technische Informationen anzeigen

`%<TECH`

Dieses Terminalkommando entspricht dem Systemkommando `TECH`.

8

%= — Zuordnen von Farben zu Feldern



Mit diesem Terminalkommando können Sie bestimmten Feldern bestimmte Farben zuweisen, und zwar für Programme, die ursprünglich ohne Berücksichtigung von Farbgebung geschrieben wurden.

Sie geben einen Feldtyp und/oder ein Feldattribut an sowie eine Farbe. Alle Felder dieses Typs/Attributs werden dann in dieser Farbe angezeigt.

Außerdem können Sie bestehende Farbzweisungen ändern, falls bereits vordefinierte Farbgebungen ungeeignet sind.

Kommando %=	Funktion
Allgemeine Einstellungen:	
Leerzeichen	Bestehende Farbzweisungen werden gelöscht.
F	Neu definierte Farbzweisungen gelten statt denen des Programms.
N	Im Programm definierte Farbzweisungen behalten ihre Gültigkeit.
Feldarten:	
O	Ausgabefelder (AD=0). Informationen zum Session-Parameter AD siehe <i>FFeldeingabe/ausgabe-Charakteristika</i> in der <i>Parameter-Referenz-Dokumentation</i> .
M	Eingabefelder; d.h. reine Eingabefelder (AD=A) sowie modifizierbare Felder (AD=M).
T	Textkonstanten.
Mögliche Feldattribute:	
B	Blinkend
C	Kursiv
D	Default, Standard
I	Intensiviert
U	Unterstrichen
V	Invers
Folgenden Bildschirmteilen kann ebenfalls eine Farbe zugewiesen werden:	
BG	Bildschirmhintergrund
WN	Vordergrund (d.h. Felder, für die keine Farbe definiert ist)
PT	Standard-Seitenüberschrift
FR	Rahmen eines Bildschirmfensters
Mögliche Farben:	
BL	Blau
GR	Grün
NE	Neutral, weiß
PI	Pink
RE	Rot
TU	Türkis
YE	Gelb

Beispiel:

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

Dieses Kommando bewirkt, dass alle intensiviert dargestellten Textfelder in Rot ausgegeben werden und alle blinkenden Ausgabefelder in Gelb.

9 %C — Seitenpuffer kopieren

`%C`

Mit diesem Terminalkommando kopieren Sie den Inhalt des Seitenpuffers (Page Buffer) in den Arbeitsbereich des Natural-Programm-Editors.

Die gegenwärtig von Natural angezeigte Seite wird in den Arbeitsbereich des Programm-Editors kopiert. Der Inhalt der Seite wird nach der letzten bereits im Editor befindlichen Sourcecode-Zeile eingefügt und kann dann mit dem Editor verändert werden.

Wollen Sie den schon im Editor befindlichen Sourcecode vorher löschen, können Sie dazu das Terminalkommando `%Z` verwenden.



Anmerkungen:

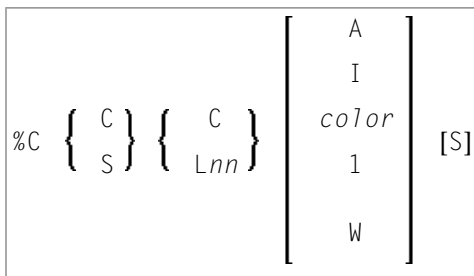
1. `%C` sollte nicht in einer Editor-Session verwendet werden. Änderungen des Sourcecode-Arbeitsbereichs außerhalb des Editor-Inhalts werden vom Editor ignoriert.
2. Der Seitenpuffer (die logische Natural-Ausgabe) ist nicht unbedingt mit dem am Bildschirm angezeigten Bildschirmpuffer identisch.
3. Wenn der Programm-Editor während der Ausführung von `%C` aktiv ist, zeigt er die neuen Zeilen sofort. Ist er nicht aktiv, werden die `%C`-Änderungen nicht angezeigt; dann muss der geänderte Editierbereich mit dem Systemkommando `EDIT` neu geladen werden.

Beispiel:

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

10 %CS und %CC — Daten in den Stack bzw. in *COM

kopieren



1 und W können nicht mit `Lnn` angegeben werden

Mit diesem Terminalkommando können Sie Teile des Bildschirms in den Natural-Stack (`%CS`) oder in die Systemvariable `*COM` (`%CC`) kopieren. Die geschützten Daten einer bestimmten Bildschirmzeile werden Feld für Feld kopiert (außer mit Option `A`; siehe im Folgenden).

Der zweite Buchstabe des Kommandos bestimmt, wohin die Daten kopiert werden:

- **%CC...**
schreibt die Daten in die Systemvariable `*COM`.
- **%CS...**
schreibt die Daten in den Natural-Stack. Die Daten werden oben auf dem Stack als Eingabedaten abgelegt (wie mit einem `STACK TOP DATA`-Statement).

Der dritte Buchstabe des Kommandos bestimmt, aus welcher Zeile Daten kopiert werden:

- **%CCC und %CSC**
kopieren alle geschützten Daten aus der Zeile, in der sich der Cursor befindet, und zwar ab dem Feld, in dem sich der Cursor befindet.

- **%CCL nn und %CSL nn**
kopieren alle geschützten Daten aus Zeile Nr. nn .

Darüber hinaus haben Sie folgende Optionen:

- **%C...A**
kopiert eine komplette Zeile, d.h. nicht nur die geschützten Daten, sondern auch die modifizierbaren Felder; die Zeile wird nicht Feld für Feld kopiert, sondern als ganzes (einschließlich der Feldattribute).
- **%C...I**
kopiert nur die intensiviert dargestellten Felder einer Zeile.
- **%C...color**
kopiert nur die in der angegebenen Farbe dargestellten Felder einer Zeile.
- **%C...C1**
kopiert nur ein Feld, und zwar das, in dem sich der Cursor befindet (ungeachtet seiner Attribute). (%C...L $nn1$ is nicht möglich.)
- **%C...CW**
kopiert nur das Wort (begrenzt durch Leer- oder Sonderzeichen in einem Feld), auf dem der Cursor steht. (%C...L nn W ist nicht möglich.)
- **%C...S**
bewirkt, dass Natural auf dem Schirm bleibt (Stay-Option), von dem die Daten kopiert werden, wenn das Kommando ausgeführt wird. Damit können Sie mehrere verschiedene Daten von einem Schirm kopieren, bevor Sie die Daten weiterverarbeiten.

Wenn Sie das Terminalkommando %C... direkt eingeben (oder es einer PF-Taste zuweisen), bezieht es sich auf den *physischen Bildschirm* innerhalb des aktiven Fensters.

11 %E= — Fehlerbehandlung ein-/ausschalten

%E=	{	ON	}
		OFF	

Mit dem Terminalkommando `%E=OFF` schalten Sie eine etwaige Fehlertransaktion sowie die `ON ERROR`-Verarbeitung aus; mit `%E=ON` werden die Fehlertransaktion und die `ON ERROR`-Verarbeitung wieder eingeschaltet.

Kommando	Funktion
<code>%E=OFF</code>	Schaltet eine etwaige Fehlertransaktion (wie durch die Systemvariable <code>*ERROR-TA</code> identifiziert oder in Natural Security definiert) sowie <code>ON ERROR</code> -Verarbeitung aus; auftretende Fehler werden dann von der normalen Natural-Fehlerbehandlung verarbeitet. Dies kann dazu eingesetzt werden, einen Fehler in der Fehlerbehandlung Ihrer Anwendung zu lokalisieren, falls es Ihnen aufgrund der Struktur der Anwendung mit verschiedenen Fehlerbehandlungsprozeduren auf verschiedenen Ebenen nicht möglich ist herauszufinden, wo genau ein Fehler ursprünglich auftrat.
<code>%E=ON</code>	Mit dem Terminalkommando <code>%E=ON</code> schalten Sie dann Fehlertransaktion und <code>ON ERROR</code> -Verarbeitung wieder ein.

Weitere Informationen siehe *Verwendung eines Fehlertransaktionsprogramm im Leitfaden zur Programmierung*

12 %FM — Freimodus für numerische Editiermasken

`%FM { + }
 { - }`

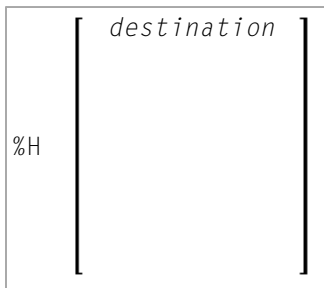
Mit diesem Terminalkommando können Sie den Freimodus für numerische Editiermasken aktivieren/deaktivieren. Es handelt sich hierbei um eine Sonderfunktion, anhand derer es möglich ist, dass Literale bei der Eingabe in ein Feld mit einer numerischen Editiermaske weggelassen werden können.

Kommando	Funktion
%FM-	Mit diesem Kommando schalten Sie den Freimodus für Editiermasken aus.
%FM+	Mit diesem Kommando schalten Sie den Freimodus für Editiermasken ein.

Die Voreinstellung beim Session-Start erfolgt über den Profilparameter `EMFM`.

Weitere Informationen zum Freimodus für Editiermasken siehe Abschnitt *Eingabe von Daten als Reaktion auf ein INPUT-Statement (Numerischer Editiermasken-Freimodus)* in der Beschreibung des INPUT-Statements in der *Statements*-Dokumentation.

13 %H — Hardcopy-Ausgabe



Dieses Kommando wird zum Drucken einer Hardcopy von Natural-Reports auf einem Drucker oder einem spezifischen Ausgabemedium (z.B. dem Arbeitsbereich des Editors) verwendet.

Standardmäßig bezieht sich ein %H-Kommando auf die aktuelle logische Seite (d.h. das aktuelle Bildschirmfenster ohne Meldungszeile, PF-Tastenleiste und Statistikzeile/Infoline).

Sie haben folgende Möglichkeiten:

Kommando	Funktion
%H	<p>Dieses Kommando aktiviert die Hardcopy-Ausgabe. Die Hardcopy-Ausgabe wird deaktiviert, wenn entweder ein INPUT-Statement mit änderbaren Feldern ausgeführt wird oder das Ende des Programms erreicht ist. Alle Seiten, die verarbeitet werden, wenn die Hardcopy-Ausgabe aktiv ist, werden an das Druck-Ausgabemedium weitergeleitet. Diese Seiten werden nicht auf dem Bildschirm angezeigt.</p> <p>Wenn %H in ein Eingabefeld eingegeben wird, wird nur die aktuelle Seite gedruckt.</p> <p>Für die Angabe des Druck-Ausgabemediums wird ein Fenster angezeigt, in dem alle verfügbaren Drucker und speziellen Ausgabemedien (z.B. SOURCE) aufgelistet werden. Von dieser Liste können Sie das Ausgabemedium auswählen, auf dem die Hardcopy ausgegeben werden soll.</p>
%H <i>destination</i>	<p>Wie %H, aber die Hardcopy wird auf dem angegebenen Druck-Ausgabemedium (<i>destination</i>) ausgegeben. <i>destination</i> kann 1 bis 8 Zeichen lang sein.</p>

Kommando	Funktion
%H!	<p>Das Statement SET CONTROL 'H!' bewirkt, dass die aktuelle Seite (d.h. die Seite vor der Ausführung des SET CONTROL-Statements) gedruckt wird. Das Kommando %H! bewirkt, dass die gerade auf dem Bildschirm angezeigte Seite gedruckt wird.</p> <p>Für die Angabe des Druck-Ausgabemediums wird ein Fenster angezeigt, in dem alle verfügbaren Drucker und speziellen Ausgabemedien (z.B. SOURCE) aufgelistet werden. Von dieser Liste können Sie das Ausgabemedium auswählen, auf dem die Hardcopy ausgegeben werden soll.</p>
%H!destination	Wie %H!, aber das Druck-Ausgabemedium (<i>destination</i>) wird mit angegeben. <i>destination</i> kann 1 bis 8 Zeichen lang sein.
%H=	<p>Dieses Kommando aktiviert/deaktiviert das Logging (Protokollieren) aller folgenden Seiten. Logging bedeutet, dass alle auf dem Bildschirm angezeigten Seiten auch an das Druckausgabemedium weitergeleitet werden. Diese Möglichkeit kann beispielsweise dazu genutzt werden, eine Abfolge von Ausgaben zu Verwaltungs-, Fehlerbeseitigungs- oder Schulungszwecken zu protokollieren.</p> <p>Für die Angabe des Druck-Ausgabemediums wird ein Fenster angezeigt, in dem alle verfügbaren Drucker und speziellen Ausgabemedien (z.B. SOURCE) aufgelistet werden. Von dieser Liste können Sie das Ausgabemedium auswählen, auf dem die Hardcopy ausgegeben werden soll.</p>
%H=destination	Wie %H=, aber mit dem Unterschied, dass mit %H=destination angegeben wird, auf welchem Hardcopy-Gerät alle logischen Ausgaben ausgegeben werden sollen. <i>destination</i> kann 1 bis 8 Zeichen lang sein.
%H#destination	Wie %Hdestination.
%H-	<p>Hält die Ausgabe der Hardcopy sofort an.</p> <p>Anmerkung: Wenn ein SET CONTROL 'H'-Statement ausgeführt wird, werden bereits in den Seitenpuffer geschriebene, aber noch nicht ausgegebene Daten nicht an den Drucker weitergeleitet. Um auch diese Daten zu drucken, müssen Sie vor dem SET CONTROL 'H'-Statement ein EJECT-Statement angeben.</p>



Anmerkung: Das EJECT-Statement wirkt sich nicht auf das %H-Kommando aus. Das %H-Kommando bewirkt bei Natural immer einen Seitenvorschub.

14 %I= — Symbol für Ausgabefenster

```
%I=icon-file-name.ICO
```

Dieses Terminalkommando können Sie zusammen mit dem `SET CONTROL`-Statement verwenden, um ein Symbol anzugeben, das an Stelle des Natural-Symbols für das Ausgabefenster verwendet werden soll. Dieses Symbol erscheint dann im Systemmenü des Ausgabefensters und in der Windows-Task-Liste.

Weitere Informationen siehe *Using Your Own Icon for the Output Window* in der *Operations*-Dokumentation.

15 %J — Helproutine aufrufen

```
%Jhelproutine
```

Mit diesem Terminalkommando können Sie eine interaktive Helproutine aufrufen.

Wenn %J verwendet wird, während eine von einem Systemkommando aufgerufene Funktion aktiv ist, sucht Natural nach der angegebenen Helproutine, und zwar in der aktiven Library des Systemkommandos oder in einer Library, die als Steplib für das Systemkommando definiert wurde.

16 %K und %KP — Simulieren von PF- und PA-Tasten

%K	{ <i>nn</i> <i>Pn</i> }
----	----------------------------------

Diese Terminalkommandos können verwendet werden, um die Funktionstasten (PF, ENTER bzw. EINGABE) und PA-Tasten zu simulieren.

Kommando	Funktion
%K <i>nn</i>	Simuliert die Funktionstaste <i>PFnn</i> (PF1 bis PF99). Dadurch können den PF-Tasten 1 bis 12 die Funktionen der PF-Tasten 13 bis 24 zugewiesen werden oder Funktionen von PF-Tasten, die auf der verwendeten Tastatur nicht vorhanden sind, aktiviert werden.
%K0	Simuliert die ENTER- bzw. EINGABE-Taste.
%KP <i>n</i>	Simuliert die Funktionstaste <i>PA<i>n</i></i> (PA1 bis PA3) siehe %K <i>nn</i> .

17 %L — Keine Umsetzung von Klein- in Großbuchstaben

`%L`

Dieses Terminalkommando bewirkt, dass bei Eingabedaten Kleinbuchstaben nicht automatisch in Großbuchstaben umgesetzt werden.

`%L` beeinflusst die interaktive Eingabe, z.B. in Zusammenhang mit einem `INPUT`-Statement. Das Kommando beeinflusst allerdings nicht die Eingaben vom Stack.

Siehe auch das Terminalkommando [%U](#).

18

%L= — Sprachcode

`%L=nn`

Mit dem Terminalkommando `%L=nn` können Sie den Sprachcode `nn` setzen, der von Natural verwendet werden soll.

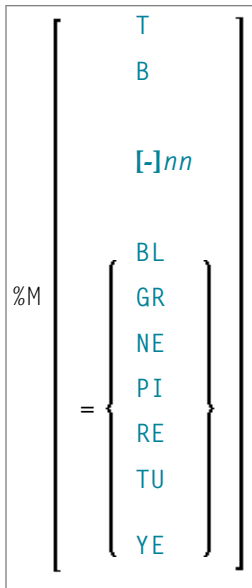
Eine Liste der möglichen Sprachcodes finden Sie bei der Beschreibung der Systemvariable `*LANGUAGE`, siehe *Systemvariablen*-Dokumentation.

Um den Sprachcode bereits zu Beginn der Session anzugeben, können Sie den Profilparameter `ULANG` benutzen.

19

%M — Steuerung der Meldungszeile

- Positionierung der Meldungszeile 44
- Farbe der Meldungszeile 45



Mit diesem Terminalkommando steuern Sie die Positionierung, den Schutz-Modus und die Farbe der Natural-Meldungszeile.

Dieses Kommando wird im Batch-Betrieb ignoriert.

Positionierung der Meldungszeile

Kommando	Funktion
%MT	Platziert die Meldungszeile am oberen Bildschirmrand (<i>Top</i>).
%MB	Platziert die Meldungszeile am unteren Bildschirmrand (<i>Bottom</i>).
%M	Bewirkt, dass die Position der Meldungszeile vom oberen zum unteren Bildschirmrand (oder umgekehrt) oder von Zeile <i>nn</i> zum unteren Bildschirmrand wechselt.
%M <i>nn</i>	Platziert die Meldungszeile in die <i>nn</i> -te Zeile des Bildschirms.
%M- <i>nn</i>	Platziert die Meldungszeile in die <i>nn</i> -te Zeile von unten auf dem Bildschirm. Ist die angegebene Zeilennummer <i>nn</i> oder <i>-nn</i> außerhalb des aktuellen Bildschirms, wird die Meldungszeile nicht angezeigt.

Farbe der Meldungszeile

Kommando	Funktion
<code>%M=color-code</code>	Zeigt die Meldungszeile in der angegebenen Farbe an (zur Bedeutung der einzelnen Farbcodes siehe Session-Parameter CD), wenn ein Farbbildschirm verwendet wird.

Siehe auch *Steuerung der Meldungszeile – Terminalkommando %M im Leitfaden zur Programmierung*.

20 %N — Aktivieren des Non-Conversational-Modus

`%N`

Dieses Terminalkommando wird mit einem `SET CONTROL`-Statement verwendet und bewirkt, dass der nächste Schirm angezeigt wird und die Verarbeitung anschließend sofort fortgesetzt wird, ohne dass es hierzu einer Benutzerreaktion bedarf; d.h. nachdem der Schirm angezeigt wurde, wird die Verarbeitung sofort fortgesetzt, ohne dass auf eine Benutzereingabe gewartet wird.

Dieses Kommando kann dazu verwendet werden, Informationen über den Stand des Programmablaufs an den Benutzer zu schicken.

21 %Q — Map-Ausgabe im Batch-Betrieb unterdrücken

%Q

Im Online-Betrieb wird %Q ignoriert.

SET CONTROL 'Q' bewirkt, dass das nächste INPUT-Statement *nicht* verarbeitet wird. Dies können Sie beispielsweise einsetzen, wenn nach Beendigung einer Helproutine die Verarbeitung bei der Rückkehr von der Helproutine zur Map sofort fortgesetzt werden soll, ohne dass der Benutzer EINGABE drücken muss.

22

%QS — Gleichzeitige Ausgabe mehrerer Schirme

%QS

Mit diesem Kommando können Sie mehrere Ausgabeschirme gleichzeitig anzeigen.

%QS bewirkt, dass die nächste Bildschirm-Eingabe/Ausgabe nicht ausgeführt wird. Der betreffende Ausgabeschirm wird intern zwischengespeichert und erst bei der darauffolgenden Eingabe/Ausgabe zusammen mit dem nächsten Schirm ausgegeben.

%QS ist also nur sinnvoll, wenn der zweite Ausgabeschirm ein Fenster ist, d.h. den ersten, mit %QS unterdrückten Schirm nicht vollständig überlagert.

Beispiel: So können Sie zum Beispiel die Ausgabe eines Schirmes A mit %QS unterdrücken; der nächste Schirm B ist ein Fenster, das Schirm A teilweise überlagert (beispielsweise ein Hilfefenster zu einem der Felder auf Schirm A); mit der nächsten Bildschirm-Eingabe/Ausgabe wird dann das Fenster B zusammen mit dem „darunterliegenden“ Schirm A gleichzeitig angezeigt.

Ein %QS-Kommando gilt nur für den jeweils nachfolgenden Schirm.



Anmerkung: Da sich durch %QS die Anzahl der Bildschirm-Eingaben/Ausgaben reduziert, spart %QS außerdem Verarbeitungszeit.

23 %R — INPUT-Statement wiederholen

`%R`

Dieses Terminalkommando bewirkt, dass ein `INPUT`-Statement wiederholt und der entsprechende Ausgabeschirm erneut generiert wird. Alle seit dem Beginn des `INPUT`-Statements generierten Ausgabedaten werden reproduziert.

24 %T — Cursor am Schirmanfang platzieren

`%T`

Dieses Kommando platziert den Cursor bei der nächsten Bildschirmausgabe in der oberen linken Bildschirmecke.

Dieses Kommando funktioniert nur dann, wenn sich in der oberen linken Bildschirmecke ein Eingabefeld (mit Feldattribut `AD=A` oder `AD=M`) befindet.

25

%TII/cc — Cursor in Zeile II, Spalte cc platzieren

`%TII/cc`

Dieses Kommando platziert den Cursor bei der nächsten Bildschirmausgabe in Zeile II, Spalte cc.

Zeilen und Spalten werden (beginnend mit 1) im physischen Bildschirmfenster gezählt, wobei die Funktionstastenleiste, die Statistikzeile/Infoline (falls aktiv) und die Meldungszeile nicht mitgezählt werden, d.h. %TII/1 positioniert immer an den Anfang der obersten Datenzeile im Fenster.

Bei manchen Geräten müssen Sie das Terminalkommando %T+ ausführen, bevor Sie den Cursor an jede gewünschte Stelle auf dem Bildschirm platzieren können.

%TII/cc funktioniert nur, wenn sich an der angegebenen Stelle ein Eingabefeld (mit Feldattribut AD=A oder AD=M) befindet.

26

%T* — Cursor außerhalb des Fensters platzieren

%T*

Ist ein Fenster aktiv, das keine Eingabefelder (Session-Parameter AD=A oder AD=M) enthält, steht der Cursor normalerweise in der oberen linken Ecke des Fensters.

Mit diesem Terminalkommando können Sie in diesem Fall den Cursor in eine *COM-Systemvariable außerhalb des Fensters platzieren.

Kommando	Funktion
%T*	Schaltet zwischen Cursor-Platzierung in Systemvariable *COM außerhalb des Fensters und Standard-Cursor-Platzierung innerhalb des Fensters hin und her.

%T* gilt jeweils nur für das nächste INPUT-Statement und muss *vor* dem betreffenden INPUT-Statement ausgeführt werden.

27 %U — Umsetzen von Klein- in Großbuchstaben

`%U`

Dieses Kommando bewirkt, dass Natural bei alphanumerischen Eingabedaten automatisch Kleinbuchstaben in Großbuchstaben umsetzt.

Dies ist die Standardeinstellung.

`%U` beeinflusst die interaktive Eingabe, z.B. in Zusammenhang mit einem `INPUT`-Statement. Das Kommando beeinflusst allerdings nicht die Eingaben vom Stack.

Die Umsetzung in Großbuchstaben wird nicht durchgeführt, wenn beim Eingabefeld `AD=W` angegeben wurde (siehe Abschnitt *Groß-/Kleinschreibung* in der Beschreibung des Session-Parameters `AD`).

Siehe auch Terminalkommando `%L`.

28

%V — Steuerung des Print-Modus

```
%V [ ON  
      OFF ]
```

Syntax-Beschreibung:

Kommando	Funktion
%VON	Setzt die Bildschirm-Leserichtung: Daten werden von rechts nach links gelesen.
%VOFF	Setzt die Bildschirm-Leserichtung: Daten werden von links nach rechts gelesen.
%V	Schaltet zwischen den Bildschirm-Leserichtungen von rechts nach links und umgekehrt hin und her. Ausführliche Informationen zur Benutzung des Terminalkommandos %V, finden Sie im Abschnitt <i>Bidirectional Language Support</i> in der <i>Unicode and Code Page Support</i> -Dokumentation.

Nähere Informationen zum Print-Modus siehe Session-Parameter PM.

29

%W — Window-Verarbeitung

- Größe und Position des Fensters auf dem physischen Bildschirm 66
- Position des Fensters auf der logischen Seite 69
- %WA und %WZ — Bildschirm speichern vor Fenster 71



Anmerkung: Es empfiehlt sich sehr, statt des %W-Kommandos das `DEFINE WINDOW`-Statement zu verwenden.

Ein Natural-„Window“ oder -Fenster ist der Ausschnitt einer von einem Natural-Programm erzeugten logischen Seite, der auf dem Bildschirm zu sehen ist.

Dieses Fenster kann mit dem Terminalkommando %W beeinflusst werden.

Das Kommando muss immer mit Parametern eingegeben werden, um die einzelnen im folgenden beschriebenen Funktionen auszuführen. Sie können mehrere Funktionen gleichzeitig ausführen, indem Sie mehrere Parameter mit einem %W-Kommando angeben. Zwischen Kommando und Parametern dürfen keine Trennzeichen oder Leerzeichen stehen.

Das Fenster ist ständig vorhanden, auch wenn Sie sich dessen nicht bewusst sind, da die Größe des Fensters — solange Sie sie nicht mit einem `DEFINE WINDOW`-Statement oder %W-Kommando ändern — mit der Größe Ihres Bildschirms identisch ist.

Weitere Informationen über Fensterverarbeitung siehe auch `DEFINE WINDOW`-Statement in der *Statements*-Dokumentation.

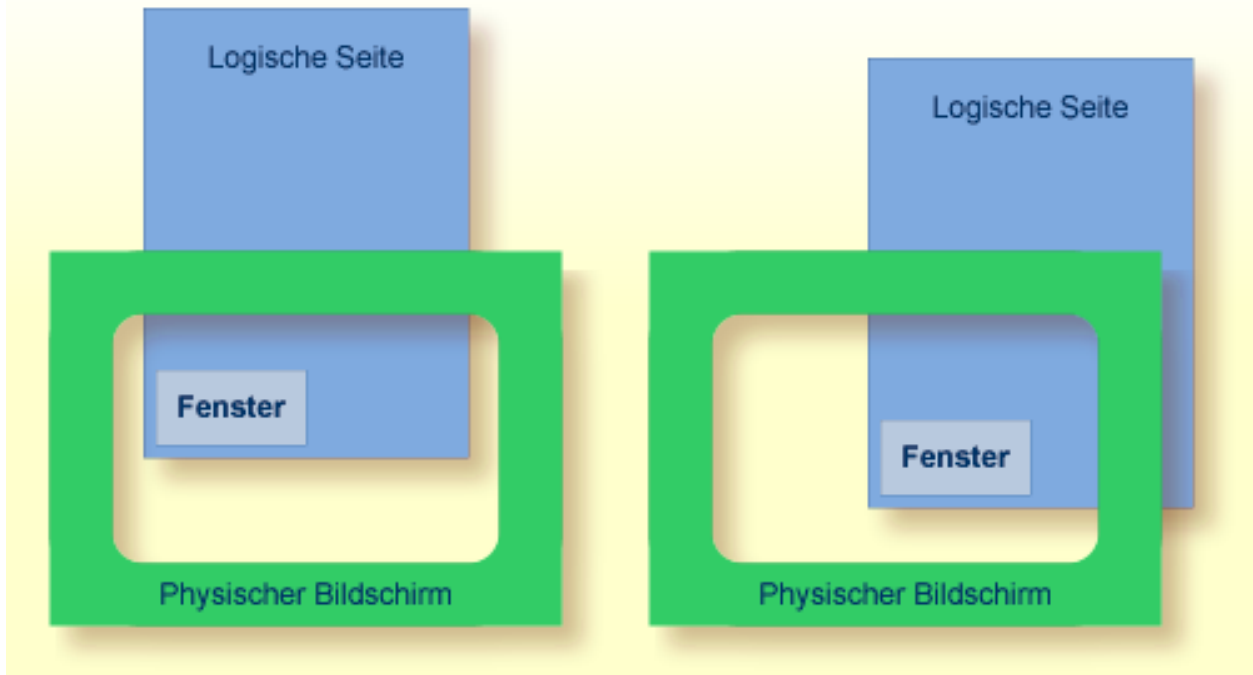
Es gibt zwei Arten von Window-Kommandos:

- Kommandos, mit denen Sie die Größe und Position des Fensters auf dem *physischen Bildschirm* Ihres Terminals verändern können.
- Kommandos, mit denen Sie die Position des Fensters auf der vom Programm erzeugten *logischen Seite* verändern können.

Größe und Position des Fensters auf dem physischen Bildschirm

Mit den folgenden Window-Kommandos bestimmen Sie die Größe und Position des Fensters auf dem physischen Bildschirm.

Wenn Sie die Position des Fensters auf dem physischen Bildschirm ändern, wird die Position des Fensters auf der logischen Seite dadurch nicht verändert:



Zu Informationen über mögliche Fenstergrößen siehe `DEFINE WINDOW`-Statement.

Kommando	Funktion
<code>%WB</code>	Die Fenstergröße (ausschließlich Rahmen) wird auf Bildschirmgröße gesetzt. Falls ein Rahmen definiert ist, ist dieser nicht sichtbar.
<code>%WB lll/ccc</code>	Die obere linke Ecke des Fensters wird in Zeile <i>lll</i> / Spalte <i>ccc</i> platziert (Zeilen und Spalten werden auf dem physischen Bildschirm gezählt). Die Größe des Fensters bleibt dabei gleich. Falls das Fenster zu groß ist, um an die angegebene Stelle platziert zu werden, wird es so nahe wie möglich an diese Stelle platziert.
<code>%WB0</code>	Das Fenster wird an die obere linke Bildschirmcke verschoben. Die Fenstergröße bleibt dabei unverändert.
<code>%W#</code>	Die obere linke Ecke des Fensters wird an die Cursor-Position platziert. Die Größe des Fensters bleibt dabei gleich. Falls das Fenster zu groß ist, um an die angegebene Stelle platziert zu werden, wird es so nahe wie möglich an diese Stelle platziert.
<code>%W?</code>	Die untere rechte Ecke des Fensters wird an die Cursor-Position platziert. Die obere linke Ecke des Fensters verschiebt sich dabei nicht, und die Größe des Fensters wird entsprechend angepasst.
<code>%WLnn</code>	Die Zeilenlänge (horizontale Ausdehnung) des Fensters (einschließlich Rahmen, falls angegeben) wird auf <i>nn</i> Stellen gesetzt. Wenn Sie <i>nn</i> nicht oder größer als auf den Bildschirm passend, angeben, wird die horizontale Ausdehnung auf größtmögliche Zeilenlänge gesetzt, d.h. bis zum rechten Bildschirmrand.

Kommando	Funktion
%WCnn	Die Spaltenlänge (vertikale Ausdehnung) des Fensters (einschließlich Rahmen, falls angegeben) wird auf nn Zeilen gesetzt. Wenn Sie nn nicht oder größer als auf den Bildschirm passend, angeben, wird die vertikale Ausdehnung auf größtmögliche Spaltenlänge gesetzt, d.h. bis zum unteren Bildschirmrand.

Zeilen- und Spaltenangaben beziehen sich auf die physische Gesamtgröße des Fensters (einschließlich Rahmen, falls angegeben), nicht auf die Größe dessen, was innerhalb des Fensters logisch sichtbar ist.

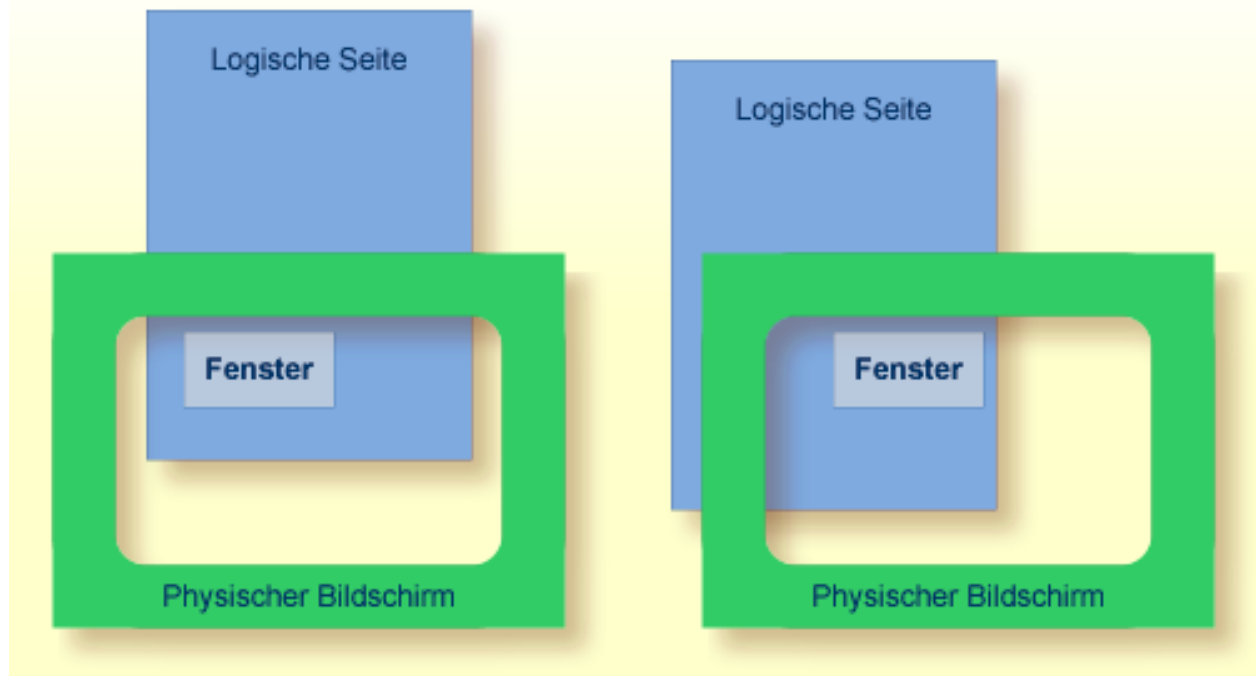
Wird ein Fenster nicht korrekt definiert, so wird ein solches Kommando entweder ignoriert oder die Fenstergröße und -position im Rahmen der physischen Möglichkeiten angepasst.

Kommando	Funktion
%WF	Schaltet die Rahmung (Frame) ein. Die Größe des Fensters wird durch einen Rahmen angezeigt. Falls das Fenster kleiner als 4 Zeilen mal 12 Spalten ist, ist der Rahmen nicht sichtbar.
%WM	Schaltet die Rahmung aus. Die Größe des Fensters wird nicht durch einen Rahmen angezeigt. Durch Ausschalten des Rahmens ändert sich die Größe des Fensters nicht (lediglich die Größe des im Fenster sichtbaren Seitenausschnitts).
%WO	Die Anzeige von Funktionstastenleiste, Meldungszeile und Statistikzeile wird unterdrückt. Dieses Kommando wirkt nur, falls das betreffende Fenster ein „echtes“ Fenster ist (d.h. kleiner als der physische Bildschirm ist). Um die Wirkung von %WO auszuschalten, geben Sie %WO noch einmal ein (oder %WD).
%WP	Standardmäßig werden die Funktionstastenleiste, die Meldungszeile und die Statistikzeile innerhalb eines Fensters angezeigt. Um sie auf dem Schirm außerhalb des Fensters anzuzeigen, verwenden Sie %WP. Um die Wirkung von %WP auszuschalten, verwenden Sie %WD.
%WD	Schaltet die Wirkung von %WF, %WO und %WP sowie der TITLE-Option des DEFINE WINDOW-Statements aus.
%WX	Befindet sich außerhalb des Fensters ein *COM-Feld, so ist dieses Feld normalerweise nicht schreibgeschützt. Mit %WX können Sie es schreibgeschützt machen.
%WY	Schaltet die Wirkung von %WX wieder aus.

Position des Fensters auf der logischen Seite

Mit den folgenden Window-Kommandos können Sie die Position des Fensters auf der aktuellen logischen Seite ändern, d.h. der vom Natural-Programm angezeigten Ausgabe oder Map. Die Größe dieser logischen Seite kann über die Größe Ihres Bildschirms hinausgehen.

Wenn Sie die Position des Fensters auf der logischen Seite ändern, bleibt die Position des Fensters auf dem physischen Bildschirm unverändert. Die logische Seite verschiebt sich also quasi „unter“ dem Fenster:



Normalerweise — wenn Sie die Position nicht durch eines der folgenden Kommandos ändern — befindet sich das Fenster an der oberen linken Ecke der logischen Seite, d.h. Sie sehen durch das Fenster den oberen linken Ausschnitt der logischen Seite.

Kommando	Funktion
<code>%W*</code>	Der durch den Cursor auf der Seite markierte Punkt wird an die obere linke Fensterecke geschoben.
<code>%W111,ccc</code>	Die Seite wird so verschoben, dass der durch Zeile 111/Spalte ccc definierte Punkt an der oberen linken Fensterecke ist. Zeilen und Spalten werden auf der logischen Seite gezählt.
<code>%W<</code>	Die Seite verschiebt sich um eine Fensterbreite nach links. Die Verschiebung entspricht der Zeilenbreite (horizontale Ausdehnung) des Fensters.

Kommando	Funktion
%W<<	Das Fenster wird bis zum linken Rand der Seite verschoben.
%W<n	Das Fenster wird um n Stellen nach links verschoben ($0 \leq n \leq$ Zeilenlänge der Seite).
%W>	Die Seite verschiebt sich um eine Fensterbreite nach rechts. Die Verschiebung entspricht der Zeilenbreite (horizontale Ausdehnung) des Fensters.
%W>>	Das Fenster wird bis zum rechten Rand der Seite verschoben.
%W>n	Das Fenster wird um n Stellen nach rechts verschoben ($0 \leq n \leq$ Zeilenlänge der Seite).
%W+	Die Seite verschiebt sich um eine Fensterhöhe nach unten. Die Anzahl der verschobenen Zeilen entspricht der Anzahl der Zeilen im Fenster. (*)
%W++	Das Fenster wird ans untere Ende der Seite verschoben. (*)
%W+n	Das Fenster wird um n Zeilen nach unten verschoben ($0 \leq n \leq$ Spaltenlänge der Seite). (*)
%W-	Die Seite verschiebt sich um eine Fensterhöhe nach oben. Die Anzahl der verschobenen Zeilen entspricht der Anzahl der Zeilen im Fenster.
%W--	Das Fenster wird ans obere Ende der Seite verschoben.
%W-n	Das Fenster wird um n Zeilen nach oben verschoben ($0 \leq n \leq$ Spaltenlänge der Seite).
%WH	Standardmäßig wird die Fenster-Position auf der logischen Seite bei einer Bildschirm-Eingabe/Ausgabe wieder auf die <i>obere linke Ecke</i> zurückgesetzt. %WH bewirkt, dass die gesetzte Fenster-Position beim nächsten Eingabe/Ausgabe nicht zurückgesetzt wird, sondern erhalten bleibt. %WH gilt jeweils nur für den nächsten Eingabe/Ausgabe.
%WS	Die STAY-Option ist eingeschaltet; d.h. die Kontrolle bleibt auf der aktuellen Seite, bis das Ende der Seite erreicht ist. Wenn eine Seite in vertikaler Richtung noch nicht vollständig angezeigt worden ist, wird Ihnen dies durch die Zeichenkette VVVV angezeigt. Jedesmal, wenn Sie EINGABE drücken, verschiebt sich die Seite um ein Fenster nach unten, bis das Seitenende erreicht ist. Mit der nächsten EINGABE wird dann die Kontrolle wieder an das Programm übergeben. Dies gilt nicht für Seiten, die über ein INPUT-Statement mit Eingabefeldern (mit Feldattribut AD=A oder AD=M) erzeugt wurden.)
%WN	Die STAY-Option ist ausgeschaltet. Wenn Sie EINGABE drücken, wird die Kontrolle an das Programm übergeben.

* Das Fenster kann höchstens bis zur letzten nicht leeren Zeile der Seite verschoben werden.



Anmerkungen:

1. Wenn Sie eines der obigen Kommandos zum Verschieben des Fensters in einem Programm verwenden wollen, weisen Sie das Kommando einer Funktionstaste zu (mit einem SET KEY-Statement).
2. Wenn Sie es mit einem SET CONTROL-Statement angeben wollen, muss diesem ein REINPUT-Statement folgen (d.h. es muss zwischen dem REINPUT-Statement und dem entsprechenden INPUT-Statement stehen); sonst kann Natural das Kommando nicht eindeutig einem bestimmten Fenster zuordnen (und ignoriert es).

3. Allerdings sollte zwischen einem `INPUT`-Statement mit Option `WINDOW='window-name'` und dem dazugehörigen `REINPUT`-Statement grundsätzlich kein `SET CONTROL 'W'`-Statement stehen.

Beispiele für Kommando-Kombinationen

Sie können die verschiedenen Parameter des `%W`-Kommandos auch miteinander kombinieren. Zum Beispiel:

<code>%W<<-</code>	Das Fenster wird an die obere linke Ecke der Seite geschoben.
<code>%W>>+</code>	Das Fenster wird an die untere rechte Ecke der Seite geschoben.
<code>%W+-</code>	Der (in vertikaler Richtung) vorletzte Fensterausschnitt der Seite wird gezeigt.
<code>%W+3>6</code>	Das Fenster verschiebt sich auf der Seite um 3 Zeilen nach unten und 6 Spalten nach rechts.
<code>%W10+></code>	Das Fenster wird an Zeile 10 der Seite geschoben, dann um ein Fenster nach unten und um ein Fenster nach rechts.
<code>%WL40C10+-3</code>	Das Fenster wird mit einer Zeilenlänge von 40 Stellen und einer Spaltenlänge von 10 Zeilen definiert, es wird ans untere Ende der logischen Seite und dann wieder 3 Zeilen nach oben geschoben.
<code>%WL30C10B3/15--<<</code>	Das Fenster wird mit einer Zeilenlänge von 30 Stellen und einer Spaltenlänge von 10 Zeilen definiert, es wird in Zeile 3 und Spalte 15 des physischen Bildschirms platziert und an die obere linke Ecke der logischen Seite geschoben.
<code>%WFS</code>	Um das Fenster wird ein Rahmen generiert, und die <code>STAY</code> -Option wird eingeschaltet.

Die Parameter werden der Reihe nach ausgewertet, so dass unterschiedliche Reihenfolgen derselben Parameter zu unterschiedlichen Ergebnissen führen können.

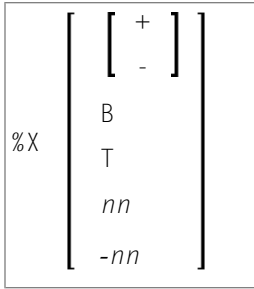
%WA und %WZ — Bildschirm speichern vor Fenster

Kommando	Funktion
<code>%WA</code>	<p>Mit <code>%WA</code> aktivieren Sie die Funktion „Bildschirm speichern vor Fenster“. Wenn diese Funktion aktiv ist und ein Fenster geöffnet wird, werden alle aktiven Bildschirmdaten, die von dem Fenster überlagert werden, vorher gespeichert. Wenn das Fenster verschoben wird, wird der gespeicherte Bildschirm rekonstruiert, bevor das Fenster an einer anderen Stelle auf dem Schirm aufgebaut wird. Es ist auch möglich, mehrere aufgerufene Fenster zu rekonstruieren, wenn das aufrufende Fenster wieder aktiv wird.</p> <p>Wenn das aktuelle <code>INPUT</code>-Statement ein Fenster verwendet, wird der Bildschirm gespeichert, bevor das Fenster ausgegeben wird. Wenn dasselbe <code>INPUT</code>-Statement wiederholt wird, wird der aktuelle Bildschirm bzw. alle anschließend gespeicherten Bildschirme rekonstruiert und wieder auf den Schirm zurückgeschrieben.</p> <p>Diese Funktion ermöglicht beispielsweise die Verwendung von Fenstern in PC-ähnlicher Form. Für ein bestimmtes Fenster können beliebig viele aufgerufene, „abhängige“ Fenster gespeichert</p>

Kommando	Funktion
	<p>werden. All diese Fenster verschwinden vom Schirm, wenn das Haupteingabefenster wiederausgeführt wird.</p> <p>Der Inhalt des Puffers (d.h. die gespeicherten Schirme) wird gelöscht, wenn Natural wieder einen ganzen Bildschirm ausgibt, wenn Natural in den Kommandomodus (NEXT) gelangt, nach einem LOGON-Kommando oder nach Drücken der CLEAR- bzw. LÖSCH-Taste.</p>
%WZ	Mit %WZ deaktivieren Sie ein vorhergegangenes %WA-Kommando.

30 %X — Steuerung der Infoline

- Infoline 74



Dieses Terminalkommando steuert die Anzeige der Natural-**Infoline**.

Kommando	Funktion
%X+	Schaltet die Anzeige der Infoline ein.
%X-	Schaltet die Anzeige der Infoline aus.
%X	Schaltet die Anzeige der Infoline ein bzw. wieder aus.
%XB	Zeigt die Infoline am unteren Bildschirmrand an (B ottom).
%XT	Zeigt die Infoline am oberen Bildschirmrand an (T op).
%Xnn	Zeigt die Infoline in der <i>nn</i> -ten Zeile des Schirms an. Ist die angegebene Zeilennummer <i>nn</i> außerhalb des aktuellen Schirms, wird die Infoline nicht angezeigt.
%X- <i>nn</i>	Zeigt die Infoline in der <i>nn</i> -ten Zeile von unten auf dem Schirm an. Ist die angegebene Zeilennummer <i>-nn</i> außerhalb des aktuellen Schirms, wird die Infoline nicht angezeigt.

Infoline

Daten können in die Infoline geschrieben werden, indem Sie `INFOLINE` als Ausgabeziel im `DEFINE PRINTER`-Statement angeben. Es kann nur eine einzelne Zeile in die Infoline geschrieben werden. Die Infoline kann zur Anzeige von Status-Informationen verwendet werden, zum Beispiel bei der Fehlersuche; sie kann auch als Trennlinie (Separator Line; wie in den SAA-Standards definiert) verwendet werden.

31 %Z — Arbeitsbereich des Editors löschen

`%Z`

Dieses Kommando löscht den Inhalt des Editor-Arbeitsbereichs.

Dieses Kommando kann nur mit einem `SET CONTROL`-Statement verwendet werden.

