

Natural for Mainframes

Terminalkommandos

Version 9.2.3

Juni 2025

Dieses Dokument gilt für Natural for Mainframes ab Version 9.2.3.

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

Copyright © 1979-2025 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein.

Nähere Informationen zu den Patenten und Marken der Software AG und ihrer Tochtergesellschaften befinden sich unter <http://documentation.softwareag.com/legal/>.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt "License Texts, Copyright Notices and Disclaimers of Third Party Products" entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices, abrufbar unter dem Abschnitt "License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products". Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

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

Dokument-ID: NATMF-NNATTCOM-923-20250606DE

Inhaltsverzeichnis

Vorwort	vii
1 Über diese Dokumentation	1
Dokumentationskonventionen	2
Online-Informationen und Support	2
Datenschutz	3
2 Einführung in Terminalkommandos	5
Wozu dienen Terminalkommandos?	6
Ändern des Terminalkommando-Steuerzeichens	6
Eingabe eines Terminalkommandos	6
Verwendung von Terminalkommandos in Programmen	7
Verwendete Begriffe	7
%? - Hilfe-Informationen für Terminalkommandos	7
3 Terminalkommandos nach Funktionsgruppen	9
Umsetzen von Klein- in Großbuchstaben	10
Kopieren, Löschen	10
Sprache, Meldungen, Fehlerbehandlung	10
Bildschirme, Terminal- und Fenster-Verarbeitung	11
Farben, Rahmung (Outlining)	11
INPUT-Statement, Stack	11
Bildschirme aufzeichnen	12
Statistikzeile und Trace-Funktion	12
Verschiedene	12
Tastenbelegungen	13
4 Terminalkommando-Tastenbelegung	15
Funktionstasten mit Terminalkommandos belegen	16
CLEAR-Taste — Aktive Operation unterbrechen	16
CTRL+D-Taste — Aktive Operation unterbrechen	16
RESET+ENTER-Taste — Aktive Schleife abbrechen	17
5 % — Fortsetzungsanzeiger für INPUT im Batch	19
6 %% und %. — Aktive Operation unterbrechen	21
%% im Online-Betrieb	22
%% im Batch-Betrieb	22
%. im Online-Betrieb	23
%. im Batch-Betrieb	23
7 %* — Anzeige von Eingabezeichen unterdrücken	25
8 %.P — Obersten Stack-Eintrag löschen	27
9 %.S — Stack-Daten lesen ohne zu löschen	29
10 %/ — End-of-File	31
11 %+ und %- — Natural Connection ein-/ausschalten	33
12 %<TECH — Technische Informationen anzeigen	35
13 %<TEST — Debugging-Funktion aufrufen	37
14 %= — Zuordnen von Farben zu Feldern	39
15 %A — Ausführen einer Bildschirm-Aufzeichnung	41

16 %B — Aktivieren/Deaktivieren des Bildschirm-Aufzeichnungsvorgangs	43
17 %B= — Library für Bildschirm-Aufzeichnung angeben	45
18 %C — Seitenpuffer kopieren	47
19 %CS und %CC — Daten in den Stack bzw. in *COM kopieren	49
20 %D — Aktivieren des Keyword/Delimiter-Modus	51
21 %D= — Outlining (Rahmung) steuern	53
22 %DUE — Speicherauszug bei spezifischem Fehler	55
23 %E — Aufgezeichnete Bildschirme anzeigen	57
24 %E= — Fehlerbehandlung ein-/ausschalten	59
25 %F — Aktivieren des Forms/Screen-Modus	61
26 %F= — Zeichen für Bildschirmfenster-Rahmen	63
27 %FM — Freimodus für numerische Editiermasken	65
28 %G — Wiedergabemodus für eine Bildschirm-Aufzeichnung	67
29 %H — Hardcopy-Ausgabe	69
30 %I — Aktuellen Bildschirm aufzeichnen	71
31 %J — Helproutine aufrufen	73
32 %K und %KP — Simulieren von PF- und PA-Tasten	75
33 %L — Keine Umsetzung von Klein- in Großbuchstaben	77
34 %L= — Sprachcode	79
35 %M — Steuerung der Meldungszeile	81
Positionierung der Meldungszeile	82
Schutz der Meldungszeile	83
Farbe der Meldungszeile	83
36 %MSGSF — Anzeigeformat von Systemfehlermeldungen	85
37 %N — Aktivieren des Non-Conversational-Modus	87
38 %O — Beenden der Aufzeichnung von Bildschirmen	89
39 %P — Aufzeichnung für nachfolgende Bildschirme aktivieren	91
40 %P= — CALL-Optionen	93
%P=S — Standard-Linkage für Aufruf	94
%P=SC — Standard-Linkage-Aufruf mit Simulation von EXEC CICS LINK	95
%P=SQ — Standard-Linkage für Aufruf von quasi-reentranten Subprogrammen	95
%P=U — Separate Arbeitseinheit für aufgerufenes Programm	96
%P=UT — Separate Arbeitseinheit für aufgerufenes Programm, Transaktions-Backout tolerieren	96
%P=V — Ausspeichern bei Aufruf	97
%P=C — Übergabe von Parameterwerten statt Adresse	97
%P=CC — Parameterwerte in einem CICS-Container übergeben	99
%P=L — Aufruf des dynamischen Hauptprogramms LE	99
%P=LS — Aufruf von LE-Subroutine	99
%P=I — Übergabe von PCB-Adressen an Aufgerufenen	100
41 %Q — Map-Ausgabe im Batch-Betrieb unterdrücken	101
42 %QO — Pseudo-konversationelle Ausgabe unterdrücken	103
43 %QS — Gleichzeitige Ausgabe mehrerer Schirme	105
44 %R — INPUT-Statement wiederholen	107

45 %<RDC+ und %<RDC- - Trace Recording aktivieren/deaktivieren	109
46 %RM — Schreibschutz von lichtstift-sensitiven Feldern	111
47 %RN — Bildschirmdaten-Komprimierung unterdrücken	113
48 %RO — Bildschirm-Optimierung ein-/ausschalten	115
49 %S — Fortsetzen der Aufzeichnung von Bildschirmen	117
50 %T — Cursor am Schirmanfang platzieren	119
51 %Tll/cc — Cursor in Zeile ll, Spalte cc platzieren	121
52 %T* — Cursor außerhalb des Fensters platzieren	123
53 %T= — Terminalspezifische Converter-Routine	125
54 %TRE — Externe Trace-Funktion aktivieren/deaktivieren	127
55 %TRI — Interne Trace-Funktion aktivieren/deaktivieren	129
56 %U — Umsetzen von Klein- in Großbuchstaben	131
57 %V — Steuerung des Print-Modus	133
58 %W — Window-Verarbeitung	135
Größe und Position des Fensters auf dem physischen Bildschirm	136
Position des Fensters auf der logischen Seite	139
%WA und %WZ — Bildschirm speichern vor Fenster	141
59 %X — Steuerung der Statistikzeile/Infoline	143
Infoline	145
Statistikzeile	145
60 %Y — Steuerung der PF-Tastenleiste	147
Anzeigeformat der Funktionstastenleiste	148
Ein- oder zweizeilige Anzeige	149
Positionierung der Funktionstastenleiste	149
Bereich der angezeigten Funktionstasten	149
Intensivierte bzw. inverse Anzeige der Funktionstastenleiste	150
Farbe der Funktionstastenleiste	150
Cursor-Sensitivität	150
61 %Z — Arbeitsbereich des Editors löschen	153
Stichwortverzeichnis	155

Vorwort

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, zu CTRL+D-Tasten und RESET+ENTER-Tasten.
Terminalkommandos in alphabetischer Reihenfolge	Beschreibungen der Terminalkommandos in alphabetischer Reihenfolge.

1 Über diese Dokumentation

■ Dokumentationskonventionen	2
■ Online-Informationen und Support	2
■ Datenschutz	3

Dokumentationskonventionen

Konvention	Beschreibung
Fettschrift	>Kennzeichnet Elemente auf einem Bildschirm.
Nichtproportionale Schrift	Kennzeichnet Namen und Orte von Diensten im Format <i>Ordner.Unterordner.Dienst</i> , Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
<i>Kursivschrift</i>	Kennzeichnet: Variablen, für die Sie situations- oder umgebungsspezifische Werte angeben müssen. Neue Begriffe, wenn sie erstmals im Text auftreten. Verweise auf andere Dokumentationsquellen.
Nichtproportionale Schrift	Kennzeichnet: Text, den Sie eingeben müssen. Meldungen, die vom System angezeigt werden. Programmcode.
{ }	Zeigt eine Reihe von Auswahlmöglichkeiten an, von denen Sie eine auswählen müssen. Geben Sie nur die innerhalb der geschweiften Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole { } ein.
	Trennt zwei sich gegenseitig ausschließende Auswahlmöglichkeiten in einer Syntaxzeile voneinander ab. Geben Sie eine der Auswahlmöglichkeiten ein. Geben Sie nicht das Symbol ein.
[]	Zeigt eine oder mehrere Optionen an. Geben Sie nur die innerhalb der eckigen Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole [] ein.
...	Zeigt an, dass Sie mehrere Auswahlmöglichkeiten desselben Typs eingeben können. Geben Sie nur die Informationen ein. Geben Sie nicht die drei Auslassungspunkte (...) ein.

Online-Informationen und Support

Produktdokumentation

Sie finden die Produktdokumentation auf unserer Dokumentationswebsite unter <https://documentation.softwareag.com>.

Zusätzlich können Sie auch über <https://www.softwareag.cloud> auf die Dokumentation für die Cloud-Produkte zugreifen. Navigieren Sie zum gewünschten Produkt und gehen Sie dann, je nach Produkt, zu „Developer Center“, „User Center“ oder „Documentation“.

Produktschulungen

Sie finden hilfreiches Produktschulungsmaterial auf unserem Lernportal unter <https://knowledge.softwareag.com>.

Tech Community

Auf der Website unserer Tech Community unter <https://techcommunity.softwareag.com> können Sie mit Experten der Software AG zusammenarbeiten. Von hier aus können Sie zum Beispiel:

- Unsere umfangreiche Wissensdatenbank durchsuchen.
- In unseren Diskussionsforen Fragen stellen und Antworten finden.
- Die neuesten Nachrichten und Ankündigungen der Software AG lesen.
- Unsere Communities erkunden.
- Unsere öffentlichen Repositories auf GitHub and Docker unter <https://github.com/softwareag> und <https://hub.docker.com/publishers/softwareag> besuchen und weitere Ressourcen der Software AG entdecken.

Produktsupport

Support für die Produkte der Software AG steht lizenzierten Kunden über unser Empower-Portal unter <https://empower.softwareag.com> zur Verfügung. Für viele Dienstleistungen auf diesem Portal benötigen Sie ein Konto. Wenn Sie noch keines haben, dann können Sie es unter <https://empower.softwareag.com/register> beantragen. Sobald Sie ein Konto haben, können Sie zum Beispiel:

- Produkte, Aktualisierungen und Programmkorrekturen herunterladen.
- Das Knowledge Center nach technischen Informationen und Tipps durchsuchen.
- Frühwarnungen und kritische Alarmer abonnieren.
- Supportfälle öffnen und aktualisieren.
- Anfragen für neue Produktmerkmale einreichen.

Datenschutz

Die Produkte der Software AG stellen Funktionen zur Verarbeitung von personenbezogenen Daten gemäß der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union zur Verfügung. Gegebenenfalls sind in der betreffenden Systemverwaltungsdokumentation entsprechende Schritte dokumentiert.

2 Einführung in Terminalkommandos

■ Wozu dienen Terminalkommandos?	6
■ Ändern des Terminalkommando-Steuerzeichens	6
■ Eingabe eines Terminalkommandos	6
■ Verwendung von Terminalkommandos in Programmen	7
■ Verwendete Begriffe	7
■ %? - Hilfe-Informationen für Terminalkommandos	7

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 ein Terminalkommando in jedem ungeschützten Feld auf dem Schirm eingeben (einschließlich der Meldungszeile, falls diese ungeschützt ist).
- Ein Terminalkommando muss in einem einzigen Feld eingegeben werden, mit Ausnahme des Steuerzeichens, das in ein vorangehendes Feld eingegeben werden kann.
- Es empfiehlt sich, nach einem Terminalkommando ein Leerzeichen einzugeben oder den übrigen Inhalt des Feldes, in das Sie das Kommando eingeben, zu löschen. Sonst kann es vorkommen, dass Natural den Feldinhalt irrtümlicherweise als Bestandteil des Terminalkommandos interpretiert.

- Sie können immer nur ein Terminalkommando zur selben Zeit ausführen. Geben Sie aber mehrere Terminalkommandos auf einmal ein, dann wird nur das erste ausgeführt; alle nachfolgenden werden ignoriert.
- Falsch eingegebene Terminalkommandos werden ignoriert, allerdings erhalten Sie keine entsprechende Fehlermeldung.
- Wenn Sie ein Terminalkommando und gleichzeitig in anderen Feldern auf dem Bildschirm Daten eingeben, wird nur das Terminalkommando ausgeführt; die Daten werden 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.

%? - Hilfe-Informationen für Terminalkommandos

`%?`

Mit diesem Terminalkommando erhalten Sie Hilfe-Informationen für die Terminalkommandos von Natural.

3

Terminalkommandos nach Funktionsgruppen

■ Umsetzen von Klein- in Großbuchstaben	10
■ Kopieren, Löschen	10
■ Sprache, Meldungen, Fehlerbehandlung	10
■ Bildschirme, Terminal- und Fenster-Verarbeitung	11
■ Farben, Rahmung (Outlining)	11
■ INPUT-Statement, Stack	11
■ Bildschirme aufzeichnen	12
■ Statistikzeile und Trace-Funktion	12
■ Verschiedene	12
■ Tastenbelegungen	13

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
%MSGSF	Anzeigeformat von Systemfehlermeldungen

Bildschirme, Terminal- und Fenster-Verarbeitung

Terminalkommando	Funktion
%F=	Rahmen-Zeichen für Fenster.
%K	Rahmen-Zeichen für Fenster.
%N	Aktivieren des Non-Conversational-Modus.
%QS	Gleichzeitige Ausgabe mehrerer Schirme.
%RM	Schreibschutz von lichtstift-sensitiven Feldern.
%RN	Bildschirmdaten-Komprimierung unterdrücken.
%RO	Bildschirm-Optimierung ein-/ausschalten.
%T and %T11/cc	Cursor-Position setzen.
%T*	Cursor außerhalb des Fensters platzieren.
%W	Natural-Window-Verarbeitung.
%Y	Steuerung der PF-Tastenzeilen.
%*	Anzeige von Eingabezeichen unterdrücken. Im Batch-Betrieb das Ausdrucken der nachfolgenden Eingabedatenzeile unterdrücken

Farben, Rahmung (Outlining)

Terminalkommando	Funktion
%D=	Rahmung (Outlining) steuern.
%=	Zuordnung von Farben zu Feldern.

INPUT-Statement, Stack

Terminalkommando	Funktion
%	Fortsetzungsanzeiger für INPUT im Batch.
%D	Aktivieren des Keyword/Delimiter-Modus.
%F	Aktivieren des Forms/Screen-Modus.
%FM	Aktivieren/Deaktivieren des Freimodus für Editiermasken.
%R	INPUT-Statement wiederholen.
%/	Erzwingen einer End-of-File-Bedingung für INPUT im Batch-Betrieb.
%.P	Obersten Stack-Eintrag löschen.

Terminalkommando	Funktion
%S	Stack-Daten lesen ohne zu löschen.

Bildschirme aufzeichnen

Terminalkommando	Funktion
%A	Ausführen einer Bildschirm-Aufzeichnung.
%B	Aktivieren/Deaktivieren des Bildschirm-Aufzeichnungsvorgangs.
%B=	Library für Bildschirm-Aufzeichnung angeben.
%E	Aufgezeichnete Bildschirme anzeigen.
%G	Wiedergabemodus für eine Bildschirm-Aufzeichnung setzen.
%I	Aktuellen Bildschirm aufzeichnen.
%O	Beenden der Aufzeichnung von Bildschirmen.
%P	Aufzeichnung für nachfolgende Bildschirme aktivieren.
%S	Fortsetzen der Aufzeichnung von Bildschirmen.

Statistikzeile und Trace-Funktion

Terminalkommando	Funktion
%TRE	Externe Trace-Funktion aktivieren/deaktivieren.
%TRI	Interne Trace-Funktion aktivieren/deaktivieren.
%X	Steuerung der Statistikzeile/Infoline.
%<TECH	Technische Informationen anzeigen.
%<TEST	Debugging-Funktion aufrufen.

Verschiedene

Terminalkommando	Funktion
%H	Hardcopy-Ausgabe.
%J	Helproutine aufrufen.
%P=	CALL-Optionen.
%Q	Map-Ausgabe im Batch-Betrieb unterdrücken.
%QO	Pseudo-konversationelle Ausgabe unterdrücken.

Terminalkommando	Funktion
%T=	Terminalspezifische Converter-Routine aktivieren.
%V	Steuerung des Print-Modus.
%?	Hilfe-Informationen für Terminalkommandos.
%% and %.	Unterbrechen der gerade aktiven Natural-Operation.
%+ and %-	Natural Connection ein/ausschalten.

Tastenbelegungen

Taste	Funktion
CLEAR	Unterbrechen der gerade aktiven Natural-Operation; Unterbrechen einer Aufzeichnung (Recording).
CTRL+D	Unterbrechen der gerade aktiven Natural-Operation.
RESET+ENTER	Beenden der aktiven Verarbeitungsschleife.

4 Terminalkommando-Tastenbelegung

■ Funktionstasten mit Terminalkommandos belegen	16
■ CLEAR-Taste — Aktive Operation unterbrechen	16
■ CTRL+D-Taste — Aktive Operation unterbrechen	16
■ RESET+ENTER-Taste — Aktive Schleife abbrechen	17

Funktionstasten mit Terminalkommandos belegen

Zum Zwecke einer benutzerfreundlicheren Bedienung können Sie Funktionstasten mit häufig benutzten Terminalkommandos belegen. Es stehen die folgenden Methoden zur Verfügung:

- In einem Programm können Sie mit dem Statement `SET KEY` Funktionstasten mit Terminalkommandos belegen.
- In der Programmier-Umgebung können Sie das Systemkommando `KEY` benutzen, um eine Funktionstaste mit einem Terminalkommando zu belegen.
- Funktionstastenbelegungen können auch über den Profilparameter `KEY` vom Natural-Administrator vorgenommen werden.



Anmerkung: Mit dem Systemkommando `KEY` vorgenommene Belegungen sind völlig unabhängig von mit einem `SET KEY`-Statement in einem Programm vorgenommene Belegungen.

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.

Hier können Sie außerdem mit der `CLEAR`-Taste ein Recording unterbrechen, das im Film-Modus abläuft. Weitere Informationen siehe *Recording Utility Natural Profiler* in der Dokumentation *Debugger und Dienstprogramme*.

CTRL+D-Taste — Aktive Operation unterbrechen

Das Drücken der Taste `CTRL+D` (bzw. `STRG+D`) hat dieselbe Wirkung wie das Terminalkommando `%%`.

RESET+ENTER-Taste — Aktive Schleife abbrechen

Diese Funktion ist nur verfügbar unter Com-plete und wenn Ihr Terminal über einen Local Controller angeschlossen ist.

Sie können die Verarbeitung einer gerade aktiven Verarbeitungsschleife abbrechen, indem Sie zuerst RESET (bzw. GRUNDSTELLUNG) und dann ENTER (bzw. EINGABE) drücken. Es muss sich um eine Schleife handeln, die einen Datenbankzugriff beinhaltet.

5 % — Fortsetzungsanzeiger für INPUT im Batch

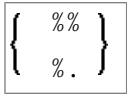


Wenn ein Prozentzeichen (%) als letztes Nicht-Leerzeichen eines Datensatzes im Batch-Betrieb verwendet wird, bewirkt es, dass der nächste Satz als Fortsetzungssatz behandelt wird.

Weitere Informationen siehe *Terminalkommandos im Batch-Betrieb* in der Beschreibung des INPUT-Statements in der *Statements-Dokumentation*.

6 %% und %. — Aktive Operation unterbrechen

■ %% im Online-Betrieb	22
■ %% im Batch-Betrieb	22
■ %. im Online-Betrieb	23
■ %. im Batch-Betrieb	23



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 Kommandoeingabe-Modus.

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

`%%` hat folgende Auswirkungen:

- Der Inhalt des Natural-Stack 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.



Anmerkung: Wenn Sie mit dem Natural-Profil- und Session-Parameter `CF` statt des Prozentzeichens (`%`) ein anderes Sonderzeichen angegeben haben, können Sie das angegebene Zeichen ebenfalls zweimal nacheinander eingeben.

%% 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.

Kommando	Funktion
	Falls weitere Daten im Eingabestrom vorhanden sind, setzt Natural die Verarbeitung mit der Zeile nach %% fort.
CC=OFF	%% 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.

7

%* — Anzeige von Eingabezeichen unterdrücken

%*

%* im Online-Betrieb

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.

%* im Batch-Betrieb

Im Batch-Betrieb wird mit %* das Ausdrucken der nachfolgenden Eingabedatenzeile unterdrückt. Die Zeile nach der Zeile, in der %* steht, wird nicht gedruckt. Damit können Sie zum Beispiel verhindern, dass Passwörter mitausgedruckt werden.

Beispiel für %* im Batch-Logon auf Natural Security:

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

Das Drucken aller Eingabedaten im Batch-Betrieb kann mit dem Profilparameter ECHO gesteuert werden.

8

%P — Obersten Stack-Eintrag löschen

`%P`

Dieses Kommando löscht den obersten Eintrag aus dem Natural-Stack.

Nähere Informationen zum Natural-Stack finden Sie im *Leitfaden zur Programmierung*.

9 **%.S — Stack-Daten lesen ohne zu löschen**

%.S

Normalerweise werden Kommandos/Daten vom Natural-Stack gelöscht, sobald sie aus dem Stack gelesen wurden.

Dieses Kommando bewirkt, dass mit dem nächsten `INPUT`-Statement der oberste Eintrag aus dem Stack gelesen wird, ohne gelöscht zu werden. Der Eintrag wird hierbei als Eingabedaten behandelt, ganz gleich ob es sich um Daten oder ein Kommando handelt.

Damit haben Sie die Möglichkeit, einen Stack-Eintrag zu prüfen und dann je nach seinem Inhalt zu entscheiden, ob sie ihn verarbeiten wollen oder nicht.

Den Inhalt des Stack können Sie auch mit der Systemvariablen `*DATA` prüfen.

Nähere Informationen zum Natural-Stack finden Sie im *Leitfaden zur Programmierung*.

10

%/ — End-of-File

%/

Dieses Terminalkommando erzwingt eine End-of-File-Bedingung, wenn es unmittelbar am Anfang eines im Batch-Betrieb von einem `INPUT`-Statement gelesenen Eingabedatensatzes steht.

11

%+ und %- — Natural Connection ein-/ausschalten

Diese Kommandos gelten nur, wenn Natural Connection installiert ist.

%	{	+ [N]	}
		-	

Mit diesen Terminalkommandos schalten Sie Natural Connection ein bzw. aus.

Kommando	Funktion
%+	Mit diesem Kommando wird die Natural-Systemvariable *DEVICE auf PC gesetzt, was die Benutzung von Natural Connection ermöglicht.
%+N	Entspricht %+. Außerdem bewirkt dieses Kommando, dass bei einem Upload/Download von Daten keine Feldnamen übertragen werden.
%-	Mit diesem Kommando wird die Natural-Systemvariable *DEVICE wieder auf den Wert gesetzt, den sie vor der Ausführung des Kommandos %+ hatte.

Weitere Informationen finden Sie in der *Natural Connection*-Dokumentation.

12

%<TECH — Technische Informationen anzeigen

%<TECH

Dieses Terminalkommando entspricht dem Systemkommando TECH.

13

%<TEST — Debugging-Funktion aufrufen

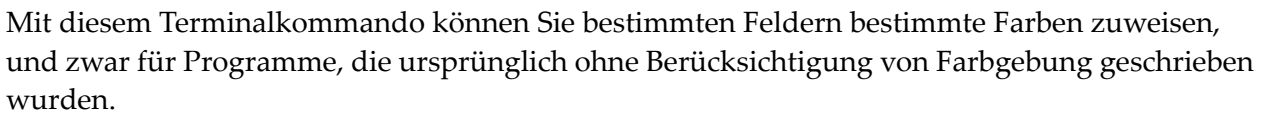
`%<TEST`

Mit diesem Terminalkommando rufen Sie die Debugging-Funktion auf.

Es entspricht dem Systemkommando `TEST`.

Nähere Informationen zur Debugging-Funktion finden Sie in der *Debugger*-Dokumentation.

%= — Zuordnen von Farben zu Feldern



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

39

Kommando %=	Funktion
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.

Siehe auch *Zuweisen von Farben zu Feldern — Terminalkommando %=* im Leitfaden zur Programmierung.

15

%A — Ausführen einer Bildschirm-Aufzeichnung

`%Aname`

Mit dem Terminalkommando `%Aname` führen Sie die unter `name` gespeicherte Aufzeichnung aus. Voraussetzung ist, dass Ihre aktuelle Library diejenige ist, in der die Aufzeichnung gespeichert ist.

Wenn Sie das Kommando `%Aname` während der Aufzeichnung einer Session eingeben, wird nicht die mit `%Aname` angegebene Aufzeichnung ausgeführt, sondern das Kommando `%Aname` in die Source, die aufgezeichnet wird, eingefügt.

So können Sie aus einer Aufzeichnung heraus eine andere Aufzeichnung ausführen und mehrere Aufzeichnungen aneinanderreihen. Allerdings können Sie Aufzeichnungen nicht schachteln; die Ausführung der Aufzeichnung, die das Kommando `%Aname` enthält, endet nach dem Kommando und wird nach der Ausführung von `name` nicht fortgesetzt.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Programmierschnittstelle (API): USR0350N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

16

%B — Aktivieren/Deaktivieren des Bildschirm-Aufzeichnungsvorgangs

`%Bname`

Der Bildschirm-Aufzeichnungsvorgang wird mit dem Terminalkommando `%Bname` eingeschaltet und mit dem Terminalkommando `%B` wieder ausgeschaltet.

Kommando	Funktion
<code>%Bname</code>	Mit dem Terminalkommando <code>%Bname</code> schalten Sie den Aufzeichnungsvorgang ein. Alle nachfolgenden Aktionen werden aufgezeichnet. Die aufgezeichneten Daten werden unter dem angegebenen Namen gespeichert. Namen von Aufzeichnungen müssen innerhalb einer Library eindeutig sein.
<code>%B</code>	Mit dem Terminalkommando <code>%B</code> schalten Sie den laufenden Aufzeichnungsvorgang aus. Die Aufzeichnung wird automatisch gespeichert, und Sie können sie beliebig oft wieder ausführen. Mit <code>%B</code> können Sie auch zusätzliche Aktionen in eine bestehende Aufzeichnung einfügen: nachdem Sie das Abspielen einer Aufzeichnung mit der LÖSCH-Taste unterbrochen haben, geben Sie das Kommando <code>%B</code> ein, und alle nachfolgenden Aktionen werden in die Aufzeichnung eingefügt, solange bis Sie erneut <code>%B</code> eingeben. Das Abspielen der Aufzeichnung wird daraufhin fortgesetzt.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Programmierschnittstelle (API): USR0350N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

17

%B= — Library für Bildschirm-Aufzeichnung angeben

`%B=library-name`

Mit dem Terminalkommando `%B=library-name` geben Sie die Library an, in der alle anschließenden Bildschirm-Aufzeichnungen gespeichert werden sollen.

Wenn Sie den Aufzeichnungsvorgang aktivieren, ohne eine Library angegeben zu haben, wird die Aufzeichnung in der Library gespeichert, deren Name dem Inhalt der Systemvariablen `*INIT-USER` zu Beginn der Aufzeichnungsvorgangs entspricht.

Wenn Sie während der Aufzeichnung einer Session die Library wechseln, bleibt die Library, in der die Aufzeichnung gespeichert wird, dieselbe (d.h. entweder die mit `%B=` angegebene oder die der Systemvariablen `*INIT-USER` entsprechende); dadurch ist es möglich, auch anwendungsübergreifende Aktionen aufzuzeichnen.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

18

%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:

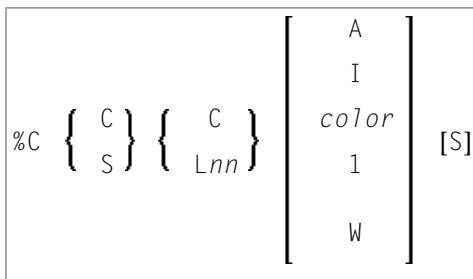
1. %C sollte nicht in einer Editor-Session verwendet werden. Änderungen des Sourcecode-Arbeitsbereichs außerhalb des Editor-Inhalts werden vom Editor ignoriert.
2. Der Seitenpuffer (die logische Natural-Ausgabe) ist nicht unbedingt mit dem am Bildschirm angezeigten Bildschirmpuffer identisch.

Beispiel:

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


19

%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 $nn1$ ist nicht möglich.)

■ **%C...CW**

kopiert nur das Wort (begrenzt durch Leer- oder Sonderzeichen in einem Feld), auf dem der Cursor steht. (%C...L nnW 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 Kommando %C... direkt eingeben (oder es einer PF-Taste zuweisen), bezieht es sich auf den gesamten *physischen Bildschirm*; d.h. Sie können vom ganzen Schirm kopieren, ganz gleich ob die Daten, die Sie kopieren möchten, innerhalb oder außerhalb des aktiven Fensters sind.

Führen Sie das Kommando über ein SET CONTROL-Statement aus, bezieht es sich auf die von Natural erzeugte *logische Seite*. In Verbindung mit der Stay-Option ist es Ihnen dadurch möglich, sämtliche benötigten Daten von einer ganzen logischen Seite (die größer als der physische Schirm sein kann) zu kopieren, bevor Sie diese Daten weiterverarbeiten.

Siehe auch *Daten vom Bildschirm kopieren* im Abschnitt *Dialog-Gestaltung* des Kapitels *Gestaltung von Benutzeroberflächen von Anwendungen* im Leitfaden zur Programmierung.

20 %D — Aktivieren des Keyword/Delimiter-Modus

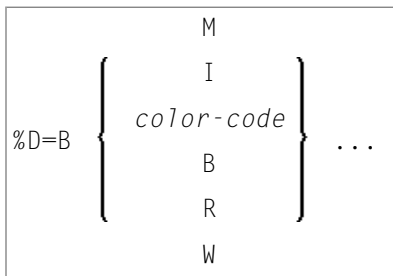
`%D`

Mit diesem Terminalkommando aktivieren Sie den Keyword/Delimiter-Modus. Dieser Modus empfiehlt sich für die Batch-Verarbeitung.

Weitere Informationen siehe `INPUT`-Statement in der Statements-Dokumentation.

21 %D= — Outlining (Rahmung) steuern

Outlining (Rahmung) ist nur auf bestimmten Terminaltypen möglich, in der Regel auf solchen, die auch Doppelbyte-Zeichensätze (z.B. Kanji) unterstützen.



Mit dem Terminalkommando `%D=B` steuern Sie das Outlining.

Outlining (auch als „Boxing“ bezeichnet) ist die Möglichkeit, bestimmte Felder auf dem Bildschirm eingerahmt (d.h. von einer Linie umgeben) anzuzeigen. Diese Form der Anzeige ist eine weitere Möglichkeit, dem Benutzer Länge und Position von Feldern auf dem Schirm deutlich zu machen.

Das Kommando `%D=B` bietet Ihnen folgende Möglichkeiten:

Kommando	Funktion
<code>%D=BM</code>	Dieses Kommando bewirkt, dass alle modifizierbaren Felder (<code>AD=M</code> oder <code>AD=A</code>) mit Rahmen angezeigt werden. Wenn dieses Kommando nochmals ausgeführt wird, wird diese Funktion wieder ausgeschaltet: mit anderen Worten, es erfolgt eine Rückkehr in den standardmäßigen Zustand.
<code>%D=BI</code>	Dieses Kommando bewirkt, dass alle intensivierten Felder (<code>AD=I</code>) mit Rahmen angezeigt werden.
<code>%D=Bcolor-code</code>	Dieses Kommando bewirkt, dass alle Felder der angegebenen Farbe mit Rahmen angezeigt werden. Gültige Farbcodes, siehe Session-Parameter <code>CD</code> .

Kommando	Funktion
%D=BB	Dieses Kommando generiert große Rahmen („big boxes“): wenn mehrere Felder mit identischen Anzeigecharakteristika untereinander angezeigt werden, wird ein einziger großer Rahmen um sie gezogen. Normalerweise (wenn %D=BB nicht angegeben wird) wird jedes Feld einzeln eingerahmt.
%D=BR	Dieses Kommando bewirkt, dass alle Rahmen-Anweisungen zurückgesetzt werden, d.h. kein Feld wird mit Rahmen angezeigt.
%D=BW	Dieses Kommando bewirkt, dass alle Fenster, die mit Rahmen angezeigt werden sollen (siehe <code>DEFINE WINDOW</code> -Statement in der <i>Statements</i> -Dokumentation), mit Outlining- Rahmen statt mit Fenster-Rahmen angezeigt werden.

Wenn Outlining aktiv ist, werden Füllzeichen in modifizierbaren Feldern nicht angezeigt, da sie die gleiche Funktion erfüllen wie Outlining, nämlich Benutzern Position und Länge von Feldern deutlich zu machen.

Weitere Informationen entnehmen Sie dem Abschnitt *Outlining (Umrahmung) — Terminalkommando %D=B* im *Leitfaden zur Programmierung*.

22

%DUE — Speicherauszug bei spezifischem Fehler

%DUE	$\left\{ \begin{array}{l} +nnnn \\ -nnnn \\ -ALL \end{array} \right\}$
------	--

Mit diesem Terminalkommando können Sie in einer Session Natural-Fehlernummern angeben, für die ein hexadezimaler Speicherauszug (Dump) gedruckt werden soll. Dies kann hilfreich sein, um einen hexadezimalen Speicherauszug für die Analyse einer spezifischen Fehlersituation durch Software AG-Personal zu erhalten. Siehe auch Profilparameter DUE.

Mit dem Kommando %DUE haben Sie die folgenden Optionen zur Verfügung:

Kommando	Funktion
%DUE+nnnn	Fügt eine Fehlernummer hinzu, für die ein hexadezimaler Speicherauszug gedruckt werden soll. Wenn die Nummer größer als 9999 ist, wird die Fehlermeldung NAT1134 ausgegeben.
%DUE-nnnn	Entfernt eine vorher angegebene Fehlernummer aus der Tabelle. Wenn die Nummer größer als 9999 ist, wird die Fehlermeldung NAT1134 ausgegeben.
%DUE-ALL	Entfernt alle vorher angegebenen Fehlernummern aus der Tabelle.

Wenn ein Fehler auftritt, der durch %DUE angegeben wurde, wird eine Programmprüfung erzwungen. Wenn der Profilparameter DU=OFF gesetzt ist, wird er auf DU=ON abgeändert. Bei der weiteren Verarbeitung wird die Einstellung des Profilparameters DU berücksichtigt.

Beispiele:

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


23

%E — Aufgezeichnete Bildschirme anzeigen

%E

Mit diesem Kommando können Sie sich Bildschirme anzeigen lassen, die mit der Utility NATPAGE aufgezeichnet wurden. Siehe auch *NATPAGE Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Dieses Kommando unterbricht die normale Verarbeitung und aktiviert den Anzeigemodus der NATPAGE-Utility: Sie erhalten eine Liste aller mit der NATPAGE-Utility (unter Verwendung der Terminalkommandos %I und %P) aufgezeichneten Schirme, jeweils mit Bildschirmnummer, Aufzeichnungszeitpunkt und Map-Name (falls der Bildschirm eine Map ist). In der Liste können Sie die Schirme auswählen, die Sie sehen möchten. Sie können auf den angezeigten Bildschirmen keine Eingaben machen.

Wenn ein Bildschirm mit %I oder %P aufgezeichnet wird, wird eine Ecke des Bildschirms mit zwei Informationen überschrieben: der Uhrzeit, wann der Bildschirm aufgezeichnet wurde, sowie einer laufenden Nummer (die Bildschirme werden in der Reihenfolge, in der sie aufgezeichnet werden, fortlaufend numeriert).

Im NATPAGE-Anzeigemodus können Sie im Eingabefeld `CMD` folgende Blätterkommandos eingeben:

Kommando	Funktion
TOP oder T	Zeigt den zuerst aufgezeichneten Schirm an.
BOT oder B	Zeigt den zuletzt aufgezeichneten Schirm an.
<i>nnn</i>	Zeigt den Schirm mit der Nummer <i>nnn</i> an..
<i>+nnn</i>	Blättert <i>nnn</i> Schirme vor.
<i>- nnn</i>	Blättert <i>nnn</i> Schirme zurück.
.	Punkt, beendet den Anzeigemodus.

Beim Versuch, zu einer nicht vorhandenen Schirmnummer zu blättern, wird - je nach Blätterrichtung
- entweder der erste oder der letzte Schirm angezeigt.

Siehe auch die Terminalkommandos [%P](#), [%O](#), [%S](#) und [%I](#).

24 %E= — Fehlerbehandlung ein-/ausschalten

$$\%E = \begin{cases} \text{ON} \\ \text{OFF} \end{cases}$$

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>	<p>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.</p> <p>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.</p>
<code>%E=ON</code>	<p>Mit dem Terminalkommando <code>%E=ON</code> schalten Sie dann Fehlertransaktion und <code>ON ERROR</code>-Verarbeitung wieder ein.</p>

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

25

%F — Aktivieren des Forms/Screen-Modus

`%F`

Mit diesem Kommando aktivieren Sie den Forms/Screen-Eingabemodus.

Weitere Informationen zu diesem Modus entnehmen Sie dem Abschnitt *Eingabe-Modi* in der Beschreibung des `INPUT`-Statements in der *Statements*-Dokumentation.

26

%F= — Zeichen für Bildschirmfenster-Rahmen

`%F=c h v`

Mit diesem Terminalkommando können Sie die Zeichen definieren, die im Rahmen eines Bildschirmfensters verwendet werden sollen.

<i>c</i>	Das erste Zeichen wird für die vier Ecken (<i>corners</i>) des Rahmens verwendet.
<i>h</i>	Das zweite Zeichen wird für die <i>horizontalen</i> Linien des Rahmens (oberer und unterer Rand) verwendet.
<i>v</i>	Das dritte Zeichen wird für die <i>vertikalen</i> Linien des Rahmens (linker und rechter Rand) verwendet.

Beispiel: Das Kommando `%F=+ - !` lässt den Rahmen des Fensters so aussehen:

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

Weitere Informationen zur Fensterverarbeitung siehe `DEFINE WINDOW`-Statement in der *States-Dokumentation*.

Siehe auch den Abschnitt *Bildschirmgestaltung (Fenster)* im Kapitel *Gestaltung der Benutzeroberflächen von Anwendungen* im *Leitfaden zur Programmierung*.

27

%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.

28

%G — Wiedergabemodus für eine Bildschirm-Aufzeichnung

%G	{	ON	}
		OFF	

Eine Bildschirm-Aufzeichnung kann in zwei Modi wiedergegeben werden:

■ Hintergrund-Modus

Im Hintergrund-Modus läuft die Wiedergabe der gesamten Aufzeichnung unsichtbar ab; alle Aktionen der Aufzeichnung werden ausgeführt, ohne dass Ihnen während der Wiedergabe der Aufzeichnung irgendetwas am Bildschirm angezeigt wird. Außerdem können Sie ein im Hintergrund-Modus ablaufende Wiedergabe nicht unterbrechen (es sei denn, die Aufzeichnung enthält das Terminalkommando %R).

■ Einzelschritt-Modus (Filmmodus)

Im Einzelschritt-Modus wird eine Aufzeichnung Schritt für Schritt wiedergegeben, und alle Aktionen werden Ihnen angezeigt. Durch Drücken von EINGABE gelangen Sie zum jeweils nächsten Schritt. Im Einzelschritt-Modus können Sie die Wiedergabe einer Aufzeichnung außerdem durch Drücken von LÖSCH unterbrechen.

Standardmäßig läuft die Wiedergabe einer Aufzeichnung im Hintergrund-Modus ab.

Kommando	Funktion
%GON	Mit diesem Kommando schalten Sie den Einzelschritt-Modus ein.
%GOFF	Mit diesem Kommando schalten Sie zurück zum Hintergrund-Modus.
%G	Mit diesem Kommando wechseln Sie von einem Modus zum anderen.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Programmierschnittstelle (API): USR0350N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

29 %H — Hardcopy-Ausgabe

%H	<i>destination</i>
	,
	.
	<i>#destination</i>
	<i>=[destination]</i>
	-
L	

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



Anmerkung: Die Kommandos sind nur gültig, falls diese Funktion implementiert ist.

Standardmäßig bezieht sich ein %H-Kommando auf die aktuelle logische Seite (d.h. das aktuelle Bildschirmfenster ohne Meldungszeile, PF-Tastenleiste und Statistikzeile/Infoline). Sie können aber auch den aktuellen Bildschirm drucken (siehe %H, und %H. unten).

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>Sie erhalten kein Auswahlfenster; hier wird der mit dem Profilparameter HCDEST angegebene Drucker genommen.</p>

Kommando	Funktion
%H <i>destination</i>	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.
%H,	Alle nachfolgenden %H-Kommandos beziehen sich auf den jeweils aktuellen <i>Bildschirminhalt</i> .
%H.	Alle nachfolgenden %H-Kommandos beziehen sich auf die jeweils <i>aktuelle logische Seite</i> (gilt standardmäßig).
%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 Hardcopy-Ausgabe wird der mit dem Profilparameter HCDEST angegebene Drucker benutzt.</p>
%H= <i>destination</i>	Wie %H=, aber mit dem Unterschied, dass mit %H= <i>destination</i> angegeben wird, auf welchem Hardcopy-Gerät alle logischen Ausgaben ausgegeben werden sollen. <i>destination</i> kann 1 bis 8 Zeichen lang sein.
%H# <i>destination</i>	Mit %H# <i>destination</i> können Sie die Hardcopy zu einem besonderen, mit einem DEFINE PRINTER-Statement definierten Ausgabemedium leiten, zum Beispiel in den Natural-Editorarbeitsbereich, nach Con-nect oder in die Infoline.
%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>
%HL	<p>Dieses Kommando gilt nur unter CMS.</p> <p>Es wird eine Datei mit Namen NATURAL LISTING A angelegt.</p>



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

30

%I — Aktuellen Bildschirm aufzeichnen

%I

Wenn Sie dieses Terminalkommando auf einem Bildschirm eingeben, wird der Bildschirm von der NATPAGE-Utility aufgezeichnet. Siehe auch NATPAGE Utility in der Dokumentation *Debugger und Dienstprogramme*.

Die maximale Anzahl von Bildschirmen, die aufgezeichnet werden können, bestimmen Sie mit dem Session-Parameter PD. Wenn diese Anzahl überschritten wird, überschreibt jeder weitere Bildschirm einen bereits aufgezeichneten, beginnend mit dem zuerst aufgezeichneten.

Um sich die so aufgezeichneten Bildschirme anzeigen zu lassen, verwenden Sie das Terminalkommando %E.

Siehe auch die Terminalkommandos %P, %O, %S und %E.

31

%J — Helproutine aufrufen

`%Jhelproutine`

Mit diesem Terminalkommando können Sie im Rahmen einer Bildschirm-Aufzeichnung eine interaktive Helproutine aufrufen.

Wenn Sie %J eingeben, nachdem Sie die Ausführung einer Bildschirm-Aufzeichnung unterbrochen haben, wird die Ausführung der Aufzeichnung nach Ausführung der Helproutine fortgesetzt.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

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.

32

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

%K	$\left\{ \begin{array}{c} nn \\ p n \end{array} \right\}$
----	---

Diese Terminalkommandos können verwendet werden, um die Funktionstasten (PF, ENTER bzw. EINGABE) und PA-Tasten zu simulieren. Bevor eine Taste simuliert werden kann, muss sie mit dem SET KEY-Statement sensibilisiert werden.

Kommando	Funktion
%Knn	Simuliert die Funktionstaste PFnn (PF1 bis PF48). 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. Dadurch können PF-Tastenfunktionen auch im Batch-Betrieb verwendet werden.
%K0	Simuliert die ENTER- bzw. EINGABE-Taste.
%KPn	Simuliert die Funktionstaste PAn (PA1 bis PA3) siehe %Knn.

33

%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.

Sie sollten sicherstellen, dass auch der verwendete TP-Monitor keine Umsetzung vornimmt, bevor Daten an Natural übergeben werden. You should also ensure that the TP monitor used does not perform any translation before data are passed to Natural.

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

Programmierschnittstelle (API): USR1005N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

34 %L= — Sprachcode

`%L=nn`

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

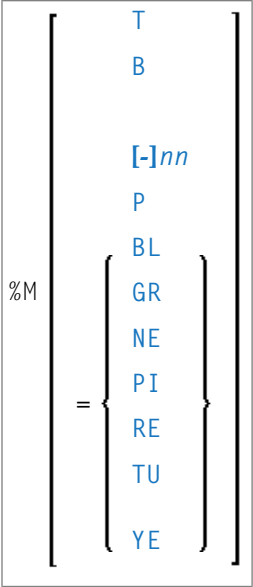
Eine Liste der möglichen Sprachcodes finden Sie bei der Beschreibung der Systemvariable `*LANGUAGE`, siehe *Systemvariablen*-Dokumentation.

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

35

%M — Steuerung der Meldungszeile

■ Positionierung der Meldungszeile	82
■ Schutz der Meldungszeile	83
■ Farbe der Meldungszeile	83



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

Dieses Kommando wird im Batch-Betrieb ignoriert.

Dieses Terminalkommando wirkt auch auf die `NEXT`-Zeile (bzw. die `MORE`-Zeile) von Natural.

Programmierschnittstelle (API): `USR1005N`. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Positionierung der Meldungszeile

Kommando	Funktion
<code>%MT</code>	Platziert die Meldungszeile am oberen Bildschirmrand (<i>Top</i>).
<code>%MB</code>	Platziert die Meldungszeile am unteren Bildschirmrand (<i>Bottom</i>).
<code>%M</code>	Bewirkt, dass die Position der Meldungszeile vom oberen zum unteren Bildschirmrand (oder umgekehrt) oder von Zeile <i>nn</i> zum unteren Bildschirmrand wechselt.
<code>%Mnn</code>	Platziert die Meldungszeile in die <i>nn</i> -te Zeile des Bildschirms.
<code>%M-<i>nn</i></code>	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.

Schutz der Meldungszeile

Kommando	Funktion
%MP	Schaltet zwischen geschützter und ungeschützter Meldungszeile hin und her.

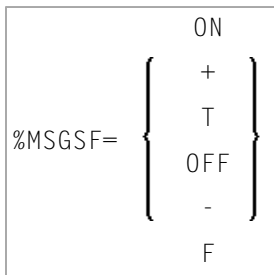
Farbe der Meldungszeile

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

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

36

%MSGSF — Anzeigeformat von Systemfehlermeldungen



Standardmäßig besteht eine Natural-Systemfehlermeldung aus dem Namen des Programms und der Nummer der Zeile, das/die den Fehler verursacht hat, gefolgt von dem eigentlichen Text der Meldung. Je nach Größe des Fensters, in dem die Meldung angezeigt wird, kann es sein, dass der eigentliche Text nicht vollständig angezeigt, sondern abgeschnitten wird. Mit dem Terminalkommando `%MSGSF` können Sie dies vermeiden.

Kommando	Funktion
<code>%MSGSF=ON</code>	Systemfehlermeldungen werden komplett angezeigt; d.h. Programmname, Zeilennummer und eigentlicher Meldungstext werden angezeigt (dies ist die Standardeinstellung).
<code>%MSGSF=+</code>	
<code>%MSGSF=T</code>	
<code>%MSGSF=OFF</code>	Systemfehlermeldungen werden in Kurzform angezeigt; d.h. es wird nur der eigentliche Meldungstext angezeigt (aber nicht der Programmname und die Zeilennummer).
<code>%MSGSF=-</code>	
<code>%MSGSF=F</code>	



Anmerkungen:

1. Anstelle von `ON` können Sie auch `+` oder `T` (true) angeben; anstelle von `OFF` können Sie auch `-` oder `F` (false) angeben.
2. Die Anzeigeform von Systemfehlermeldungen kann auch mit dem Profilparameter `MSGSF` gesteuert werden, siehe *Parameter-Referenz*.

37 %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.

Unter IMS TM:

Unter IMS TM gilt dieses Terminalkommando nicht. Wenn es verwendet wird, bewirkt es, dass der nächste logische Ausgabeschirm unterdrückt wird.

38 %O — Beenden der Aufzeichnung von Bildschirmen

%O

Mit diesem Terminalkommando beenden Sie das Aufzeichnen von Bildschirmen mit der NATPAGE-Utility, die mit dem Terminalkommando %P aktiviert wurde). Siehe auch *NATPAGE Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Der aktuelle Bildschirm wird noch aufgezeichnet. Alle seit dem letzten %P-Kommando aufgezeichneten Schirme bleiben gespeichert.

See auch die Terminalkommandos %E, %I, %P und %S.

39

%P — Aufzeichnung für nachfolgende Bildschirme aktivieren

%P

Mit diesem Terminalkommando aktivieren Sie die NATPAGE-Utility, um den aktuellen Schirm und alle nachfolgenden Schirme aufzuzeichnen. Siehe auch *NATPAGE Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Die maximale Anzahl von Bildschirmen, die mit NATPAGE aufgezeichnet werden können, bestimmen Sie mit dem Session-Parameter PD. Wenn diese Anzahl überschritten wird, überschreibt jeder weitere Schirm einen bereits aufgezeichneten, beginnend mit dem zuerst aufgezeichneten.

Alle Schirme, die mit vorhergehenden %P- bzw. %I-Kommandos aufgezeichnet wurden, werden gelöscht, wenn Sie ein %P-Kommando ausführen.

Um sich die aufgezeichnete Bildschirme anzeigen zu lassen, verwenden Sie das Terminalkommando %E.

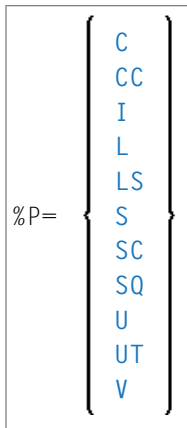
Siehe auch die Terminalkommandos %E, %I, %O und %S.

Programmierschnittstelle (API): USR0350N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

40

%P= — CALL-Optionen

■ %P=S — Standard-Linkage für Aufruf	94
■ %P=SC — Standard-Linkage-Aufruf mit Simulation von EXEC CICS LINK	95
■ %P=SQ — Standard-Linkage für Aufruf von quasi-reentranten Subprogrammen	95
■ %P=U — Separate Arbeitseinheit für aufgerufenes Programm	96
■ %P=UT — Separate Arbeitseinheit für aufgerufenes Programm, Transaktions-Backout tolerieren	96
■ %P=V — Ausspeichern bei Aufruf	97
■ %P=C — Übergabe von Parameterwerten statt Adresse	97
■ %P=CC — Parameterwerte in einem CICS-Container übergeben	99
■ %P=L — Aufruf des dynamischen Hauptprogramms LE	99
■ %P=LS — Aufruf von LE-Subroutine	99
■ %P=I — Übergabe von PCB-Adressen an Aufgerufenen	100



Jedes `%P=`-Kommando gilt nur für den nächsten Aufruf, und die `CALL`-Optionen werden ohne Bedingung nach Rückkehr von dem Aufruf zurückgesetzt.

Deshalb wird empfohlen, das `%P=`-Kommando unmittelbar vor dem betreffenden `CALL`-Statement abzusetzen.



Anmerkung: Um die `CALL`-Optionen dauerhaft zu setzen, können Sie den Profilparameter `PGP` bzw. das entsprechende Parametermakro `NTPGP` benutzen.

Wenn Sie die Natural CICS-Schnittstelle benutzen, können Sie mit den Kommandos `%P=S(C)`, `%P=V`, `%P=C(C)`, `%P=U` und `%P=UT` spezielle Optionen setzen, die dann gelten, wenn unter CICS ein Natural-Programm (über ein `CALL`-Statement) ein Nicht-Natural-Programm aufruft. In allen anderen Umgebungen werden diese Kommandos ignoriert.

Wenn Sie die Natural IMS TM-Schnittstelle benutzen, können Sie mit dem Kommando `%P=I` spezielle Optionen setzen, die gelten, wenn ein Natural-Programm (über ein `CALL`-Statement) ein Nicht-Natural-Programm aufruft. In allen anderen Umgebungen wird dieses Kommando ignoriert.

Informationen bezüglich der Unterstützung von IBM Language Environment (LE)-Subprogrammen durch Natural siehe Abschnitt *LE-Subprogramme* in der *Operations*-Dokumentation.

Einzelheiten zum Aufrufen von Nicht-Natural-Programmen siehe `CALL`-Statement in der *Statements*-Dokumentation.

%P=S — Standard-Linkage für Aufruf

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird.

Wenn ein Natural-Programm unter CICS ein Nicht-Natural-Programm aufruft, geschieht der Aufruf normalerweise über eine `EXEC CICS LINK`-Anforderung.

Wenn für den Aufruf stattdessen Standard Linkage verwendet werden soll, geben Sie das Terminalkommando `%P=S` ein. Voraussetzung ist, dass das aufgerufene Programm den Standard-Linkage-Konventionen mit Standard-Registerverwendung entspricht.

%P=SC — Standard-Linkage-Aufruf mit Simulation von EXEC CICS LINK

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird.

Wenn ein Natural-Programm normalerweise ein Nicht-Natural-Programm unter CICS aufruft, erfolgt der Aufruf über eine `EXEC CICS LINK`-Anforderung.

Wenn dieses Terminalkommando benutzt wird, ruft die Natural CICS-Schnittstelle das Nicht-Natural-Programm mittels Standard-Linkage-Konventionen auf, aber mit der Übergabe von Parametern, wie dies ein `CICS LINK` durchführt, d.h. Register 1 zeigt auf eine Parameter-Liste mit Adressen von `CICS EIB` und einer `CICS COMMAREA`.



Anmerkungen:

1. `%P=SC` kann mit `%P=C` kombiniert werden.
2. Ein über Standard Linkage aufgerufenes Programm darf nicht über `EXEC CICS RETURN` zum aufrufenden Programm zurückkehren. Assembler-Programme müssen über einen `DFHEIRET`-Macro-Aufruf zurückkehren. COBOL-Programme müssen über ein `GOBACK`-Statement zurückkehren.

%P=SQ — Standard-Linkage für Aufruf von quasi-reentranten Subprogrammen

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird und wenn die bei Ihnen installierte Natural CICS Interface-Version dieses Kommando unterstützt.

`%P=SQ` entspricht funktional dem Terminalkommando `%P=S`.

Dieses Kommando gilt nicht, wenn ein aufgerufenes Nicht-Natural-Subprogramm in Natural mit dem Profilparameter `CSTATIC` oder `RCA` definiert ist.

Wenn Natural mit dem CICS-Attribut `CONCURRENCY(REQUIRED)` definiert ist und die CICS OTE-Umgebung (Open Transaction Environment) benutzt, müssen alle mittels Standard-Linkage-Konventionen aufgerufenen Subprogramme voll reentrant sein.

Wenn ein Nicht-Natural-Subprogramm aufgerufen wird, das quasi-reentrant, aber nicht thread-sicher und voll reentrant ist, während Natural in einem offenen TCB läuft, sorgt die Einstellung `%P=SQ` dafür, dass das aufgerufene Programm unter dem CICS QR TCB ausgeführt wird. Natural

setzt dann das CICS-Kommando `EXEC CICS LINK PROGRAM('NCILINKQ')` ab, um den QR TCB umzuschalten und das Programm `NCILINKQ` auszuführen, das das Subprogramm mittels Standard-Linkage-Konventionen aufruft.

Dieses Terminalkommando gilt auch für LE-Subprogramme. Informationen bezüglich der Unterstützung von IBM Language Environment (LE)-Subprogrammen durch Natural siehe Abschnitt *LE-Subprogramme* in der *Operations-Dokumentation*.

%P=U — Separate Arbeitseinheit für aufgerufenes Programm

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird.

Dieses Kommando gibt an, dass ein über DPL aufgerufenes CICS-Programm seine eigene Arbeitseinheit hat, die bei der Rückkehr zu Natural bestätigt wird, d.h. vom aufgerufenen Programm vorgenommene Änderungen an wiederherstellbaren Ressourcen werden bestätigt oder wieder eingespeichert, unabhängig von Änderungen an wiederherstellbaren Ressourcen, die Natural vorgenommen hat. In der Sprache von CICS ausgedrückt: Dieses Kommando bewirkt ein `EXEC CICS LINK` mit der `SYNCONRETURN`-Option.

%P=UT — Separate Arbeitseinheit für aufgerufenes Programm, Transaktions-Backout tolerieren

Dieses Kommando hat fast die gleiche Bedeutung wie das Kommando `%P=U`, jedoch mit einer Ausnahme:

Bei `%P=U` führen bei der Rückkehr von dem aufgerufenen Programm alle CICS-Bedingungen, außer `NORMAL`, zu einer Natural-Fehlermeldung `NAT0920`, wobei der Reason Code der Response Code des `EXEC CICS LINK`-Befehls ist.

Bei `%P=UT` führt ein `CICS ROLLEDBACK` Response Code nicht zu einem Fehler, sondern wird toleriert und wie eine `NORMAL`-Bedingung behandelt.

%P=V — Ausspeichern bei Aufruf

Dieses Kommando gilt nur in Natural-Umgebungen, in denen sich mehrere Sessions Threads in derselben Region teilen, beispielsweise Natural CICS-Schnittstelle oder Batch-Server-Umgebungen unter z/OS.

Wenn ein Natural-Programm ein Nicht-Natural-Programm aufruft und das aufgerufene Programm im Dialog einen Terminal-I/O ausführt, ist der Natural-Thread normalerweise solange blockiert, bis der Benutzer eine Eingabe gemacht hat.

Mit %P=V können Sie vermeiden, dass der Natural-Thread blockiert ist: wenn Sie dieses Terminalkommando angeben, werden die vom Natural-Programm an das aufgerufene Programm übergebenen Parameterdaten aus dem Thread herauskopiert, und der Thread wird vor dem Aufruf ausgespeichert. Der Thread steht dann für andere Benutzer zur Verfügung. Wenn das aufgerufene Programm die Kontrolle wieder an das aufrufende Natural-Programm zurückgibt, wird der Thread wieder eingespeichert, der (modifizierte) Datenbereich wird in den Thread kopiert, und Natural setzt die Verarbeitung fort.



Anmerkung: Nur die im CALL-Statement angegebenen Parameter werden aus dem Thread heraus- und wieder hineinkopiert.

%P=C — Übergabe von Parameterwerten statt Adresse

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird.

Ruft ein Natural-Programm unter CICS ein Nicht-Natural-Programm auf, wird normalerweise die Adresse der CALL-Statement-Parameteradressenliste in der COMMAREA übergeben. Wenn Sie statt der Adresse der Parameteradressenliste die Parameterwerte selbst in der COMMAREA übergeben möchten, führen Sie vor dem Aufruf das Terminalkommando %P=C aus.

Dadurch ist zum Beispiel DPL für aufgerufene CICS-Programme möglich: der Aufruf eines CICS-Programms, das sich in einer anderen CICS-Region befindet, ist nur mit %P=C möglich; denn da die „aufgerufene“ Region nicht auf Adressen in der „aufrufenden“ Region zugreifen kann, müssen stattdessen die Parameterwerte übergeben werden.

Wenn %P=C verwendet wird, werden keine Parameter in der TWA übergeben, sondern nur Parameterwerte in der CICS COMMAREA. Alle Parameter der CALL-Parameterliste werden unmittelbar aufeinanderfolgend kopiert, unabhängig von ihrer Ausrichtung. Die sich daraus ergebende COMMAREA-Länge entspricht der Summe der einzelnen Parameterlängen (was bei Bestimmung der Anzahl der zu übergebenden Array-Ausprägungen zu berücksichtigen ist). Bei der Rückkehr vom aufgerufenen Programm werden die Parameter zurückkopiert.

Wenn sich überlappende Felder übergeben werden oder dasselbe Feld mehrmals übergeben wird, sollten diese Felder für das aufgerufene Programm schreibgeschützt werden; andernfalls kann es bei der Rückgabe von Parameterwerten an das aufrufende Programm zu unvorhersehbaren Ergebnissen kommen.

Bei %P=C gilt die Einschränkung, dass Gruppen-Arrays nicht übergeben werden können:

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

Übergeben Sie sie entweder als einzelne Arrays:

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

Oder redefinieren Sie sie:

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

und geben den Array-Namen im CALL-Statement an.



Anmerkungen:

1. Wenn Sie für einen Aufruf sowohl %P=S als auch %P=C ausführen, wird %P=C ignoriert.
2. CICS COMMAREAs sind auf 32 KB begrenzt.
3. Wenn der Parameter CNTCALL im Macro NTCICSP auf ON gesetzt ist, wird ein CICS-Container statt einer COMMAREA übergeben, wenn die Datenlänge des Parameterwertes 32 KB überschreitet.
4. Vermeiden Sie Speicherüberschreitungen in CICS, indem Sie sicherstellen, dass ein aufgerufenes Programm prüft, ob die Länge des zu übergebenden Datenbereichs wirklich zur internen Datenstruktur passt. Dies erreicht man, indem man EIBCALEN gegen die COMMAREA-Länge prüft: IF LENGTH OF DFHCOMMAREA GREATER THAN EIBCALEN ... (in COBOL) oder IF SIZE (DFHCOMMAREA) > EIBCALEN ... (in PL/1).

%P=CC — Parameterwerte in einem CICS-Container übergeben

Dieses Kommando gilt nur, wenn die Natural CICS-Schnittstelle benutzt wird.

Für diese Funktionalität ist der CICS Transaction Server for z/OS ab Version 3.1 erforderlich.

Der Unterschied zwischen %P=C und %P=CC ist der, dass die Parameterwerte sofort in einem CICS-Container übergeben werden. Alle Einschränkungen gelten noch, außer der Längenbegrenzung von 32 KB.

%P=L — Aufruf des dynamischen Hauptprogramms LE

Dieses Kommando gilt nur, wenn bei der Installation von Natural die Option zur Unterstützung von IBM Language Environment (LE)-Aufrufkonventionen gesetzt wurde.

Dieses Kommando bewirkt, dass nach dem Aufruf eines dynamischen Hauptprogramms LE die Kontrolle wieder an Natural zurückgegeben wird.

Wenn ein dynamisches Hauptprogramm LE aufgerufen wird, wird standardmäßig die Kontrolle nach Ausführung des aufgerufenen Programms nicht an Natural zurückgegeben. Damit nach der Ausführung des aufgerufenen Programms die Kontrolle an Natural zurückgegeben wird, müssen Sie vor dem Programmaufruf %P=L verwenden.

Informationen zur Unterstützung von LE-Aufrufkonventionen finden Sie im Abschnitt *LE-Subprogramme* in der *Operations*-Dokumentation.



Anmerkung: Unter CICS sind %P=L und %P=LS gleichwertig, da LE-Hauptprogramme nicht unterstützt werden.

%P=LS — Aufruf von LE-Subroutine

Dieses Kommando gilt nur, wenn bei der Installation von Natural die Option zur Unterstützung von IBM Language Environment (LE)-Aufrufkonventionen gesetzt wurde.

Dieses Kommando bewirkt, dass ein dynamisches oder statisches Nicht-Natural-Programm mittels LE-Aufrufkonventionen aufgerufen wird, d.h. über Standard-Linkage-Konventionen mit dem Register 12, das die Adresse der LE CAA enthält.

Informationen zur Unterstützung von LE-Aufrufkonventionen finden Sie im Abschnitt *LE-Subprogramme* in der *Operations*-Dokumentation.

%P=I — Übergabe von PCB-Adressen an Aufgerufenen

Dieses Kommando gilt nur, wenn die Natural IMS TM-Schnittstelle benutzt wird.

%P=I zeigt Natural an, dass PCB-Adressen an das 3GL-Programm im folgenden CALL-Statement übergeben werden sollen. Da es nicht möglich ist, die Adresse einer PCB von Natural an das 3GL-Programm direkt zu übergeben, wird eine spezielle Konvention benutzt, um Natural anzuzeigen, dass an das 3GL-Programm übergebene Parameter entweder die Position oder den Namen einer PCB enthalten können. Für Felder, die diese Konvention erfüllen, übergibt Natural die Adresse der PCB an das aufgerufene 3GL-Programm anstatt der Adresse des angegebenen Parameters.

Nur skalare Felder des Formats A8 oder A12 werden auf eine PCB-Spezifikation abgeprüft. Die Felder, die eine PCB-Spezifikation enthalten, müssen die folgenden Konventionen erfüllen:

- Ein A8-Feld muss die Zeichenkette 'PCB=nnnn' enthalten, wobei nnnn eine vierstellige Ziffer ist, die die Position der PCB in der PSB-Adressenliste darstellt.
- Ein A12-Feld muss die Zeichenkette 'PCB=name' enthalten, wobei name der acht Zeichen umfassende Name für eine PCB ist, der mit dem NTIMSPT-Makro im Verlauf der Installation der Natural IMS TM-Schnittstelle definiert wurde.

Beispiel:

Nach Ausführung von %P=I werden für das Statement

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

die folgenden Parameter an das 3GL-Programm MY3GL übergeben:

- Adresse der PCB mit dem Namen MYPCB, wie mit dem Makro NTIMSPT angegeben,
- Adresse der vierten PCB in der PSB-Adressenliste,
- Adresse des Natural-Feldes PARAM.

Die folgenden Returncodes werden von der Systemfunktion RET zurückgegeben, wenn die PCB-Spezifikation ungültig ist:

Return Code	Erläuterung
-4	Der angegebene Wert nnnn für das Format 'PCB=nnnn' ist nicht numerisch.
-8	Der angegebene PCB-Name wurde nicht mit dem Makro NTIMSPT definiert.
-12	Die angegebene PCB mit der Position nnnn ist in der PSB-Adressenliste nicht vorhanden.

41

%Q — Map-Ausgabe im Batch-Betrieb unterdrücken

%Q

Im Online-Betrieb wird %Q ignoriert.

Im Batch-Betrieb bewirkt %Q, dass Maps oder Schirme, die über INPUT-Statements erzeugt werden, nicht mit ausgegeben werden.

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.

42 %Q0 — Pseudo-konversationelle Ausgabe unterdrücken

Dieses Kommando gilt nur unter CICS.

%Q0

In einer Natural-Session, die unter CICS im Pseudo-Dialogbetrieb (Pseudo-Conversational Mode) abläuft, können Sie mit dem Statement `CALL 'CMTASK'` eine andere CICS-Transaktion starten. Damit CICS die Transaktion starten kann, ist allerdings nach dem `CALL`-Statement ein Terminal-I/O, d.h. ein `INPUT`-Statement, erforderlich.

Um die Bildschirmausgabe dieses `INPUT`-Statements (die ohnehin sofort von der gestarteten Transaktion überschrieben würde) zu unterdrücken, führen Sie vor dem `INPUT`Statement das Statement `SET CONTROL 'Q0'` aus.

43 %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.

44

%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.

%R bei einer Bildschirm-Aufzeichnung (Recording Utility)

Durch Aufzeichnen des Terminalkommandos `%R` können Sie einen einzelnen Schritt in einer Aufzeichnung während der Ausführung der Aufzeichnung verändern.

Sie können `%R` auch dazu benutzen, Eingabedaten in einer Aufzeichnung, die ausgeführt wird, zu überschreiben.

Weitere Informationen zu Bildschirm-Aufzeichnungen siehe *Recording Utility* in der Dokumentation *Debugger und Dienstprogramme*.

45

%<RDC+ und %<RDC- - Trace Recording aktivieren/deaktivieren

% { <RDC+ <RDC- }

Diese Kommandos werden in Verbindung mit der *POFILER-Utility* benutzt, um die Trace Recording-Funktion des Natural Data Collector zu aktivieren bzw. zu deaktivieren.

Weitere Informationen siehe *Natural Data Collector aktivieren und steuern* in der Dokumentation *Debugger und Dienstprogramme*.

%<RDC+	Aktiviert die Trace Recording-Funktion des Natural Data Collector.
%<RDC-	Deaktiviert die Trace Recording-Funktion des Natural Data Collector.

Siehe auch Systemkommando RDC in der *Systemkommandos*-Dokumentation und Profilparameter RDC in der *Parameter-Referenz*-Dokumentation.

46

%RM — Schreibschutz von lichtstift-sensitiven Feldern

%RM

Mit diesem Kommando machen Sie alle lichtstift-sensitiven Felder auf dem Schirm schreibgeschützt, d.h. der Benutzer kann sie mit dem Lichtstift auswählen, aber ihren Inhalt nicht überschreiben.

Um den Schreibschutz wieder auszuschalten, geben Sie das Kommando noch einmal ein.

Um lichtstift-sensitiv zu sein, muss ein Feld intensiviert (AD=I) oder blinkend (AD=B) angezeigt werden, und das erste Zeichen im Feld muss eines der Lichtstift-Funktionszeichen (Fragezeichen (?), Größer-Zeichen (>), kaufmännisches Und-Zeichen (&) oder ein Leerzeichen) sein. Wenn ein Feld mit dem Lichtstift ausgewählt wird, ändert sich das Funktionszeichen; Sie können also die Verarbeitung von mit Lichtstift ausgewählten Feldern vom Wert des Funktionszeichens abhängig machen.

Siehe auch *Light Pen Support* (in der *Operations*-Dokumentation) und den Wert `PEN` der Systemvariablen `*PF-KEY`.

47

%RN — Bildschirmdaten-Komprimierung unterdrücken

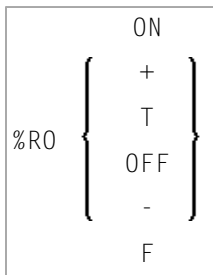
%RN

Für den nächsten Bildschirm-I/O unterdrückt dieses Kommando Natural's automatische Bildschirmdaten-Komprimierung und schickt stattdessen den kompletten Schirm.

Normalerweise schickt Natural bei einem Bildschirm-I/O nicht den kompletten Schirm, sondern nur die geänderten Bildschirmdaten. Beim Aufruf eines Nicht-Natural-Programms, das einen Bildschirm-I/O verursacht, erkennt Natural in den meisten TP-Umgebungen, dass ein Nicht-Natural-I/O stattgefunden hat, und schickt beim nächsten Bildschirm-I/O den kompletten Schirm. In manchen TP-Umgebungen erkennt Natural den Nicht-Natural-I/O jedoch nicht; in diesem Fall sollte %RN verwendet werden.

%RN gilt jeweils nur für den nächsten Bildschirm-I/O.

48 %R0 — Bildschirm-Optimierung ein-/ausschalten



Mit diesem Kommando können Sie die automatische Bildschirm-Optimierung von Natural aus- und wieder einschalten.

Die Bildschirm-Optimierung von Natural ermöglicht es, Bildschirm-Daten so komprimiert wie möglich zu senden. Falls dies zu Problemen bei der Bildschirm-Optimierung oder im Zusammenhang mit Hardware-Beschränkungen eines TP-Monitors führen sollte, können Sie die Bildschirm-Optimierung von Natural mit diesem Terminalkommando ausschalten; Bildschirmdaten werden dann in unkomprimierter Form gesendet.

Wenn Sie die Session-Parameter **BX** oder **BX=R** verwenden, sollten Sie die Bildschirm-Optimierung von Natural ausschalten.

Kommando	Funktion
%R0ON	Schaltet die Bildschirm-Optimierung ein.
%R0+	
%R0T	
%R0OFF	Schaltet die Bildschirm-Optimierung aus.
%R0-	
%R0F	
%R0	Schaltet zwischen beiden Modi hin und her.



Anmerkung: Anstelle von `ON` können Sie auch `+` oder `T` (true) angeben; anstelle von `OFF` können Sie auch `-` oder `F` (false) angeben.

49

%S — Fortsetzen der Aufzeichnung von Bildschirmen

%S

Mit diesem Terminalkommando können Sie das Aufzeichnen von Bildschirmen durch die Utility NATPAGE wiederaufnehmen. Siehe auch *NATPAGE Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Das Aufzeichnen von Bildschirmen, das mit dem Terminalkommando %P aktiviert wird, kann mit dem Terminalkommando %O unterbrochen werden. Nach einer solchen Unterbrechung können Sie das Aufzeichnen von Bildschirmen mit dem Terminalkommando %S wiederaufnehmen.

Siehe auch die Terminalkommandos %E, %I, %O und %P.

50

%T — Cursor am Schirmanfang platzieren

%T

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

51

%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. %T1/1 positioniert immer an den Anfang der obersten Datenzeile im Fenster.

52 %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.

Siehe auch *Positionierung des Cursors auf *COM – Terminalkommando %T** (im Leitfaden zur Programmierung).

53

%T= — Terminalspezifische Converter-Routine

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

Mit diesem Terminalkommando ermöglichen Sie, dass Natural die für einen bestimmten Terminaltyp geeignete Converter-Routine benutzen kann.

Falls das TP-System nicht in der Lage ist, Natural über den verwendeten physischen Terminaltyp zu informieren (wobei dann der Terminaltyp automatisch gesetzt würde), muss Natural mittels des Terminalkommandos %T= mitgeteilt werden, mit welchem Terminaltyp es arbeitet. Dadurch versetzen Sie Natural in die Lage, die für diesen Terminaltyp erforderliche Attributsequenz aufzubauen.

Standardmäßig (falls keine Informationen vom TP-System übergeben werden) benutzt Natural die Terminaltypen %T=3270 (IBM)

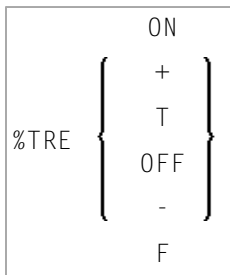
Statt des Kommandos %T= können Sie auch den Profilparameter TTYPE verwenden.



Anmerkung: Im obigen Diagramm sind nur die gebräuchlichsten Terminaltypen aufgeführt. Falls Sie einen anderen Terminaltyp benutzen, bitten Sie Ihren Natural-Administrator, im NATCONFIG-Modul zu überprüfen, ob dieser Typ mit %T= angegeben werden kann.

54

%TRE — Externe Trace-Funktion aktivieren/deaktivieren



Dieses Kommando dient dazu, die externe Trace-Funktion zu aktivieren bzw. deaktivieren.



Wichtig: Verwenden Sie dieses Kommando nicht ohne vorherige Rücksprache mit dem Software AG Support. Diese Funktion ist hauptsächlich für Software-AG-interne Verwendung gedacht und dient zur Fehlersuche. Sie schreibt Trace-Daten auf ein externes Trace-Dataset in Abhängigkeit von der TP-Umgebung, in der Natural läuft.

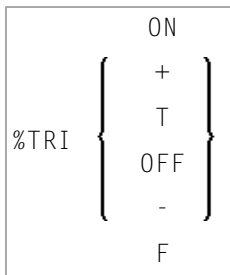


Anmerkungen:

1. Anstelle von `ON` können Sie auch `+` oder `T` (true) angeben. Anstelle von `OFF` können Sie auch `-` oder `F` (false) angeben.
2. Die externe Trace-Funktion kann auch mit dem Profilparameter `ETRACE` aktiviert / deaktiviert werden.

55

%TRI — Interne Trace-Funktion aktivieren/deaktivieren



Dieses Kommando dient dazu, die interne Trace-Funktion zu aktivieren bzw. deaktivieren.



Wichtig: Verwenden Sie dieses Kommando nicht ohne vorherige Rücksprache mit dem Software AG Support. Diese Funktion ist hauptsächlich für Software-AG-interne Verwendung gedacht und dient zur Fehlersuche. Sie übergibt Trace-Daten an die `SYSRDC`-Utility (die im Abschnitt *Debugging and Monitoring* in der *Natural Utilities*-Dokumentation beschrieben ist).



Anmerkungen:

1. Anstelle von `ON` können Sie auch `+` oder `T` (true) angeben. Anstelle von `OFF` können Sie auch `-` oder `F` (false) angeben.
2. Die interne Trace-Funktion kann auch mit dem Profilparameter `ITRACE` aktiviert / deaktiviert werden. (siehe *Parameter-Referenz*).

56

%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.

Siehe auch Terminalkommando [%L](#).

Programmierschnittstelle (API): USR0350N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

57

%V — Steuerung des Print-Modus

Dieses Terminalkommando ist nur relevant für Terminals, die invertierten Print-Modus (von rechts nach links) unterstützen. Siehe auch Profilparameter *PM* in der *Parameter Reference*-Dokumentation und *Bidirectional Language Support* in der *Unicode and Code Page Support*-Dokumentation.

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

Syntax-Beschreibung:

Kommando	Funktion
%VON	Schaltet den Print-Modus in umgekehrte Richtung, d.h. Daten werden von rechts nach links ausgegeben (entspricht <i>PM=I</i>). Alle nicht-alphanumerischen Felder, Systemvariablen und PF-Tastenzeilen werden in umgekehrter Reihenfolge dargestellt. Dadurch wird die korrekte Anzeige dieser Schirmelemente auf einem Terminal sichergestellt, das im Recht-nach-Links-Modus arbeitet.
%VOFF	Schaltet den normalen Print-Modus ein, d.h. Daten werden von links nach rechts ausgegeben (entspricht <i>PM=R</i>).
%V	Schaltet zwischen normalem und umgekehrtem Print-Modus hin und her.
%VNON	Schaltet den inversen Print-Modus für numerische Felder ein, wenn der Profilparameter <i>PM=I</i> gesetzt ist oder %VON abgesetzt wurde. Dies ist die Standardeinstellung.
%VNOFF	Schaltet den normalen Print-Modus für numerische Felder ein, wenn der Profilparameter <i>PM=I</i> gesetzt ist oder %VON abgesetzt wurde. Dies ist relevant bei sensitiven Terminals, die numerische Werte nicht invertieren.

Kommando	Funktion
%VN	Schaltet für numerische Felder zwischen normalem und umgekehrtem Print-Modus hin und her.
%VAON	Wenn der Profilparameter $PM=I$ gesetzt ist, bewirkt diese Einstellung, dass alphanumerische Eingabefelder, die mit dem Feldattribut $PM=I$ definiert sind, mit dem numerischen Feldattribut zum Terminal gesendet werden. Dadurch ist gewährleistet, dass numerische Werte in solchen Feldern, die andernfalls von rechts nach links interpretiert würden, korrekt von links nach rechts interpretiert werden. Dazu muss die entsprechende Hardware-Funktion vorhanden sein. Als Alternative zu diesem Terminalkommando kann individuell das Feldattribut ($AD=Q$) verwendet werden.
%VAOFF	Alphanumerische Eingabefelder werden ohne das numerische Feldattribut zum Terminal gesendet. Dies ist die Standardeinstellung.
%VA	Schaltet zwischen %VAON und %VAOFF hin und her.
%VKON	Numerische Felder, für die eine Helprououtine oder Editiermaske definiert ist, werden intern von Natural in alphanumerisches Format umgesetzt (sonst könnte ein Benutzer z.B. in ein solches Feld kein Fragezeichen zum Aufrufen der Helprououtine eingeben). Bei invertiertem Print-Modus würde diese interne Umsetzung jedoch bewirken, dass ein numerischer Wert in einem solchen Feld von rechts nach links interpretiert würde. Bei numerischen Feldern, die mit $PM=I$ definiert sind und eine Helprououtine oder Editiermaske haben, verhindert %VKON eine interne Umsetzung in alphanumerisches Format und gewährleistet damit eine korrekte Interpretation der Feldwerte; die korrekte Anwendung einer Editiermaske und die Eingabe eines Fragezeichens ist hierbei nach wie vor gewährleistet.
%VKOFF	Schaltet die Wirkung von %VKON aus. Dies ist die Standardeinstellung.
%VK	Schaltet zwischen %VKON und %VKOFF hin und her.

Programmierschnittstelle: `USR0350N`. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

58

%W — Window-Verarbeitung

■ Größe und Position des Fensters auf dem physischen Bildschirm	136
■ Position des Fensters auf der logischen Seite	139
■ %WA und %WZ — Bildschirm speichern vor Fenster	141



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.

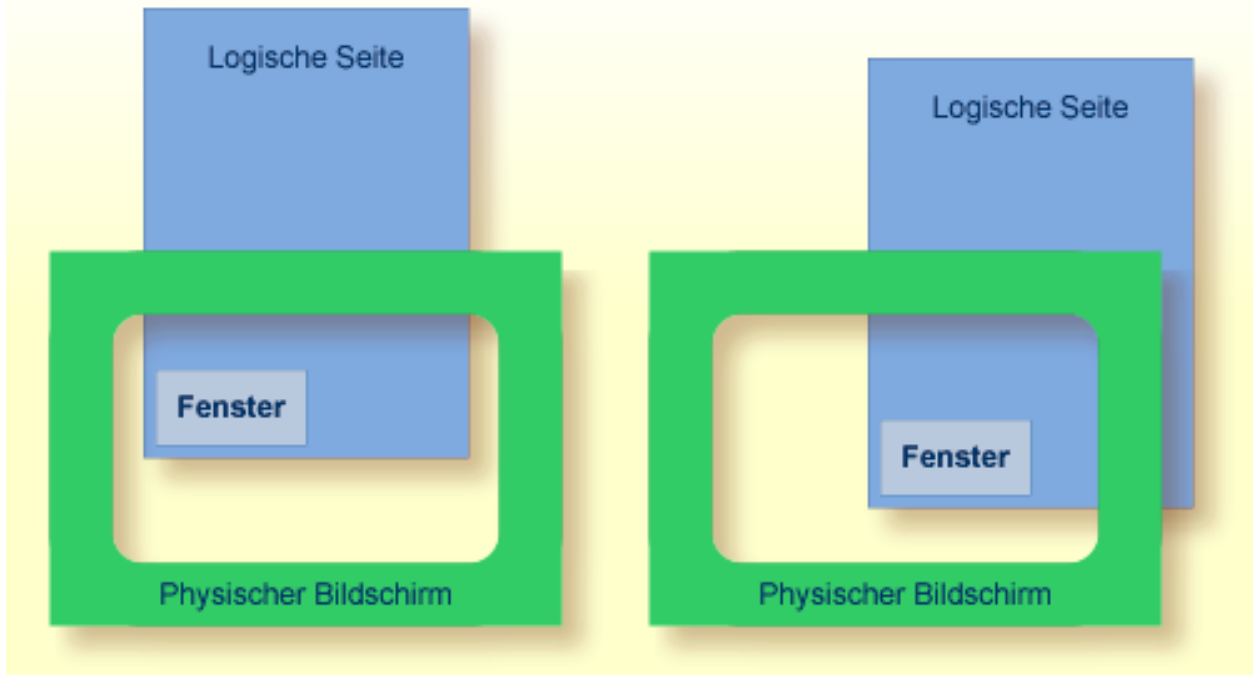
Es gibt zwei Arten von Window-Kommandos:

- 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 Bildschirmecke 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	<p>Die Spaltenlänge (vertikale Ausdehnung) des Fensters (einschließlich Rahmen, falls angegeben) wird auf <i>nn</i> Zeilen gesetzt.</p> <p>Wenn Sie <i>nn</i> nicht oder größer als auf den Bildschirm passend, angeben, wird die vertikale Ausdehnung auf größtmögliche Spaltenlänge gesetzt, d.h. bis zum unteren Bildschirmrand.</p>

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.

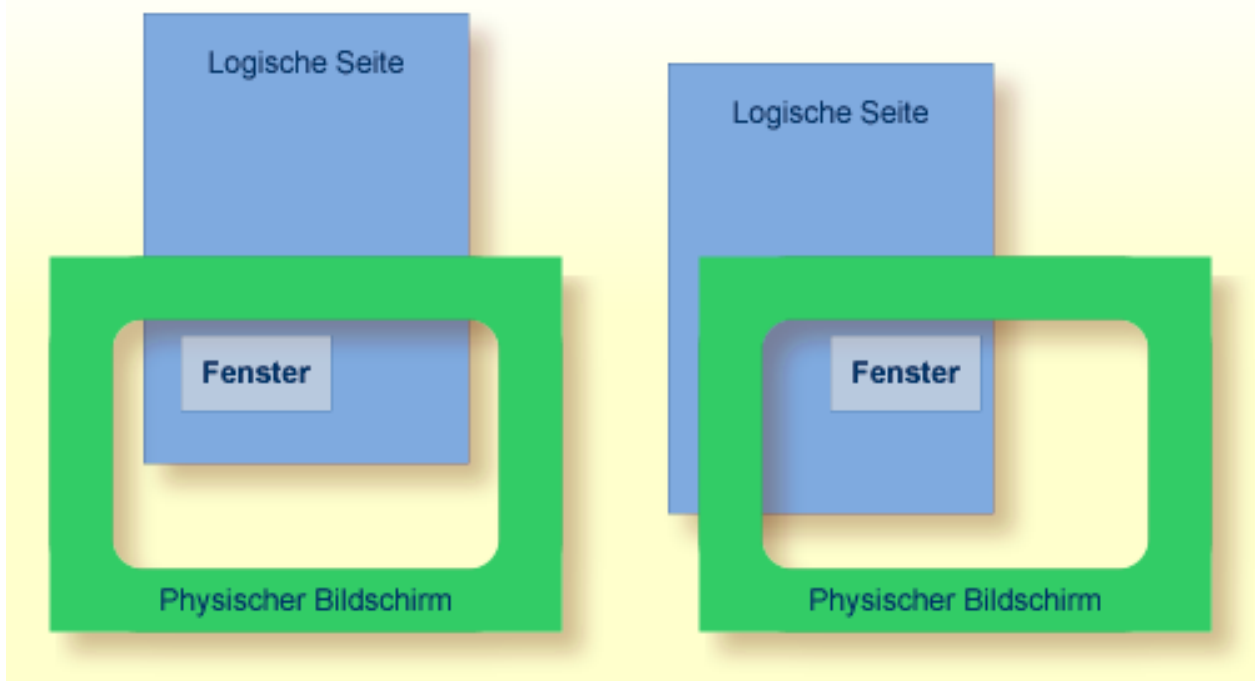
Wird ein Fenster nicht korrekt definiert, so wird ein solches Kommando entweder ignoriert oder die Fenstergröße und -position im Rahmen der physischen Möglichkeiten angepasst.

Kommando	Funktion
%WF	<p>Schaltet die Rahmung (Frame) ein. Die Größe des Fensters wird durch einen Rahmen angezeigt.</p> <p>Falls das Fenster kleiner als 4 Zeilen mal 13 Spalten ist, ist der Rahmen nicht sichtbar.</p>
%WM	<p>Schaltet die Rahmung aus. Die Größe des Fensters wird nicht durch einen Rahmen angezeigt.</p> <p>Durch Ausschalten des Rahmens ändert sich die Größe des Fensters nicht (lediglich die Größe des im Fenster sichtbaren Seitenausschnitts).</p>
%WO	<p>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).</p> <p>Um die Wirkung von %WO auszuschalten, geben Sie %WO noch einmal ein (oder %WD).</p>
%WP	<p>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.</p> <p>Um die Wirkung von %WP auszuschalten, verwenden Sie %WD.</p>
%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>%Wlll,ccc</code>	Die Seite wird so verschoben, dass der durch Zeile <i>lll</i> /Spalte <i>ccc</i> 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< <i>n</i>	Das Fenster wird um <i>n</i> Stellen nach links verschoben ($0 \leq n \leq \text{Zeilenlänge der Seite}$).
%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> <i>n</i>	Das Fenster wird um <i>n</i> Stellen nach rechts verschoben ($0 \leq n \leq \text{Zeilenlänge der Seite}$).
%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+ <i>n</i>	Das Fenster wird um <i>n</i> Zeilen nach unten verschoben ($0 \leq n \leq \text{Spaltenlänge der Seite}$). (*)
%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- <i>n</i>	Das Fenster wird um <i>n</i> Zeilen nach oben verschoben ($0 \leq n \leq \text{Spaltenlänge der Seite}$).
%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 ENTER drücken, verschiebt sich die Seite um ein Fenster nach unten, bis das Seitenende erreicht ist. Mit dem nächsten ENTER 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 ENTER 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.

Wenn Sie mehrere Parameter mit dem `%W`-Kommando angeben, beachten Sie bitte, dass Sie auf Großrechnern nach dem Prozentzeichen (%) höchstens 24 Zeichen angeben können; alle weiteren Zeichen werden ignoriert.

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

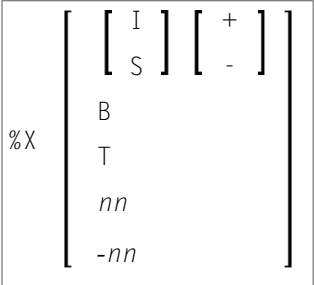
Kommando	Funktion
<code>%WA</code>	<p>Mit <code>%WA</code> aktivieren Sie die Funktion „Bildschirm speichern vor Fenster“. Wenn diese Funktion aktiv ist und ein Fenster geöffnet wird, werden alle aktiven Bildschirmdaten, die von dem Fenster überlagert werden, vorher gespeichert. Wenn das Fenster verschoben 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</p>

Kommando	Funktion
	<p>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 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.

59

%X — Steuerung der Statistikzeile/Infoline

■ Infoline	145
■ Statistikzeile	145



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

Programmierschnittstelle (API): USR1005N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Kommando	Funktion
%X+	Schaltet die Anzeige der Statistikzeile/Infoline ein.
%X-	Schaltet die Anzeige der Statistikzeile/Infoline aus.
%X	Schaltet die Anzeige der Statistikzeile/Infoline ein bzw. wieder aus.
%XI+	Zeigt die Zeile als Infoline an.
%XI-	Zeigt die Zeile als Statistikzeile an.
%XI	Schaltet zwischen Infoline- und Statistik-Anzeige hin und her.
%XS+	Zeigt zusätzliche Statistik-Informationen an (siehe unten).
%XS-	Zeigt wieder die ursprünglichen Statistik-Informationen an.
%XS	Schaltet zwischen zusätzlichen und ursprünglichen Statistik-Informationen hin und her.
%XB	Zeigt die Statistikzeile/Infoline am unteren Bildschirmrand an (B ottom).
%XT	Zeigt die Statistikzeile/Infoline am oberen Bildschirmrand an (T op).
%Xnn	Zeigt die Statistikzeile/Infoline in der <i>nn</i> -ten Zeile des Schirms an. Ist die angegebene Zeilennummer <i>nn</i> außerhalb des aktuellen Schirms, wird die Statistikzeile/Infoline nicht angezeigt.
%X- <i>nn</i>	Zeigt die Statistikzeile/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 Statistikzeile/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.

Statistikzeile

Wenn die Zeile als Statistikzeile verwendet wird, liefert sie folgende Informationen:

Kommando-Option	Funktion
IO	Anzahl der während der letzten Bildschirmoperation an den Bildschirm übergebenen Bytes.
AIO	Durchschnittliche Anzahl der je Bildschirmoperation an den Bildschirm übergebenen Bytes (seit Beginn der Natural-Session).
L	Logische Zeilennummer (L = Line) der oberen Zeile des aktiven Natural-Windows, gezählt auf der logischen Seite.
C	Logische Spaltennummer (C = Column) der äußersten linken Spalte des aktiven Natural-Windows, gezählt auf der logischen Seite.
LS	Logische Zeilenlänge (LS = Line Size) der aktuellen logischen Seite (wie mit dem Session-Parameter <code>LS</code> definiert).
PS	Logische Seitenlänge (PS = Page Size) der aktuellen logischen Seite (wie mit dem Session-Parameter <code>PS</code> definiert).
PLS	Physische Zeilenlänge des aktiven Natural-Fensters.
PCS	Physische Spaltenlänge des aktiven Natural-Fensters.
FLD	Die Anzahl der auf dem letzten Schirm generierten Felder.
CLS	Die Anzahl der Programm-Aufrufe während des letzten Terminal-I/Os.
ADA	Die Anzahl der Adabas-Aufrufe während des letzten Terminal-I/Os.

Zusätzliche Informationen, die mit `%XS+` angezeigt werden:

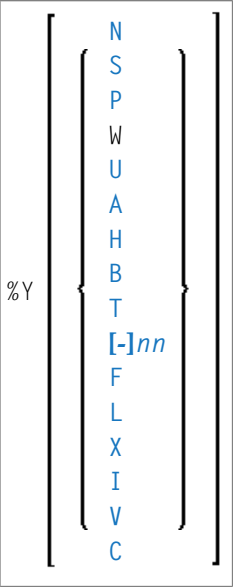
Kommando-Option	Funktion
MIN	Kürzeste Dauer (in Sekunden) der Natural-Aktivitäten zwischen zwei Terminal-I/Os während der laufenden Natural-Session.
MAX	Längste Dauer (in Sekunden) der Natural-Aktivitäten zwischen zwei Terminal-I/Os während der laufenden Natural-Session.
AVR	Durchschnittliche Dauer (in Sekunden) der Natural-Aktivitäten zwischen zwei Terminal-I/Os während der laufenden Natural-Session.
LST	Dauer (in Sekunden) der Natural-Aktivitäten zwischen dem vorletzten und dem vorvorletzten Terminal-I/O.

Weitere Informationen siehe *Statistikzeile/Infoline - Terminalkommando %X* (im *Leitfaden zur Programmierung*).

60

%Y — Steuerung der PF-Tastenleiste


■ Anzeigeformat der Funktionstastenleiste	148
■ Ein- oder zweizeilige Anzeige	149
■ Positionierung der Funktionstastenleiste	149
■ Bereich der angezeigten Funktionstasten	149
■ Intensivierte bzw. inverse Anzeige der Funktionstastenleiste	150
■ Farbe der Funktionstastenleiste	150
■ Cursor-Sensitivität	150



Alternativ können Sie Folgendes angeben:

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

Das Terminalkommando %Y steuert die Anzeige der Natural-Funktionstastenleiste.

 **Anmerkung:** Auf graphischen Benutzeroberflächen wird dieses Kommando ignoriert.

Programmierschnittstelle (API): USR1005N. Siehe auch *SYSEXT Utility* in der Dokumentation *Debugger und Dienstprogramme*.

Anzeigeformat der Funktionstastenleiste

Kommando	Funktion
%YN	Zeigt die Funktionstastenleiste im normalen tabularischen Software-AG-Format an.
%YS	Zeigt die Funktionstastenleiste im sequentiellen Format an, d.h. die Tasten werden nacheinander angezeigt, und zwar nur die Tasten, denen Namen zugewiesen wurden (PF1= <i>wert</i> , PF2= <i>wert</i> usw.).
%YP	Zeigt die Funktionstastenleiste im sequentiellen PC-Format; entspricht %YS, außer dass vor den Namen <i>F</i> <i>n</i> = statt <i>PF</i> <i>n</i> = steht.
%YW	Entspricht %YP — gilt jedoch nur, wenn die Funktionstastenleiste innerhalb eines Fensters angezeigt wird.
%YU	Hebt die Wirkung von %YW wieder auf.

Ein- oder zweizeilige Anzeige

Kommando	Funktion
%YA	(alle) Zeigt beide PF-Tastenzeilen an.
%YH	(halb) Zeigt nur eine PF-Tastenzeile an: beim normalen Software-AG-Anzeigeformat (%YN) ist dies die Zeile mit den Funktionstasten-Namen; bei den anderen beiden Anzeigeformaten (%YS und %YP) ist dies die obere Zeile.

Positionierung der Funktionstastenleiste

Kommando	Funktion
%YB	Zeigt die Funktionstastenleiste am unteren Bildschirmrand an (B = Bottom).
%YT	Zeigt die Funktionstastenleiste am oberen Bildschirmrand an (T = Top).
%Ynn	Zeigt die Funktionstastenleiste in der nn-ten Zeile des Schirms an.
%Y- <i>nn</i>	Zeigt die Funktionstastenleiste in der nn-ten Zeile von unten auf dem Schirm an. Ist die angegebene Zeilennummer <i>nn</i> oder <i>-nn</i> außerhalb des aktuellen Schirms, wird die Funktionstastenleiste nicht angezeigt.

Bereich der angezeigten Funktionstasten

Kommando	Funktion
%YF	Zeigt den ersten Bereich von Funktionstasten (d.h. in der Regel 1 bis 12) an (F = First).
%YL	Zeigt den letzten Bereich von Funktionstasten (d.h. in der Regel 13 bis 24) an (L = Last).
%YX	Schaltet zwischen beiden Anzeigen hin und her.

Intensivierte bzw. inverse Anzeige der Funktionstastenleiste

Kommando	Funktion
%YI	Zeigt die Funktionstastenleiste intensiviert an. Durch nochmaliges Eingeben von %YI erhalten Sie wieder die normale nicht intensivierte Anzeige.
%YV	Zeigt die Funktionstastenleiste invers an. Durch nochmaliges Eingeben von %YV erhalten Sie wieder die normale nicht inverse Anzeige.

Farbe der Funktionstastenleiste

Kommando	Funktion
%Y= <i>color-code(s)</i>	<p>Zeigt die Funktionstastenleiste in den angegebenen Farben an. Sie können die gleichen Farbcodes (<i>color-codes</i>) angeben wie beim Session-Parameter CD.</p> <p>Sie können bis zu drei <i>color-codes</i> angeben: Der erste <i>color-code</i> gilt für die erste Funktionstastenzeile (in der die Funktionstastennummern angezeigt werden), der zweite <i>color-code</i> gilt für die zweite Funktionstastenzeile (in der die Namen, die den Funktionstasten zugeordnet sind, angezeigt werden), und der dritte <i>color-code</i> gilt für den Hintergrund beider Zeilen.</p> <p>Mit %Y=GRPIYE würden Sie beispielsweise die Zeichen der ersten Zeile in Grün, die Zeichen der zweiten Zeile in Pink, und den Hintergrund beider Zeilen in Gelb erhalten.</p> <p>Anmerkung: Auf Großrechnern wird der dritte <i>color-code</i> ignoriert.</p>

Cursor-Sensitivität

Kommando	Funktion
%YC	<p>Dieses Kommando macht die Funktionstastenleiste cursor-sensitiv. Sie reagiert dann wie eine Aktionsleiste auf einem PC-Bildschirm: der Benutzer wählt mit dem Cursor lediglich den Namen oder die Nummer der gewünschten Funktionstaste aus und drückt ENTER, und Natural reagiert, als ob die betreffende Funktionstaste gedrückt worden wäre.</p> <p>Durch nochmaliges Eingeben von %YC schalten Sie die Cursor-Sensitivität wieder aus.</p>

Kommando	Funktion
	Durch Verwendung von %YC in Verbindung mit Software-AG-Anzeigeformat (%YN) und einzeliger Anzeige (%YH) können Sie Ihre Anwendungen mit einer sehr komfortablen Aktionsleisten-Verarbeitung ausstatten: der Benutzer wählt dann bloß noch den Namen einer Funktion mit dem Cursor aus und drückt ENTER, und die Funktion wird ausgeführt.

Um die Anzeige der Funktionstastenleiste in einem Programm zu aktivieren, verwenden Sie den Session-Parameter `KD=ON`.

Siehe auch *Steuerung der Funktionstastenleiste – Terminalkommando %Y im Leitfaden zur Programmierung*.

61

%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.

Stichwortverzeichnis

Symbole

%
 Terminalkommando, 19
%%
 Terminalkommando, 21
%*
 Terminalkommando, 25
%+
 Terminalkommando, 33
%-
 Terminalkommando, 33
%.
 Terminalkommando, 21
%.P
 Terminalkommando, 27
%.S
 Terminalkommando, 29
%/
 Terminalkommando, 31
%<RDC
 terminal command, 109
%<TECH
 Terminalkommando, 35
%<TEST
 Terminalkommando, 37
%=
 Terminalkommando, 39
%A
 Terminalkommando, 41
%B
 Terminalkommando, 43
%B=
 Terminalkommando, 45
%C
 Terminalkommando, 47
%CC
 Terminalkommando, 49
%CS
 Terminalkommando, 49
%D
 Terminalkommando, 51
%D=
 Terminalkommando, 53
%DUE
 Terminalkommando, 55
%E
 Terminalkommando, 57
%E=

 Terminalkommando, 59
%F
 Terminalkommando, 61
%F=
 Terminalkommando, 63
%FM
 Terminalkommando, 65
%G
 Terminalkommando, 67
%H
 Terminalkommando, 69
%I
 Terminalkommando, 71
%J
 Terminalkommando, 73
%K
 Terminalkommando, 75
%KP
 Terminalkommando, 75
%L
 Terminalkommando, 77
%L=
 Terminalkommando, 79
%M
 Terminalkommando, 81
%MSGSF
 Terminalkommando, 85
%N
 Terminalkommando, 87
%O
 Terminalkommando, 89
%P
 Terminalkommando, 91
%P=
 Terminalkommando, 93
%Q
 Terminalkommando, 101
%QO
 Terminalkommando, 103
%QS
 Terminalkommando, 105
%R
 Terminalkommando, 107
%RM
 Terminalkommando, 111
%RN
 Terminalkommando, 113
%RO
 Terminalkommando, 115
%S

- Terminalkommando, 117
- %T
 - Terminalkommando, 119
- %T*
 - Terminalkommando, 123
- %T=
 - Terminalkommando, 125
- %Tll/cc
 - Terminalkommando, 121
- %TRE
 - Terminalkommando, 127
- %TRI
 - Terminalkommando, 129
- %U
 - Terminalkommando, 131
- %V
 - Terminalkommando, 133
- %W
 - Terminalkommando, 135
- %X
 - Terminalkommando, 143
- %Y
 - Terminalkommando, 147
- %Z
 - Terminalkommando, 153

T

Terminalkommandos, vii