

## **Natural für Großrechner**

### **Natural benutzen**

Version 4.2.6 für Großrechner

Februar 2010

Dieses Dokument gilt für Natural für Großrechner ab Version 4.2.6 für Großrechner.

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

Copyright © 1979-2010 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, Vereinigte Staaten von Amerika, und/oder ihre Lizenzgeber..

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

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

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

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

---

## Inhaltsverzeichnis

1 Natural benutzen .....	1
2 Natural-Session aufrufen und beenden .....	3
Natural-Online-Session aufrufen .....	4
Vorgehensweise nach dem Start der Session .....	4
Natural-Online-Session beenden .....	5
Natural-Batch-Session aufrufen und beenden .....	6
3 Kommandos und Menü-Funktionen benutzen .....	7
Welche Arten von Natural-Kommandos gibt es? .....	8
Natural-Kommandos benutzen .....	9
Natural-Menüs benutzen .....	13
4 Hilfe benutzen .....	17
Allgemeine Informationen – Natural-Help-Utility .....	18
Hilfe zu Natural-Meldungen .....	21
Informationen zu Menüs und Felder .....	27
5 Natural-Libraries benutzen .....	29
In einer Library enthaltene Objekte .....	30
Library-Arten .....	31
Steplib-Libraries .....	31
Suchreihenfolge bei der Objekt-Ausführung .....	33
Standard-Library-Zuweisung .....	34
Einloggen in eine Library .....	35
Neue Library anlegen .....	36
Libraries auflisten .....	36
Objekte in einer Library auflisten .....	39
Objektliste drucken .....	40
Objekte in einer Library suchen .....	41
Libraries kopieren, verschieben und umbenennen .....	46
Objekte in einer Library löschen .....	48
6 Natural-Objekte pflegen und ausführen .....	53
Natural-Objekte – Einführung .....	54
Natural-Editoren oder Utilities benutzen .....	56
Objekte auswählen und anzeigen .....	59
Objekte erstellen und bearbeiten .....	60
Objekte prüfen und testen .....	64
Objekte speichern und katalogisieren .....	66
Objekt-Verzeichnis-Informationen anzeigen .....	69
Objekte kopieren .....	70
Objekte drucken .....	71
Objekte umbenennen .....	73
Objekte verschieben .....	75
Objekte löschen .....	76
Programme ausführen .....	78
7 Natural-Hauptmenü (Main Menu) .....	81

Natural-Hauptmenü (Main Menu) aufrufen .....	82
Development Functions (Entwicklungsfunktionen) .....	84
Development Environment Settings (Entwicklungsumgebungseinstellungen) .....	90
Maintenance and Transfer Utilities (Wartungs- und Übertragungs-Utilities) .....	91
Debugging and Monitoring Utilities (Utilities zur Fehlerbeseitigung und Überwachung) .....	92
Example Libraries (Beispielprogramm-Libraries) .....	93
Other Products (Sonstige Produkte) .....	94
8 Druck- und Arbeitsdateien (Printfiles und Workfiles) .....	95
9 Natural-Umgebung konfigurieren .....	97
Profilparameter benutzen .....	98
Session-Parameter benutzen .....	101
Parametrisierungsebenen .....	104
10 Regeln und Namenskonventionen .....	105
Standard-Zeichenzuweisungen .....	106
Namenskonventionen für Objekte .....	106
Namenskonventionen für Libraries .....	107
Namenskonventionen für Benutzervariablen .....	108
Stichwortverzeichnis .....	111

# 1 Natural benutzen







---




Diese Dokumentation ist dazu gedacht, Sie in die Benutzung von Natural auf Großrechnern einzuführen.

Die in dieser Dokumentation enthaltenen Anleitungen und Verfahren beziehen sich auf die standardmäßigen Voreinstellungen, die im Auslieferungszustand der Natural-Original-Software vorhanden sind. Es werden nicht alle Funktionsmerkmale und Optionen ausführlich behandelt. Weiterführende Informationen finden Sie in den Dokumenten, die an entsprechender Stelle referenziert sind.

Die in dieser Dokumentation gezeigten Beispielmasken und das hier beschriebene Verhalten von Natural können von den Ergebnissen in Ihrer Umgebung abweichen. So kann bei Ihnen zum Beispiel die Kommandozeile oder die Nachrichtenzeile an einer anderen Stelle auf der Maske erscheinen oder die Ausführung eines bestimmten Natural-Kommandos durch Sicherheitseinstellungen geschützt sein. Die Standardvorgaben in Ihrer Umgebung sind abhängig von den von Ihrem Natural-Administrator vorgenommenen Profilparametereinstellungen.

Diese Dokumentation ist in die folgenden Abschnitte untergliedert:

	<b>Natural-Session aufrufen und beenden</b>	Wie Sie eine Natural-Session aufrufen und beenden.
	<b>Kommandos und Menüfunktionen benutzen</b>	Wie Sie Natural-Kommandos absetzen und Menü-Funktionen aufrufen.
	<b>Hilfe benutzen</b>	Beschreibt die in Natural vorhandenen Online-Hilfefunktionen und die verschiedenen Hilfearten.
	<b>Natural-Libraries benutzen</b>	Wie Sie eine Natural-Library benutzen und wie Sie auf Libraries zugreifen und ihre Inhalte anzeigen und pflegen können.
	<b>Natural-Objekte pflegen und ausführen</b>	Wie Sie ein Natural-Objekt erstellen, zeigen, ändern, löschen und ausführen.
	<b>Natural-Hauptmenü (Main Menu)</b>	Beschreibt das Natural-Hauptmenü <b>Main Menu</b> und die dahinter liegenden Untermenüs, die den Zugang zu

		Entwicklungsfunktionen, Einstellungen, Utilities (Dienstprogramme) und Beispiel-Libraries ermöglichen.
	<b>Druck- und Arbeitsdateien (Printfiles und Workfiles)</b>	Informationen über die in Natural vorhandenen Printfiles und Workfiles (Druck- und Arbeitsdateien).
	<b>Natural-Umgebung konfigurieren</b>	Wie Sie Natural-Parameter setzen, um Ihre Natural-Umgebung an Ihre persönlichen Anforderungen anzupassen und Abläufe zu standardisieren oder zu automatisieren.
	<b>Regeln und Konventionen</b>	Natural-spezifische Namenskonventionen für Objekte, Libraries (Bibliotheken) und Benutzervariablen.



## 2 Natural-Session aufrufen und beenden

---

▪ Natural-Online-Session aufrufen .....	4
▪ Vorgehensweise nach dem Start der Session .....	4
▪ Natural-Online-Session beenden .....	5
▪ Natural-Batch-Session aufrufen und beenden .....	6

Sie können eine Natural-Session für Online- oder Batch-Betrieb aufrufen.

**Verwandte Themen:**

- [Kommandos und Menüfunktionen benutzen](#)
- [Natural-Hauptmenü \(Main Menu\)](#)

Dieser Abschnitt behandelt folgende Themen:

## Natural-Online-Session aufrufen

---

Wie Sie eine Natural-Session für den Online-Betrieb aufrufen, hängt von Ihrer System-Umgebung ab. Fragen Sie Ihren Natural-Administrator, wie Sie Natural aufrufen sollen. Falls Natural Security installiert ist, kann es sein, dass der Zugriff auf bestimmte Libraries (Bibliotheken) sowie der Funktionsumfang, der Ihnen zur Verfügung steht, eingeschränkt sind.

## Vorgehensweise nach dem Start der Session

---

Nachdem Sie Natural gemäß der an Ihrem Standort vorgesehenen Prozedur aufgerufen haben, erscheint, in Abhängigkeit von den in Ihrer Umgebung eingerichteten Standardvorgaben, entweder das Natural-Hauptmenü **Main Menu** oder ein Bildschirm mit einer NEXT-Eingabezeile oder ein vom Benutzer definiertes Startmenü.

Wenn Sie das Natural-Hauptmenü **Main Menu** benutzen, werden Sie schrittweise durch die Natural-Entwicklungsfunktionen, Umgebungseinstellungen, Utilities (Dienstprogramme) und Libraries (Bibliotheken) mit Beispielpogrammen geführt. Informationen zu den Funktionen im Natural-Hauptmenü und in den untergeordneten Menüs und Anleitungen zum Aufrufen und Verlassen des Natural-Hauptmenüs finden Sie im Abschnitt [Natural-Hauptmenü \(Main Menu\)](#).

Sie können eine Natural-Funktion wahlweise über das Natural-Hauptmenü **Main Menu** und seine Untermenüs oder direkt über die NEXT-Eingabezeile aufrufen.

Wenn Sie die NEXT-Eingabezeile benutzen, führen Sie entweder ein Natural-Systemkommando oder ein Natural-Objekt des Typs Programm aus. Weitere Informationen finden Sie unter [Natural-Kommandos benutzen](#).

## Natural-Online-Session beenden

Sie können eine Natural-Session aus dem Natural-Hauptmenü **Main Menu**, einer der Natural-Eingabeaufforderungszeilen oder einem Natural-Objekt beenden.

### ► Um eine Natural-Online-Session zu beenden

- Benutzen Sie eine der folgenden Möglichkeiten:

- Im Natural-Hauptmenü **Main Menu**: Drücken Sie PF3 (Exit).

Oder:

Im Natural-Hauptmenü **Main Menu**: Wählen Sie **Exit Natural Session**.

- In der *Kommandozeile* (Command ==>): Geben Sie eines der folgenden Kommandos ein:

```
.
```

(ein Punkt)

Oder:

```
FIN
```

Oder:

```
% %  
% %
```

- In der NEXT-Eingabezeile (siehe *NEXT- und MORE-Eingabeaufforderungszeilen*): Geben Sie eines der folgenden Kommandos ein:

```
FIN
```

Oder:

```
% %  
% %
```

Oder:

```
CLEAR
```

- In der MORE -Eingabezeile (siehe *NEXT- und MORE-Eingabeaufforderungszeilen*): Geben Sie eines der folgenden Kommandos ein:

```
FIN
```

Oder:

% / %

- Innerhalb eines Natural-Objekts:

Führen Sie ein `TERMINATE`-Statement aus.

### Verwandte Themen:

- [Natural-Kommandos benutzen.](#)
- [Natural-Objekte – Einführung](#)

## Natural-Batch-Session aufrufen und beenden

---

Natural besitzt eine Schnittstelle, über die Sie Objekt-Operationen im Batch-Betrieb ausführen können.

### ▶ Um eine Natural-Session für den Batch-Betrieb aufzurufen

- Verfahren Sie gemäß der Anleitung im Abschnitt *Subtasking Session Support for Batch Environments* im Kapitel *Natural in Batch Mode (All Environments)* in der *Operations*-Dokumentation.

Beendet wird die Natural-Batch-Session, wenn im Laufe der Session eine der folgenden Bedingungen auftritt:

- Ein `FIN`-Kommando im Natural Command Input Dataset `CMSYININ`,
- eine End-of-Input-Bedingung im Natural Command Input Dataset `CMSYININ`,
- ein `TERMINATE`-Statement in einem zurzeit ausgeführten Natural-Objekt.

### Verwandtes Thema:

- [Natural in Batch Mode - Operations-Dokumentation](#)

# 3 Kommandos und Menü-Funktionen benutzen

---

- Welche Arten von Natural-Kommandos gibt es? ..... 8
- Natural-Kommandos benutzen ..... 9
- Natural-Menüs benutzen ..... 13

Sie können während einer Natural-Session einen Bedienschritt ausführen, indem Sie entweder eine Menü-Funktion oder ein (Direkt-)Kommando benutzen. Wenn Sie ein Natural-Kommando benutzen, haben Sie den Vorteil, dass Sie einen Bedienschritt unmittelbar, d.h. ohne umständliche Navigation durch mehrere Menüs ausführen können.

Ein Natural-Objekt des Typs Programm können Sie auch ausführen, ohne ein Natural-Systemkommando zu benutzen; siehe [Programme ausführen](#).

Dieser Abschnitt stellt die verschiedenen Arten von Kommandos vor, die es bei Natural gibt, und beschreibt, wie Sie ein Kommando absetzen bzw. wie Sie ein Menü benutzen können, um eine Funktion auszuführen.

### Verwandtes Thema:

- [Natural-Hauptmenü \(Main Menu\)](#)

Dieser Abschnitt behandelt folgende Themen:

## Welche Arten von Natural-Kommandos gibt es?

---

In diesem Abschnitt werden die verschiedenen Arten von Kommandos vorgestellt, die Ihnen zur Bedienung von Natural zur Verfügung stehen:

- [Systemkommandos](#)
- [Terminalkommandos](#)
- [Editor- und Utility-Kommandos](#)

### Systemkommandos

Systemkommandos führen Funktionen aus, die Sie zum Erstellen, Pflegen oder Ausführen von Natural-Prorammiereobjekten benötigen. Außerdem gibt es Systemkommandos, die Sie zum Überwachen und Verwalten Ihrer Natural-Umgebung benutzen können.

### Verwandte Themen in der Systemkommandos-Dokumentation:

- [Systemkommandos nach Funktion](#)
- [Systemkommandos](#)
- [Kommandosyntax](#)

## Terminalkommandos

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

- die Bildschirm-Anzeige und das Bildschirm-Format, 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 festgelegt werden kann. Das Standardsteuerzeichen ist das Prozentzeichen (%).

### Verwandte Themen:

- *Bildschirmgestaltung - Leitfaden zur Programmierung*
- *Daten vom Bildschirm kopieren - Leitfaden zur Programmierung*
- *Terminalkommandos nach Funktionsgruppen - Terminalkommandos-Dokumentation*
- *Terminalkommandos (Übersicht) Terminalkommandos-Dokumentation*
- *[Session-Parameter benutzen](#)*

## Editor- und Utility-Kommandos

Zusätzlich zu den Systemkommandos und Terminalkommandos stehen Ihnen innerhalb der verschiedenen Natural-Editoren und Utilities weitere Kommandos zur Verfügung, die im Gegensatz zu den oben erwähnten Kommandos nicht global, sondern nur lokal in der entsprechenden Umgebung gültig sind. Diese Kommandos und ihre Anwendung werden in der Dokumentation der Editoren und Utilities beschrieben.

## Natural-Kommandos benutzen

Sie können die zuvor genannten, global geltenden Natural-Kommandos an den folgenden Stellen eingeben:

- In der Natural-Kommandozeile im Natural-Hauptmenü (Main Menu) oder auf dem Bildschirm einer Natural-Utility oder eines Systemkommandos (siehe auch [Beispiel](#) weiter unten).
- An einem Eingabeaufforderungszeichen in einem der Natural-Editoren, z.B. hinter dem Größer-Zeichen (>) im Programm-Editor (siehe [Beispiel](#) weiter unten).
- In der NEXT- oder MORE-Eingabeaufforderungszeile.

Bei einigen Utilities müssen Sie vor einem Systemkommando ein spezielles Zeichen eingeben, z.B. zwei nach rechts geneigte Schrägstriche (//). Entsprechende Hinweise erhalten Sie in der Dokumentation der betreffenden Utilities.

Bei manchen Systemkommandos können Sie einen oder mehrere Parameter bzw. Operanden mit angeben, die bewirken, dass eine bestimmte Funktion direkt ausgeführt wird. Andernfalls, wenn Sie ein solches Kommando ohne weitere Angaben absetzen, erscheint zunächst ein Auswahlmenü. Siehe Beispiel eines *Systemkommandos* weiter unten.

Ein Natural-Terminalkommando können Sie in der Kommandozeile, in jeder der zuvor genannten Eingabeaufforderungszeilen oder in einem beliebigen eingabefähigen Feld auf einer Bildschirmmaske eingeben.

Ein Natural-Editor- oder Natural-Utility-Kommando können Sie in der Eingabeaufforderungszeile oder in der Kommandozeile des betreffenden Editors oder der betreffenden Utility eingeben.

Bei der Eingabe eines Natural-Kommandos brauchen Sie *nicht* auf Groß-/Kleinschreibung zu achten. Natural akzeptiert auch gemischte Schreibweisen. Nach Eingabe des Kommandos drücken Sie die EINGABE-Taste, um das Kommando zu bestätigen (dieser Vorgang wird in der Natural-Dokumentation normalerweise nicht erwähnt, siehe Anmerkung weiter unten). Danach wird führt Natural das eingegebene Kommando aus oder zeigt ein spezielles Fenster an, in dem Sie die Ausführung des Kommandos (z.B. DELETE) ausdrücklich bestätigen müssen.



**Anmerkung:** Wenn Sie in der Natural-Dokumentation beispielsweise aufgefordert werden, „geben Sie ein Kommando ein“, so bedeutet dies: „tippen Sie das Kommando ein, und drücken Sie EINGABE“. Falls Sie eine andere Taste zum Bestätigen des Kommandos drücken müssen, wird das in der Dokumentation ausdrücklich erwähnt.

Dieser Abschnitt behandelt folgende Themen:

- [Kommandozeile](#)
- [NEXT- und MORE-Eingabeaufforderungszeilen](#)
- [Beispiel für ein Systemkommando](#)

### Kommandozeile

Standardmäßig befindet sich die Kommandozeile im unteren Bereich des Bildschirms, und zwar oberhalb der PF-Tastenzeilen:



```

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                               Canc

```

## NEXT- und MORE-Eingabeaufforderungszeilen

Die NEXT-Eingabeaufforderungszeile ist immer dann vorhanden, wenn noch kein Natural-Programm (z.B. das Programm, das das Natural-Hauptmenü **Main Menu** aufruft) gestartet wurde. Die NEXT-Aufforderung zeigt an, dass Natural bereit ist, Ihre nächste Eingabe entgegenzunehmen.

Die MORE-Eingabeaufforderungszeile erscheint während der Ausführung eines Programms und zeigt an, dass zusätzliche Ausgabedaten zur Verfügung stehen. Drücken Sie EINGABE, um diese Ausgabedaten anzuzeigen. Sie können auch in der MORE-Eingabeaufforderungszeile ein Kommando absetzen. Dann wird das Programm, das zurzeit ausgeführt wird, beendet und das Kommando wird ausgeführt.

Standardmäßig erscheint die NEXT- bzw. die MORE-Eingabeaufforderungszeile in der linken oberen Ecke des Bildschirms.

```
NEXT
```

```
LIB=TEST
```

## Beispiel für ein Systemkommando

In diesem folgenden Beispiel wird beschrieben, wie Sie ein Natural-Systemkommando absetzen, um ein Objekt mit einem Editor zu bearbeiten.

### ▶ Um einen Natural-Editor für ein Objekt aufzurufen

- 1 Geben Sie in einer Eingabeaufforderungszeile ein Natural-Systemkommando und gegebenenfalls einen oder mehrere Parameter oder Operanden ein.

Zum Beispiel:

```
EDIT P PROGX
```

Dabei ist `EDIT` der Name des Systemkommandos, `P` der Objekttyp (Programm) und `PROGX` der Name des zu bearbeitenden Objekts.

- 2 Drücken Sie EINGABE.

Da Sie für den Objekttyp ein P eingegeben haben, wird der Natural-Programm-Editor aufgerufen. Im Arbeitsbereich des Editors befindet sich schon der Sourcecode des zu bearbeitenden Programms PROGX, zum Beispiel:

```
>                                     > + Program   PROGX   Lib TEST
All  .....1.....2.....3.....4.....5.....6.....7...
0010 READ (3) EMPLOYEES BY NAME
0020 DISPLAY NAME
0030 END
0040
0050
0060
0070
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
0210
0220
0230
0240
0250
0260
0270
0280
.....1.....2.....3.....4.....5..... S 3   L 1
```

Zum Vergleich können Sie unter [Beispiel für eine Menü-Funktion](#) lesen, wie Sie diese Aktion unter Benutzung der entsprechenden Menüfunktionen ausführen können.

**Verwandtes Thema:**

- *Kommandosyntax-Beispiel - Systemkommandos-Dokumentation*

## Natural-Menüs benutzen

---

Jeder Natural-Menüschirm bietet Ihnen eine Anzahl von Funktionen zur Auswahl an. Wie Sie eine Funktion aufrufen, hängt davon ab, wie das Menü aufgebaut ist und welche Auswahlmöglichkeiten vorhanden sind.

Es gibt mehrere Möglichkeiten, um eine Menüfunktion auszuwählen und aufzurufen:

- Auf manchen Menüschirmen befindet sich vor jeder Funktion ein Eingabefeld.
- Auf manchen Menüschirmen steht vor jeder Funktion ein ein- oder zweibuchstabiger Code, den Sie in dem **Code**-Feld auf dem Menüschirm eingeben können.
- Außerdem sind Funktionstasten (PF-Tasten) vorhanden, die mit einer der Menüfunktionen belegt sein können.

Die PF-Tasten und die Funktion, mit der sie belegt sind, werden standardmäßig im unteren Bereich des Menüschirms angezeigt. Siehe *Standard-PF-Tasten* weiter unten.

Zusätzlich zu den Funktionen enthalten manche Natural-Menüs auch Felder, in denen Sie weitere Optionen und/oder Auswahlkriterien angeben können, z.B. können Sie im Feld **Name** im Menü **Development Functions** den **Namen** eines Natural-Objekts eingeben. Diese Felder und die jeweils gültigen Eingabewerte werden an den entsprechenden Stellen in der Dokumentation erläutert oder nach Aufrufen der Hilfefunktion (siehe *Hilfe benutzen*) am Bildschirm angezeigt.

Dieser Abschnitt behandelt folgende Themen:

- [Aufrufen einer Menü-Funktion](#)
- [Beenden einer Menü-Funktion](#)
- [Standard-PF-Tasten](#)
- [Beispiel für eine Menü-Funktion](#)

### Aufrufen einer Menü-Funktion

Dieser Abschnitt beschreibt die verschiedenen Möglichkeiten, die Sie haben, um eine Menü-Funktion aufzurufen.

#### ▶ Um eine Menü-Funktion aufzurufen

- Stellen Sie den Cursor in das Eingabefeld neben der gewünschten Menü-Funktion und drücken Sie **EINGABE**.

Oder:

Stellen Sie den Cursor in das Eingabefeld neben der gewünschten Menü-Funktion, geben Sie ein beliebiges Zeichen ein und drücken Sie **EINGABE**.

Oder:

Geben Sie im Feld **Code** den ein- oder zweibuchstabigen Funktionscode ein, der vor der gewünschten Funktion angezeigt wird und drücken Sie **EINGABE**.

Oder:

Falls entsprechend belegte PF-Tasten vorhanden sind, drücken Sie die PF-Taste, die der gewünschten Funktion entspricht (siehe *Standard-PF-Tasten* weiter unten).

Oder:

Rufen Sie die gewünschte Menü-Funktion mit einem Doppelklick auf das Eingabefeld neben der Funktion auf.

Falls in dem Menü zusätzliche Eingabefelder vorhanden sind, geben Sie dort die erforderlichen Informationen ein. Wenn Sie diese Felder nicht ausfüllen, erscheint entweder ein Auswahlfenster, aus dem Sie einen gültigen Wert übernehmen können, oder eine entsprechende Natural-Fehlermeldung.

Sie können die Hilfe-Funktion benutzen, um sich eine Erklärung der möglichen Feldeingaben anzeigen zu lassen. Geben Sie dazu ein Fragezeichen (?) in das betreffende Feld ein und drücken Sie **EINGABE**.

## Beenden einer Menü-Funktion

Dieser Abschnitt beschreibt die verschiedenen Möglichkeiten, die Sie haben, um eine Menü-Funktion ordnungsgemäß zu beenden und die vorgenommenen Änderungen zu speichern oder um sie ohne Speicherung der zuvor gemachten Änderungen abzurechnen.

### ▶ Um die Ausführung einer Funktion abzurechnen

- Geben Sie in der Kommandozeile einen Punkt (.) ein und drücken Sie **EINGABE**.

Oder:

Drücken Sie **PF12**.

Die zuvor gemachten Änderungen werden verworfen (nicht gespeichert).

### ▶ Um eine Funktion ordnungsgemäß zu beenden

- Drücken Sie **PF3**.

Die zuvor gemachten Änderungen werden gespeichert.

## Standard-PF-Tasten

Standardmäßig sind überall in Natural folgende Tasten mit folgenden Funktionen belegt:

PF Key	PF-Key Name	Explanation
PF1	<b>Help</b>	Ruft das Online-Hilfe-System auf.
PF2	<b>Menu</b>	Ruft das Natural-Hauptmenü <b>Main Menu</b> auf.
PF3	<b>Exit</b>	Beendet die aktuelle Funktion und führt die vorher gemachten Änderungen aus.
PF12	<b>Canc</b>	Bewirkt den Abbruch der aktuellen Funktion. Die die vorher gemachten Änderungen werden nicht ausgeführt.

## Beispiel für eine Menü-Funktion

Das folgende Beispiel zeigt, wie Sie ein Natural-Menü benutzen können, um ein Objekt (hier: ein Programm) im zugehörigen Natural-Editor zu bearbeiten.

### ► Um einen Natural-Editor zum Bearbeiten eines Natural-Objekts aufzurufen

#### 1 Im Menü **Development Functions**:

Geben Sie im Feld **Code** den einbuchstabigen Code, ein C, für die Funktion **Edit Object** ein.

Geben Sie im Feld **Type** den einbuchstabigen Code ein, der dem gewünschten Objekttyp entspricht (hier: P für Programm).

Geben Sie im Feld **Name** den Namen des zu bearbeitenden Natural-Objekts ein (hier der Name des Programms: PROGX).

```

13:33:16          ***** NATURAL *****                               2009-05-20
User SAG          - Development Functions -                               Library TEST
                                                           Mode Structured
                                                           Work area empty

Code  Function
C    Create Object
E    Edit Object
R    Rename Object
D    Delete Object
X    Execute Program
L    List Object(s)
S    List Subroutines Used
?    Help
.    Exit

Code .. E      Type .. P
Name .. PROGX_____

```

```
Command ===>
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---  
      Help  Menu  Exit                                     Canc
```

2 Drücken Sie EINGABE.

Der Natural-Programm-Editor wird aufgerufen. Der Sourcecode des Programms `PROGX` befindet sich im Arbeitsbereich des Editors und kann sofort bearbeitet werden (siehe auch [Beispiel für ein Systemkommando](#)).

# 4 Hilfe benutzen

---

- Allgemeine Informationen – Natural-Help-Utility ..... 18
- Hilfe zu Natural-Meldungen ..... 21
- Informationen zu Menüs und Felder ..... 27

Natural bietet Ihnen online verschiedene Hilfe-Informationen an:

- Allgemeine Informationen zu Natural-Sprachelementen (z.B. Statements, Systemvariablen) und Komponenten der Entwicklungsumgebung (z.B. Editoren und Utilities). Diese allgemeinen Informationen befinden sich in der Natural-Help-Utility, die Sie mit dem Systemkommando `HELP` aufrufen können.
- Ausführliche Informationen zu Natural-Systemfehlermeldungen und benutzerdefinierten Meldungen. Diese Informationen werden nach Absetzen der Systemkommandos `HELP` und `LASTMSG` angezeigt.
- Spezifische Informationen zu Natural-Menüs und zu Eingabefeldern, die in den Natural-Editoren und Utilities angezeigt werden. Diese Informationen werden durch umgebungsspezifische Hilfe-Funktionen angezeigt.

### Verwandte Themen:

- [HELP - Systemkommandos-Dokumentation](#)
- [LASTMSG - Systemkommandos-Dokumentation](#)

Dieser Abschnitt behandelt folgende Themen:

## Allgemeine Informationen – Natural-Help-Utility

---

Die Natural-Help-Utility bietet allgemeine Informationen zu Natural-Sprachelementen (z.B. Statements, Systemvariablen) und Komponenten der Entwicklungsumgebung (z.B. Systemkommandos, Editoren und Utilities) sowie zu Natural-Systemfehlermeldungen. Um die Hilfe zu einem bestimmten Thema aufzurufen, können Sie wahlweise durch die Natural-Help-Utility und ihre Menüs navigieren oder ein bestimmtes Hilfethema direkt aufrufen.

Dieser Abschnitt behandelt folgende Themen:

- [Natural-Help-Utility über Menü benutzen](#)
- [Hilfethemen direkt aufrufen](#)

### Natural-Help-Utility über Menü benutzen

#### ► Um das Menü der Help-Utility aufzurufen

1 Geben Sie folgendes Systemkommando ein:

```
HELP
```

oder

```
?
```



2 Drücken Sie EINGABE.

Es erscheint das **Menü** der Natural-Help-Utility mit einer Liste der zur Verfügung stehenden Hilfethemen, zum Beispiel:

```
10:58:19          ***** NATURAL HELP UTILITY *****          2009-05-20
                    - Menu -

-----

                Natural Help provides information on:

                S Natural Statements
                V Natural System Variables
                F Natural System Functions
                C Natural System Commands
                E Natural Editors
                U Natural Utilities
                P Natural Session Parameters
                T Natural Terminal Commands
                N Natural System Messages
                M User-Defined Messages
                . Exit

                Code .. _

-----

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit                                     Canc
```

3 Geben Sie im Feld **Code** den einbuchstabigen Code ein, der dem gewünschten Hilfethema entspricht.

4 Drücken Sie EINGABE.

Es erscheint entweder ein Bildschirm mit Informationen zu dem gewünschten Thema oder ein Untermenü mit weiteren Hilfethemen, das Ihnen hilft, Ihre Suche einzugrenzen.

## Hilfethemen direkt aufrufen

Anstatt durch die Menüs der Help-Utility zu navigieren, um ein bestimmtes Hilfethema anzuzeigen, können Sie auch direkt auf die Hilfe zu einem Natural-Systemkommando oder zu Elementen der Programmiersprache Natural zugreifen.

### ► Um ein Hilfethema direkt aufzurufen

- 1 Benutzen Sie folgende Systemkommando-Syntax:

```
HELP command
```

wobei *command* das Kommando ist, zu dem Sie Hilfe benötigen, z.B. das Systemkommando EDIT:

```
HELP EDIT
```

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm zum gewünschten Thema, in diesem Beispiel mit Informationen zum Systemkommando EDIT:

```
15:20:54          ***** NATURAL HELP UTILITY *****          2009-05-20
                        - EDIT -                                Page  1
-----
Function: Invoke a Natural editor for the purpose of editing a Natural object.
Parameters:
  Object-type: Copycode(C), Description(D), Global(G), Helproutine(H),
              Local(L), Map(M), Parameter(A), Program(P), Subprogram(N),
              Subroutine(S), Text(T) or Class(4).
              -> Must be specified if object-name is not specified.
  Object-name: The name of the object to be edited, which will then be
              loaded in the edit work area.
  Library-ID : May only be specified if the object is contained in a library
              other than the one to which you are currently logged on.
              -> Must not start with 'SYS' except 'SYSTEM'
              -> Must not be specified if Natural Security is installed.
-----
                                                    More ...
```

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit                -      +                                Canc

```

Falls die Informationen mehr als eine Bildschirmseite umfassen, erscheint links oben der Hinweis `Page 1` und unten links der Hinweis `More . . .`

- 3 Drücken Sie `EINGABE` oder drücken Sie `PF8`, um zu den Folgeseiten weiterzublättern.

Wenn Sie die letzte Seite des Hilfetextes erreicht haben, erscheint unten rechts der Hinweis `Last page`.

Drücken Sie `PF7`, wenn Sie rückwärts blättern und zur ersten Seite (`Page 1`) zurückkehren möchten.

## Hilfe zu Natural-Meldungen

In Natural gibt es Systemmeldungen und benutzerdefinierte Meldungen.

Natural-Systemmeldungen erkennen Sie daran, dass sie mit `NAT` beginnen und eine vierstellige Zahl `nnnn` aufweisen.

Zu jeder Natural-Systemmeldung gibt es einen Kurz- und einen Langtext:

- Der Kurztext ist die einzeilige Meldung, die beim Auftreten eines Fehlers zusammen mit der Fehlernummer ausgegeben wird.
- Der Langtext enthält eine Erläuterung und beschreibt Abhilfemaßnahmen, die Sie zur Behebung des Fehlers ergreifen können. Er wird angezeigt, wenn Sie das Systemkommando `HELP` absetzen (siehe Langtext einer Meldung anzeigen weiter unten).

Dieser Abschnitt behandelt folgende Themen:

- [Langtext einer Systemmeldung oder benutzerdefinierten Meldung anzeigen](#)

- [Informationen zum zuletzt aufgetretenen Fehler anzeigen](#)

## Langtext einer Systemmeldung oder benutzerdefinierten Meldung anzeigen

Sie können sich den Langtext einer Natural-Systemmeldung oder benutzerdefinierten Meldung anzeigen lassen wie im folgenden Abschnitt beschrieben. Siehe auch die [Anleitung](#) zur Anzeige der zuletzt aufgetretenen Systemmeldung.

### ▶ Um den Langtext zu einer Natural-Systemmeldung anzuzeigen

- 1 Benutzen Sie folgende Systemkommando-Syntax:

```
HELP NAT nnnn
```

oder

```
? nnnn
```

Wobei Sie anstelle von *nnnn* die Nummer des Fehlers angeben.

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Natural System Message** der Natural-Help-Utility mit dem Langtext zur angegebenen Fehlernummer (siehe [Beispielschirm](#) zur Natural-Systemmeldung NAT0082 weiter unten).

### ▶ Um den Langtext zu einer benutzerdefinierten Meldung anzuzeigen

- 1 Melden Sie sich in der Library an, in der die gewünschte benutzerdefinierte Meldung gespeichert ist.
- 2 Benutzen Sie folgende Systemkommando-Syntax:

```
HELP USER nnnn
```

oder

```
? Unnnn
```

Wobei Sie anstelle von *nnnn* die vierstellige Nummer des Fehlers angeben.

- 3 Drücken Sie EINGABE.

Es erscheint der Bildschirm **User Message** der Natural-Help-Utility mit dem Langtext zur angegebenen benutzerdefinierten Fehlernummer (siehe [Beispielschirm](#) weiter unten).

## Informationen zum zuletzt aufgetretenen Fehler anzeigen

Sie können sich den Kurz- und Langtext der in der aktuellen Session zuletzt aufgetretenen Natural-Systemmeldung anzeigen lassen, indem Sie entweder das Systemkommando `HELP ERROR` absetzen oder indem Sie den Cursor in die Meldungszeile stellen und wie weiter unten beschrieben vorgehen.

Sie können sich den Kurztext der zuletzt aufgetretenen Fehlermeldung(en) sowie zusätzliche Informationen zur Fehlersituation anzeigen lassen, indem Sie das Systemkommando `LASTMSG` benutzen. Die angezeigten Informationen enthalten zugehörige, eventuell vor der letzten Meldung aufgetretene Fehlermeldungen.

### ► Um den Langtext zur zuletzt aufgetretenen Systemmeldung anzuzeigen

- Benutzen Sie eine der folgenden Methoden

1. Geben Sie folgendes Systemkommando ein:

```
HELP ERROR
```

Drücken Sie `EINGABE`.

Es erscheint der Bildschirm **Natural System Message** der Natural-Help-Utility. Er enthält den Langtext zu der Systemmeldung, zum Beispiel:

```
15:18:27          ***** NATURAL HELP UTILITY *****          2009-05-20
Library SAG          - Natural System Message NAT0082 -          Page
1

      Invalid command, or Program ANTON does not exist in library.

Tx *** Short Text ***

      Invalid command, or ... .. does not exist in library.

Ex *** Explanation ***

      One of the following has occurred:

      - You entered a value in the command line which is neither a
```

```
Natural command nor the name of a Natural program contained
in the active library or in a library defined as a steplib.
- An object which is required during execution of a program,
subprogram, subroutine or helproutine is not contained in

More ...

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit      Print      +      Canc
```

2. Oder:

Wenn der Kurztext der Meldung in einem Menüschirm angezeigt wird, stellen Sie den Cursor in die Meldungszeile und drücken Sie PF1 (Help).

Es erscheint das Fenster **Current Natural Message**, zum Beispiel:

```
11:22:48          ***** NATURAL *****          2009-05-20
User SAG          - Main Menu -          Library TEST

Function

_ Development Functions
```

```

    _ Development Environment Settings
    _ Maintenance and Transfer Utilities
    _ Debugging and Monitoring Utilities
    _ Example Libraries

+----- Current Natural Message NAT0082 -----+
| Sh Invalid command, or Program ANTON does not exist in library.
|
| Tx Invalid command, or ... .. does not exist in library.
| Ex One of the following has occurred:
|   - You entered a value in the command line which is neither a
|     Natural command nor the name of a Natural program contained
|     in the active library or in a library defined as a steplib.
|   - An object which is required during execution of a program,
|     subprogram, subroutine or helproutine is not contained in
|     the active library or in a library defined as a steplib.
|   - Your Natural session is currently applying system files other
|     than those containing the object you specified.
| Ac Check to ensure that you entered a valid Natural command or the name
|   of an existing Natural object. Use the command SYSPROF to check
|   whether you are using the correct system file.
+-----+

NAT0082 Invalid command, or Program ANTON does not exist in library.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                               Exit                                     Canc

```



**Anmerkung:** Mit der User Exit Routine USR2002P (siehe *Operations*-Dokumentation) können Sie das Fenster **Current Natural Message** kundenspezifisch anpassen.

► Um die zuletzt aufgetretene(n) Systemmeldung(en) und weitere Einzelheiten anzuzeigen

1 Geben Sie folgendes Systemkommando ein:

```
LASTMSG
```

2 Drücken Sie EINGABE.

Es erscheint das Fenster **LASTMSG** mit dem Kurztext zu der bzw. den zuletzt aufgetretenen Systemmeldungen.

3 Um weitere Informationen zu der bzw. den Meldungen zu erhalten:

Im Fenster **LASTMSG**: stellen Sie den Cursor in die gewünschte Meldungszeile und drücken Sie EINGABE.

Es erscheint das Fenster **Detailed Information**, zum Beispiel:

```
+----- LASTMSG -----More: ++
! ERRST1  0080 NAT0917 Error 920 in COPYCODE ERRST2.      !
! ERRST2  0090 NAT0917 Error 920 in COPYCODE ERRST3.      !
! ERRST3  0070 NAT0920 Program HUGO cannot be loaded (00000004).  !
! +--- Detailed Information for NAT0920 ---+                !
! ! Error Number .. 920                                     !
! ! Error Line .... 70                                     !
! ! Object ..... ERRST3                                    !
! ! Object Type ... Copycode                               !
! ! Level ..... 5                                         !
! ! Library ..... SYSEXV                                  !
! ! DBID/FNR ..... 10 / 410                               !
! ! Error Class ... System                                !
! ! Error Type .... Runtime                               !
! ! Error Time .... 2003-02-27 15:58:01                   !
! +-----+                                               !
!                                                         !
! Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF1 !
!           Exit          --  ++      <<  >      Can !
+-----+
Command ==> LASTMSG

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          Exit                                Canc
```



Weitere Informationen finden Sie unter *LASTMSG* in der *Systemkommandos*-Dokumentation.

## Informationen zu Menüs und Felder

---

Zusätzlich zu den in der Natural-Help-Utility angebotenen Hilfethemen bietet Ihnen Natural spezifische Informationen zu allen Natural-Menüs und den dort vorhandenen Eingabefeldern.

### ▶ Um Hilfe zum aktuellen Menü aufzurufen

- Geben Sie in der Kommandozeile oder im Feld **Code** ein Fragezeichen (?) ein und drücken Sie `EINGABE`.

Oder:

Drücken Sie `PF1` (Help).

### ▶ Um Hilfe zu einem Feld aufzurufen

- 1 Stellen Sie den Cursor in das betreffende Feld und geben Sie ein Fragezeichen (?) ein.
- 2 Drücken Sie `EINGABE`.



# 5 Natural-Libraries benutzen

---

▪ In einer Library enthaltene Objekte .....	30
▪ Library-Arten .....	31
▪ Steplib-Libraries .....	31
▪ Suchreihenfolge bei der Objekt-Ausführung .....	33
▪ Standard-Library-Zuweisung .....	34
▪ Einloggen in eine Library .....	35
▪ Neue Library anlegen .....	36
▪ Libraries auflisten .....	36
▪ Objekte in einer Library auflisten .....	39
▪ Objektliste drucken .....	40
▪ Objekte in einer Library suchen .....	41
▪ Libraries kopieren, verschieben und umbenennen .....	46
▪ Objekte in einer Library löschen .....	48

Die in den Natural-Systemdateien `FNAT` und `FUSER` gespeicherten Natural-Objekte sind in logische Konstrukte unterteilt, die als Libraries (Bibliotheken) bezeichnet werden. Eine Natural-Library dient dazu, Objekte nach bestimmten funktionalen Kriterien zusammenzufassen. So kann eine Library beispielsweise aus einem Satz von Objekten bestehen, die eine bestimmte Aufgabe zu erledigen haben. Wenn das Add-on-Produkt Natural Security installiert ist, kann man eine Library auch verwenden, um den Zugriff auf Objekte auf bestimmte Benutzergruppen, z.B. Administratoren, einzuschränken.

Eine Natural-Anwendung kann, je nachdem wie die Umgebung eingerichtet ist, auf Objekte in mehreren Libraries zugreifen.

Alle Operationen auf Libraries werden mit Hilfe von Menüfunktionen oder entsprechenden Kommandos ausgeführt. Siehe auch den Abschnitt [Kommandos und Menü-Funktionen benutzen](#).

#### **Verwandte Themen:**

- [Natural-Systemdateien - System-Architektur-Dokumentation](#)
- [Libraries in Systemdateien - System-Architektur-Dokumentation](#)
- [Library Maintenance - Natural Security-Dokumentation](#)

Dieser Abschnitt behandelt folgende Themen:

Siehe auch [Namenskonventionen für Libraries](#).

## **In einer Library enthaltene Objekte**

---

Eine Library enthält Objektmodule (Source-Objekt und ggf. katalogisiertes Objekt) der folgenden Arten von Natural-Objekten: Programm, Subprogramm, Subroutine, Copycode, Helproutine, Class, Text, Recording, Map, Local Data Area, Global Data Area, Parameter Data Area, Command Prozessor Source, Error Message, Dialog, Debug Environment, Adapter und Resource.

#### **Verwandte Themen:**

- [Katalogisiertes Objekt](#)
- [Source-Objekt](#)

---

## Library-Arten

---

Im folgenden Abschnitt werden die verschiedenen Arten von Libraries vorgestellt:

- Benutzer-Library
- System Library

### Benutzer-Library

Eine Benutzer-Library enthält Objekte einer Benutzer-Anwendung, die speziell zu dieser Anwendung gehören und zum Ausführen dieser Anwendung benötigt werden. Benutzer-Libraries sind in der Systemdatei `FUSER` untergebracht.


### System Library

Die Pflege einer System-Library obliegt der Software AG. System-Libraries sind in der Systemdatei `FNAT` untergebracht.

In einer System-Library sind alle Objekte enthalten, die zum Ausführen einer Natural-Systemanwendung benötigt werden.

Darüber hinaus gibt es System-Libraries, die Objekte des Typs Programm enthalten, die zu Demonstrationszwecken geliefert werden, siehe *Example Libraries (Beispielprogramm-Libraries)*.

Auf die System-Libraries `SYSLIB` und `SYSLIBS` können Sie nicht zugreifen. Diese sind zur internen Verwendung durch die Software AG bestimmt. Wenn Sie es dennoch versuchen, erhalten Sie die Fehlermeldung: `Libraries SYSLIB and SYSLIBS reserved for system usage.`

 **Wichtig:** Speichern Sie keine benutzerspezifischen Objekte in einer Natural-System-Library. Diese Objekte können überschrieben werden, wenn Software AG-Installations-Datasets (zum Beispiel beim Umstieg auf einen höheren System Maintenance (SM) Level, neue Versionen oder beim Einspielen von Fix-Bändern) in die Systemdateien geladen werden.

---

## Steplib-Libraries

---

Eine Steplib ist eine Natural-Benutzer-Library oder System-Library, die mit der aktuellen Benutzer- oder System-Library verkettet ist. Eine Steplib kann als einziger Speicherort für mehrere Objekte dienen, die von verschiedenen Anwendungen gemeinsam genutzt werden. Dadurch wird die redundante Speicherung von identischen Objekten vermieden und die Strukturierung von Anwendungen erleichtert.

Eine Steplib ist eine Library, in der Natural sucht, wenn ein Objekt in der aktuellen Library nicht gefunden werden konnte (siehe auch den folgenden Abschnitt). Als Standard-Steplibs dienen die Libraries SYSTEM in den Systemdateien FUSER und FNAT.

### **Verwandte Themen in der Parameter-Referenz-Dokumentation:**

- *STEPLIB - Additional Steplib Library*
- *BPSFI - Object Search First in Buffer Pool*

Dieser Abschnitt behandelt folgende Themen:

- [Zusätzliche Steplibs bei Einsatz von Natural Security](#)
- [Zusätzliche Steplibs ohne Einsatz von Natural Security](#)
- [Überprüfen der Steplib-Einstellung](#)

### **Zusätzliche Steplibs bei Einsatz von Natural Security**

Wenn Natural Security aktiv ist, können Sie im Security-Profil zu jeder Library zusätzliche Steplibs festlegen. Die Einträge in einem Library-Security-Profil haben Vorrang vor jeglichen außerhalb von Natural Security vorgenommenen Festlegungen. Diese Steplibs werden nach einem Objekt durchsucht, bevor in den Standard-Steplibs SYSTEM in den Systemdateien FUSER und FNAT danach gesucht wird.

#### **Verwandtes Thema:**

- *Steplibs - Natural Security-Dokumentation*

### **Zusätzliche Steplibs ohne Einsatz von Natural Security**

Wenn kein Natural Security aktiv ist, können Sie beim Session-Start eine zusätzliche Steplib mit Hilfe des Natural-Profilparameters STEPLIB angeben. Darüber hinaus können Sie weitere Steplibs unter Verwendung der Programmierschnittstellen (APIs, z.B. USR1025N oder USR3025N) definieren. Diese und weitere APIs finden Sie in der System-Library SYSEXT.

Diese Steplibs werden nach einem Objekt durchsucht, bevor in den Standard-Steplibs SYSTEM in den Systemdateien FUSER und FNAT danach gesucht wird.

#### **Verwandtes Thema:**

- *STEPLIB - Parameter-Referenz-Dokumentation*

## Überprüfen der Steplib-Einstellung

### ▶ Um die Steplib-Einstellung Ihrer aktuellen Library zu überprüfen

- 1 Geben Sie folgendes Systemkommando ein:

```
TECH
```

Das Fenster **TECH** erscheint.

- 2 Blättern Sie nach unten, indem Sie solange **EINGABE** drücken, bis die Tabelle mit der Spalte **Steplib** erscheint.

### Verwandtes Thema:

- *TECH - Systemkommandos-Dokumentation*

## Suchreihenfolge bei der Objekt-Ausführung

Dieser Abschnitt beschreibt, in welcher Reihenfolge Natural-Libraries und Systemdateien nach einem angeforderten Objekt durchsucht werden, das entweder aus einer Benutzer-Library oder einer System-Library heraus ausgeführt werden soll.



**Anmerkung:** Wenn der Profilparameter **BPSFI** auf **ON** gesetzt ist (Standard-Einstellung ist **OFF**), wird zunächst nach Objekten im Buffer Pool gesucht.

### Suchreihenfolge nach Benutzer-Objekten, die aus einer Benutzer-Library ausgeführt werden sollen:

1. Die aktuelle Library in der Systemdatei **FUSER** gemäß Definition durch die Systemvariable **\*LIBRARY-ID**.
2. Die Steplibs (sequenziell) gemäß Angaben im Natural-Security-Profil für die aktuelle Library oder in der Steplib-Tabelle.
3. Die Standard-Steplib gemäß Definition in der Systemvariablen **\*STEPLIB**.
4. Die Library **SYSTEM** in der Systemdatei **FUSER**.
5. Die Library **SYSTEM** in der Systemdatei **FNAT**.

### Suchreihenfolge nach einem Natural-Objekt, das aus einer System-Library ausgeführt werden soll:

1. Die aktuelle Library in der Systemdatei **FNAT** gemäß Definition durch die Systemvariable **\*LIBRARY-ID**.

2. Die Steplibs (sequenziell) gemäß Angaben im Natural-Security-Profil für die aktuelle Library oder in der Steplib-Tabelle.
3. Die Library SYSLIBS in der Systemdatei FNAT, die Objekte enthält, die gemeinsam von Systemkommandos und Utilities verwendet werden.
4. Die Library SYSTEM in der Systemdatei FNAT.
5. Die Library SYSTEM in der Systemdatei FUSER.

Da die Systemdatei FUSER zuletzt durchsucht wird, dürfen Sie ein Objekt, das sowohl in der Systemdatei FUSER als auch in FNAT verwendet wird (z.B. eine User Exit Routine für eine Natural-Utility) nur an einer Stelle vorhalten, und zwar in FUSER.

#### Verwandte Themen in der Systemvariablen--Dokumentation:

- *\*LIBRARY-ID - Systemvariablen*
- *\*STEPLIB*

## Standard-Library-Zuweisung

---

Wenn Sie eine Natural-Session starten, loggen Sie sich automatisch in eine von Natural zugewiesene Library ein. Auf Natural-Bildschirmen ist normalerweise ein Feld vorhanden (z.B. **Library** oder **LIB=**), in dem der Name (die ID) Ihrer Library angezeigt wird, d.h. die aktuelle Library (z.B. **LIB=SYSTEM**), in der Natural-Objekte gespeichert werden und aus der sie abgerufen werden.

#### ► Um festzustellen, welche Ihre aktuelle Library ist

- Geben Sie folgendes Systemkommando ein:

```
TECH
```

Das Fenster **TECH** erscheint. Der Name (die ID) der Library, in der Sie zurzeit angemeldet sind, wird im Feld **Library** angezeigt.

Wenn der Natural-Profilparameter **AUTO** auf **OFF** gesetzt ist, weist Natural Ihnen standardmäßig die Library ID **SYSTEM** zu. Sie können beim Starten der Session die Standard-Library ändern, indem Sie **AUTO** auf **ON** setzen. **AUTO=ON** bewirkt das automatische Einloggen in die Library, die Ihrer Benutzerkennung (ID) entspricht. Wenn Ihr Natural unter Natural Security läuft, gelten besondere Regeln für das Einloggen. Diese sind in der *Natural Security*-Dokumentation beschrieben.

Es kann sein, dass Sie sich in eine andere Library einloggen müssen, wenn eine Verwaltungsfunktion ausführen oder an einer anderen Anwendung arbeiten möchten; siehe Abschnitt [Einloggen in eine Library](#).



**Verwandte Themen:**

- *TECH* - Systemkommandos-Dokumentation
- *AUTO* - Parameter-Referenz-Dokumentation
- *Logging On* - Natural Security-Dokumentation

## Einloggen in eine Library

---

Wenn Sie ein Natural-Objekt erstellen oder pflegen oder ein Natural-Programm in einer bestimmten Library ausführen möchten, müssen Sie unter Umständen zuerst die Library wechseln und sich in die Library oder Steplib (siehe [Steplib-Libraries](#)) einloggen, die das Objekt enthält bzw. aufnehmen soll.

Zum Einloggen in eine Library benutzen Sie normalerweise das Systemkommando `LOGON`. Wenn Ihr Natural unter Natural Security läuft, gelten besondere Regeln. Diese sind in der *Natural Security*-Dokumentation beschrieben.

Nach dem erfolgreichen Einloggen in eine Library, die noch keine Objekte enthält, lautet die Bestätigungsmeldung: `This library is empty` (Diese Library ist leer.).



**Anmerkung:** Das Systemkommando `LOGON` berücksichtigt keine Objekte des Typs Error Message und Debug Environment. Deshalb bezieht sich die o. g. Meldung nur auf die anderen Objekttypen. Um festzustellen, ob eine Library Objekte des Typs Error Message und Debug Environment enthält, können Sie, zum Beispiel, die Utility `SYSMAIN` oder die Utilities `SYSERR` bzw. den Debugger benutzen.

Darüber hinaus können Sie auch das Systemkommando `LIST` (siehe [Objekte in einer Library auflisten](#)) benutzen, um festzustellen, ob eine Library leer ist.

Die Library, in die Sie sich eingeloggt haben, bleibt solange aktiv, bis Sie sich in eine andere Library einloggen oder Ihre Natural-Session beenden.

### ► Um sich in eine andere Library einzuloggen

- 1 Geben Sie folgendes Systemkommando ein:

```
LOGON library-ID
```

Dabei ist `library-ID` der Name (ID) der Library, auf die Sie zugreifen wollen.

Oder:

Im Natural-Hauptmemü (**Main Menu**) oder einem der untergeordneten Menüs:

Überschreiben Sie in der rechten oberen Ecke des Bildschirms im Feld Library die dort vorhandene **Library** ID mit der ID der gewünschten Library (siehe auch [Beispiel für eine Menü-Funktion](#)).

- 2 Drücken Sie EINGABE.

Vorausgesetzt, die von Ihnen eingegebene Library ID erfüllt **Namenskonventionen für Libraries**, dann erscheint die folgende Meldung zur Bestätigung: Logon in Library *library-ID* accepted.

#### Verwandte Themen:

- *LOGON* und *LOGOFF* - Systemkommandos-Dokumentation
- *Logging On* - Natural Security-Dokumentation

## Neue Library anlegen

---

Um eine neue Library anzulegen, benutzen Sie entweder das Systemkommando `LOGON`, mit dem Sie eine leere Library erhalten, in der Sie ein Objekt erstellen können, oder Sie verwenden die Move- oder Copy-Funktion einer Natural-Utility, wenn Sie Objekte übertragen.

### ▶ Um eine neue Library anzulegen

- 1 Wenn Sie das Systemkommando `LOGON` benutzen, folgen Sie der Beschreibung im Abschnitt [Einloggen in eine Library](#). Geben Sie die ID der neuen Library an. Diese enthält noch keine Objekte. Erstellen Sie in dieser Library mindestens ein katalogisiertes Objekt oder ein Source-Objekt.
- 2 Wenn Sie Objekte aus einer vorhandenen Library in eine neue Library verschieben oder kopieren möchten, folgen Sie der Beschreibung im Abschnitt [Libraries kopieren, verschieben und umbenennen](#).

## Libraries auflisten

---

Eine Liste aller in Ihrer aktuellen Natural-Systemumgebung vorhandenen Libraries erhalten Sie, zum Beispiel, indem Sie die Utility `SYSMAIN` benutzen.

Die folgende Anleitung enthält Beispiele dafür, wie Sie entweder die `SYSMAIN`-Menü-Funktionen oder die entsprechenden Kommandos benutzen können, um Libraries aufzulisten.

► Um mittels der **SYSMAIN**-Menü-Funktionen alle Libraries aufzulisten

- 1 Wählen Sie im Natural-Hauptmenü (**Main Menu**) den Eintrag **Maintenance and Transfer Utilities** aus.
- 2 Drücken Sie EINGABE.

Der Bildschirm **Maintenance and Transfer Utilities** erscheint.

- 3 Wählen Sie **Transfer Objects to Other Libraries**.

Oder:

Geben Sie das folgende Systemkommando ein:

```
SYSMAIN
```

- 4 Drücken Sie EINGABE.

Das Hauptmenü (**Main Menu**) der Utility **SYSMAIN** erscheint, zum Beispiel:

```
17:41:46          ***** NATURAL SYSMAIN UTILITY *****          2009-05-20
User SAG                      - Main Menu -

          Code  Object                                Code  Function
          A    Programming Objects                    C    Copy
          D    Debug Environments                      D    Delete
          E    Error Message Texts                    F    Find
          P    Profiles                                L    List
          R    Rules                                    M    Move
          S    DL/I Subfiles                            R    Rename
          V    DDMs                                     ?    Help
          ?    Help                                     .    Exit
          .    Exit

Object Code .. A                                Function Code .. _

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren
```

- 5 Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um alle Libraries aufzulisten, die Natural-Objekte (Fehlermeldungen und Debug-Umgebungen) enthalten. (Für Fehlermeldungen müssen Sie ein E eingeben, für Debug-Umgebungen ein D).

Geben Sie im Feld **Function Code** ein L (für List) ein.

- 6 Drücken Sie EINGABE.

Es erscheint der Bildschirm **List Programming Objects**.

- 7 Geben Sie im Feld **Code** ein A ein, um nach Libraries zu suchen, die alle Arten von Objektmodulen enthalten: katalogisierte Objekte und Source-Objekte.

Geben Sie im Feld **Source Library** einen Stern (\*) ein, um nach allen Libraries zu suchen.

(Informationen zu gültigen Namensbereichen siehe *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Lassen Sie die übrigen Eingabefelder unverändert.

- 8 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Library Selection**. Er enthält eine Liste aller in der aktuellen Systemdatei vorhandenen Libraries.

Drücken Sie EINGABE, um in der Liste nach unten zu blättern bzw. PF11, um nach rechts und PF12, um nach links zu blättern.

Auf dem Bildschirm **Library Selection** können Sie eine bestimmte Library auswählen und die in ihr enthaltenen Objekte auflisten. Weitere Informationen finden Sie im Abschnitt **Library Selection**, siehe *Verwandtes Thema* weiter unten.

#### ▶ Um alle Libraries mit einem Kommando aufzulisten

- 1 Geben Sie das folgende Kommando ein:

```
SYSMAIN LIST ALL * IN LIBRARY *
```

- 2 Drücken Sie EINGABE.

Der Bildschirm **Library Selection** erscheint mit einer Liste aller in der aktuellen Systemdatei vorhandenen Libraries.

#### Verwandtes Thema:

- *Listing and Selecting Libraries - SYSMAIN Utility*-Dokumentation

## Objekte in einer Library auflisten

Eine Liste der in einer Library enthaltenen Objekte erhalten Sie, indem Sie entweder das Systemkommando LIST oder die entsprechende Funktion der Natural-Utility SYSMAIN oder den Object Handler benutzen.



**Anmerkung:** Das Systemkommando LIST kann keine Objekte des Typs Fehlermeldung oder Debug-Umgebung verarbeiten. Letztere können Sie, zum Beispiel, mit der Utility SYSMAIN auflisten.

Dieser Abschnitt beschreibt, wie Sie sich eine Liste von Objekten mit Hilfe des Systemkommandos LIST anzeigen lassen können.

### ► Um Objekte mit LIST aufzulisten

- 1 Geben Sie das folgende Kommando ein:

```
LIST *
```

Dabei bedeutet die Angabe eines Sterns (\*), dass alle Source-Objekte und alle katalogisierten Objekte aufgelistet werden sollen.

(siehe auch: *object-name-range* im Abschnitt LIST in der Systemkommandos-Dokumentation).

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm **LIST Objects in a Library** (siehe Beispiel weiter unten). Er enthält eine Liste der Source-Objekte und der katalogisierten Objekte, die in der aktuellen Library (in diesem Beispiel: TEST) in der aktuellen Systemdatei vorhanden sind.

```
12:22:23          ***** NATURAL LIST COMMAND *****          2009-05-20
User SAG          - LIST Objects in a Library -          Library TEST

Cmd  Name          Type          S/C  SM  Version  User ID  Date          Time
---  *-----*-----*--*  *  *-----*-----*-----*-----*-----*-----*-----*
___  DBTEST          Program       S/C  S   4.2.01   SAG     2002-02-27   16:46:36
___  DEMOPROG        Program       S/C  S   4.1.02   SAG     2000-12-20   11:40:46
___  DEMOSPRG        Subprogram    S/C  S   4.1.02   SAG     2000-12-22   11:32:34
___  DEMOTEXT        Text          S     4.1.02   SAG     2000-12-20   13:59:26
___  EMP-L           Local         S/C     4.1.02   SAG     2002-07-29   12:46:43
___  EMPL            Program       S/C  S   4.1.02   SAG     2002-07-29   12:46:30
___  GDATEST         Global        S/C     4.1.03   SAG     2004-11-22   13:24:32
___  LDATEST         Local         S     4.2.01   SAG     2004-05-21   17:18:13
___  MAPTEST         Map           S/C  S   4.2.03   SAG     2000-12-20   11:40:46
___  PDATEST         Parameter     S/C     4.1.03   SAG     2004-05-05   11:46:24
___  PDATEST2        Parameter     S/C     4.1.03   SAG     2004-11-23   10:12:07
___  PROCTEST        Processor     S/C     4.2.01   SAG     2004-05-05   11:25:22
```

```

SUBRTEST  Subroutine  S/C  S  4.1.03  SAG      2002-07-23  14:37:16
TESTPGM   Subprogram  S    S  4.1.03  SAG      2000-12-20  11:40:45
                                         14 Objects found
Top of List.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit  Sort      --    -    +    ++      >    Canc

```

Sortiert ist die Liste nach dem Objektname (in der Spalte **Name**). Sie enthält zusätzlich Informationen zu jedem Objekt (z. B. Program), zur Art des vorhandenen Objektmoduls (in der Spalte **S/C**: S = Source-Objekt, C = katalogisiertes Objekt) und die ID des Benutzers, der das Objekt angelegt oder geändert hat.

Auf dem Bildschirm **LIST Objects in a Library** können Sie ein Objekt zur weiteren Verarbeitung auswählen. Sie können sich z. B. den Sourcecode eines Objekts anzeigen lassen oder Objekte aus einer Library löschen; siehe [To delete objects using LIST](#).

Weitere Informationen zum Bildschirm **LIST Objects in a Library** sowie zu weiteren Optionen, die Ihnen beim Systemkommando `LIST` zur Verfügung stehen, siehe *Verwandte Themen* unten.

**Verwandte Themen:**

- *object-name-range - LIST, Systemkommandos-Dokumentation*
- *Erklärung der Spaltenüberschriften - LIST, Systemkommandos-Dokumentation*
- *Inhalt der Auswahlliste sortieren - LIST, Systemkommandos-Dokumentation*
- *Eine Funktion auf einem Objekt ausführen - LIST, Systemkommandos-Dokumentation*

## Objektliste drucken

Sie können sich eine Liste der in der aktuellen Library in der aktuellen Systemdatei enthaltenen Objekte ausdrucken lassen.

► **Um eine Objektliste zu drucken**

- 1 Rufen Sie die Funktion **LIST Objects in a Library** auf; siehe [Schritte 1 und 2](#) von *Um Objekte mit LIST aufzulisten* .
- 2 Drücken Sie PF2 (Print).

Es erscheint das Fenster **PRINT**.

- 3 Geben Sie im Feld **Destination** einen gültigen Druckernamen ein. (Fragen Sie ggf. Ihren Natural-Administrator nach dem Namen eines in Ihrer Umgebung zur Verfügung stehenden Druckers.) Falls Sie dies wünschen, können Sie die Seitenlänge ändern (standardmäßig hat eine Seite 60 Zeilen).

4 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Printout Specification**, auf dem Sie Druckereinstellungen, z. B. die Anzahl der zu druckenden Exemplare vornehmen können.

5 Drücken Sie EINGABE.

Die Liste der in der Library enthaltenen Objekte wird auf dem angegebenen Drucker gedruckt.

#### Verwandte Themen:

- [Objekte drucken](#)
- [LIST - Systemkommandos-Dokumentation](#)

## Objekte in einer Library suchen

---

Sie können die in der Liste angezeigten Objekte weiter eingrenzen, indem Sie zusätzliche Auswahlkriterien angeben, z. B. die ID des Benutzers, der ein bestimmtes Objekt oder eine bestimmte Zeichenkette in einer Objekt-Source angelegt hat.

Dieser Abschnitt beschreibt, wie Sie unter Angabe von Suchkriterien Objekte suchen können. Dazu können Sie entweder die Utility `SYSMAIN` oder das Systemkommando `SCAN` benutzen.

- [Objekte suchen mit SYSMAIN](#)
- [Objekte suchen mit SCAN](#)

### Objekte suchen mit SYSMAIN

Der folgende Abschnitt enthält Beispiele dafür, wie Sie Objekte durch Angabe von Suchkriterien suchen können, indem Sie entweder die Funktionen des `SYSMAIN`-Menüs oder das entsprechende Kommando benutzen.

#### ► Um Objekte mit Hilfe der Menüfunktionen zu suchen

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility `SYSMAIN` auf; siehe [Schritte 1 bis 4](#) von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um nach allen Objekttypen zu suchen.

(Ausnahmen: Für Fehlermeldungen geben Sie ein E, für Debug-Umgebungen ein D.)

Geben Sie im Feld **Function Code** ein F für (Find) ein.

- 3 Drücken Sie EINGABE.

Der Bildschirm **Find Programming Objects** erscheint.

- 4 Geben Sie im Feld **Code** ein A ein, um nach allen Arten von Objektmodulen zu suchen: katalogisierte Objekte und Source-Objekte.

Geben Sie im Feld **Object Name** einen Stern (\*) ein, um nach allen Objektnamen zu suchen. Dies ist die Standardeinstellung.

(Gültige Namensbereiche siehe *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Geben Sie im Feld **Source Library** die ID der Library ein, die die zu suchenden Objekte enthält.

Überschreiben Sie im Feld **Criteria** das N (No) durch Y (Yes). N ist die Standardeinstellung.

Lassen Sie alle übrigen Eingabefelder unverändert.

- 5 Drücken Sie EINGABE.

Es erscheint das Fenster **Additional Criteria**, in dem Sie Suchkriterien eingeben können, zum Beispiel:

```

18:23:16          ***** NATURAL SYSMAIN UTILITY *****          2009-05-20
User SAG          - Find Programming Objects -

          +-----+
          !      --- Additional Criteria ---      !
          !                                         !
          ! Object Type ..... PM_____          !
          ! Date/Time From .. 2006-05-09 _____ !
          ! Date/Time To .... 2009-05-01 _____ !
          ! User ID ..... SAG_____          !
          ! Terminal ID ..... _____          !
          !                                         !
          !                                         !
          ! Command ==>                          !
          Code .. !                               !
          Object Name .. +-----+
    
```



```

Source  Library ... TEST____ Database .... 10____ File .. 32____

Options                                Criteria .... Y

Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit Copy Del Find List Move Ren Fsec Fdic Fnat

```

- 6 Geben Sie eines oder mehrere Suchkriterien ein und drücken Sie EINGABE.

Das Fenster verschwindet. Ein Plus-Zeichen (+) vor dem Feld **Criteria** zeigt an, dass in dem Fenster außer dem Objekttyp noch andere Suchkriterien angegeben wurden.

- 7 Drücken Sie noch einmal EINGABE.

Der Bildschirm **Find Selection** erscheint. Er enthält eine Liste aller Source-Objekte und katalogisierten Objekte, die in der angegebenen Library in der aktuellen Systemdatei vorhanden sind. In dem obigen Beispiel werden alle Objekte der Typen P (Programm) und M (Map) angezeigt, die zwischen 2006-05-09 und 2009-05-01 durch den Benutzer SAG angelegt oder geändert wurden.

- 8 Drücken Sie EINGABE, um in der Liste nach unten zu blättern, bis die folgende Meldung erscheint: Library has been successfully listed (Library erfolgreich aufgelistet).

Auf dem Bildschirm **Find Selections** haben Sie die Möglichkeit, ein Objekt zur Weiterverarbeitung auszuwählen, zum Beispiel, können Sie sich den Sourcecode eines Objekts anzeigen lassen; siehe *Selection Lists for Programming Objects (SYSMAIN Utility-Dokumentation)*.

#### Verwandtes Thema:

- *Using Menu Functions and Commands - SYSMAIN Utility-Dokumentation*

► **Um Objekte mittels Kommando zu suchen**

- 1 Geben Sie das folgende Kommando ein:

```
SYSMAIN FIND ALL * TYPE PN WITH USER user-ID IN library-ID
```

Dabei gibt ALL an, dass alle Source-Objekte und katalogisierten Objekte in die Suche einbezogen werden sollen.

Der Stern (\*) gibt an, dass alle Objektnamen in die Suche einbezogen werden sollen (siehe auch: *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation).

PN gibt die Objekttypen an, nach denen gesucht werden soll (P bedeutet Programm, N bedeutet Subprogramm).

*library-ID* ist die ID der zu durchsuchenden Library.

*user-ID* ist die ID des Benutzers, nach dem gesucht werden soll.

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Find Selection**. Er enthält eine Liste aller Source-Objekte und katalogisierten Objekte der Typen Programm und Subprogramm, die von dem angegebenen Benutzer angelegt oder geändert wurden.

**Verwandtes Thema:**

- *Keywords and Variables in Direct Commands - SYSMAIN Utility*-Dokumentation

**Objekte suchen mit SCAN**

Der folgende Abschnitt enthält Beispiele dafür, wie Sie Objekte finden können, indem Sie ihre Sourcen nach einer bestimmten Zeichenkette mit Hilfe des Systemkommandos SCAN durchsuchen.



**Anmerkung:** Das Systemkommando SCAN verarbeitet keine Objekte des Typs Error Message und Debug-Environment. Zum Durchsuchen von Fehlermeldungen können Sie die Funktion **Scan in messages** der Natural-Utility SYSERR verwenden.

► **Um Sourcen mit Hilfe von Menüfunktionen zu durchsuchen**

- 1 Geben Sie das folgende Systemkommando ein:

```
SCAN
```

- 2 Drücken Sie EINGABE.

Der Bildschirm **Scan Objects in Libraries** erscheint (siehe folgendes Beispiel). Sie können hier einen Objektbereich und einen Suchwert eingeben:

```

18:24:53          ***** NATURAL SCAN COMMAND *****          2009-05-20
User SAG          - Scan Objects in Libraries -          Library TEST

                Code  Function

                T    Statistics
                L    List of Objects Containing Scan Value
                S    Object Lines with Scan Value
                ?    Help
                .    Exit

Code ..... L
Scan value ..... LOCAL_____
Replace value ... GLOBAL_____
Library ..... TESTLIB_
Object name ..... *_____ Selection list .. N
Object type(s) .. *_____
Absolute scan ... N Trace ..... N

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Exit

```

Geben Sie im Feld **Code** ein L (für **List of Objects Containing Scan Value**) ein und geben Sie einen Suchwert, eine Library und einen Objektbereich ein. Mit Stern-Notation (\*) können Sie alle Objektnamen und Objekttypen auswählen.

Geben Sie im Feld **Replace value** eine Zeichenkette (z. B. GLOBAL) ein, durch die der angegebene Suchwert (z. B. LOCAL) ersetzt werden soll.

### 3 Drücken Sie EINGABE.

Der Bildschirm **Select Objects for Library** erscheint. Er enthält für die angegebene Library in der aktuellen Systemdatei eine Liste aller Source-Objekte, die die angegebenen Suchkriterien erfüllen. Im Fall des obigen Beipiels werden alle Objekte in der Library TESTLIB ausgewählt, die die Zeichenkette LOCAL enthalten.

Auf dem Bildschirm **Select Objects for Library** können Sie ein Objekt auswählen und die Source-Zeile, die den Suchwert enthält, anzeigen oder ändern.

▶ **Um Sourcen mit Hilfe eines Kommandos zu durchsuchen**

- 1 Geben Sie das folgende Kommando ein:

```
SCAN FUNC=L,SVAL=scan-value,LIB=library-ID,*
```

Dabei ist:

*scan-value* die Zeichenkette (z. B. LOCAL), nach der die Sourcen durchsucht werden sollen.

*library-ID* library-ID die ID der Library, die durchsucht werden soll (z. B. TESTLIB).

Verwenden Sie Stern-Notation (\*), wenn nach allen Source-Objekten in der Library gesucht werden soll.

- 2 Drücken Sie EINGABE.

**Verwandtes Thema:**

- *SCAN - Systemkommandos*-Dokumentation

## Libraries kopieren, verschieben und umbenennen

---

Sie können eine Library kopieren oder verschieben, indem Sie alle Natural-Objekte von einer Quell-Library in eine vorhandene oder neue Ziel-Library kopieren oder verschieben.

Beim Umbenennen einer Library sind dieselben Schritte wie beim Verschieben einer Library nötig. Sobald alle Objekte einer Library in eine neue Ziel-Library verschoben worden sind, wird die alte Library automatisch gelöscht.

Sie können einzelne oder mehrere Objekte von einer Library in eine andere Library kopieren oder verschieben, indem Sie entweder die Natural-Utility SYSMAIN oder den Object Handler benutzen. Die folgende Anleitung enthält Beispiele für das Kopieren oder Verschieben aller Objekte mit SYSMAIN, und zwar mit Hilfe von Menüfunktionen oder den entsprechenden Kommandos.

▶ **Um alle Objekte mit Hilfe von Menüfunktionen zu kopieren oder zu verschieben**

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe **Schritte 1 bis 4** von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um alle Objekttypen auszuwählen.

(Ausnahmen: für Error Messages geben Sie ein E, für Debug-Umgebungen ein D ein.)

Geben Sie im Feld **Function Code** ein C (für Copy) oder ein M (für Move) ein.

- 3 Drücken Sie EINGABE.

Je nachdem, welchen Funktionscode Sie eingegeben haben, erscheint entweder der Bildschirm **Copy Programming Objects** oder der Bildschirm **Move Programming Objects**.

- 4 Geben Sie im Feld **Code** ein A ein, um alle Arten von Objektmodulen auszuwählen: katalogisierte Objekte und Source-Objekte.

Überschreiben Sie im Feld **Sel. List** (Selection List) das Y (Yes) mit einem N (No). Y ist die Standardeinstellung.

Geben Sie im Feld **Object Name** einen Stern (\*) ein, um alle Objektnamen auszuwählen. Dies ist die Standardeinstellung.

(Informationen zu gültigen Namensbereichen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Geben Sie im Feld **Source Library** die ID der Library ein, die die zu kopierenden oder zu verschiebenden Objekte enthält.

Geben Sie im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in die Sie die Objekte kopieren oder verschieben wollen.

Alle anderen Eingabefelder lassen Sie unverändert.

- 5 Drücken Sie EINGABE.

Alle Source-Objekte und katalogisierten Objekte werden von der angegebenen Quell- in die Ziel-Library in der aktuellen Systemdatei kopiert bzw. verschoben. Danach erscheint die Meldung `Function completed successfully` (Funktion erfolgreich beendet).

#### ► Um alle Objekte mit Hilfe von Kommandos zu kopieren oder zu verschieben

- 1 Geben Sie eines der folgenden Kommandos ein:

```
SYSMAIN COPY ALL * FM old-library TO new-library
```

(kopiert Objekte)

bzw.

```
SYSMAIN MOVE ALL * FM old-library TO new-library
```

(verschiebt Objekte)

Dabei gibt `ALL` an, dass alle Source-Objekte und katalogisierten Objekte ausgewählt werden sollen.

Der Stern (\*) gibt an, dass alle Objektnamen ausgewählt werden sollen (siehe auch *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation).

`old-library` ist die ID der Library, die die zu kopierenden oder zu verschiebenden Objekte enthält.

`new-library` ist die ID einer vorhandenen oder neuen Library (siehe auch [Namenskonventionen für Libraries](#)).

## 2 Drücken Sie EINGABE.

Wenn der Kopier- oder Verschiebevorgang erfolgreich beendet wurde, wird das verarbeitete Kommando angezeigt, und die folgende Meldung erscheint: `Function completed successfully` (Funktion erfolgreich beendet).

Weitere Informationen zu den Funktionen, die Ihnen bei der Utility *SYSMAIN* zur Verfügung stehen, zum Beispiel zum Ersetzen und Umbenennen von Objekten, finden Sie in der *Utilities*-Dokumentation.

## Objekte in einer Library löschen

---

Eine Library wird in einer Natural-Umgebung nur so lange gepflegt, wie sie wenigstens ein Source-Objekt oder ein katalogisiertes Objekt enthält. Wenn Sie in einer Library alle Objekte löschen, ist diese Library nicht mehr vorhanden.

Sie können einzelne oder mehrere Objekte aus einer Library löschen, indem Sie das Systemkommando `LIST` oder eine Natural-Utility benutzen.

Die folgende Anleitung enthält Beispiele dafür, wie Sie Objekte mit Hilfe des Systemkommandos `LIST` aus der aktuellen Library löschen können und wie Sie unter Verwendung der Menüfunktionen bzw. Kommandos der Utility *SYSMAIN* Objekte aus einer anderen Library löschen können.

### ► Um Objekte mit `LIST` zu löschen

#### 1 Geben Sie folgendes Kommando ein:

```
LIST *
```

Der Stern (\*) gibt an, dass alle Source-Objekte und katalogisierten Objekte aufgelistet werden sollen.

(siehe auch *object-name-range* im Abschnitt *LIST* in der *Systemkommandos*-Dokumentation).

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm **LIST Objects in a Library**. Er enthält eine Liste der in der aktuellen Library in der aktuellen Systemdatei vorhandenen Source-Objekte und katalogisierten Objekte.

- 3 Geben Sie in der Spalte **Cmd** neben dem oder den gewünschten Objekt oder Objekten folgendes Zeilenkommando ein:

```
DE
```

Siehe auch den Beispielschirm weiter unten.

- 4 Drücken Sie EINGABE.

Es erscheint ein **DELETE**-Fenster (ähnlich dem unten gezeigten):

```
12:22:23          ***** NATURAL LIST COMMAND *****          2009-05-20
User SAG          - LIST Objects in a Library -          Library TEST

Cmd  Name          Type          S/C  SM Version  User ID  Date          Time
---  *          *          *   *   *          *          *          *
DE  DBTEST          Pr +-----DELETED-----+ 002-02-27  16:46:36
DE  DEMOPROG          Pr !          ! 000-12-20  11:40:46
DE  DEMOSPRG          Su !          ! 000-12-22  11:32:34
DE  DEMOTEXT          Te !          ! 000-12-20  13:59:26
___  EMP-L            Lo !          ! 002-07-29  12:46:43
___  EMPL              Pr !          ! 002-07-29  12:46:30
___  GDATEST           Gl !          ! 004-11-22  13:24:32
___  LDATEST           Lo !          ! 004-05-21  17:18:13
___  MAPTEST           Ma +-----+ 000-12-20  11:40:46
___  PDATEST           Parameter    S/C      4.1.03   SAG      2004-05-05  11:46:24
___  PDATEST2          Parameter    S/C      4.1.03   SAG      2004-11-23  10:12:07
___  PROCTEST          Processor    S/C      4.2.01   SAG      2004-05-05  11:25:22
___  SUBRTEST          Subroutine   S/C      S  4.1.03   SAG      2002-07-23  14:37:16
___  TESTPGM           Subprogram   S        S  4.1.03   SAG      2000-12-20  11:40:45
                                          14 Objects found

Top of List.
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Print Exit Sort      --      -      +      ++      >      Canc
```

- 5 Wählen Sie eine Option aus, indem Sie ein beliebiges Zeichen in dem Feld neben der gewünschten Option eingeben und EINGABE drücken:

Wenn Sie **Confirm each deletion** wählen, erscheint ein Fenster zu jedem zu löschenden Objekt, in dem Sie jeweils den Namen des Objekts eingeben müssen, um die Löschung zu bestätigen.

Wenn Sie **Delete without confirmation** wählen, erfolgt die Löschung sofort.

Wenn Sie **Exit** wählen, erfolgt keine Löschung, und das Fenster verschwindet.

► **Um Objekte mit SYSMAIN-Menüfunktionen zu löschen**

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility `SYSMAIN` auf; siehe **Schritte 1 bis 4** von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein `A` ein (Standardeinstellung), um alle Typen von Objekten auszuwählen.

(Ausnahmen: Für Error Messages geben Sie ein `E`, für Debug-Umgebungen ein `D` ein.)

Geben Sie im Feld **Function Code** ein `D` (Delete) ein.

- 3 Drücken Sie `EINGABE`.

Der Bildschirm **Delete Programming Objects** erscheint.

- 4 Geben Sie im Feld **Code** ein `A` ein, um alle Arten von Objekten auszuwählen: katalogisierte Objekte und Source-Objekte.

Überschreiben Sie im Feld **Sel. List** (Selection List) das `Y` (Yes) mit einem `N` (No). `Y` ist die Standardeinstellung.

Geben Sie im Feld **Object Name** einen Stern (`*`) ein, um alle Objektnamen auszuwählen. Dies ist die Standardeinstellung.

(Bezüglich gültiger Namensbereiche siehe *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Geben Sie im Feld **Source Library** die ID der Library ein, die Sie löschen möchten.

Die übrigen Eingabefelder lassen Sie unverändert.

- 5 Drücken Sie `EINGABE`.

In der angegebenen Library in der aktuellen Systemdatei werden alle Source-Objekte und katalogisierten Objekte gelöscht, und es erscheint die folgende Meldung: `Function completed successfully` (Funktion erfolgreich ausgeführt).

► **Um alle Objekte mit einem SYSMAIN-Kommando zu löschen**

- 1 Geben Sie folgendes Kommando ein:

```
SYSMAIN DELETE ALL * FM library-ID
```

Dabei gibt `ALL` an, dass alle Source-Objekte und katalogisierten Objekte ausgewählt werden sollen.

Der Stern (`*`) gibt an, dass alle Objektnamen ausgewählt werden sollen (siehe auch *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation).



*library-ID* ist die ID der Library, die die zu löschen Objekte enthält.

2 Drücken Sie EINGABE.

Wenn der Löschvorgang erfolgreich beendet wurde, wird das verarbeitete Kommando angezeigt, und die folgende Meldung erscheint: `Function completed successfully` (Funktion erfolgreich beendet).



# 6 Natural-Objekte pflegen und ausführen

---

▪ Natural-Objekte — Einführung .....	54
▪ Natural-Editoren oder Utilities benutzen .....	56
▪ Objekte auswählen und anzeigen .....	59
▪ Objekte erstellen und bearbeiten .....	60
▪ Objekte prüfen und testen .....	64
▪ Objekte speichern und katalogisieren .....	66
▪ Objekt-Verzeichnis-Informationen anzeigen .....	69
▪ Objekte kopieren .....	70
▪ Objekte drucken .....	71
▪ Objekte umbenennen .....	73
▪ Objekte verschieben .....	75
▪ Objekte löschen .....	76
▪ Programme ausführen .....	78

Ein Objekt ist Bestandteil einer Anwendung. Eine Natural-Anwendung besteht aus einem Satz Objekte, die interagieren, um eine bestimmte Aufgabe zu erfüllen.

Zu den für das Einrichten und Verwalten einer Natural-Anwendung zur Verfügung stehenden Objekten gehören Natural-Objekte und Nicht-Natural-Objekte.

Nicht-Natural-Objekte sind Objekte, die nicht mit einer Natural-Entwicklungsfunktion erstellt wurden und die außerhalb einer Natural- und Adabas-Umgebung gespeichert werden. Beispiele für Nicht-Natural-Objekte sind Bitmaps, XML-Sourcen, HTML-Dateien, DL/I-Subdateien und Predict-Regeln.

Dieser Abschnitt enthält allgemeine Informationen zu Natural-Objekten und beschreibt die Schritte, die zum Erstellen, Pflegen, Löschen und Ausführen eines Objekts erforderlich sind.

Alle Aktionen, die an einem Natural-Objekt ausgeführt werden können, erfolgen mit Hilfe von Natural-Kommandos und/oder Natural-Menüfunktionen. Informationen zur Benutzung von Kommandos und Menüfunktionen finden Sie im Abschnitt [Kommandos und Menü-Funktionen benutzen](#).

**Verwandte Themen im Abschnitt Libraries benutzen:**

- [Objekte in einer Library auflisten](#)
- [Objekte in einer Library suchen](#)
- [Objekte in einer Library löschen](#)

Dieser Abschnitt behandelt folgende Themen:

## Natural-Objekte — Einführung

---

Folgende Merkmale sind typisch für ein Natural-Objekt:

- Es wird in einer Library in einer Natural-Systemdatei gespeichert.
- Es besteht aus einem katalogisierten Objekt und/oder einem Source-Objekt.
- Es wird mit einem der Natural-Editoren oder einer der Natural-Utilities erstellt.

Dieser Abschnitt behandelt folgende Themen:

- [Katalogisiertes Objekt](#)
- [Source-Objekt](#)

- Objekttypen

### Katalogisiertes Objekt

Ein katalogisiertes Objekt ist die ausführbare (kompilierte) Form eines Natural-Objekts. Erstellt wird es durch den Natural-Compiler und gespeichert wird es als Objektmodul in einer Natural-Systemdatei. Unter dem Katalogisieren versteht man das Kompilieren des Sourcecodes und das Erstellen eines katalogisierten Objekts. Ausgelöst wird dieser Vorgang durch Absetzen des Natural-Systemkommandos `CATALOG` oder `STOW`.

Zur Ausführungszeit wird das katalogisierte Objekt in den Natural Buffer Pool geladen und dann durch das Natural-Laufzeitsystem ausgeführt. Natural-Objekte können nur dann ausgeführt werden oder sich gegenseitig referenzieren, wenn sie als katalogisierte Objekte in einer Natural-Systemdatei gespeichert worden sind.

Ein katalogisiertes Objekt kann nicht geändert oder dekompiert werden.

### Source-Objekt

Ein Source-Objekt (oder ein gespeichertes Objekt) enthält die menschenlesbare Form eines Natural-Sourcecodes. Der Sourcecode wird als Source-Objekt in einer Natural-Systemdatei gespeichert. Dies geschieht durch Absetzen des Systemkommandos `SAVE` oder `STOW`.

Um den in einem Source-Objekt enthaltenen Sourcecode ausführen zu können, müssen Sie den Sourcecode kompilieren, um generierten Objektcode zu erzeugen, der vom Natural-Laufzeitsystem interpretiert und ausgeführt werden kann.

### Verwandte Themen:

- *Natural-Systemdateien* - System-Architektur-Dokumentation
- *Editors*-Dokumentation
- *Utilities*-Dokumentation

### Objekttypen

In einer Natural-Anwendung können mehrere Arten von Natural-Objekten zum Einsatz kommen, um eine effiziente Anwendungsstruktur zu erstellen und um speziellen Programmierungs- und Anforderungserfordernissen gerecht zu werden. Zu den wichtigsten Natural-Objekten gehören Programme, Subprogramme, Routinen und Data Areas. Eine vollständige Übersicht und eine Beschreibung aller verfügbaren Objekttypen finden Sie im Abschnitt *Objekttypen* im *Leitfaden zur Programmierung*.

Informationen zu DDMs siehe *Natural-DDMs* in der *System-Architektur-Dokumentation* und im Abschnitt *Zugriff über Datendefinitionsmodule* im *Leitfaden zur Programmierung*.

## Natural-Editoren oder Utilities benutzen

---

Wenn Sie ein Natural-Objekt anlegen, warten oder löschen, benutzen Sie entweder einen Natural-Editor oder eine Natural-Utility.

Es gibt Wartungsfunktionen, die nicht für alle Objekttypen benutzt werden können. Zum Beispiel können Sie keine Objekte des Typs Adapter bearbeiten.

Ein Natural-Editor wird für alle Objekttypen aufgerufen, die beim Systemkommando `EDIT` oder auf dem Bildschirm **Development Functions** angegeben werden können. Je nach angegebenem Objekttyp ruft Natural den passenden Editor auf: den Programm-Editor, den Data-Area-Editor oder den Map-Editor. So ruft Natural zum Beispiel für ein Objekt des Typs Programm automatisch den Programm-Editor auf.

Eine Natural-Utility wird bei Objekten benutzt, die entweder zusätzliche Verwaltungsarbeiten erfordern oder die nicht in einer Library gewartet werden (z.B. DDMs). Eine Utility hat einen eigenen Editor.

Die folgende Tabelle enthält eine Aufstellung der Natural-Objekttypen und der zugehörigen Editoren bzw. Utilities:

Objekttype	Editor oder Utility
Local Data Area, Global Data Area, Parameter Data Area	Data-Area-Editor
Map (Bildschirm-Layout) Map-Editor-Profil Device-Profil	Map-Editor
Programm Subprogramm Subroutine Copycode Helproutine Class Text Programm-Editor-Profil	Programm-Editor
Error Message (Fehlermeldung)	SYSERR-Utility
Data Definition Module (DDM)	SYSDDM-Utility
Command Processor Source	SYSNCP-Utility
Parameterprofil	SYSARM-Utility
Debug-Umgebung	Debugger

**Verwandte Themen:**

- *Editors*-Dokumentation
- *Utilities*-Dokumentation

Dieser Abschnitt behandelt folgende Themen:

- [Einen Natural-Editor aufrufen](#)
- [Eine Natural-Utility aufrufen](#)
- [Persönliches Editor-Profil erstellen](#)

**Einen Natural-Editor aufrufen****▶ Um einen Natural-Editor aufzurufen**

- Benutzen Sie das Systemkommando `EDIT`.

Ein Beispiel für die Benutzung des Systemkommandos `EDIT` finden Sie unter [Beispiel für ein Systemkommando](#).

Oder:

Rufen Sie im Natural-Hauptmenü (**Main Menu**) das Menü **Development Functions** auf (siehe *Natural-Hauptmenü (Main Menu)*) und wählen Sie entweder die Funktion **Create Object** oder **Edit Object**.

Ein Beispiel für das Aufrufen eines Editors finden Sie unter [Beispiel für eine Menü-Funktion](#).

**Verwandtes Thema:**

- *EDIT - Systemkommandos*-Dokumentation

**Eine Natural-Utility aufrufen****▶ Um eine Natural-Utility aufzurufen**

- Geben Sie eines der folgenden Systemkommandos ein:

```
SYSERR
```

(für Fehlermeldungen)

```
SYSDDM
```

(für DDMs)

```
SYSNCP
```

(für Kommandoprozessor-Sourcen)

SYSPARM

(für Parameterprofile)

TEST

(für Debug-Umgebungen)

Oder:

Rufen Sie im Hauptmenü (**Main Menu**) das entsprechende Menü auf und wählen Sie die entsprechende Utility aus:

**Maintenance and Transfer Utilities** für SYSERR, SYSDDM und SYSNCP,

**Development Environment Settings** für SYSPARM und

**Debugging and Monitoring Utilities** für TEST.

#### Verwandtes Thema:

- *Natural-Hauptmenü (Main Menu)*

#### Persönliches Editor-Profil erstellen

Wenn Sie mit dem Natural-Programm-Editor oder mit dem Data-Area-Editor arbeiten, können Sie die Profilfunktion des Editors benutzen, um sich die aktuellen Einstellungen anzeigen zu lassen und persönliche Profileinstellungen vorzunehmen, die wirksam werden, wenn Sie Sourcecode bearbeiten.

#### ► Um die Editorprofileinstellungen anzuzeigen oder zu ändern

- 1 Geben Sie folgende Kommando in der Kommandoeingabezeile des Programm-Editors oder des Data-Area-Editors ein:

PROFILE

- 2 Drücken Sie EINGABE.

Der Bildschirm **Editor Profile** erscheint.

Informationen zu den Feldern und Optionen auf diesem Bildschirm finden Sie unter *Editor Profile* in der *Editors*-Dokumentation.



## Objekte auswählen und anzeigen

Sie können sich ein Source-Objekt anzeigen lassen, um den Sourcecode zu lesen oder zu kopieren, ohne dabei das Source-Objekt zu ändern. Der Sourcecode des angegebenen Objekts wird dann im Arbeitsbereich des zugehörigen Editors angezeigt.

Dazu können Sie entweder ein Objekt aus einer Liste auswählen oder den Namen des Objekts, das Sie anzeigen möchten, angeben.

Dieser Abschnitt beschreibt, wie Sie den Sourcecode mit dem Systemkommando `LIST` anzeigen können. Alternativ zu `LIST` können Sie die Funktion **List Object(s)** im Menü **Development Functions** benutzen. Siehe *Natural-Hauptmenü (Main Menu)*.

### ▶ Um ein Objekt aus einer Liste von Objekten auszuwählen

- 1 Rufen Sie die Funktion **LIST Objects in a Library** auf. Siehe **Schritte 1 und 2** von *Um Objekte mit LIST aufzulisten*.
- 2 Geben Sie in der Spalte **Cmd** neben dem gewünschten Objekt folgendes Kommando ein:

```
LI
```

- 3 Drücken Sie `EINGABE`.

Der Sourcecode des ausgewählten Objekts wird angezeigt.

### ▶ Um den Sourcecode eines angegebenen Objekts anzuzeigen

- 1 Geben Sie folgendes Kommando ein:

```
LIST object-name
```

Dabei ist *object-name* der Name des anzuzeigenden Objekts.

Wenn Sie für *object-name* keinen Namen angeben, wird der zurzeit im Arbeitsbereich befindliche Sourcecode angezeigt.

- 2 Drücken Sie `EINGABE`.

Der Sourcecode des angegebenen Objekts wird angezeigt.

### Verwandte Themen:

- *Objekte in einer Library auflisten*
- *LIST - Systemkommandos-Dokumentation*

## Objekte erstellen und bearbeiten

---

Dieser Abschnitt beschreibt die Schritte, die zum Erstellen und Bearbeiten eines Natural-Objekts mit einem Natural-Editor erforderlich sind. Informationen zu den zuvor erwähnten Utilities finden Sie in der *Utilities*-Dokumentation.

- Aktuelle Umgebung prüfen
- Programmiermodus einstellen
- Natural-Programmiersprache benutzen
- Sourcecode erstellen
- Source-Objekt bearbeiten
- Objekttyp festlegen

### Aktuelle Umgebung prüfen

Erstellt wird ein Natural-Objekt in der aktuellen Library in der aktuellen Systemdatei. Bevor Sie damit beginnen, ein Objekt zu erstellen oder zu bearbeiten, sollten Sie prüfen, ob Sie in der Library angemeldet sind, in der Sie das Objekt speichern oder auffinden möchten.

Eine Anleitung, wie Sie eine Library zuordnen und zwischen Libraries umschalten, finden Sie unter *Standard-Library-Zuweisung* und *Einloggen in eine Library*.

### Programmiermodus einstellen

Falls erforderlich, können Sie den Programmiermodus von Structured Mode auf Reporting Mode umschalten; siehe *Um den Programmiermodus zu wechseln*.

Weitere Informationen zum Programmiermodus finden Sie auch im Abschnitt *Natural-Programmiermodi* im *Leitfaden zur Programmierung*.

### Natural-Programmiersprache benutzen

Die Natural-Programmiersprache besteht aus Statements, Systemfunktionen und Systemvariablen.

Natural-Statements sind Programmieranweisungen, die zum Erstellen einer Natural-Programmsource verwendet werden.

Natural-Systemfunktionen werden beispielsweise eingesetzt, um mathematische Funktionen auszuführen.

Natural-Systemvariablen sind Standardvariablen, die von Natural zur Verfügung gestellt und erzeugt werden. Systemvariablen werden beispielsweise eingesetzt, um das Datum und die Uhrzeit zu erhalten.

**Verwandte Themen:**

- *Statements*-Dokumentation (Übersicht)
- *Systemfunktionen*-Dokumentation
- *Systemvariablen*-Dokumentation

**Sourcecode erstellen**

Dieser Abschnitt beschreibt an einem Beispiel, wie Sie Sourcecode erstellen. Verwendet wird das Systemkommando `EDIT` und der Programm-Editor. Darüber hinaus sind Beispiele für Editorkommandos und Anleitungen zum Navigieren in einer Source vorhanden.

Alternativ zum Systemkommando `EDIT` können Sie die Funktion **Create Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

**▶ Um neuen Sourcecode einzugeben**

- 1 Geben Sie folgendes Kommando ein:

```
EDIT object-type
```

Dabei ist *object-type* der Typ des Objekts, das Sie anlegen wollen.

Beispiel: Um ein Objekt des Typs Programm anzulegen, müssen Sie Folgendes eingeben:

```
EDIT PROGRAM
```

Wenn Sie für *object-type* keine Angabe machen, wird standardmäßig der Programm-Editor aufgerufen.

(Siehe auch *Objekttyp festlegen*.)

- 2 Drücken Sie `EINGABE`.

Es erscheint der Arbeitsbereich des Programm-Editors. In der obersten Zeile wird der Typ des anzulegenden Objekts (hier: Programm) angezeigt, zum Beispiel:

```
>                                     > + Program                               Lib SYSTEM
All  .....1.....2.....3.....4.....5.....6.....7..
0010
0020
0030
0040
0050
```

```
0060
0070
0080
0090
0100
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+... S 0 L 1
```

- 3 Sollte der Arbeitsbereich nicht leer sein, geben Sie am Eingabeaufforderungszeichen des Editors (>) folgendes Editorkommando ein:

```
CLEAR
```

und drücken Sie EINGABE.

CLEAR löscht den gesamten Inhalt des Arbeitsbereichs.

- 4 Die Eingabezeilen sind nummeriert. Beginnen Sie in der ersten Zeile (0010) des leeren Arbeitsbereichs, indem Sie den Sourcecode mit Hilfe der Copy- und Paste-Funktionen einer Terminal-Emulation (z. B. Entire Connection) einfügen oder ihn selbst eintippen (siehe auch [Objekte kopieren](#)). Wenn Sie die automatisch stattfindende Umwandlung von Klein- in Großbuchstaben nicht wünschen, können Sie die Standardeinstellung im Editor-Profil entsprechend ändern; siehe *General Defaults* in der *Editors*-Dokumentation.
- 5 Wenn alle angezeigten Zeilen im Eingabebereich des Editors vollgeschrieben sind, geben Sie folgendes Editorkommando ein:

```
ADD
```

und drücken Sie EINGABE.

Das Editorkommando ADD bewirkt, dass weitere neun Leerzeilen hinzugefügt werden. In die Programm-Source werden allerdings nur die Zeilen aufgenommen, in die Sie tatsächlich etwas schreiben. Frei gelassene Zeilen werden beim nächsten Drücken von EINGABE entfernt. Sie können diese Standardeinstellung im Editor-Profil ändern, siehe *Editor Defaults* in der *Editors*-Dokumentation. Die zur Verfügung stehenden Programm-Editor-Kommandos sind in der *Program Editor*-Dokumentation beschrieben.

► **Um in einer Source zu blättern**

- 1 Um an den Anfang des Sourcecodes zurückzukehren, geben Sie folgendes Editorkommando ein:

```
TOP
```

- Um an das Ende des Sourcecodes zu gelangen, geben Sie folgendes Editorkommando ein:

```
BOT
```

- Um im Sourcecode eine Seite nach unten zu blättern, drücken Sie PF8 oder EINGABE.
- Um im Sourcecode eine Seite nach oben zu blättern, drücken Sie PF7.

Die zur Verfügung stehenden Programm-Editor-Kommandos sind im Abschnitt *Editor Commands for Positioning* in der *Program Editor*-Dokumentation beschrieben.

### Source-Objekt bearbeiten

Sobald der Sourcecode als Source-Objekt gespeichert worden ist (siehe [Objekte speichern und katalogisieren](#)), rufen Sie durch Angabe des Source-Object-Namens den zugehörigen Natural-Editoren auf.

#### ► Um den Sourcecode eines Source-Objekts zu bearbeiten

- Geben Sie folgendes Kommando ein:

```
EDIT object-name
```

Dabei ist *object-name* der Name eines vorhandenen Source-Objekts, das sich in der aktuellen Library in der aktuellen Systemdatei befindet.

- Drücken Sie EINGABE.

Der Sourcecode des angegebenen Source-Objekts wird im Arbeitsbereich des zugehörigen Editors im Ändern-Modus angezeigt.

Alternativ zu `EDIT` können Sie die Funktion **Edit Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

Alternativ to `EDIT` können Sie auch das Systemkommando `READ` benutzen wie in [Objekte kopieren](#) beschrieben.

### Verwandtes Thema:

- *EDIT - Systemkommandos*-Dokumentation

## Objekttyp festlegen

Der Objekttyp wird beim Anlegen eines Objekts angegeben (die Standardeinstellung ist Programm) oder automatisch beim Einlesen eines vorhandenen Source-Objekts in den Arbeitsbereich festgelegt. Wenn Sie mit dem Programm-Editor oder dem Data-Area-Editor arbeiten, können Sie den Objekttyp jederzeit ändern, indem Sie das Editorkommando `SET TYPE` absetzen. Das gilt natürlich nur für Objekttypen, die mit demselben Editor bearbeitet werden.

### ▶ Um den Objekttyp zu ändern

- 1 Geben Sie folgendes Editorkommando ein:

```
SET TYPE object-type
```

Zum Beispiel:

```
SET TYPE SUBPROGRAM
```

- 2 Drücken Sie EINGABE.

Der mit dem Kommando angegebene neue Objekttyp wird am Bildschirm angezeigt (hier: Subprogram).

### Verwandte Themen:

- *SET TYPE - Program Editor-Dokumentation*  
*SET TYPE - Data Area Editor-Dokumentation*

## Objekte prüfen und testen

---

Beim Kompilieren (Katalogisieren) des Sourcecodes wird eine Syntaxprüfung vorgenommen und ausführbarer Objektcode erzeugt. Sie können den im Arbeitsbereich enthaltenen Sourcecode kompilieren, ohne ihn zuerst als katalogisiertes Objekt gespeichert zu haben (siehe [Objekte speichern und katalogisieren](#)). Zusätzlich können Sie das Kompilieren des Sourcecodes für Objekte des Typs Programm mit der Ausführung des Programms kombinieren. Siehe auch [Programme ausführen](#).

### ▶ Um Sourcecode zwecks Syntaxprüfung zu kompilieren

- 1 Geben Sie folgendes Systemkommando ein:

```
CHECK
```

- 2 Drücken Sie EINGABE.

Wird kein Syntaxfehler gefunden, dann wird der im Arbeitsbereich enthaltene Sourcecode kompiliert.

### ► Um Sourcecode zwecks Programmausführung zu kompilieren

- 1 Geben Sie folgendes Systemkommando ein:

```
RUN
```

- 2 Drücken Sie EINGABE.

Wird kein Syntaxfehler gefunden, dann wird der im Arbeitsbereich enthaltene Sourcecode kompiliert, und der generierte Code wird ausgeführt.

### Verwandte Themen:

- *CHECK* - Systemkommandos-Dokumentation
- *RUN* - Systemkommandos-Dokumentation

### Online-Hilfe bei Syntaxfehlern

Wenn nach dem Kompilieren keine Fehlermeldung erscheint, war die Sourcecode-Kompilierung erfolgreich.

Falls Natural jedoch beim Kompilieren einen Syntaxfehler feststellt, erscheint am Bildschirm eine Fehlermeldung. Außerdem wird die Programmzeile, die den Fehler enthält, mit einem E (Error) markiert und hell hervorgehoben, zum Beispiel:

```
>                                     > + Program      PGM01      Lib SYSTEM
      ....+....1....+....2....+....3....+....4....+....5....+....6....+....7...
0250  RD1. READ EMPLOYEES-VIEW BY NAME
0260      STARTING FROM #NAME-START
0270      THRU #NAME-END
0280 *
0290  IF LEAVE-DUE >= 20
0300      PERFORM MARK-SPECIAL-EMPLOYEES
0310  ELSE
0320      RESET #MARK
0330  END-IF
0340 *
E 0350  DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20 #MARK
0360 *
0370  END-READ
0380 *
0390  IF *COUNTER (RD1.) = 0
0400      REINPUT 'PLEASE TRY ANOTHER NAME'
0410  END-IF
0420 *
```

```
0430 END-REPEAT
0440 *
.....1.....2.....3.....4.....5..... S 49 L 25
NAT0305 Text string must begin and end on the same line.
```

Sie müssen den Fehler berichtigen, um das betreffende Objekt kompilieren zu können. Handelt es sich um einen Syntaxfehler, können Sie den Sourcecode nur als Source-Objekt speichern (siehe den folgenden Abschnitt). Informationen zum vorliegenden Fehler sowie zu Abhilfemaßnahmen erhalten Sie, wenn Sie die Online-Hilfe-Funktion benutzen.

### ► Um Hilfe zu Fehlermeldungen zu erhalten

- 1 Benutzen Sie eines der folgenden Kommandos:

```
HELP nnnn
```

or

```
? nnnn
```

Dabei steht *nnnn* für die vierstellige Fehlernummer.

Zum Beispiel:

```
HELP NAT0305
```

- 2 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Natural System Message** mit einer Erläuterung zu dem angegebenen Fehler.

Weitere Informationen zur Online-Hilfe finden Sie unter [Hilfe zu Natural-Meldungen](#).

## Objekte speichern und katalogisieren

---

Sie können den zurzeit als Source-Objekt im Arbeitsbereich enthaltenen Sourcecode speichern, indem Sie das Systemkommando `SAVE` benutzen. `SAVE` bewirkt kein Katalogisieren (Kompilieren) des Sourcecodes und führt folglich auch keine Syntaxprüfung durch.

Den zurzeit im Arbeitsbereich enthaltenen Sourcecode können Sie als Source-Objekt *und* als katalogisiertes (kompiliertes) Objekt speichern, indem Sie das Systemkommando `STOW` benutzen.

Sie können den zurzeit im Arbeitsbereich enthaltenen Sourcecode nur katalogisieren und als katalogisiertes Objekt speichern, indem Sie das Systemkommando `CATALOG` benutzen. `CATALOG` bewirkt *keine* Speicherung des Sourcecodes als Source-Objekt, das bearbeitet werden kann.



► **Um Sourcecode als Source-Objekt zu speichern**

- 1 Geben Sie folgendes Kommando am Editor-Eingabeaufforderungszeichen ein:

```
SAVE object-name
```

Dabei ist *object-name* der Name des Source-Objekts, das Sie anlegen wollen. Der Name des Objekts muss eindeutig sein und muss den **Namenskonventionen für Objekte** entsprechen.

- 2 Drücken Sie EINGABE.

Der Sourcecode wird als Source-Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert.

► **Um Sourcecode als Source-Objekt und/oder katalogisiertes Objekt zu speichern**

- 1 Geben Sie am Editor-Eingabeaufforderungszeichen eines der folgenden Kommandos ein:

```
STOW object-name
```

oder

```
CATALOG object-name
```

Dabei ist *object-name* der Name des Source-Objekts und/oder des katalogisierten Objekts, das Sie anlegen wollen. Der Name des Objekts muss eindeutig sein und muss den **Namenskonventionen für Objekte** entsprechen.

- 2 Drücken Sie EINGABE.

Wenn Sie `STOW` benutzen, wird der Sourcecode als Source-Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert. Zusätzlich wird der generierte Objektcode als katalogisiertes Objekt in derselben Library in derselben Systemdatei gespeichert.

Wenn Sie `CATALOG` benutzen, wird der Sourcecode nur als katalogisiertes Objekt unter dem angegebenen Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert. Dabei wird der Sourcecode nicht als Source-Objekt in der aktuellen Systemdatei mitgespeichert (oder aktualisiert, wenn das Kommando auf einem schon vorhandenen Source-Objekt ausgeführt wird). Der Sourcecode wird nur mit `SAVE` oder `STOW` gespeichert oder aktualisiert.

Eine Beschreibung, wie Sie feststellen können, ob ein Objekt schon als Source-Objekt und/oder katalogisiertes Objekt gespeichert worden ist, finden Sie unter **[Um Objekt-Verzeichnisinformationen anzuzeigen](#)**.

## Mehrere Objekte kompilieren

Mit dem Systemkommando `CATALL` können Sie mehrere, in der aktuellen Library enthaltene Source-Objekte kompilieren bzw. neu kompilieren.

### ► Um mehrere Objekte zu kompilieren

- 1 Geben Sie folgendes Systemkommando ein:

```
CATALL
```

- 2 Drücken Sie `EINGABE`.

Es erscheint der Bildschirm **Catalog Objects in Library** (siehe Beispiel weiter unten). Hier können Sie die zu verarbeitenden Objekte, das auszuführende Kompilierungskommando und zusätzliche Optionen, z. B. das Erstellen eines Fehlerberichts, angeben.

```

10:10:55          ***** NATURAL CATALL COMMAND *****          2009-05-20
User SAG          - Catalog Objects in Library -          Library SAGTEST

Catalog Objects from .. *_____ (start value, range, input list)
                    to .... _____ (end value)

Select object types:
X Global data areas
X Parameter data areas
X Local data areas
X Copycodes
X Texts
X External subroutines
X Subprograms
X Help routines
X Maps
X Adapter
X Programs
X Classes
Command ==>

X Recatalog only existing modules
_ Catalog all sources
Select function:
Save
X Catalog
Stow
Check
Select options:
Condition code in batch
X Renumber source-code lines
Keep result list
X Processing information
X Error report
Extended error report

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Menu Exit AddOp Sel.                               Canc
    
```

Weitere Informationen zu den Optionen auf dem Bildschirm **Catalog Objects in Library** finden Sie unter *CATALL* in der *Systemkommandos*-Dokumentation.

### Verwandtes Thema:

- *Beispiel für eine Kompilierung - System-Architektur*

## Objekt-Verzeichnis-Informationen anzeigen

Im Verzeichnis (Directory) zu einem Natural-Objekt finden Sie allgemeine Informationen zu dem Objekt, z. B. Name des Objekts, Name der Library, in es untergebracht ist, und Datum des Anlegens und Änderns.

### ► Um Objekt-Verzeichnisinformationen anzuzeigen

- 1 Geben Sie folgendes Kommando ein:

```
LIST DIR object-name
```

Zum Beispiel:

```
LIST DIR PGMTEST
```

- 2 Drücken Sie EINGABE.

Der Bildschirm **List Directory** erscheint. Beispiel für das Programm PGMTEST:

```
09:34:31          ***** NATURAL LIST COMMAND *****          2009-05-20
User SAG          - List Directory -          Library SAGTEST

Directory of Program PGMTEST          Saved on ... 2008-05-14 13:30:33
-----
Library .... SAGTEST    User-ID ..... SAG      Mode ..... Structured
TP-System .. COMPLETE  Terminal-ID .. 1      24
Op-System .. MVS/ESA    Transaction .. NAT42
NAT-Ver .... 4.2.4      Code page ... IBM01140
Source size .....      1046 Bytes

Directory of Program PGMTEST          Cataloged on 2006-05-23 16:36:12
-----
Library .... SAGTEST    User-ID ..... SAG      Mode ..... Structured
TP-System .. COMPLETE  Terminal-ID .. 1      16
Op-System .. MVS/ESA    Transaction .. NAT42
NAT-Ver .... 4.2.1      Code page ... IBM01140
Used GDA ...           Options ..... PCHECK DBSHORT PSIGNF GFID TQMARK
Size of global data ... 0 Bytes  Size in DATSIZE ..... 720 Bytes
Size in buffer pool ... 3416 Bytes
```

```
Size of OPT-Code ..... 0 Bytes
Initial OPT string ....
ENTER to continue
```

Weitere Informationen zum Bildschirm **List Directory** finden Sie unter *Directory-Informationen anzeigen* im Abschnitt *LIST* in der *Systemkommandos*-Dokumentation.

## Objekte kopieren

---

Sie können neue Objekte anlegen, indem Sie entweder den im Arbeitsbereich enthaltenen Sourcecode kopieren oder die Kopierfunktion einer Natural-Utility, z. B. SYSMAIN, verwenden.

### ▶ Um Sourcecode aus dem Arbeitsbereich zu kopieren

- 1 Lesen Sie den Sourcecode ein, den Sie kopieren möchten. Benutzen Sie dazu folgendes Kommando:

```
READ object-name
```

Dabei ist *object-name* der Name des Objekts, das den zu kopierenden Sourcecode enthält.

- 2 Drücken Sie EINGABE.

Der Sourcecode des angegebenen Objekts wird in den Arbeitsbereich eingelesen.

- 3 Geben Sie eines der folgenden Kommandos ein:

```
SAVE object-name
```

oder

```
STOW object-name
```

Dabei ist *object-name* der Name des Objekts, das Sie anlegen möchten.

- 4 Drücken Sie EINGABE.

Das neue Objekt wird in der aktuellen Library in der aktuellen Systemdatei als Source-Objekt gespeichert (mit SAVE) und als katalogisiertes Objekt (mit STOW).

### ▶ Um ein einzelnes oder mehrere Objekte mit SYSMAIN zu kopieren

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe [Schritte 1 bis 4](#) von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. E bei Fehlermeldungen (Error messages).

Geben Sie im Feld **Function Code** ein C ein (für Copy).

- 3 Drücken Sie EINGABE.

Der Bildschirm **Copy Programming Objects** erscheint.

- 4 Geben Sie im Feld **Code** ein A ein, um alle Arten von Objektmodulen auszuwählen: katalogisierte Objekte und Source-Objekte.

Überschreiben Sie im Feld **Sel. List** (Selection List) das Y (Yes) mit einem N (No). Y ist die Standardeinstellung.

Geben Sie im Feld **Object Name** den Namen des Objekts ein, das Sie kopieren möchten, oder geben Sie einen Bereich von Namen an. Durch Eingabe eines Sterns (\*) können Sie alle Objektnamen auswählen. Dies ist die Standardeinstellung

(Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *YSYMAIN Utility*-Dokumentation.)

Geben Sie im Feld **Source Library** die ID der Library ein, die die zu kopierenden Objekte enthält.

Geben Sie im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in die Sie die Objekte kopieren wollen.

Lassen Sie alle übrigen Felder unverändert.

- 5 Drücken Sie EINGABE.

Alle Source-Objekte und katalogisierten Objekte werden von der angegebenen Quell-Library in die angegebene Ziel-Library in der aktuellen Systemdatei kopiert. Danach erscheint die folgende Meldung: `Function completed successfully` (Funktion erfolgreich beendet).

## Objekte drucken

Sie können den Sourcecode eines Source-Objekts drucken, indem Sie das Systemkommando `LIST` benutzen.

Außerdem können Sie eine Liste der in einer Library enthaltenen Objekte drucken; siehe [Objektliste drucken](#).

### ► Um den Sourcecode eines Source-Objekts zu drucken

- 1 Sie haben die Wahl zwischen den folgenden Methoden:
  - Wählen Sie ein Objekt aus einer Liste aus, indem Sie die Funktion **LIST Objects in a Library** aufrufen; siehe [Schritte 1 und 2](#) von *Um Objekte mit LIST aufzulisten*.

Geben Sie in der Spalte **Cmd** neben dem gewünschten Objekt folgendes Kommando ein:

```
PR
```

Drücken Sie EINGABE.

■ Oder:

Geben Sie folgendes Kommando ein:

```
LIST object-name
```

Dabei ist *object-name* der Name des zu druckenden Objekts.

Drücken Sie EINGABE.

Der Sourcecode des angegebenen Objekts wird angezeigt.

Drücken Sie PF2.

Das Fenster **PRINT** erscheint.

- 2 Geben Sie im Feld **Destination** einen gültigen Druckernamen ein (ggf. fragen Sie Ihren Natural-Systemadministrator nach dem Namen eines Druckers, der in Ihrer derzeitigen Umgebung zur Verfügung steht). Wenn Sie möchten, können Sie die Seitenlänge ändern (standardmäßig sind 60 Zeilen eingestellt).

- 3 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Printout Specification**. Hier können Sie Druckereinstellungen vornehmen, z. B. Anzahl der Ausdrücke.

- 4 Drücken Sie EINGABE.

Das angegebene Source-Objekt wird auf dem angegebenen Drucker gedruckt.

**Verwandte Themen:**

- [Objektliste drucken](#)
- [LIST - Systemkommandos-Dokumentation](#)

## Objekte umbenennen

Sie können entweder einzelne Objekte mit dem Systemkommando `RENAME` umbenennen oder mehrere Objekte mit Hilfe der Natural-Utility `SYSMAIN`.

Alternativ zu `RENAME` können Sie die Funktion **Rename Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

### ► Um ein Objekt mit `RENAME` umzubenenen

- 1 Geben Sie folgendes Kommando ein:

```
RENAME object-name
```

Dabei ist *object-name* der Name des Objekts, das Sie umbenennen wollen.

- 2 Drücken Sie EINGABE.
- 3 Das Fenster **Rename Object** erscheint. Im Feld **Name** den Namen steht bereits der Name des mit dem Kommando angegebenen Objekts.
- 4 Geben Sie im Feld **New Name** den neuen Objektnamen ein.

Falls gewünscht, können Sie noch im Feld **New Type** einen neuen Objekttyp angeben.

- 5 Drücken Sie EINGABE.

Die folgende Meldung erscheint: `Object renamed successfully` (Objekt erfolgreich umbenannt).

### ► Um ein einzelnes oder mehrere Objekte mit Hilfe von `SYSMAIN` umzubenenen

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility `SYSMAIN` auf; siehe **Schritte 1 bis 4** von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein `A` (Standardeinstellung) ein, um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. `E` bei Fehlermeldungen (Error messages).

Geben Sie im Feld **Function Code** ein `R` ein (für Rename).

- 3 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Rename Programming Objects**.

- 4 Geben Sie im Feld **Code** ein `A` ein, um alle Arten von Objektmodulen auszuwählen: katalogisierte Objekte und Source-Objekte.

Geben Sie im Feld **Name** den Namen des umzubenennenden Objekts ein oder geben Sie einen Bereich von Namen an (im Beispiel weiter unten: TEST\*). Durch Eingabe eines Sterns (\*) können Sie alle Objektnamen auswählen. Dies ist die Standardeinstellung.

(Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Wenn Sie nur ein einzelnes Objekt umbenennen wollen, geben Sie im Feld **New Name** einen neuen Namen ein, und überschreiben Sie im Feld **Sel. List** das Y (Yes) durch ein N (No).

Geben Sie im Feld **Source Library** die ID der Library ein, die die umzubenennenden Objekte enthält.

Geben Sie ggf. im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in der Sie die Objekte speichern wollen.

Lassen Sie alle übrigen Felder unverändert.

5 Drücken Sie EINGABE.

Es erscheint ein Fenster, in dem Sie ein Y (Yes) eingeben können, wenn Sie eine Kopie des Objekts oder der Objekte, die umbenannt werden sollen, behalten möchten.

6 Drücken Sie EINGABE.

Wenn Sie einen Bereich von Objekten angegeben haben, erscheint das Fenster **Rename Selection** (siehe Beispiel weiter unten) mit einer Liste aller Objekte, die die angegebenen Auswahlkriterien (im Beispiel: TEST\*).

Geben Sie in der Spalte **C** neben dem bzw. den gewünschten Objekten ein A ein, um sowohl Source-Objekte als auch katalogisierte Objekte umzubenennen. Geben Sie in der Spalte **New Name** den neuen Namen ein; siehe unten:

```

16:39:39          ***** NATURAL SYSMAIN UTILITY *****          2009-05-20
User SAG          -      Rename Selection      -

RENAME ALL TEST* WITH XREF N IN SAGTEST WHERE DBID 10 FNR 32

  C  Name      Type  S/C  New Name      C  Name      Type  S/C  New Name
  -  - - - - -  - - -  - -  - - - - -  -  - - - - -  - - -  - -  - - - - -
A  TEST+     Progrm S    PGMT+___      _  TEST+2     Progrm S    _____
A  TESTCHAR  Progrm S/C  CHARTEST      A  TESTDIR    Progrm S    PGMDIR__
_  TESTDISP  Progrm S/C  _____    _  TESTDIS2   Progrm S/C  _____
_  TESTMM0   Proc   S/C  _____    A  TESTPGM_   Progrm S/C  PGMTTEST_
_  TESTTEST  Progrm S    _____    _  TESTXXX2   Progrm S    _____
A  TEST1     Subpgm S/C  SUBTEST1      A  TEST10     Subpgm S/C  SUB10___
A  TEST2     Subpgm S/C  SUBTEST2      _  TEST666    Progrm S/C  _____
    
```



```

Enter New Name and options, or '?' (Help) or '.' (Exit): _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren          Canc

```

- 7 Drücken Sie EINGABE.

Es erscheint eine Spalte **Message Text**, in der neben jedem umbenannten Objekt eine Bestätigungsmeldung angezeigt wird. Je nachdem, ob Sie die Option, eine Originalkopie des umzubenennenden Objekts zu behalten, markiert hatten oder nicht, erscheint entweder die Meldung `Renamed as` oder `Copied as` (Umbenannt als bzw. Kopiert als).

## Objekte verschieben

Sie können mit einer Natural-Utility, z. B. SYSMAIN, Objekte von einer Library in eine andere verschieben.

### ► Um Objekte mit Hilfe von SYSMAIN-Menüfunktionen zu verschieben

- 1 Rufen Sie das Hauptmenü (**Main Menu**) der Utility SYSMAIN auf; siehe [Schritte 1 bis 4](#) von *Um mittels der SYSMAIN-Menü-Funktionen alle Libraries aufzulisten*.
- 2 Geben Sie im Feld **Object Code** ein A ein (Standardeinstellung), um alle Objekttypen auszuwählen. Bei Objekttypen, die separat auf dem Menüschirm aufgelistet sind, geben Sie einen anderen Code ein, z. B. E bei Fehlermeldungen (Error messages).

Geben Sie im Feld **Function Code** ein M ein (für Move).

- 3 Drücken Sie EINGABE.

Es erscheint der Bildschirm **Move Programming Objects**.

- 4 Geben Sie im Feld **Code** ein A ein, um alle Arten von Objektmodulen auszuwählen: Source-Objekte und katalogisierte Objekte.

Überschreiben Sie im Feld **Sel. List** (Selection List) das Y (Yes) durch ein N (No). Y ist die Standardeinstellung.

Geben Sie im Feld **Object Name** den Namen des Objekts ein, das Sie verschieben möchten, oder geben Sie einen Bereich von Namen an. Durch Eingabe eines Sterns (\*) wählen Sie alle Objektnamen aus. Dies ist die Standardeinstellung.

(Informationen zu gültigen Namen finden Sie unter *Specifying a Range of Names* in der *SYSMAIN Utility*-Dokumentation.)

Geben Sie im Feld **Source Library** die ID der Library ein, die die zu verschiebenden Objekte enthält.

Geben Sie im Feld **Target Library** die ID einer vorhandenen oder einer neuen Library ein, in die Sie das oder die Objekte verschieben wollen.

Lassen Sie alle übrigen Felder unverändert.

- 5 Drücken Sie EINGABE.

Es erscheint ein Bestätigungsfenster.

- 6 Drücken Sie EINGABE, um die Verschiebeaktion zu bestätigen, oder einen Punkt (.), um die Aktion abubrechen.

Wenn die Verschiebeaktion erfolgreich ausgeführt wurde, sind alle angegebenen Source-Objekte und katalogisierten Objekte von der angegebenen Quell-Library in die angegebene Ziel-Library verschoben worden, und die folgende Meldung erscheint: `Function completed successfully` (Funktion erfolgreich ausgeführt).

## Objekte löschen

---

Sie können Objekte löschen, indem Sie entweder das Systemkommando `DELETE`, das Systemkommando `LIST` oder eine Natural-Utility, z. B. `SYSMAIN`, benutzen. Eine Anleitung, wie Sie Objekte mit `LIST` oder `SYSMAIN` löschen, finden Sie unter [Objekte in einer Library löschen](#).

Als Alternative zu `DELETE` können Sie die Funktion **Delete Object** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

### ► Um ein einzelnes oder mehrere Objekte mit `DELETE` zu löschen

- 1 Geben Sie eines der folgenden Kommandos ein:

```
DELETE object-name
```

oder

```
DELETE object-name*
```

oder

**DELETE \***

Dabei ist:

*object-name* der Name des zu löschenden Objekts.

*object-name\** ein spezieller Bereich auszuwählender Objekte. Zum Beispiel werden mit `TEST*` alle Objekte ausgewählt, die mit der Zeichenfolge `TEST` beginnen.

Durch Eingabe eines Sterns (\*) werden alle in der aktuellen Library in der aktuellen Systemdatei enthaltenen Objekte ausgewählt.

2 Drücken Sie `EINGABE`.

- Wenn Sie ein einzelnes Objekt angegeben haben, erscheint das Fenster **DELETE**.

Geben Sie den Namen des Objekts ein, um die Löschaktion zu bestätigen.

- Wenn Sie einen Bereich von Objekten eingegeben haben, erscheint das Fenster **Delete Sources and Objects**.

Geben Sie in der Spalte **M** neben dem oder den zu löschenden Objekten ein **B**, um sowohl das Source-Objekt als auch die katalogisierten Objekte zu löschen.

Drücken Sie `EINGABE`.

Das Fenster **DELETE** erscheint

Markieren Sie eine Option, indem Sie ein beliebiges Zeichen neben der gewünschten Option eingeben.

**Confirm each deletion** ruft das Fenster **DELETE** für das erste zu löschende Objekt auf. Geben Sie den Namen des Objekts ein, und drücken Sie `EINGABE`, um die Löschung zu bestätigen und das Fenster **DELETE** für die Löschung des nächsten Objekts zu öffnen.

**Delete without confirmation** bewirkt die sofortige Ausführung des Lösches oder der Löschen. **Exit (no deletion)** bewirkt den Abbruch des oder der Löschvorgänge.

3 Drücken Sie `EINGABE`.

Es erscheint der Bildschirm **Delete Sources and Objects**. Dort wird neben den für die Löschung ausgewählten Objekten eine Meldung. Die Meldung besagt, dass das Objekt gelöscht wurde (`deleted`) bzw. dass der Löschvorgang abgebrochen wurde (`not deleted`)

### Verwandtes Thema:

- *DELETE* - Systemkommandos-Dokumentation

## Programme ausführen

---

Sie können ein Objekt des Typs Programm mit einem Systemkommando ausführen. Alle anderen Objekttypen können dagegen nur ausgeführt oder aufgerufen werden, wenn Sie in diesem Programm oder in einem untergeordneten Objekt referenziert werden. Siehe auch *Mehrere Stufen (Levels) aufgerufener Objekte* im Leitfaden zur Programmierung.

Um ein Programm auszuführen, können Sie entweder das Systemkommando `RUN` oder `EXECUTE` benutzen.

Als Alternative zu `EXECUTE` können Sie die Funktion **Execute Program** im Menü **Development Functions** benutzen; siehe *Natural-Hauptmenü (Main Menu)*.

`RUN` führt den zurzeit im Arbeitsbereich befindlichen Sourcecode oder ein in einer Systemdatei gespeichertes katalogisiertes Objekt aus.

`EXECUTE` führt nur katalogisierte Objekte aus. Im Gegensatz zu `RUN` werden bei `EXECUTE` keine letzten Änderungen berücksichtigt, die an dem entsprechenden Code im Arbeitsbereich vorgenommen worden sein können. Denn diese Änderungen werden nur dann berücksichtigt, nachdem das Source-Objekt aktualisiert und neu kompiliert worden ist.

Durch die Ausführung eines katalogisierten Objekts wird der zurzeit im Arbeitsbereich befindliche Sourcecode nicht beeinflusst.

### ► Um ein Programm mit `RUN` auszuführen

- 1 Geben Sie folgendes Kommando ein:

```
RUN
```

oder

```
RUN program-name
```

Dabei ist *program-name* der Name des Source-Objekts vom Typ Programm, das in den Arbeitsbereich eingelesen wird.

- 2 Drücken Sie `EINGABE`.

Wenn kein Syntaxfehler gefunden wurde, wird der im Arbeitsbereich enthaltene Sourcecode kompiliert und ausgeführt.

### ► Um ein Programm mit `EXECUTE` auszuführen

- 1 Geben Sie folgendes Kommando ein:

```
EXECUTE program-name
```

Dabei ist *program-name* der Name eines katalogisierten Objekts des Typs Programm.

Das Schlüsselwort EXECUTE ist optional; es genügt, wenn Sie einfach nur den Namen des Programms (*program-name*) angeben. it is sufficient to specify.

2 Drücken Sie EINGABE.

Das Objekt wird ausgeführt.

**Verwandte Themen:**

- *RUN - Systemkommandos-Dokumentation*
- *EXECUTE - Systemkommandos-Dokumentation*
- *Objektausführung - System-Architektur-Dokumentation*
- *Suchreihenfolge bei der Objekt-Ausführung*
- *Beispiel für das Laden und Ausführen eines Objekts - System-Architektur-Dokumentation*



# 7 Natural-Hauptmenü (Main Menu)

---

▪ Natural-Hauptmenü (Main Menu) aufrufen .....	82
▪ Development Functions (Entwicklungsfunktionen) .....	84
▪ Development Environment Settings (Entwicklungsumgebungseinstellungen) .....	90
▪ Maintenance and Transfer Utilities (Wartungs- und Übertragungs-Utilities) .....	91
▪ Debugging and Monitoring Utilities (Utilities zur Fehlerbeseitigung und Überwachung) .....	92
▪ Example Libraries (Beispielprogramm-Libraries) .....	93
▪ Other Products (Sonstige Produkte) .....	94

Das Natural-Hauptmenü (**Main Menu**) ermöglicht Ihnen den Zugang zu den Natural-Entwicklungsfunktionen, Umgebungseinstellungen, Utilities und Beispiel-Libraries.

Der folgende Abschnitt enthält Informationen zu den Funktionen und Eingabemöglichkeiten, die auf dem Natural-Hauptmenü (**Main Menu**) und den untergeordneten Menüs vorhanden sind.

## Natural-Hauptmenü (Main Menu) aufrufen

---

Sie haben zwei Möglichkeiten, das Natural-**Hauptmenü** aufzurufen:

- Sie können eine Standardeinstellung definieren, indem Sie den Menü-Modus ein- oder ausschalten. Bei eingeschaltetem Menü-Modus wird das Natural-**Hauptmenü** beim Start der nächsten Session automatisch angezeigt.
- Sie können innerhalb einer Natural-Session das Natural-**Hauptmenü** aufrufen und schließen, wann immer Sie dies möchten:

### ▶ Um den Menü-Modus vor dem Session-Start ein- oder auszuschalten

- Geben Sie beim Start von Natural folgenden Profilparameter an:

`MENU=ON` (aktiviert den Menü-Modus)

oder

`MENU=OFF` (deaktiviert den Menü-Modus)

Siehe auch Profilparameter `MENU` in der *Parameter-Referenz*-Dokumentation.

### ▶ Um das Natural-Hauptmenü in einer Natural-Session aufzurufen oder zu schließen

- 1 Geben Sie folgendes Kommando ein:

```
MAINMENU
```

(ruft das Menü auf)

oder

```
MAINMENU OFF
```

(schließt das Menü)

- 2 Drücken Sie `EINGABE`.



Das Natural-Hauptmenü (**Main Menu**) sieht in etwa so aus wie das folgende Beispiel:

```

10:20:23          ***** NATURAL *****          2009-05-20
User SAG          - Main Menu -          Library TEST

                Function
                _ Development Functions
                _ Development Environment Settings
                _ Maintenance and Transfer Utilities
                _ Debugging and Monitoring Utilities
                _ Example Libraries
                _ Other Products
                _ Help
                _ Exit Natural Session

Command ===>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                               Canc

```

Zu Beginn Ihrer Natural-Session weist Ihnen Natural eine Standard-Library zu, die auf dem Bildschirm angezeigt wird. Im obigen Beispiel wird die ID (d.h. der Name) der Library im Feld **Library** in der linken oberen Ecke des Bildschirms angezeigt. Siehe auch [Standard-Library-Zuweisung](#).

Wie Sie eine [Menüfunktion ausführen](#) erfahren Sie im Abschnitt *Kommandos und Menü-Funktionen benutzen*.

Jede im Natural-Hauptmenü (**Main Menu**) aufgeführte Funktion ruft ein Menü mit entsprechendem Namen auf, in dem Sie weitere Funktionen auswählen können.

Funktion im Natural-Hauptmenü	Funktionen im entsprechenden Menü
<b>Development Functions</b>	Erstellen und Pflegen von Programmen, Maps (Masken), Data Areas (Datenbereichen) und sonstigen Komponenten, aus denen eine Natural-Anwendung besteht.
<b>Development Environment Settings</b>	Anzeigen und Ändern verschiedener Einstellungen, die Einfluß auf Ihre Natural-Session haben.
<b>Maintenance and Transfer Utilities</b>	Aufrufen einer Natural-Utility zum Erstellen oder Pflegen bestimmter Objekte oder zum Verschieben von Objekten in eine andere Umgebung.
<b>Debugging and Monitoring Utilities</b>	Aufrufen einer Natural-Utility zum Überwachen Ihrer Natural-Anwendungen und zum Auffinden von Fehlern in deren Verarbeitungsfluss.
<b>Example Libraries</b>	Auswahl von Libraries mit Beispielprogrammen und Programmierschnittstellen (APIs).
<b>Other Products</b>	Aufrufen anderer Software AG-Produkte.



**Anmerkung:** Die Position und die Farbe der Meldungszeile und der PF-Tastenzeilen im Natural-Hauptmenü und den untergeordneten Menüs kann über die User Exit Routine USR2003P geändert werden; siehe *Operations*-Dokumentation.

---

## Development Functions (Entwicklungsfunktionen)

---

Bei den im Menü **Development Functions** aufgelisteten Funktionen handelt es sich um Funktionen, die Sie häufig benötigen, wenn Sie mit Natural eine Anwendung entwickeln. Die Funktionen betreffen alle Natural-Objekte, die in der Library vorhanden sind, in der Sie zurzeit angemeldet sind.

### Verwandte Themen:

- *Leitfaden zur Programmierung*
- *Kommandos und Menü-Funktionen benutzen*
- *Natural-Objekte pflegen und ausführen*

Die folgende Tabelle enthält die Beschreibung der im Menü **Development Functions** vorhandenen Felder:

Feld	Beschreibung
<b>User</b>	<p><b>Benutzer</b></p> <p>Die Kennung (ID) des Natural-Benutzers, der sich für die aktuelle Session angemeldet hat.</p>
<b>Library</b>	<p><b>Library (Natural-Bibliothek)</b></p> <p>Die zurzeit aktive Library.</p> <p>Siehe auch <a href="#">Natural-Libraries benutzen</a>.</p>
<b>Mode</b>	<p><b>Programmiermodus</b></p> <p>Reporting oder Structured Mode. Siehe <a href="#">Programmiermodus</a>.</p>
<b>Work area empty</b>	<p><b>Arbeitsbereich leer</b></p> <p>Zeigt an, dass noch keine Source in den Arbeitsbereich geladen worden ist.</p> <p>Wurde bereits eine Source in den Arbeitsbereich geladen, dann werden hier der <b>Typ</b> und der <b>Name</b> des betreffenden Objekts angezeigt, zum Beispiel: Program PROGX.</p>
<b>Code</b>	<p><b>Funktionscode</b></p> <p>Der Code, der der gewünschten Funktion entspricht, zum Beispiel: C für <b>Create Object</b> (Objekt erstellen).</p> <p>Siehe auch <a href="#">Aufrufen einer Menü-Funktion</a>.</p>
<b>Type</b>	<p><b>Objekttyp</b></p> <p>Der Typ des Objekts, zum Beispiel: P für Programm. Weitere Informationen finden Sie unter <i>Objekttypen</i> im <i>Leitfaden zur Programmierung</i>.</p> <p>Wenn Sie den Namen eines schon vorhandenen Natural-Objekts angeben, können Sie das Feld <b>Type</b> leer lassen.</p> <p>Wie Sie den Objekttyp ändern können, erfahren Sie im Abschnitt <a href="#">Objekttyp festlegen</a>.</p>
<b>Name</b>	<p><b>Name des Objekts</b></p> <p>Informationen zu gültigen Objektnamen siehe <a href="#">Namenskonventionen für Objekte</a>.</p>
<b>Command ==&gt;</b>	<p><b>Kommandozeile</b></p> <p>Ein Eingabefeld, in das Sie ein Natural-Kommando eingeben können.</p> <p>Beispiel:</p> <p>Um ein Programm mit dem Namen PROGX zu bearbeiten, geben Sie folgendes Kommando ein:</p> <pre>EDIT PROGX</pre>

Feld	Beschreibung
	Siehe auch <i>Kommandos und Menü-Funktionen benutzen</i> .
PF (function keys)	<p><b>Funktionstasten</b></p> <p>PF-Tasten können Sie als Alternative zur Benutzung von Kommandos und Menüfunktionen benutzen. In den PF-Tastenzeilen am unteren Bildschirmrand ist angegeben, welche Taste mit welcher Funktion belegt ist.</p> <p>Siehe auch <i>Standard-PF-Tasten</i>.</p>

Die folgende Tabelle enthält Informationen zu den im Menü **Development Functions** vorhandenen Funktionen. Für die meisten dieser Menüfunktionen gibt es auch ein entsprechendes Natural-Systemkommando. Diese alternativ zu verwendenden Systemkommandos sind ebenfalls in der Tabelle aufgeführt. Eine ausführliche Beschreibung finden Sie in der *Systemkommandos*-Dokumentation.

Funktion	Code	Funktionsbeschreibung
<b>Create Object</b>	C	<p><b>Objekt anlegen</b></p> <p>Ruft den <b>Natural-Editor</b> auf, in dem Sie ein neues Objekt erstellen können, zum Beispiel: Programm, Map, Data Area.</p> <p>Geben Sie den Typ und den <b>Namen</b> des zu erstellenden Objekts an. Zur Auswahl eines Objekttyps können Sie ein Fragezeichen (?) im Feld <b>Type</b> eingeben. Es erscheint eine Liste der zu dieser Funktion zur Verfügung stehenden Objekttypen.</p> <p>Siehe auch <i>Objekte erstellen und bearbeiten</i>.</p>
<b>Edit Object</b>	E	<p><b>Objekt ändern</b></p> <p>Ruft einen <b>Natural-Editor</b> auf und zeigt die Source des angegebenen Objekts im Ändern-Modus an.</p> <p>Geben Sie den <b>Namen</b> eines vorhandenen zu bearbeitenden Objekts an. Zur Auswahl eines Objektes können Sie eine Liste der vorhandenen Objekte aufrufen; siehe <i>Objektbereiche angeben</i>.</p> <p>Zur Auswahl eines Objekttyps können Sie ein Fragezeichen (?) im Feld <b>Type</b> eingeben. Es erscheint eine Liste der zu dieser Funktion zur Verfügung stehenden Objekttypen.</p> <p>Siehe auch <i>Objekte erstellen und bearbeiten</i>.</p> <p>Entsprechendes Systemkommando: EDIT</p>
<b>Rename Object</b>	R	<p><b>Objekt umbenennen</b></p> <p>Ruft das Fenster <b>Rename Objects</b> auf, in dem Sie den Namen des angegebenen Objekts und/oder den <b>Typ</b> ändern können.</p> <p>Siehe auch <i>Objekte umbenennen</i>.</p>

Funktion	Code	Funktionsbeschreibung
		Entsprechendes Systemkommando: RENAME
Delete Object	D	<p><b>Objekt löschen</b></p> <p>Ruft das Fenster <b>Delete</b> für das angegebene Objekt auf. In diesem Fenster müssen Sie die Löschung bestätigen, indem Sie den Namen des Objekts im entsprechenden Eingabefeld erneut eingeben.</p> <p>Zur Auswahl eines Objektes können Sie eine Liste der vorhandenen Objekte aufrufen; siehe <i>Objektbereiche angeben</i>. In dieser Liste können Sie ein oder mehrere Objekte zum Löschen auswählen.</p> <p>Siehe auch <i>Objekte löschen</i>.</p> <p>Entsprechendes Systemkommando: DELETE</p>
Execute Program	X	<p><b>Objekt ausführen</b></p> <p>Führt ein Objekt des <b>Typs</b> Programm aus.</p> <p>Geben Sie den Namen des auszuführenden Objekts an.</p> <p>Es können nur Objekte des Typs Programm ausgeführt werden. Die übrigen Objekttypen müssen aus einem anderen Objekt heraus aufgerufen werden.</p> <p>Siehe auch <i>Programme ausführen</i>.</p> <p>Entsprechendes Systemkommando: EXECUTE</p>
List Object(s)	L	<p><b>Objekt(e) anzeigen</b></p> <p>Zeigt den Sourcecode des angegebenen Objekts an.</p> <p>Geben Sie den Namen des anzuzeigenden Objekts an. Zur Auswahl eines Objektes können Sie eine Liste der vorhandenen Objekte aufrufen; siehe <i>Objektbereiche angeben</i>.</p> <p>Zur Auswahl eines Objekttyps können Sie ein Fragezeichen (?) im Feld <b>Type</b> eingeben. Es erscheint eine Liste der zu dieser Funktion zur Verfügung stehenden Objekttypen.</p> <p>Siehe auch <i>Objekte in einer Library auflisten</i> .</p> <p>Entsprechendes Systemkommando: LIST</p>
List Subroutines Used	S	<p><b>Verwendete Subroutinen anzeigen</b></p> <p>Zeigt an, welche Objekte welche externen Subroutinen und Klassen verwenden.</p> <p>Entsprechendes Systemkommando: ROUTINES</p>

Dieser Abschnitt behandelt folgende Themen:

- [Programmiermodus](#)

- [Natural-Editoren](#)
- [Objektbereiche angeben](#)

## Programmiermodus

Natural bietet zwei Programmiermodi: Reporting Mode und Structured Mode. Grundsätzlich wird empfohlen, ausschließlich im Structured Mode zu programmieren, weil dieser Modus übersichtlicher strukturierte Anwendungen ergibt. Deshalb beziehen sich auch alle Erläuterungen und Beispiele in den Dokumentationen *Natural benutzen*, *Erste Schritte* und *Editors* auf den Structured Mode. Besonderheiten, die nur für den Reporting Mode gelten, werden nicht berücksichtigt.

Weitere Informationen finden Sie im Abschnitt *Natural-Programmiermodi* im *Leitfaden zur Programmierung*.

Das Feld **Mode** in der rechten oberen Ecke des Menüs **Functions** zeigt den zurzeit eingeschalteten Programmiermodus.

### ► Um den Programmiermodus zu wechseln

- 1 Überschreiben Sie im Feld **Mode** in der rechten oberen Ecke des Menüs **Development Functions** die erste Stelle mit einem S, um in den Structured Mode zu wechseln, oder einem R, um in den Reporting Mode zu wechseln.

Oder:

Geben Sie eines der folgenden Kommandos ein:

```
GLOBALS SM=ON
```

(Structured Mode ein)

oder

```
GLOBALS SM=OFF
```

(Reporting Mode ein)

- 2 Drücken Sie EINGABE.

Der Inhalt des Feldes **Mode** ändert sich von Reporting in Structured oder umgekehrt.

### Verwandte Themen:

- [Natural-Programmiermodi - Leitfaden zur Programmierung](#)
- [GLOBALS - Systemkommandos-Dokumentation](#)

## Natural-Editoren

Je nach Objekttyp ruft Natural den passenden Editor auf: den Programm-Editor, den Map-Editor oder den Data-Area-Editor. Weitere Information zu diesen Editoren finden Sie in den entsprechenden Abschnitten der *Editors*-Dokumentation.

### Objektbereiche angeben

Bei den Funktionen **Edit Object**, **List Object(s)** und **Delete Object** haben Sie die Möglichkeit, entweder den Namen eines einzelnen Objekts oder einen Bereich von Objektnamen anzugeben. Wenn Sie einen Namensbereich angeben, wird eine Objektliste angezeigt, aus der Sie ein oder mehrere Objekte auswählen, die Sie bearbeiten oder auflisten oder zur Löschung markieren wollen.

#### ▶ Um alle Objekte aufzulisten

- 1 Geben Sie im Feld **Name** einen Stern (\*) ein:

\*

- 2 Drücken Sie EINGABE.

Es wird die Liste aller in der aktuellen Library vorhandenen Objekte angezeigt.

#### ▶ Um Objekte mit Startwertangabe aufzulisten

- 1 Geben Sie im Feld **Name** einen Startwert und dahinter einen Stern (\*) ein:

Diese Art der Wertangabe wird als Stern-Notation bezeichnet.

Beispiel:

AB\*

- 2 Drücken Sie EINGABE.

Es wird für die aktuelle Library eine Liste mit Namen angezeigt, die mit AB beginnen (zum Beispiel: AB, AB1, ABC, ABEZ).

Die Liste enthält, zum Beispiel, keine Objektnamen, die mit AA1 oder ACB beginnen.



**Anmerkung:** Die Funktion **List Object(s)** bietet noch andere Optionen zur Angabe von Objekt-Namensbereichen; siehe Beschreibung des entsprechenden Systemkommandos LIST.

## Development Environment Settings (Entwicklungsumgebungseinstellungen)

Die folgende Tabelle enthält eine kurze Beschreibung der Funktionen im Menü **Development Environment Settings** und die Systemkommandos, die diesen Funktionen entsprechen. Eine ausführliche Beschreibung dieser Kommandos finden Sie in der *Systemkommandos*-Dokumentation.

Funktion	Funktionsbeschreibung	Entsprechendes Systemkommando
<b>Function-Key Settings</b>	<b>Funktionstastenbelegung</b> Dient zum Belegen von PF-Tasten mit Funktionen, die in Ihrer Natural-Session Verwendung finden sollen.	KEY
<b>Compilation Settings</b>	<b>Kompilierungsoptionen</b> Dient zum Setzen von Optionen, die die Kompilierung der Natural-Objekte beeinflussen.	COMPOPT
<b>Session Parameter Settings</b>	<b>Session-Parametereinstellungen</b> Dient zum Ändern der Session-Parametereinstellungen.  Siehe auch <a href="#">Natural-Umgebung konfigurieren</a> und den Abschnitt <i>Session-Parameter — Einführung</i> in der <i>Parameter-Referenz</i> -Dokumentation.	GLOBALS
<b>Profile Parameter Settings</b>	<b>Profilparametereinstellungen</b> Dient zum Ändern der Profilparametereinstellungen.  Beschreibungen der Profilparameter finden Sie in der <i>Parameter-Referenz</i> -Dokumentation und im Abschnitt <i>Profile Parameter Usage</i> in der <i>Operations</i> -Dokumentation.  Mit dem Systemkommando SYSPARM können Sie eine Utility gleichen Namens aufrufen, die in der <i>Utilities</i> -Dokumentation beschrieben wird.	SYSPARM
<b>Technical Session Information</b>	<b>Technische Informationen zur Session</b> Dient zur Anzeige von technischen Informationen zu Ihrer Natural-Session, zum Beispiel, Benutzerkennung, Library ID und Betriebssystem:	TECH
<b>System File Information</b>	<b>Systemdatei-Informationen</b> Dient zur Anzeige der derzeitigen Definitionen für die Natural-Systemdateien.  Siehe auch <i>Natural-Systemdateien</i> in der <i>System-Architektur</i> -Dokumentation.	SYSPROF



Funktion	Funktionsbeschreibung	Entsprechendes Systemkommando
<b>Product Installation Information</b>	<b>Informationen zu installierten Produkten</b> Dient zur Anzeige einer Liste der in Ihrer Umgebung installierten Produkte und zugehöriger Informationen.	SYSPROD
<b>Security Profile Information</b>	<b>Security-Profil-Informationen</b> Nur verfügbar, wenn Natural Security installiert ist. Dient zur Anzeige des zurzeit wirksamen Security-Profiles.	PROFILE

## Maintenance and Transfer Utilities (Wartungs- und Übertragungs-Utilities)

Die folgende Tabelle enthält eine kurze Beschreibung der Funktionen im Menü **Maintenance and Transfer Utilities** und die Systemkommandos, die diesen Funktionen entsprechen. Jedes dieser Kommandos ruft eine Natural-Utility gleichen Namens auf, die in der *Utilities*-Dokumentation beschrieben wird.

Funktion	Utility-Kurzbeschreibung	Entsprechendes Systemkommando
<b>Maintain Error Messages</b>	<b>Fehlermeldungen verwalten</b> Dient zum Anlegen und Verwalten von Meldungen, die Sie in Ihren Natural-Anwendungen ausgeben möchten.	SYSERR
<b>Maintain DDMs</b>	<b>DDMs verwalten</b> Dient zum Anlegen und Verwalten von DDMs (Data Definition Modules).	SYSDDM
<b>Maintain Command Processors</b>	<b>Kommandoprozessoren verwalten</b> Dient zum Anlegen und Verwalten von Kommandoprozessoren, die Sie in Ihren Natural-Anwendungen verwenden möchten.	SYSNCP
<b>Maintain Remote Procedure Calls</b>	<b>RPCs verwalten</b> Dient zum Einrichten und Verwalten von Remote Procedure Calls und bietet die Einstellmöglichkeiten zum Ausführen eines auf eine Remote Server befindlichen Natural-Subprogramms.	SYSRPC
<b>Transfer Objects to Other Libraries</b>	<b>Objekte in andere Libraries übertragen</b> Dient zum Übertragen von Natural-Objekten zwischen verschiedenen Libraries.	SYSMAN
<b>Transfer Objects to Other System Files</b>	<b>Objekte in andere Systemdateien übertragen</b> Dient zum Entladen oder Laden von Natural-Objekten.	SYSunLD

Funktion	Utility-Kurzbeschreibung	Entsprechendes Systemkommando
	<p>Sie können entweder das Systemkommando <code>YSUNLD</code> benutzen, um das erste Utility-Menü zum Entladen oder Laden von Objekten aufzurufen, oder Sie können das Systemkommando <code>NATUNLD</code> oder <code>NATLOAD</code> benutzen, um die untergeordnete Lade- oder Entlade-Utility direkt aufzurufen.</p> <p>Mit der Utility <code>NATUNLD</code> können Sie Natural-Objekte von einer Natural-Systemdatei in ein Workfile entladen.</p> <p>Mit der Utility <code>NATLOAD</code> können Sie Natural-Objekte von einem Workfile in eine Natural-Systemdatei laden.</p> <p><b>Anmerkung:</b> Die Funktionalität der Utilities <code>NATUNLD</code> und <code>NATLOAD</code> wird durch den Natural Object Handler abgedeckt. Wir empfehlen Ihnen, anstelle dieser Utilities den Object Handler zu benutzen.</p>	
<p><b>Transfer Objects to Other Platforms</b></p>	<p><b>Objekte auf andere Plattformen übertragen</b></p> <p>Dient zum Übertragen von Natural-Objekten und Adabas FDTs von einer Hardware-Plattform auf eine andere.</p> <p><b>Anmerkung:</b> Die Funktionalität der Utility <code>SYSTRANS</code> wird durch den Natural Object Handler abgedeckt. Wir empfehlen Ihnen, anstelle dieser Utilities den Object Handler zu benutzen.</p>	<p><code>SYSTRANS</code></p>
<p><b>Transfer Objects to Other Systems</b></p>	<p><b>Objekte in andere Systeme übertragen</b></p> <p>Dient zum Aufrufen des Object Handler, zum Übertragen und Verteilen von Natural-Objekten und Nicht-Natural-Objekten in Natural-Umgebungen.</p>	<p><code>YSOBJH</code></p>

## Debugging and Monitoring Utilities (Utilities zur Fehlerbeseitigung und Überwachung)

Die folgende Tabelle enthält eine kurze Beschreibung der Funktionen im Menü **Debugging and Monitoring Utilities** und die Systemkommandos, die diesen Funktionen entsprechen. Jedes dieser Kommandos ruft eine Natural-Utility gleichen Namens auf, die in der *Utilities*-Dokumentation beschrieben wird.

Funktion	Utility-Kurzbeschreibung	Entsprechendes Systemkommando
<b>Debugging</b>	Dient zur Suche nach Fehlern im Verarbeitungsfluss von Programmen.	TEST
<b>Logging of Database Calls</b>	Dient zum Protokollieren von Datenbank-Kommandos.	TEST DBLOG
<b>Issuing Adabas Calls</b>	Dient zur direkten Übergabe von Adabas-Kommandos an die Datenbank.	SYSADA
<b>Buffer Pool Maintenance</b>	Dient zur Überwachung und Anpassung des Natural Buffer Pool an Ihre Erfordernisse.	SYSBPM
<b>Editor Buffer Pool Maintenance</b>	Dient zur Überwachung und Anpassung des Buffer Pool des Software AG Editors an Ihre Erfordernisse.	SYSEDT
<b>TP-Specific Monitoring</b>	Dient zur Überwachung und und Steuerung von TP-Monitor-spezifischen Eigenschaften von Natural.	SYSTP
<b>Data Collection and Tracing</b>	Dient zum Sammeln von Überwachungs- und Accounting-Daten über den Verarbeitungsfluss einer Natural-Anwendung.	SYSRDC
<b>Error Information on Abnormal Termination</b>	Liefert Informationen, die der Technische Support der Software AG zur Fehlerdiagnose benötigt.	DUMP

## Example Libraries (Beispielprogramm-Libraries)

Wenn Sie im Natural-Hauptmenü den Eintrag **Example Libraries** wählen, wird eine Liste von Libraries angezeigt. Diese Libraries enthalten Beispielprogramme für Demonstrationszwecke und APIs, die von der Software AG ausgeliefert werden.

Library	Inhalt
<b>SYSEXP</b>	Beispielprogramme, auf die im <i>Leitfaden zur Programmierung</i> Bezug genommen wird.
<b>SYSEXR</b>	Beispielprogramme, auf die in der <i>Statements</i> -Dokumentation und der <i>Systemvariablen</i> -Dokumentation Bezug genommen wird.
<b>SYSEXV</b>	Beispielprogramme, die neue Natural-Funktionalität veranschaulichen.
<b>SYSEXT</b>	APIs und Beispielprogramme zum Einsatz der APIs. Siehe auch Systemkommando <i>SYSEXT</i> in der <i>Systemkommandos</i> -Dokumentation.
<b>SYSEXTP</b>	Beispielprogramme und APIs zu speziellen Funktionen, die nur unter bestimmten TP-Monitoren verfügbar sind.

## Other Products (Sonstige Produkte)

---

Wenn Sie im Natural-Hauptmenü den Eintrag **Other Products** wählen, erscheint eine Liste von Software AG Add-on-Produkten. Diese Produkte sind in Ihrer Umgebung installiert und können aus diesem Menü aufgerufen werden.

# 8

## Druck- und Arbeitsdateien (Printfiles und Workfiles)

---

Druckdateien (Printfiles) und Arbeitsdateien (Workfiles) werden als logische Dateien für eine Natural-Online- oder Batch-Umgebung definiert und physisch zu einer Datei oder einem Drucker zugeordnet. Dies geschieht mit Hilfe eines Natural-Parameters und/oder eines Control Statements für das darunter liegende Betriebssystem bzw. den TP-Monitor. Eine Änderung dieser Zuordnungen ist für die aktuelle Session zur Laufzeit möglich.

Zum Lesen und Schreiben der Daten in ein Workfile und zum Schreiben in ein Printfile gibt es entsprechende Natural-Statements.

Physische Printfile- bzw. Workfile-Zuordnungen erfolgen unabhängig von den Objekten, die in der entsprechenden Natural-Umgebung gepflegt werden. Deshalb brauchen Sie, wenn eine Zuordnung geändert wird, keine Statements in Objekten zu ändern, die Printfiles und/oder Workfiles referenzieren.

Allgemeine Informationen zu Printfiles und Workfiles finden Sie im Abschnitt *Printfiles und Workfiles* in der *System-Architektur-Dokumentation*.

Informationen zur Benutzung von Printfiles und Workfiles finden Sie in den entsprechenden Abschnitten der folgenden Dokumente.

### **Operations-Dokumentation:**

- *Datasets Used by Natural in z/OS Batch Mode*
- *Natural Datasets Used under a BS2000/OSD Batch Mode Session*
- *NATVSE Print and Work File Support for z/VSE Library Members*
- *Print File and Work File Support (VM/CMS-Umgebungen)*
- *Print and Work File Handling with External Datasets in a Server Environment - Natural as a Server under z/OS*
- *Output Reports and Workfiles - Profile Parameters Grouped by Function*

**Statements-Dokumentation:**

- *Verarbeitung von Arbeitsdateien/PC-Dateien*
- *CLOSE WORK FILE*
- *DEFINE WORK FILE*
- *READ WORK FILE*
- *WRITE WORK FILE*

**Parameter-Referenz-Dokumentation:**

- *PRINT - Print File Assignments*
- *WORK - Work File Assignments*

# 9 Natural-Umgebung konfigurieren

---

▪ Profilparameter benutzen .....	98
▪ Session-Parameter benutzen .....	101
▪ Parametrisierungsebenen .....	104

Die Konfiguration Ihrer Natural-Umgebung erfolgt mit Hilfe von Natural-Parametern.

Natural-Parameter dienen zur Standardisierung und Automatisierung von Entwicklungs- und Produktionsabläufen oder zum Einrichten von Standardeinstellungen entsprechend den persönlichen Anforderungen einzelner Benutzer. Verwendet werden Natural-Parameter zum Beispiel zum Setzen von Standardvorgaben für die Berichterstellung, zum Festlegen der Größe eines Reports oder der Größe eines benötigten Speicherbereichs, z.B. des Arbeitsspeichers eines Editors.

Die meisten Eigenschaften einer Natural-Umgebung werden schon durch die Software AG festgelegt. Ihr Natural-Administrator kann davon abweichende Standardeinstellungen konfigurieren, die für alle Natural-Benutzer gelten. Jeder Benutzer kann die Einstellungen entsprechend seinen eigenen Erfordernissen umkonfigurieren, indem er die Standard-Umgebungseinstellungen mit Hilfe eines oder mehrerer dynamischen Profil- oder Session-Parameters überschreibt.

Dieser Abschnitt behandelt folgende Themen:

## Profilparameter benutzen

---

Profilparameter werden statisch oder dynamisch angegeben.

Statische Parameter werden bei der Installation von Natural im Natural-Parametermodul NATPARM angegeben. Die dort vorgenommenen Einstellungen dienen als Standardvorgaben für jede Natural-Session.

Dynamische Parameter werden beim Starten einer Natural-Session angegeben. Sie können mit der Utility SYSPARM einen Satz dynamischer Parameter vordefinieren.

Ein vordefinierter Satz dynamischer Parameter ist ein Objekt des Typs Parameterprofil.

Beim Starten einer Natural-Session können Sie einen oder mehrere Parameterprofile und ein oder mehrere einzelne Profilparameter angeben; siehe [Um Profilparameter beim Session-Start zu setzen](#) weiter unten.

Durch die beim Session-Start (mit oder ohne Parameterprofil) angegebenen Parameter werden die Wertangaben in den entsprechenden Parametern in einem Standard-Parameterprofil überschrieben, das zu der Session zugeordnet und/oder im Natural-Parametermodul NATPARM definiert wurde; siehe auch [Parametrisierungsebenen](#).

### ► Um Profilparameter in NATPARM anzugeben

- Verfahren Sie entsprechend der Anleitung im Abschnitt *Using a Natural Parameter Module* in der *Operations*-Dokumentation.



► **Um ein Parametermodul mit SYSPARM anzulegen**

- 1 Rufen Sie die Utility SYSPARM auf, indem Sie das folgende Systemkommando eingeben:

```
SYSPARM
```

- 2 Drücken Sie EINGABE.

Das **Menü** der Natural SYSPARM Utility erscheint.

- 3 Geben Sie im Feld **Code** ein A (für **Add New Profile**) ein, und im Feld **Profile** den Namen des Parametermoduls, das Sie anlegen wollen, zum Beispiel:

```
11:36:19          ***** NATURAL SYSPARM UTILITY *****          2009-05-20
User SAG                      - Menu -

          Code  Function

          L    List Profiles
          D    Display Profile
          A    Add New Profile
          M    Modify Profile
          C    Copy Profile
          X    Delete Profile
          ?    Help
          .    Exit

Code ..... A
Profile .. TESTPROF  DBID ..   10  FNR ..... 1640
Copy to .. _____ Password ..
                               Cipher ....
```

```

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                                     Canc
    
```

4 Drücken Sie EINGABE.

Es erscheint der Editierbereich der Utility SYSPARM.

5 Geben Sie die einzelnen Profilparameter mit den gewünschten Werten ein (siehe Beispiel eines Parameterprofils weiter unten). Die möglichen Parameterwerte entnehmen Sie der Beschreibung *Profilparameter* in der *Parameter-Referenz-Dokumentation*.

```

16:57:37          ***** NATURAL SYSPARM UTILITY *****          2009-05-20
> FUSER=(10,32)                                           <
> LS=250,PS=60                                           <
> WORK=((6-8),AM=PC)                                     <
> DB=(ADAV7,*)                                          <
> PC=ON                                                  <
> AUTO=ON                                                <
> MENU=OFF                                               <
>                                                         <
>                                                         <
>                                                         <
>                                                         <
Help with parameters .. _____ (Profile name: TESTPROF)
Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit  Check Save                               Insrt Del   Copy   Canc
    
```

6 Drücken Sie PF4 (Check), um die Syntax auf Richtigkeit zu prüfen und sie ggf. zu korrigieren.

Informationen zur Syntax finden Sie unter *Dynamic Assignment of Parameter Values* in der *Operations*-Dokumentation.

- 7 Drücken Sie PF5 (Save), um das Parameterprofil als Source-Objekt unter dem angegebenen Namen in der aktuellen Systemdatei (standardmäßig ist das FNAT) zu speichern.

#### ▶ Um Profilparameter beim Session-Start zu setzen

- Geben Sie einen oder mehrere Parameterprofile (falls vorhanden) und/oder einen oder mehrere Profilparameter ein. Die Eingabe kann in beliebiger Reihenfolge erfolgen. Falls Sie jedoch einen Parameter mehr als einmal eingeben (FUSER im Beispiel weiter unten), wird der zuletzt angegebene Wert genommen (im Beispiel ist das der Wert 10, 245).

```
PROFILE=TESTPROF FUSER=(10,123) DSIZE=40 LS=250,PS=50 FUSER=(10,245)
```

#### Verwandte Themen:

- *Profile Parameter Usage* - *Operations*-Dokumentation
- *Profile Parameters Grouped by Function* - *Operations*-Dokumentation
- *Assignment of Parameter Values* - *Operations*-Dokumentation
- *SYSPARM Utility*-Dokumentation

## Session-Parameter benutzen

Session-Parameter können in einer aktiven Natural-Session und/oder in einem Natural-Objekt angegeben werden. Der Hauptzweck von Session-Parametern ist die Steuerung der Ausführung von Natural-Programmen.

#### ▶ Um einen Session-Parameter zu setzen

- Verfahren Sie entsprechend der Anleitung im Abschnitt *Setzen von Session-Parametern* in der *Parameter-Referenz*-Dokumentation.

#### ▶ Um die Parametereinstellungen für die aktuelle Session zu prüfen oder zu ändern

- Wählen Sie eine der folgenden Möglichkeiten:
  - Geben Sie folgendes Systemkommando ein:

```
GLOBALS
```

Drücken Sie EINGABE.

Es erscheint ein **Session Parameters**-Bildschirm ähnlich dem Beispiel weiter unten. Darin werden die aktuellen Session-Parameter-Einstellungen angezeigt:

```

10:28:03          ***** NATURAL GLOBALS COMMAND *****          2009-05-20
                    - Session Parameters -

(CC) Cond.Prog.Execution ..... OFF (LT) Limit ..... _99999999
(CF) Term.Control Character .... % (MT) Max.CPU Time ..... _0
(CPCVERR) CP Conversion Error .. ON (NC) Nat.Sys.Commands ..... OFF
(DC) Dec. Character ..... . (OPF) Overw.Prot.Fields ..... ON
(DFOUT) Date Format Output ..... S (PD) Page Dataset ..... _50
(DFSTACK) Date Format Stack Cmd .. S (PM) Print Mode ..... RP
(DFTITLE) Date Format Title ..... S (PS) Page Size ..... _31
(DO) Data to Display Order ..... L (REINP) Reinput on Error ..... ON
(DU) Dump Generation ..... OFF (SA) Sound Alarm ..... OFF
(EJ) Page Eject ..... ON (SF) Spacing Factor ..... _1
(FS) Default Format ..... OFF (SL) Source Line Length ..... _72
(FCDP) FC on Dyn.Prot.Fields ... ON (SM) Structured Mode ..... OFF
(IA) Input Assign ..... = (THSEPCH) Thousands Separator .. ,
(ID) Input Delimiter ..... , (TS) Translate Sys.Prog ..... OFF
(IM) Input Mode ..... F (WH) Wait on Hold ..... OFF
(LE) Limit Error ..... OFF (ZD) Zero Division ..... ON
(LS) Line Size ..... _80 (ZP) Zero Printing ..... ON

Command ==>
```

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Print Exit          Updat                               Canc

```

Die Buchstaben in Klammern sind die Namen der Session-Parameter; siehe auch *Liste der Parameter* in der Beschreibung des Systemkommandos *GLOBALS* in der *Systemkommandos-Dokumentation*.

Sie können die Einstellung eines Parameters ändern, indem Sie den vorgegebenen Wert neben dem Namen des Parameters überschreiben. Die gültigen Werte können Sie der entsprechenden *Session-Parameterbeschreibung* in der *Parameter-Referenz-Dokumentation* entnehmen.

Beispiel:

Um die Einstellung des Datumsformat-Parameters `DFOUT` zu ändern, überschreiben Sie den Wert im Eingabefeld neben **Date Format Output** (im obigen Beispiel ein `S`) mit einem `I`.

Drücken Sie `PF5` (`Updat`), um die Änderung zu speichern.

Das Format der Datumsvariablen ändert sich von `yy-mm-dd` (zum Beispiel: 2009-06-16) in `yyymdd` (zum Beispiel: 20090616)

■ Oder:

Geben Sie folgendes Kommando ein:

```
GLOBALS parameter=value
```

Wobei:

`parameter` der Session-Parameter ist (im Beispiel unten `DFOUT`).

`value` ein gültiger Wert für diesen Session-Parameter ist (im Beispiel unten `I`).

Sie können mehrere Parameter und Werte angeben.

Beispiel:

```
GLOBALS DFOUT=I PS=60
```

Drücken Sie EINGABE.

**Verwandte Themen:**

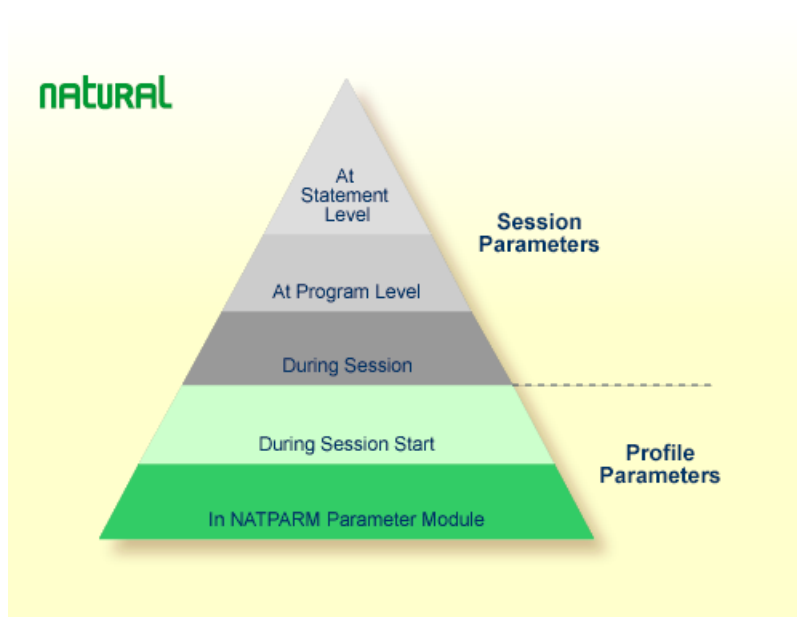
- *Session-Parameter — Einführung - Parameter-Referenz-Dokumentation*
- *GLOBALS - Systemkommandos-Dokumentation*

## Parametrisierungsebenen

---

Die Ebenen, auf denen Natural-Parameter angegeben werden können, sind hierarchisch aufgebaut. Ein auf einer höheren Ebene gesetzter Parameterwert hat Vorrang vor dem auf einer niedrigeren Ebene gesetzten Wert. Wenn Sie beispielsweise einen Parameter dynamisch angeben, überschreibt der neue Parameterwert die statische Angabe beim entsprechenden Parameter im Natural-Parametermodul NATPARM.

Das folgende Diagramm veranschaulicht, wann ein Parameter gesetzt werden kann, und stellt die Hierarchie von der untersten Ebene an der Basis der Pyramide bis zur höchsten Ebene an ihrer Spitze dar:



**Verwandtes Thema:**

- *Natural Parameter Hierarchy - Operations-Dokumentation*

# 10 Regeln und Namenskonventionen

---

▪ Standard-Zeichenzuweisungen .....	106
▪ Namenskonventionen für Objekte .....	106
▪ Namenskonventionen für Libraries .....	107
▪ Namenskonventionen für Benutzervariablen .....	108

Dieses Kapitel beschreibt Natural-spezifische Regeln und Namenskonventionen.

## Standard-Zeichenzuweisungen

---

In den Tabellen in diesem Abschnitt sind die Standard-Zeichenzuweisungen aufgeführt, die für Natural gelten. Diese Zuweisungen stimmen mit der IBM-Codepage 1140 (US English) überein, die die Zuordnung von Codepoints zu grafischen Zeichen festlegt. Codepoints werden durch hexadezimale Werte dargestellt. Weitere Informationen zu IBM-Codepages entnehmen Sie der entsprechenden IBM-Dokumentation

Die Einzeigesteuereinheit oder der Emulator legt fest, welches Zeichen für einen bestimmten hexadezimalen Wert angezeigt werden soll. Wenn Sie nicht die Codepage 1140, sondern eine andere Codepage verwenden, kann das folgende Auswirkung haben:

- Ein Zeichen liefert einen hexadezimalen Wert, der von Natural nicht akzeptiert wird.
- Ein von der Standard-Zeichenzuweisung abweichendes Zeichen liefert denselben hexadezimalen Wert, der für das Standardzeichen festgelegt wurde.

Beispiele:

Das bei der deutschen Codepage 1141 verwendete Zeichen @ (at-Zeichen) liefert nicht den hexadezimalen Wert 7C. Das bei der UK-englischen Codepage 1146 verwendete Zeichen L (Pfund-Zeichen) liefert denselben hexadezimalen Wert wie das bei der US-englischen Codepage 1140 verwendete Zeichen \$ (Dollar-Zeichen). Das bei der deutschen Codepage 1141 verwendete Zeichen § (Paragraphen-Zeichen) liefert denselben hexadezimalen Wert wie das bei der US-englischen Codepage 1140 verwendete Zeichen @.

## Namenskonventionen für Objekte

---

Dieser Abschnitt beschreibt die Namenskonventionen, die beim Speichern und/oder Katalogisieren eines Natural-Objekts in eine Natural-Systemdatei gelten.

Der Name eines Natural-Objekts kann 1 bis 8 Zeichen lang sein (die Zeichen sind in der Tabelle unten aufgeführt). Das *erste* Zeichen muss eines der folgenden Zeichen sein:

- ein Großbuchstabe aus dem Alphabet
- ein Nummernzeichen (#)
- ein Pluszeichen (+)

Wenn das erste Zeichen ein Nummernzeichen (#) oder ein Pluszeichen (+) ist, dann muss der Name mindestens noch ein weiteres Zeichen enthalten.



Ausnahme: Der Name eines Natural-DDMs kann 1 bis 32 Zeichen lang sein (die Zeichen sind in der Tabelle unten aufgeführt). Das *erste* Zeichen muss ein Großbuchstabe aus dem Alphabet sein.

Der Name eines Natural-Objekts oder DDMs kann aus den folgenden Zeichen bestehen:

Zeichen	EBCDIC Hex Value (EBCDIC Hexadezimaler Wert)	ISO-Zeichename (ISO Character Name)	Anmerkung
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Großbuchstabe aus dem Alphabet.
0 - 9	F0-F9	Digit zero - digit nine	Numerisches Zeichen.
-	60	Hyphen-minus	Bindestrich.
_	6D	Low line	Unterstrich.
/	61	Solidus	Schrägstrich.
@	7C	Commercial at	Kaufmännisches „At“-Zeichen.
\$	5B	Dollar sign	Dollar-Zeichen.
&	50	Ampersand	Kaufmännisches „Und“-Zeichen. Nur in Sprachcodes erlaubt.  Siehe auch <i>Definition der Sprache eines Natural-Objektes</i> im Abschnitt <i>Bildschirmgestaltung (Gestaltung der Benutzeroberflächen von Anwendungen)</i> im <i>Leitfaden zur Programmierung</i> .
#	7B	Number sign	Nummernzeichen.
+	4E	Plus sign	Pluszeichen. Nur als erstes Zeichen erlaubt.

## Namenskonventionen für Libraries

Dieser Abschnitt beschreibt die Namenskonventionen, die für eine Natural-Library gelten.

Der Name (ID) einer Library kann 1 bis 8 Zeichen lang sein und darf nicht mit "SYS" beginnen. Das Präfix "SYS" ist reserviert für die Natural-System-Libraries.

Der Library-Name muss mit einem Großbuchstaben aus dem Alphabet beginnen. Alle weiteren Zeichen aus dem Alphabet müssen auch in Großbuchstaben sein. Der Library-Name darf keine Leerzeichen enthalten.

Ein Library-Name kann aus den folgenden Zeichen bestehen:

Zeichen	EBCDIC Hex Value (EBCDIC Hexadezimaler Wert)	ISO-Zeichename (ISO Character Name)	Anmerkung
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Großbuchstabe aus dem Alphabet.
0 - 9	F0-F9	Digit zero - digit nine	Numerisches Zeichen.
-	60	Hyphen-minus	Bindestrich.
_	6D	Low line	Unterstrich. Nicht erlaubt als letztes Zeichen.

## Namenskonventionen für Benutzervariablen

---

Dieser Abschnitt beschreibt die Namenskonventionen, die für eine Benutzervariable gelten. Weitere Informationen zu Benutzervariablen finden Sie unter *Benutzervariablen* im *Leitfaden zur Programmierung*.

Die folgenden Themen werden behandelt:

- [Länge der Variablennamen](#)
- [Beschränkungen für Variablennamen](#)
- [Erlaubte Zeichen im Variablennamen](#)
- [Erstes Zeichen im Variablennamen](#)
- [Kleinbuchstaben im Variablennamen](#)

### Länge der Variablennamen

Der Name einer Benutzervariablen kann 1 bis 32 Zeichen lang sein.

Sie können Variablennamen benutzen, die länger als 32 Zeichen sind (zum Beispiel in komplexen Anwendungen, bei denen längere, bedeutungsvolle Variablennamen die Lesbarkeit der Programme erhöhen); jedoch werden nur die ersten 32 Zeichen berücksichtigt. Diese Zeichenfolge muss eindeutig sein. Alle übrigen Zeichen werden von Natural ignoriert.

## Beschränkungen für Variablennamen

Der Name einer Benutzervariablen darf kein reserviertes Natural-Schlüsselwort sein.

Innerhalb eines Natural-Programms dürfen Sie denselben Namen nicht gleichzeitig für eine Benutzervariable und ein Datenbankfeld benutzen, weil dies zu Zuordnungsfehlern führen kann (siehe *Datenstrukturen qualifizieren* im Abschnitt *Benutzervariablen im Leitfaden zur Programmierung*).

## Erlaubte Zeichen im Variablennamen

Der Name einer Benutzervariablen kann aus den folgenden Zeichen bestehen:

Zeichen	EBCDIC Hex Value (EBCDIC Hexadezimaler Wert)	ISO-Zeichename (ISO Character Name)	Anmerkung
A - Z	C1-C9 D1-D9 E2-E9 81-89 91-99 A2-A9	Latin capital and/or small letter A - Z	Groß- und/oder Kleinbuchstabe aus dem Alphabet. Ein Kleinbuchstabe ist nicht als erstes Zeichen erlaubt.
0 - 9	F0-F9	Digit zero - digit nine	Numerisches Zeichen.
-	60	Hyphen-minus	Bindestrich.
_	6D	Low line	Unterstrich.
/	61	Solidus	Schrägstrich.
@	7C	Commercial at	Kaufmännisches „At“-Zeichen.
\$	5B	Dollar sign	Dollar-Zeichen.
&	50	Ampersand	Kaufmännisches „Und“-Zeichen.
#	7B	Number sign	Nummernzeichen.
+	4E	Plus sign	Pluszeichen. Nur als erstes Zeichen erlaubt.

## Erstes Zeichen im Variablennamen

Das erste Zeichen im Namen muss eines der folgenden Zeichen sein:

Zeichen	EBCDIC Hex Value (EBCDIC Hexadezimaler Wert)	ISO-Zeichename (ISO Character Name)	Anmerkung
A - Z	C1-C9 D1-D9 E2-E9	Latin capital letter A - Z	Großbuchstabe aus dem Alphabet.
&	50	Ampersand	Kaufmännisches „Und“-Zeichen.
#	7B	Number sign	Nummernzeichen.

Zeichen	EBCDIC Hex Value (EBCDIC Hexadezimaler Wert)	ISO-Zeichename (ISO Character Name)	Anmerkung
+	4E	Plus sign	Pluszeichen.

Wenn das erste Zeichen ein Nummernzeichen (#), ein Pluszeichen (+) oder ein kaufmännisches „Und“-Zeichen (&) ist, dann muss der Name mindestens noch ein weiteres Zeichen enthalten.

Variablen in einer Global Data Area (GDA), die ein Pluszeichen (+) als erstes Zeichen haben, müssen auf Ebene 1 definiert werden; siehe auch *Global Data Area* im Abschnitt *Datenbereiche (Data Areas)* im *Leitfaden zur Programmierung*. Andere Ebenen werden nur in einer Redefinition benutzt.

Ein Pluszeichen (+) ist nur bei anwendungsunabhängigen Variablen (AIVs) und bei Variablen in einer Global Data Area (GDA) als erstes Zeichen im Namen erlaubt. Die Namen von anwendungsunabhängigen Variablen (AIVs) müssen mit einem Pluszeichen (+) beginnen; siehe auch *Definition von anwendungsunabhängigen Variablen* bei der Beschreibung des Statements `DEFINE DATA` in der *Statements-Dokumentation*.

Ein kaufmännisches „Und“-Zeichen (&) als erstes Zeichen im Namen wird bei dynamischen Sourcecode-Änderungen benutzt (siehe die Beschreibung des Statements `RUN` in der *Statements-Dokumentation*) und als dynamisch ersetzbares Zeichen beim Definieren von Processing-Rules (Verarbeitungsregeln); siehe die Beschreibung des Map-Editor in der *Editors-Dokumentation*.

### Kleinbuchstaben im Variablennamen

Kleinbuchstaben können nur als zweites und als darauf folgende Zeichen eines Variablennamens eingegeben werden. Kleinbuchstaben, die als Teil eines Variablennamens eingegeben werden, werden intern in Großbuchstaben umgesetzt, wenn die Option `LOWSRCE` des Systemkommandos `COMPOPT` auf `ON` gesetzt ist (siehe *Systemkommandos-Dokumentation*). Kleinbuchstaben werden nicht in Großbuchstaben umgesetzt. Kleinbuchstaben und die entsprechenden Großbuchstaben werden als unterschiedliche Zeichen interpretiert, wenn

- die Option `LOWSRCE` des Systemkommandos `COMPOPT` auf `OFF` (Standardwert) gesetzt ist, und
- die Eingaben in den Editor nicht in Großbuchstaben umgesetzt werden (die Umsetzung in Großbuchstaben im Editor wird durch Editor-Profiloptionen und durch betriebssystemabhängige Optionen gesteuert).

Dies hat zum Beispiel zur Folge, dass die Namen `#FIELD` und `#fielD` als zwei verschiedene Feldnamen interpretiert.



**Anmerkung:** Aus Kompatibilitätsgründen sollten Sie dieses Merkmal nicht nutzen, wenn Sie planen, Natural-Anwendungen, die auf Großrechnern entwickelt wurden, auf Windows, UNIX oder OpenVMS zu portieren. Wenn Sie Kleinbuchstaben als Teil des Variablennamens verwenden, empfehlen wir Ihnen dringend, dafür Sorge zu tragen, dass die Namen von Variablen, egal ob in Groß- oder Kleinbuchstaben, immer eindeutig sind.

## Stichwortverzeichnis

---

