

# **Natural**

## **Systemkommandos**

Version 8.2.7

Oktober 2018

Dieses Dokument gilt für Natural ab Version 8.2.7.

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

Copyright © 1979-2018 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

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

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

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

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

**Dokument-ID: NATMF-NNATSYSCOM-827-20180523DE**

# Inhaltsverzeichnis

Vorwort .....	vii
1 Kommandoausführung .....	1
Kommandoeingabe .....	2
Kommandozeile .....	2
NEXT-Zeile .....	2
MORE-Zeile .....	3
2 Systemkommando-Syntax .....	5
Syntax-Elemente .....	6
Kommandosyntax-Beispiel .....	7
3 Systemkommandos sortiert nach Kategorien .....	9
In Natural navigieren .....	10
Natural-Entwicklungsumgebung .....	10
Anwendungen mit Natural-Programmobjekten pflegen .....	11
Monitor-, Debug- und Trace-Funktionen .....	13
Natural mit anderen Software AG-Produkten .....	14
4 AIV .....	17
5 BUS .....	19
6 CATALL .....	21
Objekte katalogisieren von/bis .....	22
Predict Set und Set Benutzer .....	23
Vorhandene Module neu katalogisieren oder Alle Sourcen katalogisieren .....	23
Objektypen auswählen .....	23
Funktion auswählen .....	24
Optionen auswählen .....	24
Auswahlliste .....	25
Direktkommando-Syntax .....	26
7 CATALOG .....	31
8 CHECK .....	33
9 CLEAR .....	35
10 COMPOPT .....	37
Syntax-Erklärung .....	38
Compiler-Schlüsselwortparameter angeben .....	38
Allgemeine Compiler-Optionen .....	39
Kompilierungsoptionen für Versions- und Plattformkompatibilität .....	53
11 CPINFO .....	57
12 DELETE .....	61
Syntax-Erklärung .....	62
Auswahlliste .....	63
Schutz vor versehentlichem Löschen .....	63
Beispiele .....	63
13 DUMP .....	65
14 EDIT .....	67
Syntax 1 .....	68

Syntax 2 .....	70
Syntax 3 .....	70
15 EDT .....	73
Syntax-Erklärung .....	74
EDT-Kommandos .....	74
EDT-Funktionstasten .....	75
16 EXECUTE .....	77
Syntax-Erklärung .....	78
Beispiele für das EXECUTE-Kommando .....	79
17 FIN .....	81
18 GLOBALS .....	83
Syntax-Erklärung .....	84
Liste der Parameter .....	84
Zusammenhang zwischen GLOBALS, SET GLOBALS und anderen Statements .....	85
19 HELP .....	87
20 INPL .....	89
21 KEY .....	91
Kommandos zuweisen .....	93
Aktivieren/Deaktivieren aller Tasten - KEY ON/OFF .....	93
Aktivieren/Deaktivieren einzelner Tasten — KEY key=ON/OFF .....	94
22 LAST .....	95
23 LASTMSG .....	97
24 LIST .....	99
Syntax-Übersicht .....	100
Inhalt des Arbeitsbereichs auflisten .....	107
Sourcecode eines einzelnen Objekts anzeigen .....	108
Anzahl und Größe von Objekten anzeigen .....	108
Sourcecode mehrerer Objekte nacheinander anzeigen .....	109
Liste von Objekten anzeigen .....	109
Vorsortierte Liste von ausgewählten Objekten anzeigen .....	109
Langnamen katalogisierter Subroutinen und Klassen anzeigen .....	110
NOC-Optionen katalogisierter Objekte anzeigen .....	110
Compiler-Optionen katalogisierter Objekte anzeigen .....	110
Compiler Output zu einem katalogisierten Objekt anzeigen .....	110
Directory-Informationen anzeigen .....	112
DDMs (Views) anzeigen .....	113
Optionen .....	113
Objekt-Auswahlliste .....	118
Source-Liste .....	126
Individuelles List-Profil erstellen .....	131
25 LIST XREF .....	137
26 LISTDBRM .....	139
27 LISTSQL .....	143
28 LISTSQLB .....	149

29 LOGOFF .....	153
30 LOGON .....	155
31 MAIL .....	157
32 MAINMENU .....	159
33 NATQVS .....	161
34 NOCOPT .....	163
35 NOCSHOW .....	165
36 NOCSTAT .....	167
37 PROFILE .....	169
38 PROFILER .....	171
39 RDC .....	173
40 READ .....	175
41 RENAME .....	177
42 RENUMBER .....	179
43 RETURN .....	181
44 ROUTINES .....	183
45 RPCERR .....	185
46 RUN .....	187
47 SAVE .....	189
48 SCAN .....	191
Menü-Optionen .....	192
SCAN-Editierkommandos .....	195
SCAN mittels Direktkommando .....	195
Individuelles SCAN-Profil definieren .....	199
SCAN unter Natural Security .....	200
49 SCRATCH .....	201
50 SETUP .....	203
Syntax-Erklärung .....	204
Beispiel für SETUP/RETURN .....	205
51 SHOWDBS .....	207
Entsprechendes API USR8211N .....	209
52 SHOWSU .....	211
53 SQLDIAG .....	213
54 SQLERR .....	217
55 STOW .....	221
56 STRUCT .....	223
Generate Structured Source into Work Area .....	224
Display Structure of Source .....	227
Print Structure of Source .....	228
Write Structure of Source into Work Area .....	229
57 SYSADA .....	231
58 SYSAPI .....	233
59 SYSBPM .....	235
60 SYSCP .....	237
61 SYSDB2 .....	239

62	SYSDDM .....	241
63	SYSEDT .....	243
64	SYSERR .....	245
65	SYSEXT .....	247
66	SYSEXV .....	249
67	SYSFILE .....	251
68	SYSMAIN .....	253
69	SYSNCP .....	255
70	SYSOBJH .....	257
71	SYSARM .....	259
72	SYSPROD .....	261
	Kommando SYSPROD im Online-Modus benutzen .....	262
	Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen) .....	263
	Anwendungsprogrammierschnittstellen (APIs) .....	263
73	SYSPROF .....	265
74	SYSRPC .....	267
75	SYSTP .....	269
76	TECH .....	271
77	TEST .....	273
78	TEST DBLOG .....	275
79	UNCATALOG .....	277
80	UNLOCK .....	279
	Natural-Objekte entsperren .....	280
	UNLOCK-Parameter-Beschreibungen .....	281
	Parameterverarbeitung und Anzeige der gefundenen Objekte .....	282
	Batch-Verarbeitung .....	283
81	UPDATE .....	285
82	XREF .....	287
83	ZIIP .....	289
	Syntax der ziip-subfunction-and-option .....	290
	Syntax-Element-Beschreibung .....	290
	ziIP-Berichte .....	292
	ZIIP-Kommando-Menü für Berichtausgabe-Optionen .....	295
	User Exit für ZIIP PRINT-Kommando-Optionen .....	295
	Anwendungsprogrammierschnittstelle (API) für ZIIP-Kommando-Funktionen .....	295
	Stichwortverzeichnis .....	297

---

# Vorwort

---

Diese Dokumentation beschreibt die Natural-Systemkommandos.

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

Diese Dokumentation ist in die folgenden Abschnitte untergliedert:

## **Kommandoausführung**

Beschreibt die Regeln, die für die Eingabe eines Systemkommandos gelten.

## **Kommando-Syntax**

Erklärt die Symbole, die in den Diagrammen verwendet werden, in denen die Syntax der Systemkommandos beschrieben ist.

## **Systemkommandos sortiert nach Kategorien**

Liefert eine Übersicht über die nach Kategorien eingeteilten Systemkommandos.

Systemkommandos in alphabetischer Reihenfolge

Beschreibungen der Systemkommandos in alphabetischer Reihenfolge.

## **Notation *vrs* bzw. *vr***

Die in diesem Dokument verwendete Notation *vrs* bzw. *vr* steht als Platzhalter für die betreffende Produktionversion (siehe auch *Version* im *Glossary*).





# 1 Kommandoausführung

---

■ Kommandoeingabe .....	2
■ Kommandozeile .....	2
■ NEXT-Zeile .....	2
■ MORE-Zeile .....	3

## Kommandoeingabe

---

Sie können ein Systemkommando an folgenden Stellen absetzen:

- in der **Kommandozeile**;
- in einer **NEXT**- oder **MORE**-Zeile.

Folgende Regeln gelten:

- Bei Kommandoeingaben wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- Kommandoeingaben sind kontextabhängig.
- Manche Systemkommandos betreffen nicht das zur Zeit aktive Objekt, sondern ein anderes Objekt.

Eine Erklärung der in den Syntaxbeschreibungen der Systemkommandos enthaltenen Symbole finden Sie im Abschnitt **Kommando-Syntax**.

## Kommandozeile

---

Sie können ein Systemkommando in der Natural-Kommandozeile absetzen, die mit einem Pfeil (==>) als Eingabeaufforderungszeichen gekennzeichnet ist.

Manche Systemkommandos können auch über PF-Tasten oder über das Hauptmenü abgesetzt werden.

## NEXT-Zeile

---

Die NEXT-Zeile erscheint in einer Natural-Anwendung oder einem Natural-Programm, wenn keine weitere Ausgabe zu erwarten ist.

## MORE-Zeile

---

Die MORE-Zeile wird am unteren Rand eines Ausgabeschirms angezeigt, um darauf hinzuweisen, dass keine weitere Ausgabe mehr zu erwarten ist. Wenn Sie in der MORE-Zeile ein Systemkommando absetzen, wird die Ausführung des aktuellen Programms unterbrochen, und das eingegebene Systemkommando wird ausgeführt.



## 2 Systemkommando-Syntax

---

■ Syntax-Elemente .....	6
■ Kommandosyntax-Beispiel .....	7

## Syntax-Elemente

---

In den Diagrammen, in denen die Syntax der Systemkommandos beschrieben ist, werden folgende Symbole verwendet:

Element	Beschreibung
ABCDEF	Bei Angaben, die in Großbuchstaben dargestellt sind, handelt es sich um Natural-Schlüsselwörter, die Sie genauso eingeben müssen wie angegeben.
<u>ABCDEF</u>	Ist von mehreren wahlweise verwendbaren Elementen, die in Großbuchstaben dargestellt sind, eins unterstrichen (kein Hyperlink!), handelt es sich um das jeweils gültige Standardelement. Lassen Sie das Element weg, gilt der unterstrichene Wert.
<u>ABCDEF</u>	Ist ein Teil eines Wortes in Großbuchstaben unterstrichen (kein Hyperlink!), kann der unterstrichene Teil als Abkürzung für das jeweilige Wort verwendet werden.
<i>abcdef</i>	Bei Angaben, die in <i>kursiven</i> Kleinbuchstaben dargestellt sind, handelt es sich um variable Informationen, die Sie selbst angeben können.
[ ]	Syntax-Elemente, die in eckigen Klammern stehen, sind nicht unbedingt erforderlich, sie können gegebenenfalls weggelassen werden.  Wenn innerhalb der eckigen Klammern mehrere Zeile übereinander stehen, dann enthält jede Zeile eine optionale Alternative. Sie dürfen nur eine dieser Alternativen eingeben.
{ }	Von mehreren Elementen, die in einer geschweiften Klammer untereinander stehen, muß genau eines angegeben werden.
...	Drei Punkte hinter einem Element bedeuten, dass Sie das Element mehrmals angeben dürfen. Gegebenenfalls gibt eine Zahl hinter den drei Punkten an, wie oft das Element angegeben werden kann.  Ist das Element vor den drei Punkten in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
,...	Ein Komma und drei Punkte hinter einem Element bedeuten, dass Sie das Element mehrmals angeben dürfen, wobei die einzelnen Elemente durch Kommas voneinander getrennt werden müssen. Gegebenenfalls gibt eine Zahl hinter dem Komma und den drei Punkten an, wie oft das Element angegeben werden darf.  Ist das Element vor dem Komma und den drei Punkten in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
:...	Ein Doppelpunkt und drei Punkte hinter einem Element bedeuten, dass Sie das Element mehrmals angeben dürfen, wobei die einzelnen Elemente durch Doppelpunkte voneinander getrennt werden müssen. Gegebenenfalls gibt eine Zahl hinter dem Doppelpunkt und den drei Punkten an, wie oft das Element angegeben werden darf.

Element	Beschreibung
	Ist das Element vor dem Doppelpunkt und den drei Punkten in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
Andere Symbole (außer <code>[] {} ... ,... :...</code> )	<p>Alle anderen Symbole außer den in dieser Tabelle definierten müssen genau so eingegeben werden wie angegeben.</p> <p><b>Ausnahme:</b></p> <p>Der skalare SQL-Verkettungsoperator wird durch zwei senkrechte Striche dargestellt, die genauso eingegeben werden müssen, wie sie in der Syntax-Definition erscheinen.</p>

## Kommandosyntax-Beispiel

```
CATALOG [object-name [library-id]]
```

- **CATALOG** ist ein Natural-Schlüsselwort, das Sie genauso eingeben müssen, wie es dasteht. Die Unterstreichung bedeutet, dass Sie es auch in abgekürzter Form als **CAT** eingeben können.
- *object-name* und *library-id* sind variable Operanden, an deren Stelle Sie den gewünschten Programmnamen und die ID der Library, in der das Programm enthalten ist, eingeben.
- Die eckigen Klammern bedeuten, dass Sie *object-name* und *library-id* angeben können, aber nicht müssen. Die Anordnung der Klammern besagt, dass Sie **CATALOG** alleine eingeben können oder gefolgt von entweder nur einem Programmnamen oder einem Programmnamen und einer Library-ID; allerdings können Sie keine Library-ID eingeben, ohne gleichzeitig einen Programmnamen einzugeben.





# 3

## Systemkommandos sortiert nach Kategorien

---

■ In Natural navigieren .....	10
■ Natural-Entwicklungsumgebung .....	10
■ Anwendungen mit Natural-Programmobjekten pflegen .....	11
■ Monitor-, Debug- und Trace-Funktionen .....	13
■ Natural mit anderen Software AG-Produkten .....	14

Dieses Kapitel liefert eine Übersicht über die nach Kategorien eingeteilten Systemkommandos.

## In Natural navigieren

---

Kommando	Funktion
FIN	Beendet eine Natural-Session
LOGOFF	Setzt die Library-ID auf SYSTEM und das Adabas-Passwort auf Leerzeichen. Der Source-Inhalt des Editor-Arbeitsbereichs bleibt erhalten.
LOGON	Wählt eine bestimmte Library für den Benutzer. In dieser Library werden dann alle von Ihnen während der Session in Source- und/oder Objektform erstellten Objekte gespeichert (es sei denn, Sie geben bei einem SAVE-, CATALOG- oder STOW-Kommando ausdrücklich eine andere Library an).
MAINMENU	Schaltet die Anzeige des Natural-Hauptmenüs ein- bzw. aus oder ruft ein Programm auf, das einen eigenen Menüschirm anzeigt.
SETUP	Ermöglicht es, einen Rückkehrpunkt zu setzen, zu dem Sie dann später mit dem Systemkommando RETURN zurückkehren können. Das erlaubt es Ihnen, während einer Natural-Session problemlos von einer Anwendung in eine andere zu gelangen.
RETURN	Ermöglicht die Rückkehr zu einer bestimmten vorherigen Natural-Anwendung (oder der Ausgangsanwendung), die als Rückkehrpunkt mit dem SETUP Kommando gesetzt wurde.

## Natural-Entwicklungsumgebung

---

Kommando	Funktion
AIV	Zeigt an, welche anwendungsunabhängigen Variablen (Application-Independent Variables, AIVs) zur Zeit aktiv sind.
COMPOPT	Mit diesem Kommando können Sie verschiedene Kompilierungsoptionen setzen. Die Optionen werden wirksam, wenn ein Natural-Programmierobjekt kompiliert wird.
CPINFO	Dient zum Anzeigen der relevanten Natural Codepage-Einstellungen, z. B. Inhalt der Systemvariablen *LOCALE, *CODEPAGE, und zum Anzeigen der im Konfigurationsmodul NATCONFIG definierten Codepages.
GLOBALS	Mit diesem Kommando können Sie Natural-Session-Parameter setzen.
KEY	Mit diesem Kommando können Sie bestimmten Funktionstasten eine Funktion zuweisen.
LAST	Mit diesem Kommando können Sie sich die zuletzt ausgeführten Systemkommandos anzeigen lassen und sie erneut ausführen.
LIST XREF	Mit diesem Kommando können Sie sich alle aktiven Referenzdaten für die aktuelle Library anzeigen lassen.  Das Kommando ist nur verfügbar, wenn Predict installiert ist.

Kommando	Funktion
NATQVS	Mit diesem Kommando können Sie sich Informationen über die Eigenschaften der Maschine (z.B. die CPU-Kennung und logische Partitionen) anzeigen lassen, auf der zurzeit Natural ausgeführt wird.
SYSAPI	Betrifft nur Software AG-Produkte, die über Programmierschnittstellen (APIs) verfügen.  Mit diesem Kommando rufen Sie die SYSAPI-Utility auf. Diese dient zum Auffinden von APIs, wenn die in Ihrer Umgebung installierten Software AG-Produkte solche enthalten.
SYSCP	Mit diesem Kommando rufen Sie die SYSCP-Utility auf, die Code-Page-Informationen liefert und benutzt werden kann, um Codepages für Natural-Source-Objekte zu verwalten.
SYSEXT	Mit diesem Kommando rufen Sie die SYSEXT-Anwendung auf. Sie enthält verschiedene Natural-Programmierschnittstellen (API).
SYSE XV	Mit diesem Kommando rufen Sie die SYSE XV-Anwendung auf. Sie enthält Beispiele der neuen Merkmale der aktuellen Natural-Versionen.
SYSFILE	Mit diesem Kommando können Sie die Natural Print/Work Files (SYSFILE)-Funktion der SYSTP-Utility aufrufen. Diese Funktion liefert Informationen zu den verfügbaren Arbeitsdateien und Druckdateien (Work und Print Files).
SYSPARM	Mit diesem Kommando rufen Sie die SYSPARM-Utility auf. Sie können diese Utility benutzen, um Zeichenketten mit Natural-Profilparametern zu erstellen und zu pflegen, die als Natural-Profile gespeichert werden.
SYSPROD	Mit diesem Kommando können Sie feststellen, welche Produkte in Ihrer Natural-Umgebung installiert sind: d.h. Natural selbst, sowie Produkte, die mit bzw. unter Natural laufen.
SYSPROF	Mit diesem Kommando können Sie sich die gegenwärtigen Definitionen der Natural-Systemdateien anzeigen lassen.
UPDATE	Mit diesem Kommando können Sie verhindern (bzw. ermöglichen), dass ein auszuführendes Programm Datenänderungen auf der Datenbank durchführt.
XREF	Mit diesem Kommando können Sie die Verwendung der Predict-Funktion „Active Cross-References“ steuern. Mit Hilfe der aktiven Referenzen wird im Predict Data Dictionary automatisch dokumentiert, welche Objekte von einem Programm bzw. einer Data Area referenziert werden.  Ist nur verfügbar, wenn Predict installiert ist.

## Anwendungen mit Natural-Programmobjekten pflegen

Kommando	Funktion
CATALL	Mit diesem Kommando können Sie <i>alle</i> Objekte in der aktuellen Library katalogisieren (kompilieren).
CATALOG	Mit diesem Kommando können Sie den im Arbeitsbereich eines Editors befindlichen Sourcecode katalogisieren (kompilieren) und nach erfolgreicher Syntaxprüfung das resultierende katalogisierte Objekt in der Natural-Systemdatei speichern.

Kommando	Funktion
CHECK	Dieses Kommando dient dazu, den gerade im Arbeitsbereich des Editors befindlichen Sourcecode auf Syntaxfehler zu überprüfen.  Eine Syntaxprüfung erfolgt außerdem während der Ausführung der Kommandos <b>RUN</b> , <b>CATALL</b> , <b>CATALOG</b> und <b>STOW</b> .
CLEAR	Mit diesem Kommando löschen Sie den Inhalt des Editor-Arbeitsbereichs.
DELETE	Dieses Kommando dient dazu, Source-Objekte und/oder das entsprechende katalogisierte Objekt aus der Natural-Systemdatei zu löschen.
EDIT	Mit diesem Kommando rufen Sie einen der Natural-Editoren auf, um Sourcecode zu bearbeiten.
EXECUTE	Dieses Kommando dient dazu, ein katalogisiertes (kompiliertes) Programm auszuführen, das als katalogisiertes Objekt in der Natural-Systemdatei gespeichert worden ist.
INPL	Ruft die INPL-Utility auf. Diese Utility dient <i>nur</i> zum Laden von Software-AG-Installations-Datasets in die Systemdateien.
LIST	Mit diesem Kommando können Sie sich eines oder mehrere in Ihrer aktuellen Library gespeicherte Objekte auflisten lassen oder sich den Inhalt des Editor-Arbeitsbereichs anzeigen lassen.
READ	Mit diesem Kommando können Sie ein Source-Objekt in den Arbeitsbereich des entsprechenden Editors einlesen.
RENAME	Mit diesem Kommando können Sie ein Objekt umbenennen und den Objekttyp ändern.
RENUMBER	Mit diesem Kommando erreichen Sie, dass die Sourcecode-Zeilen des Objekts, das sich gerade im Editor-Arbeitsbereich befindet, neu durchnummeriert werden.
ROUTINES	Mit diesem Kommando können Sie feststellen, welche Objekte in der aktuellen Library welche externen Subroutinen verwenden.
RUN	Mit diesem Kommando können Sie ein Source-Programm, das sich im Editor-Arbeitsbereich befindet, kompilieren und ausführen.
SAVE	Mit diesem Kommando können Sie den Sourcecode, der sich im Editor-Arbeitsbereich befindet, als Source-Objekt in der Natural-Systemdatei speichern.
SCAN	Mit diesem Kommando können Sie den Sourcecode nach einer bestimmten Zeichenkette absuchen; es besteht die Möglichkeit, die gesuchte Zeichenkette durch eine andere Zeichenkette zu ersetzen.
STOW	Mit diesem Kommando können Sie Sourcecode zugleich als Source-Objekt und als katalogisiertes Objekt in der aktuellen Natural-Systemdatei katalogisieren (kompilieren) und speichern.
STRUCT	Dieses Kommando können Sie verwenden, um die Sourcecode-Zeilen eines Programms entsprechend der Programmstruktur einzurücken. Diverse zusätzliche Anzeigen verdeutlichen die Programmstruktur und erlauben es Ihnen so, etwaige Inkonsistenzen in der Struktur aufzuspüren und zu beseitigen.
SYSDDM	Mit diesem Kommando können Sie Natural Data Definition Modules (DDMs) erstellen und pflegen.
SYSERR	Mit diesem Kommando rufen Sie die SYSERR-Utility auf, mit der Sie Ihre eigenen anwendungsspezifischen Meldungen schreiben und pflegen können.
SYSMAIN	Mit diesem Kommando können Sie Natural-Objekte innerhalb des Natural-Systems von einer Library in eine andere Library übertragen.

Kommando	Funktion
<b>SYSNCP</b>	Mit diesem Kommando rufen Sie die SYSNCP-Utility auf, mit der Sie die in Ihren Anwendungen zu verwendenden Kommando-Prozessoren erstellen und pflegen können.
<b>SYSOBJH</b>	Mit diesem Kommando rufen Sie den Object Handler auf. Dort können Sie Natural- und Nicht-Natural-Objekte zwecks Verteilung in Natural-Umgebungen handhaben.
<b>SYSRPC</b>	Mit diesem Kommando rufen Sie die SYSRPC-Utility auf. Diese Utility bietet Funktionen zum Verwalten von Natural Remote Procedure Calls (Natural RPC).
<b>UNLOCK</b>	Mit diesem Kommando können Sie gesperrte Natural-Objekte anzeigen und entsperren.

## Monitor-, Debug- und Trace-Funktionen

Kommando	Funktion
<b>BUS</b>	<p>Mit diesem Systemkommando rufen Sie die Funktion Buffer Usage Statistics der SYSTP-Utility auf. Diese Funktion liefert Ihnen Informationen darüber, welche Puffer mit welcher Größe für die aktive Natural-Session angelegt wurden und wieviel Puffer-Platz tatsächlich benutzt wird.</p> <p><b>Anmerkung:</b> Das Kommando BUS hat die gleiche Funktion wie das Kommando SYSBUS, das nicht mehr zur Verfügung steht.</p>
<b>DUMP</b>	Liefert Informationen, die das technische Personal der Software AG benötigt, um einen Fehler zu finden, der zum Abbruch (Abend) des Natural-Systems geführt hat.
<b>HELP</b>	Ruft das Natural-Hilfesystem auf. Es liefert Informationen zu Natural-Statements, -Kommandos und Fehlermeldungen.
<b>LASTMSG</b>	Zeigt zusätzliche Informationen zu der zuletzt aufgetretenen Fehlersituation an.
<b>PROFILER</b>	Mit diesem Systemkommando können Sie die Profiler Utility aufrufen. Diese Utility dient dazu, mit dem Natural Data Collector aufgezeichnete Trace-Daten zu kontrollieren und zu verwalten. Siehe auch Systemkommando RDC.
<b>RDC</b>	Dieses Kommando wird in Verbindung mit der PROFILER-Utility benutzt, um die Trace Recording-Funktion des Natural Data Collector zu aktivieren bzw. zu deaktivieren. Siehe auch Systemkommando PROFILER.
<b>RPCERR</b>	Zeigt die Nummer und die Meldung des letzten Natural-Fehlers, falls dieser den Natural RPC (Remote Procedure Call) betrifft, sowie den letzten Reason Code des Brokers und die zugehörige Meldung an.
<b>SYSADA</b>	<p>Mit diesem Kommando rufen Sie die ADACALL-Utility auf, die sich in der Library SYSADA befindet. Mit der Utility können Sie von einem Großrechner-Natural aus Adabas-Direktkommandos (Native Commands) direkt an eine Adabas-Datenbank absetzen.</p> <p>Dieses Kommando steht nur zur Verfügung, wenn Adabas installiert ist.</p>
<b>SYSBPM</b>	Mit diesem Kommando rufen Sie die SYSBPM-Utility auf. Diese liefert statische Informationen über den aktuellen Status des Natural Buffer Pool, über den Buffer Pool Cache und über die Objekte, die sich gerade im Buffer Pool und im Buffer Pool Cache befinden.

Kommando	Funktion
<b>SYSEDT</b>	Mit diesem Kommando rufen Sie die SYSEDT-Utility for Editor Buffer Pool Services auf. Die SYSEDT-Utility ist nur für Natural-Administratoren gedacht.
<b>SYSTP</b>	Mit diesem Kommando rufen Sie die SYSTP-Utility auf, mit der Sie verschiedene TP-Monitor-spezifische Charakteristika von Natural überwachen und steuern können.
<b>TECH</b>	Mit diesem Kommando können Sie sich technische und andere Informationen über Ihre Natural-Session anzeigen lassen.
<b>TEST</b>	Mit diesem Kommando rufen Sie die Debugging-Funktion auf.
<b>TEST DBLOG</b>	Mit diesem Kommando rufen Sie die DBLOG-Utility auf, mit der Sie Datenbankaufrufe protokollieren können.

## Natural mit anderen Software AG-Produkten

Die folgenden Systemkommandos stehen nur in Verbindung mit anderen, in Ihrer Umgebung installierten Software AG-Produkten zur Verfügung.

Produkt	Systemkommando	Kurzbeschreibung
Adabas	<b>SYSADA</b>	Mit diesem Kommando rufen Sie die ADACALL-Utility auf, um von einem Großrechner-Natural aus Adabas-Direktkommandos (Native Commands) direkt an eine Adabas-Datenbank abzusetzen.
Natural for DB2	<b>LISTSQLB</b>	Ruft die Funktion <b>Explain all SQL Statements</b> in den <b>Natural Tools for DB2</b> auf.  Siehe <i>Using Natural Tools for DB2</i> in der <i>Database Management System Interfaces</i> -Dokumentation.
	<b>SQLDIAG</b>	Liefert Diagnoseinformationen über das zuletzt ausgeführte SQL-Statement (anders als das GET DIAGNOSTICS-Statement). Diese Informationen werden gesammelt, während das vorhergehende Statement ausgeführt wird. Einige der beim GET DIAGNOSTICS-Statement verfügbaren Informationen sind auch in der SQLCA vorhanden.
	<b>SYSDB2</b>	Ruft die <b>Natural Tools for DB2</b> mit Funktionen auf, die zur Pflege von DB2-basierten Anwendungen benötigt werden.  Siehe <i>Using Natural Tools for DB2</i> in der <i>Database Management System Interfaces</i> -Dokumentation.
Natural for DB2 Natural for SQL/DS	<b>LISTDBRM</b>	Dient (bei Natural for DB2) zur Anzeige von vorhandenen Datenbankverwaltungssystemen (DBRMs) oder (bei Natural for SQL) zur Anzeige von vorhandenen Paketen mit Natural-Programmen oder zur Anzeige von Natural-Programmen, welche ein gegebenes DBRM referenzieren.
	<b>LISTSQL</b>	Listet diejenigen Natural-Statements im Sourcecode eines Objekts auf, die in Verbindung mit einem Datenbankzugriff stehen, sowie die entsprechenden SQL-Kommandos, in die diese Natural-Statements

Produkt	Systemkommando	Kurzbeschreibung
		übersetzt wurden. Damit ist es möglich, den generierten SQLCODE zu sichten, bevor das Natural-Programm, welches auf eine SQL-Tabelle zugreift, ausgeführt wird.
	SQLERR	Dient zur Anzeige von Informationen über den zuletzt aufgetretenen SQL-Fehler.
Natural Optimizer Compiler	NOCOPT	Mit diesem Kommando können Sie die beim Start von Natural gesetzten Optionen für den Natural Optimizer Compiler anzeigen und ändern.
	NOCSHOW	Liefert Buffer-Informationen zur Ausgabe der PGEN-Option des Natural Optimizer Compiler. Diese Option erstellt generierten Code und interne Natural-Strukturen.
	NOCSTAT	Liefert Statistikdaten zu Programmen, die für die Verarbeitung durch den Natural Optimizer Compiler geeignet sind.
Natural Security	MAIL	Ruft eine Mailbox auf, um ihren Inhalt sowie das Ablaufdatum zu ändern. Eine Mailbox ist eine Art „Schwarzes Brett“ zur Übermittlung von Informationen an Natural-Benutzer.
	PROFILE	Zeigt Ihr gegenwärtig gültiges Benutzungsprofil an. Dieses Profil informiert Sie über die Bedingungen und Voraussetzungen, die in Ihrer aktuellen Natural-Umgebung für Sie gelten.
Natural SQL Gateway	LISTSQL	Listet diejenigen Natural-Statements im Sourcecode eines Objekts auf, die in Verbindung mit einem Datenbankzugriff stehen, sowie die entsprechenden SQL-Kommandos, in die diese Natural-Statements übersetzt wurden. Damit ist es möglich, den generierten SQLCODE zu sichten, bevor das Natural-Programm, welches auf eine SQL-Tabelle zugreift, ausgeführt wird.
	SQLERR	Dient zur Anzeige von Informationen über den zuletzt aufgetretenen SQL-Fehler.
Predict	LIST XREF	Zeigt alle aktiven Referenzdaten für die aktuelle Library an.
	XREF	Steuert die Verwendung der Predict-Funktion „Active Cross-References“. Mit Hilfe der aktiven Referenzen wird im Predict Data Dictionary automatisch dokumentiert, welche Objekte von einem Programm bzw. einer Data Area referenziert werden.
Natural zIIP Enabler	ZIIP	Mit diesem Kommando rufen Sie Statistiken zu IBM System z Integrated Information Processors (zIIPs), die in einer Natural z/OS-Umgebung verwendet werden.
Verschiedene Produkte	SYSAPI	Ruft die SYSAPI-Utility auf. Diese dient zum Auffinden von APIs, wenn die in Ihrer Umgebung installierten Software AG-Produkte solche enthalten.





# 4 AIV

---

## AIV

Mit diesem Systemkommando können Sie sich anzeigen lassen, welche anwendungsunabhängigen Variablen (Application-Independent Variables, AIVs) zur Zeit aktiv sind.

Wenn Sie das Kommando ausführen, wird eine Liste mit Namen, Format und Länge aller AIVs angezeigt. Auf der Liste können Sie eine AIV mit dem Kommando `DI` (Display) markieren, um sich den Inhalt der AIV anzeigen zu lassen. Sie können sich den Inhalt in alphanumerischer und in hexadezimaler Darstellung anzeigen lassen. Mit den Tasten `PF10` und `PF11` können Sie zwischen alphanumerischer und hexadezimaler Anzeige hin- und herschalten.

Weitere Informationen siehe:

- `DEFINE DATA`-Statement (*Definition von anwendungsunabhängigen Variablen*) in der *Statements-Dokumentation*
- *Benutzervariablen im Leitfaden zur Programmierung*



# 5 BUS

---

BUS

Mit diesem Systemkommando rufen Sie die Funktion **Buffer Usage Statistics** der SYSTP-Utility auf. Diese Funktion liefert Ihnen Informationen darüber, welche Puffer mit welcher Größe für die aktive Natural-Session angelegt wurden und wieviel Puffer-Platz tatsächlich benutzt wird. Näheres zu dieser Funktion finden Sie in der Utilities-Dokumentation.



**Anmerkung:** Das Kommando BUS hat die gleiche Funktion wie das Kommando SYSBUS, das nicht mehr zur Verfügung steht.

Näheres zu der Funktion Buffer Usage Statistics finden Sie in der *Utilities*-Dokumentation.

Die vom Systemkommando BUS gelieferten Informationen erhalten Sie auch mit Hilfe der Programmierschnittstelle (API) USR1019N. Siehe *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.



## 6 CATALL

---

■ Objekte katalogisieren von/bis .....	22
■ Predict Set und Set Benutzer .....	23
■ Vorhandene Module neu katalogisieren oder Alle Sourcen katalogisieren .....	23
■ Objekttypen auswählen .....	23
■ Funktion auswählen .....	24
■ Optionen auswählen .....	24
■ Auswahlliste .....	25
■ Direktkommando-Syntax .....	26

CATALL	{	<i>object-name</i> [TO <i>object-name</i> ] {XSET / XREFSET} <i>set-number</i> [SETUSER <i>set-user</i> ]	}	[	RECAT ALL	]	[TYPES <i>types</i> ]	[	SAVE CATALOG STOW CHECK	]	[ <i>options</i> ...]
		<i>text-name</i>									

Mit dem Systemkommando CATALL können Sie einzelne oder mehrere Natural-Objekte in der aktuellen Natural-Library katalogisieren (kompilieren) und speichern.

Wenn Sie das Kommando CATALL ohne zusätzliche Optionen eingeben, erscheint der Schirm **Catalog Objects in Library**, auf dem Sie die nachfolgend beschriebenen Funktionen auswählen können. Sie können das Kommando CATALL aber auch als Direktkommando unter Verwendung der oben dargestellten **Syntax** benutzen.

Alle Objekte werden in der Reihenfolge verarbeitet, in der sie im Schirm **Catalog Objects in Library** aufgelistet sind (siehe auch Informationen zu **TYPES** im Abschnitt *Direktkommando-Syntax*).

Außerdem können Sie mit Hilfe des Subprogramms CATALLU2 standardmäßig Funktionen auf dem **Catalog Objects in Library**-Schirm auswählen. Darüber hinaus können Sie auswählen, ob CATALLU2 im Batch Mode oder im Command Mode aufgerufen wird. Dieses Subprogramm wird in Source-Form in der Library SYSTEM (FNAT) mitgeliefert. Um das Subprogramm zu aktivieren, müssen Sie es entsprechend den Angaben in der Source modifizieren, es anschließend katalogisieren und in die Library SYSLIB kopieren. Das Subprogramm wird aufgerufen, bevor der Schirm **Catalog Objects in Library** ausgegeben wird.

Siehe auch *Namenskonventionen für Objekte* im Dokument *Natural benutzen*.

Dieses Dokument behandelt folgende Themen:

## Objekte katalogisieren von/bis

---

Wenn Sie CATALL für *alle* Objekte der ausgewählten Typen in der aktuellen Library ausführen möchten, geben Sie einen Stern (\*) als Objektname im **from**-Feld ein.

Wenn Sie CATALL nur für einen bestimmten Bereich von Objekten ausführen möchten, können Sie im **from**-Feld Stern-Notation (\*) und Wildcard-Notation (?) für den Namen verwenden, und zwar in der gleichen Weise wie beim Systemkommando LIST beschrieben.

Sie können auch einen Anfangs- und Endwert für einen bestimmten Bereich von Objekten angeben, indem Sie in den beiden Feldern **from** und **to** entsprechende Objektnamen (ohne Stern- oder Wildcard-Notation) eingeben.

Statt in diese Felder Werte einzugeben, können Sie auch Objekte von einer Auswahlliste auswählen (siehe unten).

Sie haben außerdem die Möglichkeit, im **from**-Feld den Namen eines Objekts vom Typ Text anzugeben, das eine Liste von CATALL-Kommandos enthält. Die in dem Text enthaltenen CATALL-Kommandos werden dann ausgeführt. Sie können einen solchen Text entweder von Hand erstellen oder unter Benutzung der [Auswahlliste](#) (siehe unten) automatisch erstellen.

## Predict Set und Set Benutzer

Wenn Sie CATALL für die Objekte eines Predict Set ausführen möchten, geben Sie im Feld **Predict Set** eine ein- bzw. zweistellige Zahl an, die den beibehaltenen Set („Retained Set“) bezeichnet, welcher für die Namen der zu verarbeitenden Objekte verwendet werden soll.

Wenn Sie einen Predict Set verwenden, dürfen Sie die Felder **from** und **to** nicht benutzen. Zusätzlich können Sie im Feld **Set User** die Benutzerkennung des Benutzers angeben, der den Predict Set erstellt hat. Wenn Sie keine Benutzerkennung angeben, wird der Wert der Systemvariablen \*USER verwendet.

Ein beibehaltener Set wird mit der `Save Set`-Option des Systemkommandos `LIST XREF` erstellt.



**Anmerkung:** Predict muss installiert sein. Weitere Informationen zu Predict Sets siehe *Predict*-Dokumentation.

## Vorhandene Module neu katalogisieren oder Alle Sourcen katalogisieren

Diese Option bewirkt Folgendes:

- Wenn Sie das Feld **Select cataloged source objects only** markieren, werden nur die Source-Objekte in der aktuellen Library katalogisiert, die bereits früher katalogisiert worden waren (ein entsprechendes katalogisiertes Objekt existiert).
- Wenn Sie das Feld **Select all source objects** markieren, werden *alle* Source-Objekte (mit oder ohne entsprechende katalogisierte Objekte) in der aktuellen Library katalogisiert.

## Objekttypen auswählen

Standardmäßig gilt CATALL für Objekte jeglichen Objekttyps in der aktuellen Library (alle Objekttypen sind mit X markiert).

Wenn Sie bestimmte Objekttypen von der Verarbeitung durch CATALL ausnehmen möchten, überschreiben Sie das betreffende X mit einem Leerzeichen.

## Funktion auswählen

Sie können eine der folgenden Funktionen auswählen, die für die ausgewählten Objekte ausgeführt werden soll: **SAVE**, **CATALOG**, **STOW** oder **CHECK**. Die Funktionen entsprechen den gleichnamigen Systemkommandos.



**Anmerkung:** Objekte des Typs Copycode und des Typs Text werden immer mit **SAVE** gespeichert, auch wenn Sie die Funktion **STOW** auswählen. Sie werden jedoch nicht gespeichert, wenn Sie **CATALOG** auswählen.

## Optionen auswählen

Sie können eine oder mehrere der folgenden Optionen für die CATALL-Verarbeitung auswählen:

Option	Erklärung
Condition Code im Batch	Wenn Sie CATALL im Batch-Betrieb ausführen und diese Option mit einem Zeichen markieren, wird Condition Code 55 zurückgegeben, falls entweder bei der CATALL-Ausführung ein Syntaxfehler entdeckt wird oder innerhalb des angegebenen Bereiches zu verarbeitender Objekte keine Objekte gefunden wurden (gilt nur bei <b>CATALOG</b> und <b>STOW</b> ).
Sourcecode-Zeilen neu numerieren	Standardmäßig werden mit den Funktionen <b>SAVE</b> und <b>STOW</b> die Sourcecode-Zeilen von Source-Objekten, die mit <b>SAVE</b> oder <b>STOW</b> bearbeitet wurden, auch neu numeriert.  Wenn Sie keine automatische Neunummerierung der Zeilen wünschen, überschreiben Sie das X in diesem Feld mit einem Leerzeichen.
Ergebnisliste speichern	CATALL erzeugt eine Ergebnisliste. Wenn Sie diese Liste für den weiteren Gebrauch aufheben möchten, markieren Sie dieses Feld mit einem Zeichen.  Die Library SYSEXT enthält eine Anwendungsprogrammierschnittstelle (API) USR1024N, mit der Sie die Ergebnisliste ausgeben können.  Sie können die Ergebnisliste auch mit einem weiteren CATALL-Kommando wieder zur Anzeige bringen. Weil die Parameter ebenfalls in der Ergebnisliste gespeichert werden, sind die Parameter des CATALL-Kommandos, mit dem die Ergebnisliste erstellt wurde, gültig. In diesem Fall bewirkt das Batch-CATALL-Kommando die Ausgabe einer Meldung, und weil keine Objekte katalogisiert werden, wird der Job mit dem Condition Code 56 beendet. Online werden Sie, falls eine Library eine Ergebnisliste enthält, gefragt, ob die vorherige Ergebnisliste angezeigt oder ein neuer CATALL-Lauf gestartet werden soll.
Verarbeitungsinformationen	Während der Online-Verarbeitung zeigt CATALL eine fortlaufende Anzeige von Verarbeitungsstatus-Informationen.



Option	Erklärung
	<p>Während der Batch-Verarbeitung gibt CATALL nur diejenigen Objekte aus, die einen Fehler verursacht haben.</p> <p>Wenn Sie diese Anzeige nicht wünschen, überschreiben Sie das X in diesem Feld mit einem Leerzeichen.</p>
Fehlerreport	<p>Am Ende der Verarbeitung zeigt CATALL eine Liste der aufgetretenen Fehler an.</p> <p>Wenn Sie die Anzeige dieser Fehlerliste nicht wünschen, überschreiben Sie das X in diesem Feld mit einem Leerzeichen.</p>
Erweiterter Fehlerreport	<p>Der Fehlerbericht wird in erweiterter Form ausgegeben, einschließlich Verzeichnisinformationen, Fehlerzeile und Fehlermeldung.</p> <p>Wenn Sie den erweiterten Fehlerbericht wünschen, markieren Sie dieses Feld mit einem X.</p>
PF4 AddOp	<p>Nach Drücken von PF4 erscheint ein Fenster, in dem Sie zusätzliche Optionen auswählen oder eingeben können.</p> <p><b>Report-Text-Objekt:</b> Geben Sie hier den Namen eines Natural-Text-Objekts ein. Es wird ein Bericht über die bei einem CATALL-Lauf aufgetretenen Fehler in dieses Text-Objekt geschrieben.</p>

## Auswahlliste

Wenn Sie CATALL nur für bestimmte Objekte ausführen möchten, können Sie die gewünschten Objekte von einer Auswahlliste auswählen.

Hierzu machen Sie zunächst die gewünschten Angaben unter **Select Function** und **Select Options** und drücken dann PF5. Es wird eine Liste der in der aktuellen Library gespeicherten Objekte angezeigt.

Die Liste entspricht der des Systemkommandos **LIST**. Auch das Blättern in der Auswahlliste und die Angabe neuer Kriterien für die Auswahlliste erfolgt in der gleichen Weise wie beim **LIST**-Kommando.

Aus der Liste wählen Sie die gewünschten Objekte aus, indem Sie sie in der Spalte **Cmd** mit einem Zeichen markieren. Um *alle* Objekte der aktuellen Auswahlliste gleichzeitig auszuwählen, drücken Sie PF5. Anschließend können Sie in der Liste blättern, andere Auswahlkriterien angeben und/oder weitere Objekte auswählen.

Nachdem Sie alle gewünschten Objekte ausgewählt haben, drücken Sie PF3.

Es erscheint ein Fenster, das Ihnen die Möglichkeit bietet, die getroffene Objektauswahl zu speichern, um sie für spätere CATALL-Verarbeitungen wiederzuverwenden:

- Falls Sie in das Fenster einen Namen eingeben, wird die getroffene Auswahl (in Form von CATALL-Kommandos) automatisch in einem Objekt vom Typ Text dieses Namens gespeichert. Dieser Text-Name kann später im Feld **Objekte katalogisieren von** im Bildschirm **Objekte in Library katalogisieren** eingegeben werden.
- Falls Sie dies nicht wünschen, geben Sie in das Fenster nichts ein, und drücken Sie EINGABE.

Anschließend beginnt CATALL, die ausgewählten Objekte zu verarbeiten.

## Direktkommando-Syntax

---

Für die verschiedenen Angaben, die Sie auf dem **Objekte in Library katalogisieren**-Schirm machen können, gibt es auch entsprechende Optionen, die Sie direkt mit dem Systemkommando CATALL angeben können:

<i>object-name</i> TO <i>object-name</i>	Entspricht den Feldern <b>Objekte katalogisieren von</b> und <b>bis</b> des <b>Objekte in Library katalogisieren</b> -Schirms, siehe <i>Objekte katalogisieren von/bis</i> .
{XSET / XREFSET} <i>set-number</i>	Entspricht dem Feld <b>Predict Set</b> im Schirm <b>Objekte in Library katalogisieren</b> , siehe <i>Predict Set und Set Benutzer</i> .
[SETUSER <i>set-user</i> ]	Entspricht dem Feld <b>Set Benutzer</b> im Schirm <b>Objekte in Library katalogisieren</b> , siehe <i>Predict Set und Set Benutzer</i> .
RECAT / ALL	Entspricht den Optionen <b>Vorhandene Module neu katalogisieren</b> oder <b>Alle Sourcen katalogisieren</b> des <b>Objekte in Library katalogisieren</b> -Schirms. RECAT ist der Standardwert, siehe <i>Vorhandene Module neu katalogisieren oder Alle Sourcen katalogisieren</i> .

TYPES <i>types</i>	<p>Entspricht den markierten Objekttypen des <b>Catalog Objects in Library</b>-Schirms. Mögliche (in der unten aufgeführten Reihenfolge verarbeitete) <i>types</i> sind:</p> <p>G - Global Data Areas  L - Local Data Areas  A - Parameter Data Areas  C - Copycodes  T - Texte  7 - Funktionen (Functions)  N - Subprogramme  S - Externe Subroutinen  H - Helprouinen  M - Maps  8 - Adapter  P - Programme  4 - Klassen  * - Alle Typen (gilt standardmäßig)</p> <p>Die <i>types</i> müssen als <i>eine</i> Zeichenkette angegeben werden (z.B. LAG für Local, Parameter und Global Data Areas). Standardmäßig gilt CATALL für alle Objektarten in der aktuellen Library.</p>
SAVE / CATALOG / STOW / CHECK	<p>Entspricht den gleichnamigen Funktionen des <b>Objekte in Library katalogisieren</b>-Schirms, siehe <a href="#">Funktion auswählen</a>. CATALOG ist der Standardwert.</p>

<i>options</i>	Diese Optionen entsprechen den <b>Optionen auswählen</b> auf dem <b>Objekte in Library katalogisieren</b> -Schirm, siehe <a href="#">Optionen auswählen</a> . Mögliche <i>options</i> sind:	
	CC	Condition Code wird zurückgegeben.
	NOREN	Keine automatische Neunummerierung von Sourcecode-Zeilen.
	KEEP	Ergebnisliste wird aufbewahrt.
	NOScroll	Online: Keine laufende Anzeige von Verarbeitungsstatus-Informationen. Batch: Ausgabe derjenigen Objekte, die einen Fehler verursacht haben.
	NOREPORT	Keine Fehlerliste.
	FULL	Erweiterte Fehlerliste.
	EL <text-object> [R]	
	EL text-object	Ausgabe der Fehlerliste in ein Natural-Text-Objekt.
	R	Wenn ein Objekt schon existiert, ist der EL-Parameter funktionslos, wenn kein R (Replace/Ersetzen) nach text-object angegeben wird.
	<b>Anmerkung:</b> Bei Angabe von NOREPORT und NOScroll, gilt automatisch auch KEEP.	
<i>text-name</i>	Entspricht der Angabe eines Text-Namens im <b>Objekte katalogisieren von</b> -Feld des <b>Objekte in Library katalogisieren</b> -Schirms, siehe <a href="#">Objekte katalogisieren von/bis</a> .	

Beispiele:

➤ Um nur die Objekte in Source- und Objektform zu speichern, für die schon ein katalogisiertes Objekt vorhanden ist:

- Geben Sie folgendes Kommando ein:

```
CATALL * STOW KEEP CC NOREN ↵
```

Dieses Kommando enthält ein implizites RECAT und bewirkt dasselbe wie folgendes Kommando:

```
CATALL * RECAT STOW KEEP CC NOREN ↵
```

➤ Um alle Objekte in Source- und Objektform zu speichern

- Geben Sie folgendes Kommando ein:

```
CATALL * ALL STOW KEEP CC NOREN
```



**Anmerkung:** Die einzelnen Bestandteile des Kommandos müssen durch ein Leerzeichen oder durch das Eingabebegrenzungszeichen (wie mit dem Session-Parameter `ID` festgelegt) voneinander getrennt werden.



# 7 CATALOG

---

CATALOG [*object-name* [*library-id*]]

Verwandte Kommandos: [SAVE](#) | [STOW](#).

Mit dem Systemkommando CATALOG katalogisieren (kompilieren) Sie das im Arbeitsbereich eines Editors befindliche Source-Objekt und speichern (nach erfolgreich verlaufener Syntax-Prüfung) das resultierende Objektmodul in der aktuellen Natural-Systemdatei.

Siehe auch:

- *Natural-Compiler in Natural-System-Architektur*
- *Namenskonventionen für Objekte in Natural benutzen*



**Wichtig:** Es kann sein, dass Sie das CATALOG-Kommando nicht verwenden können, weil der Profilparameter RECAT auf ON gesetzt ist. In diesem Fall benutzen Sie stattdessen das Systemkommando [STOW](#), um das Objekt zu kompilieren und zu speichern.

CATALOG	Falls Sie keinen <i>object-name</i> angeben, wird das Programm in der aktuellen Library unter dem Namen des Source-Objekts gespeichert, das zuletzt in den Arbeitsbereich gelesen wurde (z.B. mit einem EDIT-, READ- oder RUN-Kommando). Eventuell vorhandener Objektcode wird ersetzt.
CATALOG <i>object-name</i>	Ein neues Objekt wird angelegt. Als <i>object-name</i> geben Sie den Namen an, unter dem das neue Objekt katalogisiert werden soll. Es wird in der aktuellen Library gespeichert. Falls das Objekt schon vorhanden ist, wird das Kommando zurückgewiesen.
CATALOG <i>object-name</i> <i>library-id</i>	Wollen Sie das neue Objekt in einer anderen Library katalogisieren, müssen Sie zusätzlich zum Objektnamen die <i>library-id</i> der gewünschten Library angeben.



**Anmerkung:** Falls im Parametermodul eine ungültige Systemdatei *FDIC* angegeben ist, erscheint eine entsprechende Natural-Fehlermeldung, wenn das CATALOG-Kommando abgesetzt wird.





## 8 CHECK

---

### CHECK

Dieses Kommando dient dazu, den gerade im Arbeitsbereich des Programm-Editors befindlichen Sourcecode auf Syntaxfehler zu überprüfen.

Siehe auch:

- *Natural-Compiler in Natural System-Architektur*
- *Namenskonventionen für Objekte in Natural benutzen*

Modus	Reaktion
Online	Die Syntaxprüfung wird abgebrochen, sobald ein Syntaxfehler entdeckt wird, und die fehlerhafte Sourcecode-Zeile wird im Programm-Editor mit E markiert.
Batch	Das Programm wird vollständig auf Syntaxfehler untersucht. Alle Fehler werden in der Ausgabeliste mit ausgegeben.



**Anmerkung:** Die Syntaxprüfung erfolgt auch im Rahmen der Ausführung der Kommandos [RUN](#), [STOW](#), [CATALOG](#) und [CATALL](#).



## 9 CLEAR

---

CLEAR

Mit diesem Kommando löschen Sie den Inhalt des Editor-Arbeitsbereichs. Falls Sie ein neues Objekt erstellen wollen, und es befindet sich noch ein anderes Source-Objekt im Arbeitsbereich, empfiehlt es sich, den Arbeitsbereich mit diesem Kommando zu löschen.



# 10 COMPOPT

---

■ Syntax-Erklärung .....	38
■ Compiler-Schlüsselwortparameter angeben .....	38
■ Allgemeine Compiler-Optionen .....	39
■ Kompilierungsoptionen für Versions- und Plattformkompatibilität .....	53

COMPOPT [*option=value* ...]

Mit diesem Kommando können Sie verschiedene Kompilierungsoptionen setzen. Die Optionen werden wirksam, wenn ein Natural-Programmierobjekt kompiliert wird.

Wenn Sie das Kommando COMPOPT ohne Optionen eingeben, wird ein Schirm angezeigt, auf dem Sie die unten beschriebenen Optionen ein- bzw. ausschalten können.

Die Standardwerte für die einzelnen Optionen werden mit den Schlüsselwort-Subparametern im Parametermakro NTCMPO im Natural-Parametermodul bzw. im entsprechenden Profilparameter CMPO gesetzt. Wenn Sie die Library wechseln, werden die COMPOPT-Optionen wieder auf ihre Standardwerte zurückgesetzt.

## Syntax-Erklärung

---

COMPOPT	Wenn Sie nur das Kommando COMPOPT eingeben, wird der <b>Compilations Options</b> -Schirm angezeigt, auf dem Sie die unten beschriebenen Optionen ein- bzw. ausschalten können. Die dort vorhandenen Schlüsselwörter werden nachfolgend beschrieben.
COMPOPT <i>option=value</i>	Die Schlüsselwörter für die einzelnen Optionen werden nachfolgend beschrieben.  Die einer Compiler-Option zugewiesene Einstellung bleibt solange gültig, bis Sie das nächste LOGON-Kommando absetzen, um in eine andere Library zu wechseln. Beim LOGON werden wieder die mit dem Parametermakro NTCMPO und/oder dem Profilparameter CMPO eingestellten Standardeinstellungen gültig.

## Compiler-Schlüsselwortparameter angeben

---

Compiler-Schlüsselwortparameter können Sie auf verschiedenen Ebenen angeben:

### 1. Als Standardeinstellungen

Die Standardeinstellungen der einzelnen Compiler-Schlüsselwortparameter werden mit dem Parameter-Makro NTCMPO vorgenommen. Diese Einstellungen werden im Natural-Parametermodul gespeichert.

### 2. Zu Beginn einer Natural-Session

Zu Beginn einer Natural-Session können Sie die Compiler-Schlüsselwortparameter überschreiben, indem Sie die Werte im entsprechenden Profilparameter CMPO setzen.

### 3. Während einer aktiven Natural-Session

Während einer aktiven Natural-Session haben Sie zwei Möglichkeiten, die Compiler-Parameterwerte mit dem Systemkommando `COMPOPT` zu ändern: Entweder direkt mittels Kommando-zuordnung (`COMPOPT option=value`) oder indem Sie das Kommando `COMPOPT` ohne Optionen absetzen, woraufhin der Schirm **Compilation Options** erscheint.

Bei einem `LOGON` auf eine andere Library, werden die mit dem Makro `NTCMPO` und/oder dem Profilparameter `CMPO` vorgenommenen Standardeinstellungen (siehe Punkt 1) wieder hergestellt. Beispiel:

```
OPTIONS KCHECK=ON
DEFINE DATA LOCAL
1 #A (A25) INIT <'Hello World'>
END-DEFINE
WRITE #A
END ↵
```

### 4. In einem Natural-Programmierobjekt

In einem Natural-Programmierobjekt (z.B. Programm, Subprogramm) können Sie Compiler-Parameter mit dem `OPTIONS`-Statement setzen. Beispiel:

```
OPTIONS KCHECK=ON
WRITE 'Hello World'
END ↵
```

Die in einem `OPTIONS`-Statement angegebenen Compiler-Optionen betreffen nur die Kompilierung dieses Programmierobjekts, sie aktualisieren jedoch nicht die mit dem Systemkommando `COMPOPT` gesetzten Optionen.

## Allgemeine Compiler-Optionen

- `CHKRULE` - Validierung von `INCDIR`-Statements in Masken
- `CPAGE` - Codepage-Unterstützung bei alphanumerischen Konstanten
- `DBSHORT` - Interpretation von Datenbankfeld-Kurznamen
- `DB2ARRAY` - Unterstützung von DB2-Arrays in SQL `SELECT`- und `INSERT`-Statements
- `DB2BIN` - Generierung von SQL Binärdatentypen bei Natural-Binärfeldern
- `DB2PKYU` – Primäre Schlüsselfelder in das Natural DML Statement `UPDATE` stellen
- `DB2TSTI` – Generierung des SQL `TIMESTAMP`-Datentyps für Natural-TIME-Felder
- `ECHECK` - Existenzprüfung bei objektaufrufenden Statements
- `GDASC` - GDA-Signaturprüfung
- `GFID` - Generierung globaler Formatkennungen
- `KCHECK` - Schlüsselwort-Prüfung
- `LOWSRCE` - Source in Kleinschreibung

- MAXPREC – Maximale Anzahl Nachkommastellen
- MEMOPT - Speicheroptimierung für lokal deklarierte Variablen
- PCHECK - Parameter-Prüfung bei CALLNAT-Statements
- PSIGNF - Interne Darstellung des positiven Vorzeichens bei gepackten Zahlen
- THSEP - Dynamisches Tausender-Trennzeichen
- TQMARK - Umsetzung von Anführungszeichen
- TSENABL - Gültigkeit des Profilparameters TS

Diese Optionen entsprechen den gleichnamigen Schlüsselwortparametern des Profilparameters CMPO bzw. des Parameter-Makros NTCMPO.

### CHKRULE - Validierung von INCDIR-Statements in Masken

Mit der Option `CHKRULE` können Sie eine während des Katalogisierungsvorgangs für Masken (Maps) stattfindende Validierungsprüfung ein- bzw. ausschalten.

ON	<p>Die <code>INCDIR</code>-Validierung ist eingeschaltet. Wenn die bzw. das im <code>INCDIR</code>-Kontrollstatement referenzierte Datei (DDM) bzw. Feld nicht existiert, wird zur Kompilierungszeit ein Syntaxfehler (NAT0721) ausgegeben.</p> <p>Beim Anlegen einer Map können Sie Felder einfügen, die schon in einem anderen existierenden Programmierobjekt definiert sind. Das funktioniert bei fast allen Arten von Objekten, in denen Sie Variablen definieren können, und ebenso bei DDMs. Wenn es sich bei dem eingefügten Feld um eine Datenbankvariable handelt, fügt der Map Editor automatisch (neben dem eingefügten Feld) ein <code>INCDIR</code>-Statement in den Map-Statement-Rumpf ein, um damit das Laden und die Übernahme einer Predict-Regel auszulösen, wenn die Map (per <code>STOW</code>-Kommando) kompiliert wird.</p> <p>Die Funktionsweise ist ähnlich dem Vorgang bei der Verarbeitung eines <code>INCLUDE</code>-Statements. Anstatt jedoch Sourcecode-Zeilen aus einem Copycode-Objekt zu übernehmen, werden sie in diesem Fall aus Predict übergeben. Als Suchschlüssel zum Auffinden der Regel(n) dienen der DDM-Name (der als der Feld-Name betrachtet wird) und der Feld-Name. Beide werden im <code>INCDIR</code>-Statement angegeben. Eine während der Kompilierung angeforderte <code>INCDIR</code>-Regel muss aber nicht zwangsläufig in Predict gefunden werden, weil für ihr Vorhandensein durchaus keine Notwendigkeit besteht. Das bedeutet, dass keine Fehlersituation vorliegt, wenn eine gesuchte Regel nicht gefunden wird.</p> <p>Bei Übernahme von Feldern aus einer DDM in eine Map, werden entsprechende <code>INCDIR</code>-Statements erzeugt, die den aktuelle DDM- und Feldnamen als „Suchschlüssel“ enthalten, mit dem eventuell existierende Regeln aus Predict abgerufen werden. Falls aber die DDM nach dem Kopiervorgang umbenannt wird, bleibt der alte (nicht mehr gültige) DDM-Name im <code>INCDIR</code>-Statement erhalten, was zur Folge hat, dass keine Regel geladen und der Programmierer darüber nicht in Kenntnis gesetzt wird. Eine solche Situation tritt aber nicht nur im Falle einer DDM-Umbenennung auf. Wahrscheinlicher ist, dass sie durch versehentliche Zuordnung einer falschen FDIC-Datei verursacht wird. In diesem Fall ist der DDM-Name zwar gültig, kann aber in der aktuellen Predict-Systemdatei nicht gefunden werden. Das Ergebnis ist das gleiche wie wenn die DDM überhaupt nicht existiert, das heißt, die erwartungsgemäß aus Predict hinzuzufügenden Verarbeitungsregeln werden nicht eingefügt.</p>
OFF	Die <code>INCDIR</code> -Validierung ist ausgeschaltet. Dies ist der Standardwert.



## CPAGE - Codepage-Unterstützung bei alphanumerischen Konstanten

Mit der CPAGE-Option können Sie eine Konvertierungsroutine aktivieren, die, wenn das Objekt zur Laufzeit gestartet wird, alle alphanumerischen Konstanten von der Codepage, die bei der Kompilierung aktiv war, in die Codepage, die zur Laufzeit aktiv ist, umsetzt.

Siehe auch Abschnitt *CPAGE Compiler Option* in der Dokumentation *Unicode and Code Page Support*.

ON	Codepage-Unterstützung für alphanumerische Zeichenketten ist eingeschaltet.
OFF	Codepage-Unterstützung für alphanumerische Zeichenketten ist ausgeschaltet. Dies ist der Standardwert.

## DBSHORT - Interpretation von Datenbankfeld-Kurznamen

Interpretation von Datenbankfeld-Namen in Programmierobjekten.

Ein in einem DDM definiertes Datenbankfeld wird durch zwei Namen beschrieben:

- durch den Kurznamen mit einer Länge von zwei Zeichen, der von Natural für die Kommunikation mit der Datenbank (insbesondere mit Adabas) verwendet wird;
- durch den Langnamen mit einer Länge von 3 bis 32 Zeichen (1 bis 32 Zeichen, wenn die darunter liegende Datenbank DB2/SQL ist), der zum Referenzieren des Feldes im Natural-Programmcode verwendet werden soll.

Unter speziellen Bedingungen können Sie in einem Natural-Programm ein Datenbankfeld mit seinem Kurznamen statt mit dem Langnamen referenzieren. Dies gilt, falls Sie Natural im Reporting Mode ohne Natural Security benutzen und falls das für den Datenbankzugriff verwendete Statement statt einer Referenz auf ein View eine Referenz auf ein DDM enthält.

Die Entscheidung, ob ein Feldname als eine Kurznamen-Referenz betrachtet wird, erfolgt abhängig von der Länge des Namens. Wenn die Feldkennung (Identifizier) aus zwei Zeichen besteht, wird davon ausgegangen, dass eine Kurznamen-Referenz vorliegt; ein Feldname mit einer anderen Länge wird als Langnamen-Referenz betrachtet. Diese standardmäßige Interpretation können Sie zusätzlich beeinflussen, indem Sie die Compiler-Option DBSHORT auf ON oder OFF setzen:

ON	<p>Die Verwendung eines Kurznamens zum Referenzieren eines Datenbankfeldes ist erlaubt, jedoch wird die Verwendung eines Datenbank-Kurznamens <i>nicht</i> generell gestattet (selbst wenn DBSHORT=ON)</p> <ul style="list-style-type: none"> <li>■ für die Definition eines Feldes, wenn ein View erstellt wird;</li> <li>■ wenn im Programmcode ein View verwendet wird;</li> <li>■ wenn im Programmcode ein <code>DEFINE DATA LOCAL</code>-Statement zur Definition von Variablen verwendet wird;</li> <li>■ wenn Natural Security aktiv ist.</li> </ul> <p>Dies ist der Standardwert.</p>
----	---

OFF	<p>Ein Datenbankfeld kann nur über seinen Langnamen referenziert werden. Jede Datenbankfeldkennung wird unabhängig von ihrer Länge als Langnamen-Referenz betrachtet.</p> <p>Wenn ein zwei Zeichen langer Name geliefert wird, der nur als Kurzname und nicht als Langname gefunden werden kann, wird bei der Kompilierung ein Syntaxfehler NAT0981 ausgegeben.</p> <p>Dies ermöglicht die Verwendung von Langnamen, die in einer DDM mit einer Kennungslänge von 2 Byte definiert wurden. Diese Option ist wichtig, wenn die darunter liegende Datenbank, auf die Sie mit diesem DDM zugreifen wollen, vom Typ SQL (DB2) ist und wenn dort Tabellenspalten mit einem zwei Zeichen langen Namen existieren. Dagegen wird bei allen anderen Datenbanktypen (zum Beispiel Adabas) jeder Versuch, ein Langnamenfeld mit einer Namenslänge von 2 Bytes zu definieren bei der DDM-Generierung zurückgewiesen.</p> <p>Darüber hinaus wird das Programm, wenn keine Kurznamenreferenzen verwendet werden (was durch DBSHORT=OFF erzwungen werden kann), unabhängig davon, ob es unter Natural Security kompiliert wird oder nicht.</p>
-----	---

**Beispiele:**

Gegeben sei die folgende Datenbankfelddefinition in dem DDM EMPLOYEES:

Kurzname	Langname
AA	PERSONNEL-ID

**Beispiel 1:**

```

OPTIONS DBSHORT=ON
READ EMPLOYEES
  DISPLAY AA      /* Datenbankfeldkurzname AA ist erlaubt
END

```

**Beispiel 2:**

```

OPTIONS DBSHORT=OFF
READ EMPLOYEES
  DISPLAY AA      /* Syntaxfehler NAT0981, weil DBSHORT=OFF
END

```

**Beispiel 3:**

```

OPTIONS DBSHORT=ON
DEFINE DATA LOCAL
1 V1 VIEW OF EMPLOYEES
  2 PERSONNEL-ID
END-DEFINE
READ V1 BY PERSONNEL-ID
  DISPLAY AA      /* Syntaxfehler NAT0981, weil PERSONNEL-ID im View definiert ist;
                  /* (selbst wenn DBSHORT=ON)
END-READ
END

```

### DB2ARRY - Unterstützung von DB2-Arrays in SQL SELECT- und INSERT-Statements

Mit der Option `DB2ARRY` können Sie die Recherche und/oder das Einfügen in mehreren Reihen in DB2 mit nur einer Ausführung eines SQL `SELECT`- oder `INSERT`-Statements aktivieren. Dies ermöglicht die Angabe von Arrays als Zielfelder im SQL `SELECT`-Statement oder als Quellfelder im SQL `INSERT`-Statement. Wenn `DB2ARRY` auf `ON` gesetzt ist, ist es nicht mehr möglich, alphanumerische Natural-Arrays für DB2 `VARCHAR`/`GRAPHIC`-Spalten zu verwenden. Stattdessen müssen Sie lange alphanumerische Natural-Variablen verwenden.

ON	DB2-Array-Unterstützung ist eingeschaltet.
OFF	DB2-Array-Unterstützung ist ausgeschaltet. Dies ist der Standardwert.

### DB2BIN - Generierung von SQL Binärdatentypen bei Natural-Binärfeldern

Die Option `DB2BIN` kann benutzt werden, um die DB2-Datentypen `BINARY` und `VARBINARY` zu unterstützen.

Wenn `DB2BIN` auf `OFF` gesetzt ist, werden die Natural-Binärfelder (Format `B(n)`) als SQL-Datentyp `CHAR` ( $n \leq 253$ ) oder `VARCHAR` ( $253 < n \leq 32767$ ) generiert, so wie es in früheren Releases der Fall war. `DB2BIN=OFF` ist für jene Anwender gedacht, die Natural-Binärfelder wie SQL `CHAR`-Felder benutzten. `B2` und `B4` werden als SQL `SMALLINT` und `INTEGER` behandelt.

Wenn `DB2BIN` auf `ON` gesetzt ist, werden die Natural-Binärfelder (Format `B(n)`) als SQL-Datentyp `BINARY` ( $n \leq 255$ ) oder `VARBIN` ( $255 < n \leq 32767$ ) generiert. `DB2BIN=ON` ist für jene Anwender gedacht, die SQL-Binärspalten benutzen wollen. `B2` und `B4` werden ebenfalls als SQL `BINARY(2)` und `BINARY(4)` behandelt.



**Anmerkung:** Die Einstellung von `DB2BIN` wird am Ende der Kompilierung für das gesamte Natural-Objekt benutzt. Sie kann nicht für Teile des Natural-Objekts geändert werden.

ON	Die SQL-Typen BINARY und VARBIN werden für Natural-Felder im Binärformat generiert.
OFF	Die SQL-Typen CHAR und VARCHAR werden für Natural-Felder im Binärformat außer B2 und B4 generiert. Letztere werden als SQL-Datentyp SMALLINT und INTEGER behandelt.  Dies ist der Standardwert.

### DB2PKYU – Primäre Schlüsselfelder in das Natural DML Statement UPDATE stellen

Die Option DB2PKYU gilt nur, wenn die in Ihrer Umgebung installierte Version von Natural for DB2 diese Option unterstützt.

Die Option DB2PKYU kann benutzt werden, um primäre DB2-Schlüsselfelder mit einem Natural DML Statement UPDATE zu aktualisieren. Bei den primären DB2-Schlüsselfeldern handelt es sich um Felder, deren Kurznamen im DDM mit dem Zeichen 0 beginnen.



**Anmerkung:** Die Einstellung der Option DB2PKYU am Ende der Kompilierung wird für das gesamte Natural-Objekt benutzt. Sie kann nicht für Teile eines Natural-Objekts geändert werden.

ON	Primäre DB2-Schlüsselfelder werden aktualisiert.  Primäre DB2-Schlüsselfelder, die innerhalb des Natural-Programms aktualisiert werden, werden in das resultierende Positioned UPDATE-Statement eines Natural DML UPDATE-Statement gestellt. Der für dieses Positioned UPDATE zurückgegebene SQLCODE +535 DB2 wird vom Natural for DB2-Laufzeitsystem als 0 (Null) behandelt.
OFF	Primäre DB2-Schlüsselfelder werden nicht aktualisiert.  Primäre DB2-Schlüsselfelder, die innerhalb des Natural-Programms aktualisiert werden, werden nicht in das resultierende Positioned UPDATE-Statement gestellt.  Dies ist der Standardwert.

### DB2TSTI – Generierung des SQL TIMESTAMP-Datentyps für Natural-TIME-Felder

Die Option DB2TSTI kann benutzt werden, um Natural-Variablen vom Typ TIME statt auf den SQL-Datentyp TIME auf den SQL-Datentyp TIMESTAMP abzubilden.

ON	Der SQL-Datentyp TIMESTAMP wird für Natural-TIME-Felder des Natural-Datenformats T generiert.  Dies gilt für das gesamte Natural-Objekt. Durch Setzen der Option DB2TSTI ist es nicht möglich, ein Objekt nur teilweise zu generieren.
OFF	Der SQL-Datentyp TIME wird für Natural-TIME-Felder des Natural-Datenformats T generiert.  Dies ist der Standardwert.



**Anmerkung:** Die Genauigkeit eines Natural-TIME-Feldes beträgt nur Zehntelsekunden, wohingegen eine SQL-TIMESTAMP-Spalte eine viel höhere Genauigkeit haben kann. Deshalb kann es passieren, dass der aus der Datenbank gelesene TIMESTAMP-Wert abgeschnitten wird, wenn DB2TSTI=ON gesetzt ist.

## ECHECK - Existenzprüfung bei objektaufrufenden Statements

ON	<p>Der Compiler prüft das Vorhandensein eines Programmierobjekts, welches in einem objektaufrufenden Statement angegeben ist, zum Beispiel in <code>FETCH [RETURN/REPEAT]</code>, <code>RUN [REPEAT]</code>, <code>CALLNAT</code>, <code>PERFORM</code>, <code>INPUT USING MAP</code>, <code>PROCESS PAGE USING</code>, Function Call, Helproutine-Aufruf.</p> <p>Die Existenzprüfung basiert auf der Suche nach dem katalogisierten Objekt oder nach der Source des Objekts, wenn es von einem <code>RUN [REPEAT]</code>-Statement aufgerufen wird.</p> <p>Dazu ist erforderlich, dass der Name des aufzurufen bzw. auszuführenden (RUN) Objekts als alphanumerische Konstante (und nicht als alphanumerische Variable) definiert ist.</p> <p>Andernfalls hat die Einstellung <code>ECHECK=ON</code> keine Auswirkung.</p> <p><b>Fehler-Behandlung bei ECHECK=ON</b></p> <p>Die Existenzprüfung wird nur dann ausgeführt, wenn das Objekt keine syntaktischen Fehler enthält. Die Existenzprüfung wird bei jedem Statement ausgeführt, welches ein Objekt aufruft.</p> <p>Gesteuert wird die Existenzprüfung durch den Profilparameter <code>PECK</code> (siehe <i>Parameter-Referenz-Dokumentation</i>).</p> <p><b>Probleme bei der Benutzung des CATAL-Systemkommandos mit ECHECK=ON</b></p> <p>Wenn ein <code>CATAL</code>-Systemkommando zusammen mit der Einstellung <code>ECHECK=ON</code> benutzt wird, ist folgendes zu beachten:</p> <p>Wenn ein <code>CATAL</code>-Vorgang aufgerufen wird, hängt die Reihenfolge, in der die Programmierobjekte kompiliert werden, in erster Linie vom Objekttyp und in zweiter Linie vom alphabetischen Namen des Objekts ab. Die Reihenfolge der Objekttypen beim Kompilieren ist:</p> <p>GDA's, LDA's, PDA's, Functions, Subprogramme, externe Subroutinen, Helproutrinen, Maps, Adapter, Programme, Klassen.</p> <p>Bei Objekten desselben Typs bestimmt die alphabetische Reihenfolge der Namen die Abfolge, in der sie katalogisiert werden.</p> <p>Wie zuvor erwähnt wird die erfolgreiche Ausführung des das Objekt aufrufenden Statements gegen die kompilierte Form des aufgerufenen Subprogramms geprüft. Wenn das aufrufende Objekt (das momentan kompiliert wird und das das Objekt aufrufende Statements enthält) vor dem aufgerufenen Objekt katalogisiert wird, kann das <code>ECHECK</code>-Ergebnis falsch sein, falls das aufgerufene Objekt nicht zuvor katalogisiert worden war. In diesem Fall ist das Objekt-Image des aufgerufenen Objekts noch nicht durch das Systemkommando <code>CATAL</code> erstellt worden.</p> <p>Lösung:</p>
----	---

	<ul style="list-style-type: none"> <li>■ Setzen Sie die Compiler-Option <code>ECHECK</code> auf <code>OFF</code>.</li> <li>■ Führen Sie mit dem <code>CATALL</code>-Kommando eine generelle Kompilierung in der gesamten Library durch oder, wenn nur ein einzelnes oder nur wenige Objekte geändert wurden, führen Sie eine separate Kompilierung dieser Objekte durch.</li> <li>■ Setzen Sie die Compiler-Option auf <code>ECHECK=ON</code>.</li> <li>■ Führen Sie mit der Funktion <code>CHECK</code> des <code>CATALL</code>-Kommandos eine generelle Kompilierung in der gesamten Library durch.</li> </ul>
<code>OFF</code>	Es erfolgt keine Existenzprüfung. Dies ist der Standardwert.

### GDASC - GDA-Signaturprüfung

Diese Option dient dazu, Informationen über die Struktur einer GDA (Global Data Area) zu speichern, um so zu bestimmen, ob ein Natural-Fehler ausgegeben werden soll, wenn eine unveränderte GDA katalogisiert wird.

Die GDA-Informationen (GDA-Signatur) ändern sich nur dann, wenn eine GDA geändert wird. Die GDA-Signatur ändert sich nicht, wenn eine GDA (versehentlich) katalogisiert wird, aber nicht geändert wurde.

Die Signatur der GDA und die in allen Natural-Objekten, welche diese GDA referenzieren, gespeicherten GDA-Signaturen werden zur Ausführungszeit verglichen, und zwar zusätzlich zu den Zeitstempeln der Objekte.

<code>ON</code>	GDA-Signaturen werden gespeichert und bei der Ausführung verglichen. Natural gibt nur dann eine Fehlermeldung aus, wenn die Signaturen nicht identisch sind.
<code>OFF</code>	GDA-Signaturen werden nicht gespeichert. Dies ist der Standardwert.

### GFID - Generierung globaler Formatkennungen

Mit dieser Option können Sie Naturals interne Generierung globaler Formatkennungen (Global Format IDs) steuern, um damit das Leistungsverhalten von Adabas bei der Wiederverwendbarkeit von Format-Buffer-Übersetzungen zu beeinflussen.

<code>ON</code>	Global Format IDs werden für alle Views generiert. Dies ist der Standardwert.
<code>VID</code>	Global Format IDs werden nur für in Local/Global Data Areas definierte Views generiert, aber nicht für innerhalb von Programmen definierte Views.
<code>OFF</code>	Es werden keine Global Format IDs generiert.

Weitere Informationen zu Global Format IDs siehe Adabas-Dokumentation.

## Regeln für die Generierung von GLOBAL FORMAT-IDs in Natural

### ■ Für Natural-Nukleus-interne Systemdateiaufrufe:

GFID=abccdde

dabei steht	für
a	x'F9'
b	x'22' oder x'21' in Abhängigkeit vom DB-Statement
cc	physische Datenbanknummer (2 Bytes)
dd	physische Dateinummer (2 Bytes)
ee	zur Laufzeit generierte Nummer (2 Bytes)

### ■ Für Benutzerprogramme oder Natural Utilities:

GFID=abbbbbbc

dabei steht	für
a	x'F8' oder x'F7' oder x'F6'
bbbbbb	Bytes 1-6 des STOD-Werts
c	physische Dateinummer



**Anmerkung:** STOD ist der Rückmeldewert der Store Clock Machine Instruction (STCK).

## KCHECK - Schlüsselwort-Prüfung

Prüfung auf Schlüsselwörter.

ON	Felddeklarationen in einem Programmierobjekt werden gegen einen Satz kritischer Natural-Schlüsselwörter geprüft. Falls ein Variablenname mit einem dieser Schlüsselwörter übereinstimmt, wird ein Syntaxfehler ausgegeben, wenn das Programmierobjekt geprüft oder katalogisiert wird.
OFF	Es erfolgt keine solche Prüfung. Dies ist der Standardwert.

Der Abschnitt *Prüfung der für Natural reservierten Schlüsselwörter durchführen* im Leitfaden zur Programmierung enthält eine Liste der Natural-Schlüsselwörter, die von der KCHECK-Option abgeprüft werden.

Der Abschnitt *Alphabetische Liste der für Natural reservierten Schlüsselwörter* im Leitfaden zur Programmierung enthält eine Übersicht über alle Natural-Schlüsselwörter und reservierten Wörter.

## LOWSRCE - Source in Kleinschreibung

Diese Option unterstützt die Verwendung von Programmen in Sourceform mit Kleinschreibung oder gemischter Schreibweise auf Großrechnerplattformen und erleichtert so die Übertragung von derartigen Programmen von anderen Plattformen in eine Großrechnerumgebung.

ON	Gestattet jede Art von Zeichen in Klein-/Großschreibung in Source-Programmen.
OFF	Gestattet nur Großschreibung. Das bedeutet, dass Schlüsselwörter, Variablenamen und Identifier in Großbuchstaben definiert werden müssen. Dies ist der Standardwert.

Wenn Sie Zeichen in Kleinschreibung mit `LOWSRCE=ON` verwenden, müssen Sie folgendes berücksichtigen:

- Die Syntaxregeln für Variablenamen erlauben es, ab der zweiten Stelle Zeichen in Kleinschreibung zu benutzen. Sie können deshalb zwei Variablen gleichen Namens definieren, indem Sie die eine in Kleinschreibung und die andere in Großschreibung angeben, zum Beispiel:

```
DEFINE DATA LOCAL
1 #Vari (A20)
1 #VARI (A20) ↵
```

Mit `LOWSRCE=OFF` werden die im Beispiel angegebenen Variablen als zwei verschiedene Variablen behandelt.

Mit `LOWSRCE=ON` unterscheidet der Compiler nicht zwischen Klein- und Großschreibung. Dies führt zu einem Syntaxfehler, weil eine doppelte Namensvergabe bei Variablen nicht erlaubt ist.

- Bei Verwendung des Session-Parameters `EM` (Edit Mask) in einem I/O-Statement oder in einem `MOVE EDITED`-Statement gibt es Zeichen, die das Daten-Layout, das einer Variablen (`EM`-Steuerzeichen) zugeordnet ist, und Zeichen, die Textfragmente in das Datenfeld einfügen.

Beispiel:

```
#VARI := '1234567890'
WRITE #VARI (EM=XXXXXXxXXXXX) ↵
```

Bei `LOWSRCE=OFF` ist die Ausgabe `12345xx67890`, weil bei Variablen im Alpha-Format nur X-, H- und Zirkumflex-(`)-Zeichen in Großschreibung verwendet werden können.

Bei `LOWSRCE=ON` ist die Ausgabe `1234567890`, weil das Zeichen `x` wie der Großbuchstabe `C` behandelt und deshalb als `EM`-Steuerzeichen für dieses Feldformat interpretiert wird. Um dieses Problem zu vermeiden, sollten Sie Textkonstantenfragmente in Apostrophe (') einschließen.

Beispiel:



```
WRITE #VARI(EM=XXXXX'xx'XXXXX)
```

Das Textfragment wird, unabhängig von den LOWSRCE-Einstellungen, *nicht* als ein EM-Steuerzeichen betrachtet.

- Da bei der Einstellung LOWSRCE=ON alle Variablennamen in Großbuchstaben umgesetzt werden, ist die Anzeige von Variablennamen in I/O-Statements (INPUT, WRITE oder DISPLAY) unterschiedlich.

Beispiel:

```
MOVE 'ABC' to #Vari
DISPLAY #Vari ↵
```

Bei LOWSRCE=OFF ist die Ausgabe:

```
#Vari ----- ABC ↵
```

Bei LOWSRCE=ON ist die Ausgabe:

```
#VARI ----- ABC ↵
```

## MAXPREC – Maximale Anzahl Nachkommastellen

Mit dieser Option können Sie die maximale Anzahl der Nachkommastellen festlegen, die der Compiler bei den Ergebnissen von arithmetischen Operationen erzeugt.

7 - 29	<p>Dieser Wert gibt die maximale Anzahl der Nachkommastellen an, die der Compiler bei den Ergebnissen von arithmetischen Operationen erzeugt.</p> <p>Der Vorgabewert ist 7. Er sorgt für Aufwärtskompatibilität bei vorhandenen Anwendungen. Werden solche Anwendungen mit der Einstellung MAXPREC=7 katalogisiert, so liefern sie das gleiche Ergebnis wie frühere Versionen. Objekte, die mit einer Natural-Version katalogisiert werden, welche die Option MAXPREC nicht unterstützt, werden so ausgeführt, als ob MAXPREC=7 gesetzt worden wäre.</p> <p>Falls bei Zwischenergebnissen eine höhere Genauigkeit benötigt wird, sollte der Wert von MAXPREC erhöht werden.</p> <p>Die Einstellung von MAXPREC begrenzt nicht die Anzahl der Nachkommastellen, welche für benutzerdefinierte Felder und Konstanten angegeben werden können. Jedoch wird durch die Genauigkeit solcher Felder und Konstanten die Genauigkeit der Ergebnisse bei arithmetischen Operationen beeinflusst. Dadurch ist es möglich, bei ausgewählten Berechnungen von der erhöhten Genauigkeit zu profitieren, ohne die Compiler-Option MAXPREC auf einen Wert setzen zu müssen, der sich unbeabsichtigt andere Berechnungen auswirkt. Deshalb kann das folgende Beispiel auch dann katalogisiert und ausgeführt werden, wenn die Einstellung MAXPREC=7 wirksam ist.</p>
-----------	---

	<pre> DEFINE DATA LOCAL 1 P (P1.15) END-DEFINE P := P + 0.1234567890123456 END </pre>
	Weitere Informationen siehe Genauigkeit von Ergebnissen bei arithmetischen Operationen im <i>Leitfaden zur Programmierung</i> .



**Vorsicht:** Wenn Sie den beim Katalogisieren eines Natural-Objekts verwendeten Wert der Option MAXPREC ändern, kann das auch dann zu unterschiedlichen Ergebnissen führen, wenn Sie die Objekt-Source nicht geändert haben. Siehe folgendes Beispiel:

Beispiel:

```

DEFINE DATA LOCAL
1 #R (P1.7)
END-DEFINE
#R := 1.0008 * 1.0008 * 1.0008
IF #R = 1.0024018 THEN ... ELSE ... END-IF

```

Der Wert von #R nach der Berechnung und Ausführung des IF-Statements ist abhängig von der Einstellung der Option MAXPREC:

Beim Kompilieren wirksame Einstellung von MAXPREC	Wert von #R	Ausgeführte Klausel des IF-Statements
MAXPREC=7	1.0024018	THEN-Klausel
MAXPREC=12	1.0024019	ELSE-Klausel

## MEMOPT - Speicheroptimierung für lokal deklarierte Variablen

Diese Option bestimmt, ob für unbenutzte, auf Programmebene 1 befindliche Felder oder Gruppen, die lokal definiert (DEFINE DATA LOCAL) wurden, Speicher zugeordnet wird oder nicht.

ON	Speicher wird zugeordnet nur für <ul style="list-style-type: none"> <li>■ ein auf Programmebene 1 befindliches Feld, falls auf das Feld oder die Redefinition dieses Feldes zugegriffen wird;</li> <li>■ eine Gruppe, wenn auf die Gruppe oder mindestens ein Gruppenfeld zugegriffen wird.</li> </ul>
OFF	Datenspeicher wird für alle Gruppen und Felder zugeordnet, die lokal benannt wurden. Dies ist der Standardwert.

## PCHECK - Parameter-Prüfung bei CALLNAT-Statements

Prüfung der Parameter bei Objekt aufrufenden Statements.

ON	<p>Der Compiler prüft Anzahl, Format, Länge und Array-Index-Grenzen der Parameter, die in einem Objekt aufrufenden Statement (zum Beispiel CALLNAT, PERFORM, INPUT USING MAP, PROCESS PAGE USING, Function Call, Helproutine-Aufruf), angegeben sind. Darüber hinaus wird bei der Parameterprüfung das OPTIONAL-Feature des DEFINE DATA PARAMETER-Statements berücksichtigt.</p> <p>Die Parameterprüfung basiert auf dem Vergleich der Parameter des Objekt aufrufenden Statements mit den DEFINE DATA PARAMETER-Definitionen in dem aufzurufenden Subprogramm.</p> <p>Dazu ist folgendes erforderlich:</p> <ul style="list-style-type: none"> <li>■ Der Name des aufzurufenden Subprogramms muss als alphanumerische Konstante (nicht als alphanumerische Variable) definiert sein.</li> <li>■ Das aufzurufende Subprogramm muss als katalogisiertes Objekt zur Verfügung stehen.</li> </ul> <p>Andernfalls hat die Einstellung PCHECK=ON keine Auswirkung.</p> <p><b>Fehler-Behandlung bei ECHECK=ON</b></p> <p>Die Parameterprüfung wird nur dann ausgeführt, wenn das Objekt keine syntaktischen Fehler enthält. Die Parameterprüfung wird bei jedem Statement ausgeführt, welches ein Objekt aufruft.</p> <p>Gesteuert wird die Parameterprüfung durch den Profilparameter PECK (siehe <i>Parameter-Referenz-Dokumentation</i>).</p> <p><b>Probleme bei der Benutzung des CATAL-Kammandos mit PCHECK=ON</b></p> <p>Wenn ein CATAL-Kommando zusammen mit der Einstellung PCHECK=ON benutzt wird, ist folgendes zu beachten:</p> <p>Wenn ein CATAL-Vorgang aufgerufen wird, hängt die Reihenfolge, in der die Programmierobjekte kompiliert werden, in erster Linie vom Objekttyp und in zweiter Linie vom alphabetischen Namen des Objekts ab. Die Reihenfolge der Objekttypen beim Kompilieren ist: GDAs, LDAs, PDAs, Functions, Subprogramme, externe Subroutinen, Help Routinen, Maps, Adapter, Programme, Klassen. Bei Objekten desselben Typs bestimmt die alphabetische Reihenfolge der Namen die Abfolge, in der sie katalogisiert werden.</p> <p>Wie zuvor erwähnt werden die Parameter des des Objekt aufrufenden Statements gegen die kompilierte Form des aufgerufenen Subprogramms geprüft. Wenn das rufende Objekt (das momentan kompiliert wird und das Objekt aufrufende Statements enthält) vor dem aufgerufenen Objekt katalogisiert wird, kann das PCHECK-Ergebnis falsch sein, falls die Parameter im CALLNAT-Statement und im aufgerufenen Subprogramm geändert wurden.</p> <p>Das hat zur Folge, dass das <i>neue</i> Parameterlayout im Objekt aufrufenden Statement mit dem <i>alten</i> Parameterlayout im DEFINE DATA PARAMETER-Statement des aufgerufenen Subprogramms verglichen wird.</p> <p>Lösung:</p>
----	--

	<ul style="list-style-type: none"><li>■ Setzen Sie die Compiler-Option auf PCHECK=OFF.</li><li>■ Führen Sie mit dem CATAL- Kommando eine generelle Kompilierung in der gesamten Library durch oder, wenn ein einzelnes oder nur wenige Objekte geändert wurden, führen Sie eine separate Kompilierung dieser Objekte durch.</li><li>■ Setzen Sie die Compiler-Option auf PCHECK=ON.</li><li>■ Führen Sie mit der Funktion CHECK des CATAL- Kommandos eine generelle Kompilierung in der gesamten Library durch.</li></ul>
OFF	Es erfolgt keine Parameterprüfung. Dies ist der Standardwert.

### PSIGNF - Interne Darstellung des positiven Vorzeichens bei gepackten Zahlen

ON	Das positive Vorzeichen einer gepackten Zahl wird intern als H'F' dargestellt. Dies ist der Standardwert.
OFF	Das positive Vorzeichen einer gepackten Zahl wird intern als H'C' dargestellt.

### THSEP - Dynamisches Tausender-Trennzeichen

Mit dieser Option können Sie beim Kompilieren die Verwendung des Tausender-Trennzeichens ermöglichen oder unterdrücken. Siehe auch Profil- und Session-Parameter THSEPC sowie Abschnitt *Trennzeichen-Angaben standardisieren (im Leitfaden zur Programmierung)*.

ON	Das Tausender-Trennzeichen wird verwendet. Jedes Tausender-Trennzeichen, das nicht Teil eines Zeichenketten-Literals ist, wird intern durch ein Steuerzeichen ersetzt.
OFF	Das Tausender-Trennzeichen wird nicht verwendet, d.h. der Compiler erzeugt kein Steuerzeichen für Tausender-Trennzeichen. Dies ist die Kompatibilitätseinstellung.

### TQMARK - Umsetzung von Anführungszeichen

Mit dieser Option können Sie Anführungszeichen (") in Apostrophe (') umsetzen.

ON	Jedes Anführungszeichen (") in einer Textkonstante wird als Apostroph (') ausgegeben. Dies ist der Standardwert.
OFF	Anführungszeichen (") in einer Textkonstante werden nicht umgesetzt, sondern als Anführungszeichen beibehalten.

Beispiel:

```
RESET A (A5) A:= 'AB"CD'
WRITE '12"34' / A / A (EM=H(5))
END ↵
```

Mit TQMARK ON sieht die Ausgabe so aus:

```
12'34 AB'CD C1C27DC3C4 ↵
```

Mit TQMARK OFF sieht die Ausgabe so aus:

```
12"34 AB"CD C1C27FC3C4 ↵
```

### TSEABL - Gültigkeit des Profilparameters TS

Diese Option bestimmt, ob der Profilparameter TS für Natural-System-Libraries (d.h. für Libraries, deren Namen mit SYS beginnt, außer SYSTEM) gilt oder auch für alle Benutzer-Libraries.

Natural-Objekte, die mit der Einstellung TSEABL=ON katalogisiert werden, bestimmen die Verwendung des TS-Parameters auch dann, wenn sie sich nicht in einer System-Library befinden.

ON	Der Profilparameter TS gilt für alle Libraries.
OFF	Der Profilparameter TS gilt nur für Natural-System-Libraries. Dies ist der Standardwert.

## Kompilierungsoptionen für Versions- und Plattformkompatibilität

- LUWCOMP - Auf UNIX oder Windows nicht verfügbare Syntax unterbinden
- MASKCME - MASK-Option kompatibel mit MOVE EDITED

Diese Optionen entsprechen den gleichnamigen Schlüsselwortparametern des Profilparameters CMPO bzw. des Parameter-Makros NTCMPO.

### LUWCOMP - Auf UNIX oder Windows nicht verfügbare Syntax unterbinden

Die Option LUWCOMP prüft, ob die Syntax der seit Natural für Großrechner Version 8.2 zur Verfügung stehenden Funktionalität auch durch Natural für UNIX Version 8.3 und Natural für Windows Version 8.3 unterstützt wird. Falls Syntaxunverträglichkeiten zwischen dem Großrechner und UNIX oder Windows festgestellt werden, wird die Kompilierung unter Natural für Großrechner Version 8.2 mit entsprechender Fehlermeldung und Reason Code abgebrochen.

Mögliche Werte:

ON	Wenn ein Programm kompiliert wird, wird jeder Versuch, ein Syntaxkonstrukt zu verwenden, das von Natural für Großrechner, aber nicht von Natural für UNIX oder Natural für Windows unterstützt wird, mit dem Syntaxfehler NAT0598 und einem entsprechenden <b>Reason Code</b> (siehe unten) zurückgewiesen.
OFF	Es erfolgt keine Syntaxprüfung. Inkonsistenzen zwischen dem Großrechner und UNIX oder Windows werden ignoriert. Dies ist die Standardeinstellung.

### Reason Codes bei Syntaxfehlern

Die folgenden Reason Codes zeigen an, welche Syntaxteile nicht unter UNIX oder Windows unterstützt werden:

Reason Code	Ungültige Syntax auf UNIX oder Windows
001	<p>Es wird eine Variable mit Format P/N oder eine numerische Konstante mit mehr als 7 Dezimalstellen definiert.</p> <p>Beispiel:</p> <pre>DEFINE DATA LOCAL   1 #P(P5.8)</pre>
004	<p>Eine der beiden folgenden Compiler-Optionen wird benutzt:</p> <ul style="list-style-type: none"> <li>■ MEMOPT</li> <li>■ MAXPREC</li> </ul> <p>Beispiel:</p> <pre>OPTIONS MAXPREC=10</pre>
007	<p>In einem MOVE ALL-Statement wird eine SUBSTR-Option beim Quell- oder Zielfeld benutzt.</p> <p>Example:</p> <pre>MOVE ALL 'X' TO SUBSTR(#A, 3, 5)</pre>

Reason Code	Ungültige Syntax auf UNIX oder Windows
011	<p>Die ADJUST-Option wird in einem READ WORK FILE-Statement benutzt, um die Größe eines X-Array-Feldes beim Zugriff automatisch zu ändern.</p> <p>Beispiel:</p> <pre>READ WORK FILE 1 #XARR(*) AND ADJUST</pre>
012	<p>Das in der REINPUT ... MARK-Klausel referenzierte Feld wird mit einer (CV=...)-Option geliefert.</p> <p>Beispiel:</p> <pre>REINPUT 'text' MARK *#FLD (CV=#C)</pre>
013	In der List der Felder bei einem WRITE WORK FILE-Statement werden Systemvariablen referenziert. .
014	<p>In einem READ- oder FIND-Statement wird Folgendes verwendet:</p> <ul style="list-style-type: none"> <li>■ eine IN SHARED HOLD-Klausel oder</li> <li>■ eine SKIP RECORDS IN HOLD-Klausel</li> </ul>
015	<p>Eines der folgenden beiden Statements wird benutzt:</p> <ul style="list-style-type: none"> <li>■ READLOB oder</li> <li>■ UPDATELOB</li> </ul>
016	<p>Das Quellfeld in einem SEPARATE-Statement war als ein Array definiert.</p> <p>Beispiel:</p> <pre>SEPARATE #TEXT(*) INTO ...</pre>
017	Die POSITION-Klausel wird in einem SEPARATE-Statement benutzt.
019	<p>Eine der folgenden neuen Systemvariablen wurde benutzt:</p> <ul style="list-style-type: none"> <li>■ *REINPUT-TYPE or</li> <li>■ *LINEX</li> </ul>

**MASKCME - MASK-Option kompatibel mit MOVE EDITED**

Mit dieser Option können Sie die MASK-Option mit dem MOVE EDITED-Statement kompatibel machen.

ON	Der Bereich der gültigen Jahreswerte, die zu den YYYY-Maskenzeichen passen, ist 1582 bis 2699, damit die MASK-Option mit dem MOVE EDITED-Statement kompatibel wird. Wenn der Profilparameter MAXYEAR auf 9999 gesetzt ist, erstreckt sich der Bereich der gültigen Jahreswerte von 1582 bis 9999.
OFF	<p>Der Bereich der gültigen Jahreswerte, die zu den YYYY-Maskenzeichen passen, ist 0000 - 2699. Dies ist der Standardwert.</p> <p>Wenn der Profilparameter MAXYEAR auf 9999 gesetzt ist, erstreckt sich der Bereich der gültigen Jahreswerte von 0000 bis 9999.</p>



# 11

## CPINFO

---

Dieses Kommando dient zum Anzeigen der relevanten Natural Codepage-Einstellungen, z. B. Inhalt der Systemvariablen `*LOCALE`, `*CODEPAGE`, aktuelle Codepage des Arbeitsbereichs des Editors, aktuelle Einstellungen der relevanten Parameter, ICU-Version, Unicode-Version usw., und zum Anzeigen der im Modul `NATCONFIG` definierten Codepages.

### Codepage-Informationen

Auf dem ersten Bildschirm werden folgende Informationen angezeigt:

Feld	Erläuterung
<code>*LOCALE</code>	Zeigt die Sprache und das Land der aktuellen Region. Siehe auch Systemvariable <code>*LOCALE</code> .
<code>*CODEPAGE</code>	Zeigt den IANA-Namen der zurzeit für Konvertierungen in und von Unicode benutzten Codepage, und wird mit dem Natural-Profilparameter <code>CP</code> gesetzt. Siehe auch Systemvariable <code>*CODEPAGE</code> .
Source area CP	Zeigt die aktuelle Codepage des Source-Bereichs. Siehe auch Profilparameter <code>CP</code> .
<code>CPOBJIN</code>	Gibt die Codepage an, in der die Batch-Eingabedatei für Daten kodiert ist. Siehe auch Profilparameter <code>CPOBJIN</code> .
<code>CPSYNIN</code>	Gibt die Codepage an, in der die Batch-Eingabedatei für Kommandos kodiert ist. Siehe auch Profilparameter <code>CPSYNIN</code> .
<code>CPPRINT</code>	Gibt die Codepage an, in der die Batch-Ausgabedatei für Daten kodiert werden soll. Siehe auch Profilparameter <code>CPPRINT</code> .
<code>CPCVERR</code>	Specifies whether a conversion error that occurs when converting from U format to A format or from A format to U format or between A formats with different code pages results in a Natural error or not. Siehe auch Profilparameter <code>CPCVERR</code> .
<code>SRETAIN</code>	Gibt an, dass alle existierenden Quelldaten in ihrem ursprünglichen Codierformat gespeichert werden sollen. Siehe auch Profilparameter <code>SRETAIN</code> .
<code>CFICU</code>	Schaltet die Unicode-Unterstützung bei diversen Unicode-Einstellungen ein. Siehe auch Profilparameter <code>CFICU</code> .

Feld	Erläuterung
OPRB ACODE/WCODE	<p>Diese Werte stellen den Schlüssel der gelieferten Kodierung für A-Felder und W-Felder während der Benutzer-Session für eine gegebene Datenbankkennung (DBID) dar.</p> <p>Um die DBID für OPRB ACODE/WCODE zu ändern, müssen Sie PF4 drücken.</p> <p>Es erscheint ein Fenster, in dem Sie die DBID für OPRB ACODE/WCODE ändern können.</p>
CMPO CPAGE	Zeigt die aktuelle Einstellung der CPAGE-Option. Diese Option kann benutzt werden, um eine Konvertierungsroutine zu aktivieren, die alle numerischen Konstanten (von der beim Kompilieren aktiven Codepage in die zur Laufzeit aktive Codepage) übersetzt, wenn das Objekt zur Laufzeit gestartet wird. Siehe auch Schlüsselwort-Subparameter CPAGE des Profilparameters CMPO.
WEBIO	Enthält entweder den Wert der Systemvariablen *BROWSER-IO oder den Wert OFF. Siehe auch Profilparameter WEBIO.
ICU version	Zeigt die aktuelle ICU-Version.
Unicode version	Zeigt die aktuelle Unicode-Version.
ICU module type	Zeigt die aktuelle Implementierung der ICU-Library.

Drücken Sie **Enter**, um weiterzublättern.

Auf dem zweiten Bildschirm werden folgende Informationen angezeigt:

Feld	Erläuterung
Product	Software AG-Produktname.
Product code	Software AG-Produktcode.
Product version	Version des Software AG-Produkts (SAGICU).
Revision	Revisionsnummer des SAGICU-Nukleus. Diese Nummer gilt nicht für Treiber.
Revision date	Erstellungsdatum der Revision.
z/OS driver	Erstellungsdatum des Treibers für z/OS.
BS2000 driver	Erstellungsdatum des Treibers für BS2000.
z/VSE driver	Erstellungsdatum des Treibers für z/VSE.
z/VM driver	Erstellungsdatum des Treibers für z/VM.
ICU version	Aktuelle ICU-Version.
Unicode version	Aktuelle Unicode-Version.
Data library	Name der ICU-Daten-Library gemäß Zuweisung mittels Profilparameter: CFICU=(DATFILE=value.

## In NATCONFIG definierte Codepages anzeigen

Setzen Sie das Feld `Display code pages defined in NATCONFIG` (Y/N) auf Y, und drücken Sie Enter.

Oder drücken Sie PF5 (NTCPG).

Es werden alle Codepages angezeigt, die im Makro `NTCPAGE` des Source-Moduls `NATCONFIG` vordefiniert sind.

Feld	Erläuterung
Container Name	Der interne ICU-Name (nur die ersten 20 Zeichen).
IANA Name	Der Standard-Name der Codepage (nur die ersten 20 Zeichen).
CCSID	Der „Coded Character Set Identifier“. Dieser Bezeichner kennzeichnet den Zeichensatz gemäß Kennung seitens IBM.
First Alias Name	Erster alternativer Name für die Codepage (nur die ersten 20 Zeichen).
D/E	In dieser Spalte wird der Status der Codepage im Source-Modul <code>NATCONFIG</code> angezeigt.  E Codepage ist in <code>NATCONFIG</code> definiert und eingeschaltet. D Codepage ist in <code>NATCONFIG</code> definiert, aber ausgeschaltet.

## Weitere Informationen

Siehe auch:

- *Unicode and Code Page Support*
- *SYSCP Utility - Code Page Administration*

---

# 12


## DELETE

---

▪ Syntax-Erklärung .....	62
▪ Auswahlliste .....	63
▪ Schutz vor versehentlichem Löschen .....	63
▪ Beispiele .....	63

```
DELETE [ [TYPE object-type ...] [ { SOURCE } { OBJECT } { BOTH } ] object-name ] ...
```

Das DELETE-Kommando dient dazu, Natural-Programmierobjekte aus der Natural-Systemdatei zu löschen.

 **Anmerkung:** Das DELETE-Kommando hat keine Auswirkung auf das momentan im Arbeitsbereich des Editors befindliche Source-Programm .

Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural benutzen*.

Syntax-Erklärung

<i>object-name</i>	Als <i>object-name</i> geben Sie den Namen des Objekts bzw. die Namen der Objekte an, das bzw. die Sie löschen wollen.	
	Zusätzlich können Sie angeben, ob das Source-Objekt, das katalogisierte Objekt oder beide gelöscht werden sollen:	
	SOURCE	Source-Objekt
	OBJECT	Katalogisiertes Objekt
	BOTH	Sowohl Source-Objekt als auch katalogisiertes Objekt. Dies ist der Standardwert.
<i>object-type</i>	Eine SOURCE/OBJECT/BOTH-Angabe gilt für alle nachfolgenden Objektnamen, d.h. solange, bis die nächste SOURCE/OBJECT/BOTH-Angabe auftritt.	
	Um alle Objekte zu löschen, deren Namen mit einer bestimmten Zeichenkette beginnt, können Sie für <i>object-name</i> Stern-Notation (*) verwenden.	
	In Verbindung mit der Stern-Notation (*) für <i>object-name</i> können Sie auch einen Objekttyp ( <i>object-type</i> ) angeben, wenn Sie nur Objekte eines bestimmten Typs löschen möchten.	
	Die möglichen Angaben für <i>object-type</i> entsprechen denen des Systemkommandos <b>EDIT</b> . Außerdem können Sie X (= Global, Local und Parameter Data Areas) und U (= Subprogramme, Subroutinen und Helproutinen) angeben.	
	<b>Anmerkung:</b> Wenn Sie die vollständigen Namen bestimmter Objekte angeben, erübrigt sich die Angabe des Objekttyps.	

---

## Auswahlliste

---

Wenn Sie Stern-Notation (\*) verwenden, erhalten Sie eine Auswahlliste, auf der Sie die Objekte, die Sie löschen möchten, markieren. Dabei können Sie für jedes einzelne Objekt bestimmen, ob sein Sourcecode, sein Objektmodul oder beides gelöscht werden soll, indem Sie das Objekt mit dem entsprechenden Buchstaben (S, O oder B) markieren.

Wenn Sie nur das DELETE-Kommando selbst eingeben, erhalten Sie ebenfalls eine Auswahlliste, und zwar aller in Ihrer aktuellen Library gespeicherten Objekte.

---

## Schutz vor versehentlichem Löschen

---

Zum Schutz vor versehentlichem Löschen erscheint automatisch ein Fenster, in dem Sie das Löschen eines Objekts durch Eingabe seines Namens bestätigen müssen.

Wenn Sie mehr als ein Objekt angegeben bzw. ausgewählt haben, erscheint zusätzlich ein Fenster, indem Sie angeben können, ob Sie das Löschen für jedes Objekt einzeln bestätigen möchten oder ob alle angegebenen/ausgewählten Objekte ohne Bestätigung gelöscht werden sollen.

---

## Beispiele

---

Mit diesem Kommando löschen Sie drei Programmierobjekte namens TOM, DICK und HARRY:

```
DELETE TOM DICK HARRY
```

Mit diesem Kommando löschen Sie das Source-Objekt und das katalogisierte Objekt JOHN, die Source-Objekte PAUL und GEORGE und das katalogisierte Objekt RINGO:

```
DELETE JOHN SOURCE PAUL GEORGE OBJECT RINGO
```

Mit diesem Kommando erhalten Sie eine Auswahlliste aller Objekte in der aktuellen Library:

```
DELETE
```

Mit diesem Kommando erhalten Sie eine Auswahlliste der Source-Objekte aller Masken (Maps) in der aktuellen Library:

```
DELETE TYPE M SOURCE *
```

Mit diesem Kommando erhalten Sie eine Auswahlliste aller in Source- und/oder Objektform in der aktuellen Library gespeicherten Global, Local und Parameter Data Areas, deren Namen mit D anfangen:

## DELETE

---

```
DELETE TYPE GLA D*
```

Mit diesem Kommando erhalten Sie eine Auswahlliste aller katalogisierten Objekte in der aktuellen Library, deren Namen mit YYZ anfangen:

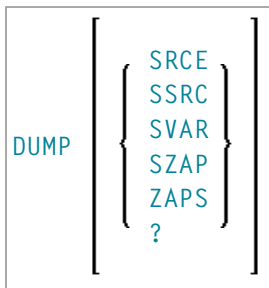
```
DELETE OBJECT YYZ*
```

Mit diesem Kommando löschen Sie die Source-Objekte und katalogisierten Objekte der Masken (Maps) TOM und DICK, das Source-Objekt der Map HARRY, das Source-Objekt des Programms JOHN und das katalogisierte Objekt des Programms PAUL:

```
DELETE TYPE M TOM DICK SOURCE HARRY TYPE P JOHN OBJECT PAUL
```



# 13 DUMP



Das Systemkommando **DUMP** liefert Informationen, die das technische Personal der Software AG benötigt, um einen Fehler zu finden, der zum Abbruch (Abend) des Natural-Systems geführt hat. Übergeben Sie diese Informationen zwecks Fehlerdiagnose und -korrektur dem Software AG Technical Support.

DUMP	Zeigt Abbruchinformationen (Kernspeicher-Inhalt) an.
DUMP SRCE	Zeigt den Bestand an Source-Änderungen, die am aktuellen Natural-Nukleus vorgenommen wurden, sortiert nach Produkten, die an Ihrem Standort installiert sind.
DUMP SSRC	Zeigt den Bestand an speziellen Source-Änderungen, die am aktuellen Natural-Nukleus vorgenommen wurden, sortiert nach Produkten, die an Ihrem Standort installiert sind.
DUMP SVAR	Zeigt TP-Monitor-abhängige und betriebssystemspezifische Systemvariablen und zusätzliche Informationen.
DUMP SZAP	Zeigt eine Liste aller im aktuellen Natural-Nukleus eingespielten Spezial-Zaps, sortiert nach Produkten, die an Ihrem Standort installiert sind.
DUMP ZAPS	Zeigt eine Liste aller im aktuellen Natural-Nukleus eingespielten Zaps, sortiert nach Produkten, die an Ihrem Standort installiert sind.
DUMP ?	Das DUMP-Kommando bietet noch eine Reihe weiterer Optionen (die auf den Hilfescreen erklärt sind, die Sie durch Eingabe eines Fragezeichens (?) auf dem DUMP-Menü erhalten). Falls es für die Fehlerdiagnose erforderlich ist, erfahren Sie vom Software AG Technical Support, wann und wie Sie diese Optionen einsetzen.



# 14

## EDIT

---

▪ Syntax 1 .....	68
▪ Syntax 2 .....	70
▪ Syntax 3 .....	70

Mit dem Systemkommando `EDIT` rufen Sie einen der Natural-Editoren auf, um eine Natural-Source zu erstellen.

Es gibt drei verschiedene Formen der Kommandosyntax. Diese sind nachfolgend in getrennten Abschnitten dokumentiert.

Verwandtes Kommando: [READ](#)

Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural benutzen*.

## Syntax 1

---

```
EDIT [object-type] [object-name] [library-id]
```

*object-type*

Folgende Objekttypen können editiert werden:

```
{  
  {  
    CLASS  
    4  
  }  
  COPYCODE  
  GLOBAL  
  HELPROUTINE  
  LOCAL  
  MAP  
  PARAMETER  
  PROGRAM  
  {  
    SUBPROGRAM  
    N  
  }  
  SUBROUTINE  
  TEXT  
  
  7 (für Function)  
}
```

Welcher Editor aufgerufen wird, hängt von dem zu editierenden Objekt ab:

- Ist das zu editierende Objekt eine Local Data Area, Global Data Area oder Parameter Data Area, wird der Data Area Editor aufgerufen.

- Ist das zu editierende Objekt eine Map oder eine Helproutine, die eine Map ist, wird der Map Editor aufgerufen.
- Ist das zu editierende Objekt eine Klasse, wird der Program Editor aufgerufen.
- Alle anderen Objekttypen (PROGRAM, SUBPROGRAM, SUBROUTINE, 7 (für Function), HELPROUTINE, COPYCODE, TEXT, DESCRIPTION) werden im Programm-Editor editiert.



**Anmerkung:** Das Textobjekt DESCRIPTION ist eine Programmbeschreibung, die im Predict-Datendiktionär gespeichert ist und gepflegt wird; diese Objekte können nur editiert werden, wenn Predict installiert ist.

Die Objekttypen werden im *Leitfaden zur Programmierung* beschrieben. Die Editoren werden in der *Editors-Dokumentation* beschrieben.

Wenn Sie den Namen des Objekts, das Sie editieren möchten, angeben, brauchen Sie keinen Objekttyp anzugeben.

*object-name*

Mit dem Systemkommando EDIT geben Sie den Namen des Objekts an, das Sie editieren möchten. Die maximale Länge des Objektnamens beträgt 8 Zeichen.

Natural lädt dann das Objekt in den Arbeitsbereich des entsprechenden Editors, wo Sie es editieren können. Wenn Sie das Objekt anschließend unter demselben Namen speichern wollen, brauchen Sie bei einem anschließenden SAVE-, CATALOG- oder STOW-Kommando keinen Namen anzugeben.

Falls der Arbeitsbereich des Editors nicht leer ist, wird das angegebene Objekt in den entsprechenden Editor geladen.



**Anmerkung:** Bei EDIT DESCRIPTION muß der *object-name* dem betreffenden Natural-Member-Namen in der Predict-Programmdefinition entsprechen.

Falls Sie keinen *object-name* angeben und es befindet sich kein Objekt im Arbeitsbereich, erhalten Sie den leeren Programm-Editor-Schirm, in dem Sie ein Programm erstellen können.

*library-id*

Befindet sich das Objekt in einer anderen Library als der, in der Sie gerade arbeiten, so müssen Sie die *library-id* der Library angeben, in der das zu editierende Objekt enthalten ist.



**Anmerkung:** Eine *library-id*, die mit SYS beginnt, darf nicht angegeben werden (Ausnahme: SYSTEM).

Wenn Natural Security aktiv ist, können Sie keine *library-id* angeben, d.h. Sie können nur Objekte aus ihrer aktuellen Library editieren

## Syntax 2

---

```
EDIT [ *  
      object-type ] { *  
                    object-name }
```

Wenn Sie den Namen des Objekts, das Sie editieren möchten, nicht wissen, haben Sie mit dieser Form des EDIT-Kommandos die Möglichkeit, von einer Liste von Objekten das gewünschte Objekt auszuwählen.

EDIT *	Liefert eine Liste aller Objekte, die in Ihrer aktuellen Library gespeichert sind.
EDIT object-type *	Liefert eine Liste aller Objekte des angegebenen Typs aus Ihrer aktuellen Library.

Um ein Objekt aus einem bestimmten Bereich von Objekten auszuwählen, können Sie Stern-Notation (\*) und Wildcard-Notation (?) für den *object-name* verwenden, und zwar in der gleichen Weise wie beim Systemkommando [LIST](#) beschrieben.

## Syntax 3

---

```
EDIT FUNCTION subroutine-name
```

Mit dem Kommando EDIT FUNCTION können Sie eine Subroutine unter ihrem internen Namen (also dem Namen, unter dem sie aufgerufen wird, nicht dem Namen, unter dem das Objekt gespeichert ist, in dem sie enthalten ist) zum Editieren aufrufen.

Der *subroutine-name* darf maximal 32 Zeichen lang sein.



**Anmerkung:** Bitte beachten Sie, dass das in dieser Syntax verwendete Schlüsselwort FUNCTION nicht mit dem oben aufgelisteten Natural-**Objekttyp** 7 (für Function) identisch ist. Siehe Beschreibung des Objekttyps Function im *Leitfaden zur Programmierung*.

Beispiel:

```
DEFINE SUBROUTINE CHECK-PARAMETERS  
...  
END-SUBROUTINE  
END ↵
```

Angenommen, obige Subroutine ist unter dem Objektnamen CHCKSUB gespeichert, dann haben Sie folgende Möglichkeiten, um die Subroutine CHECK-PARAMETERS aufzurufen:

Entweder mit

```
EDIT S CHKSUB
```

oder mit

```
EDIT F CHECK-PARAMETERS
```





# 15

## EDT

---

■ Syntax-Erklärung .....	74
■ EDT-Kommandos .....	74
■ EDT-Funktionstasten .....	75

Es empfiehlt sich, statt des EDT-Kommandos das **EDIT**-Kommando zu verwenden.

EDT [*object-name* [*library-id*]]

Mit dem Systemkommando EDT rufen Sie den Natural-Programm-Editor zum zeilenorientierten Editieren auf, d.h. Sie können ein Natural-Objekt (Programm, Subprogramm, Subroutine, Help-routine oder Copycode) Zeile für Zeile editieren, hierbei aber jeweils nur eine Zeile zur Zeit editieren. Hierzu können Sie die unten aufgeführten Kommandos und PF-Tasten verwenden.

Mit dem Kommando .E beenden Sie das zeilenorientierte Editieren.

## Syntax-Erklärung

---

<i>object-name</i>	<p>Als <i>object-name</i> geben Sie den Namen des Objekts an, das Sie editieren möchten (maximal 8 Zeichen lang). Wenn Sie <i>object-name</i> angeben, lädt Natural das Objekt in den Source-Arbeitsbereich, wo Sie es editieren können. Wenn Sie das Objekt anschließend unter demselben Namen speichern wollen, brauchen Sie bei einem <b>SAVE</b>-, <b>CATALOG</b>-, oder <b>STOW</b>-Kommando keinen Namen anzugeben.</p> <p>Wenn Sie keinen <i>object-name</i> angeben, können Sie das gerade im Editor befindliche Objekt im EDT-Modus editieren; befindet sich kein Objekt im Editor, können Sie Zeile für Zeile ein neues Objekt erstellen und erhalten hierzu zunächst Zeile 0010.</p>
<i>library-id</i>	<p>Befindet sich das zu editierende Objekt in einer anderen Library als der, in der Sie gerade arbeiten, so müssen Sie die <i>library-id</i> dieser Library angeben.</p> <p>Eine Library-ID, die mit SYS beginnt, darf nicht angegeben werden (Ausnahme: SYSTEM).</p> <p>Wenn Natural Security aktiv ist, können Sie keine <i>library-id</i> angeben, d.h. Sie können nur Objekte aus Ihrer aktuellen Library editieren</p>

## EDT-Kommandos

---

Im EDT-Modus können Sie folgende Kommandos verwenden:

Kommando	Funktion
.B	Blättert zur letzten Zeile.
.Cnnnn( <i>m</i> )	Kopiert <i>m</i> Zeilen, und zwar ab Zeile <i>nnnn</i> .
.C' <i>text</i> '( <i>m</i> )	Kopiert <i>m</i> Zeilen, und zwar ab der Zeile, die mit <i>text</i> beginnt.
.D	Löscht die Zeile.
.D( <i>n</i> )	Löscht die Zeile und die <i>n</i> minus 1 nachfolgenden Zeilen.
.E	Beendet das zeilenorientierte Editieren.

Kommando	Funktion
.I	Fügt eine Zeile ein.
.I( <i>program</i> )	Fügt das Objekt <i>program</i> ein.
.Mnnnn	Verschiebt Zeile <i>nnnn</i> .
.M' <i>text</i> '( <i>m</i> )	Verschiebt <i>m</i> Zeilen, und zwar ab der Zeile, die mit <i>text</i> beginnt.
.R	Entspricht dem Systemkommando RENUMBER.
.S' <i>text</i> '	Sucht die Zeile mit der Zeichenkette <i>text</i> .
.T	Blättert zur ersten Zeile.
.nnnn	Blättert zu Zeile <i>nnnn</i> .
.+ <i>n</i>	Blättert <i>n</i> Zeilen vor.
.- <i>n</i>	Blättert <i>n</i> Zeilen zurück.

## EDT-Funktionstasten

Im EDT-Modus können Sie folgende Funktionstasten benutzen:

Taste	Kommando	Funktion
PF1	.-18	Blättert 18 Zeilen zurück.
PF2	.T	Blättert zur ersten Zeile.
PF3	.B	Blättert zur letzten Zeile.
PF4	.+5	Blättert 5 Zeilen vor.
PF5	.+10	Blättert 10 Zeilen vor.
PF6	.+18	Blättert 18 Zeilen vor.
PF7	.R	RENUMBER-Kommando.
PF8	.I	Fügt Leerzeile ein.
PF9	.E	Beendet das zeilenorientierte Editieren.
PF10	.E,RUN	Beendet das zeilenorientierte Editieren und führt das Programm aus.
PF11	.E,SAVE,RUN	Beendet das zeilenorientierte Editieren, speichert das Programm in Sourceform und führt es aus.
PF12	.E,CAT,SAVE,EX	Beendet das zeilenorientierte Editieren, speichert das Programm in Source- und Objektform und führt es aus.



# 16

## EXECUTE

---

■ Syntax-Erklärung .....	78
■ Beispiele für das EXECUTE-Kommando .....	79

```
{ EXECUTE [REPEAT]      program-name      [library-id] }  
  program-name [parameter ...]
```

Das Systemkommando EXECUTE dient dazu, ein Natural-Objektmodul des Typs Programm auszuführen.

Das Objektmodul muß in der Natural-Systemdatei katalogisiert (d.h. in Objektform gespeichert) oder in den Natural-Nukleus eingebunden sein.

Die Ausführung eines Objektmoduls hat keinen Einfluß auf die Source, die sich gerade im Editor-Arbeitsbereich befindet.

## Syntax-Erklärung

---

EXECUTE	<p>Das Schlüsselwort EXECUTE ist nicht erforderlich; es genügt, <i>program-name</i>, d.h. den Namen des auszuführenden Programms anzugeben.</p> <p><b>Vorsicht:</b> Wenn Sie es in der Kommandozeile des Programm-Editors eingeben, können Sie das Systemkommando EXECUTE nicht mit EX abkürzen, da der Programm-Editor dies als das Editierkommando EX interpretieren würde.</p>
REPEAT	<p>Wenn das auszuführende Programm mehrere Ausgabeschirme erzeugt und Sie möchten, dass die Schirme unmittelbar nacheinander, d.h. ohne zwischengeschaltete Eingabezeilen, ausgegeben werden, verwenden Sie das Schlüsselwort EXECUTE zusammen mit dem Schlüsselwort REPEAT.</p>
<i>program-name</i>	<p>Der Name des Programms, das Sie ausführen möchten. Geben Sie keine Library-ID an, so kann das Programm nur ausgeführt werden, wenn es entweder in Ihrer aktuellen Library oder der aktuellen Steplib-Library (die Standard-Steplib ist SYSTEM) gespeichert ist.</p>
<i>library-id</i>	<p>Befindet sich das Objekt in einer anderen Library als der, in der Sie gerade arbeiten, so müssen Sie die Library-ID dieser Library angeben. Das Programm kann nur ausgeführt werden, wenn es auch tatsächlich in der angegebenen Library gespeichert ist.</p> <p>Eine Library-ID, die mit SYS beginnt, darf nicht angegeben werden (Ausnahme: SYSTEM).</p> <p>Wenn Natural Security aktiv ist, ist es nicht möglich, eine Library-ID anzugeben, d.h. ein Objekt aus einer anderen Library auszuführen.</p>
<i>parameter</i>	<p>Wenn Sie ein Programm ausführen, indem Sie den Programmnamen ohne das Schlüsselwort EXECUTE eingeben, haben Sie die Möglichkeit, Parameter an das Programm zu übergeben. Diese Parameter werden dann vom ersten INPUT-Statement des ausgeführten Programms gelesen.</p> <p>Sie können die Parameter als positionelle Parameter oder als Schlüsselwortparameter angeben, wobei die einzelnen Angaben durch Leerzeichen oder das (mit dem Session-Parameter ID bestimmte) Eingabe-Begrenzungszeichen voneinander getrennt werden müssen.</p>

<b>Anmerkung:</b> Die Parameterwerte werden immer in Großbuchstaben umgesetzt (unabhängig von dem Terminalkommando %L oder dem Profilparameter LC=ON).
--

## Beispiele für das EXECUTE-Kommando

```
EXECUTE PROG1
```

```
EXECUTE PROG1 ULIB1
```

```
PROG1
```

```
PROG1 VALUE1 VALUE2 VALUE3
```

```
PROG1 VALUE1, VALUE2, VALUE3
```

```
PROG1 PARM1=VALUE1, PARM2=VALUE2, PARM3=VALUE3
```

```
PROG1 PARM3=VALUE3 PARM1=VALUE1 VALUE2
```





# 17

## FIN

---

`FIN`

Das Systemkommando `FIN` dient dazu, eine Natural-Session zu beenden. Es gilt für Online- wie für Batch-Sessions.

Eine Batch-Session wird auch beendet, sobald in den Kommando-Eingabedaten eine „End-of-File“-Bedingung entdeckt wird.



# 18

## GLOBALS

---

■ Syntax-Erklärung .....	84
■ Liste der Parameter .....	84
■ Zusammenhang zwischen GLOBALS, SET GLOBALS und anderen Statements .....	85

GLOBALS [*parameter=va*lue ...]

Mit dem Systemkommando GLOBALS können Sie Natural-Session-Parameter setzen.

## Syntax-Erklärung

GLOBALS	Falls Sie das GLOBALS-Kommando ohne Parameter eingeben, erhalten Sie einen Schirm, der Ihnen die gegenwärtig gültigen Parameterwerte zeigt. Dort haben Sie auch die Möglichkeit, diese Werte zu ändern.
<i>parameter</i>	Die einzelnen Session-Parameter-Einstellungen können in beliebiger Reihenfolge angegeben werden; sie müssen jeweils durch ein Leerzeichen voneinander getrennt werden.  Falls der Platz in der Kommandoeingabezeile nicht ausreicht, um alle gewünschten Parameter anzugeben, führen Sie gegebenenfalls mehrere GLOBALS-Kommandos nacheinander aus, siehe Beispiel.

Beispiel:

```
GLOBALS DC=, ID=.
```

## Liste der Parameter

Die folgende Tabelle enthält eine Liste der Session-Parameter, die Sie mit dem Systemkommando GLOBALS angeben können.

Parameter	Funktion (Kurztext)
CC	Fehlerverarbeitung im Batch-Modus
CF	Steuerzeichen für Terminalkommandos
CPCVERR	Codepage-Umsetzungsfehler
DC	Dezimalstellenzeichen
DFOUT	Datumsformat für Ausgabe
DFSTACK	Datumsformat für Stack
DFTITLE	Datumsformat in Standard-Seitenüberschrift
D0	Anzeige-Reihenfolge von Ausgabedaten
DU	Dump-Generierung
EJ	Seitenvorschub
FCDP	Füllzeichen für dynamisch geschützte Felder
FS	Format-Spezifikation für Benutzervariablen
IA	Input-Zuweisungszeichen

Parameter	Funktion (Kurztext)
ID	Input-Begrenzungszeichen
IM	Input-Modus
LE	Reaktion auf Limit-Überschreitung bei Verarbeitungsschleifen
LS	Zeilenlänge
LT	Limit für Verarbeitungsschleifen
MT	Maximale CPU-Zeit
NC	Verwendung von Systemkommandos
OPF	Überschreiben geschützter Felder durch Helprouninen
PD	Seiten-Limit für NATPAGE
PM	Druck-Modus
PS	Länge einer Reportseite
REINP	Interner REINPUT bei ungültigen Daten
SA	Terminal-Warnton
SF	Spaltenabstand
SL	Sourcecode-Zeilenlänge
SM	Programmierung im Structured Mode
THSEPCH	Tausender-Trennzeichen
TS	Umsetzung von Systemdatei-Programmausgaben
WH	Warten auf Datensatz im Hold
ZD	Division durch Null
ZP	Anzeige von Nullwerten

## Zusammenhang zwischen GLOBALS, SET GLOBALS und anderen Statements

### SET GLOBALS-Statement

Das Systemkommando GLOBALS und das Statement SET GLOBALS bieten dieselben Parameter und können beide in derselben Natural-Session verwendet werden.

Die mit einem GLOBALS-Kommando angegebenen Parameterwerte gelten solange, bis Sie sie mit einem neuen GLOBALS-Kommando überschreiben, die Session beenden oder ein Logon in eine andere Library ausführen.

### **Andere Statements, die die Session-Parameter-Einstellungen beeinflussen**

Für ein einzelnes Programm oder Teile eines Programms können Sie zum Teil Parameterwerte angeben, die von den sessionweit gültigen abweichen, und zwar mittels `LIMIT-`, `EJECT-` und `FORMAT-`Statements und mittels Formatangaben, die Sie in `INPUT-`, `DISPLAY-`, `PRINT-` und `WRITE-`Statements machen.

Informationen zu diesen Statements finden Sie in der *Statements*-Dokumentation.

# 19

## HELP

```
{ HELP } [ parameter
  ?      [NAT]nnnn
         USER[nnnn][library-name]
         ERROR ]
```

Mit dem Systemkommando **HELP** rufen Sie das Natural-Hilfesystem auf. Es liefert Informationen zu Natural-Statements, -Kommandos und -Fehlermeldungen.

HELP	Zeigt das Hilfemenü an.	
HELP <i>parameter</i>	Zeigt Informationen zu dem betreffenden Parameter an. Mögliche Parameter sind:	
	<b>Parameter</b>	<b>Kommando-Beispiele</b>
	<i>statement</i>	HELP FIND HELP DEFINE WORK FILE
	<i>system-variable</i>	HELP *CODEPAGE
	<i>system-function</i>	HELP SORTKEY
	<i>system-command</i>	HELP LIST
	<i>session-parameter</i>	HELP AD
	<i>terminal-command</i>	HELP %<TEST
	<b>Anmerkung:</b> <ol style="list-style-type: none"> <li>1. Als <i>parameter</i> geben Sie nur die Grundform eines Statements, einer Systemvariablen, eines Systemkommandos usw. an. Gültiges Beispiel: HELP AD; ungültiges Beispiel: HELP AD=0</li> <li>2. Bei einem ungültigen Kommando wird ein Parameterfehler angezeigt; zum Beispiel, wenn Sie Hilfe zu einem nicht unterstützten Hilfethema anfordern: Parameter Error: SOSI</li> </ol>	

HELP [NAT] <i>nnnn</i>	Wenn Sie HELP und eine (bis zu vierstellige) Nummer (wahlweise mit NAT davor) eingeben, erhalten Sie die Erklärung zu der betreffenden Fehlernummer, d.h. den Langtext der Natural-Systemfehlermeldung NAT <i>nnnn</i> .
HELP NAT	Zeigt Informationen zu allen Fehlermeldungen an.
HELP USER <i>nnnn</i>	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer <i>nnnn</i> aus der aktuellen Library an.
HELP USER <i>nnnn</i> [ <i>library-name</i> ]	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer <i>nnnn</i> aus der angegebenen Library an.
HELP USER	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer <i>nnnn</i> aus der aktuellen Library an.
HELP USER [ <i>library-name</i> ]	Zeigt eine Auswahlliste <i>aller</i> library-spezifischen Meldungen der aktuellen Library an.
HELP <u>ERROR</u>	Zeigt den Langtext des zuletzt aufgetretenen Fehlers an.



## 20 INPL

---

### INPL [R]

Mit dem `INPL`-Kommando rufen Sie die `INPL`-Utility auf. Diese Utility dient *nur* zum Laden von Software-AG-Installationsdatasets in die Systemdateien (wie in der `INPL`-Online-Hilfe und in den plattform-spezifischen Installationsschritten der Installationsdokumentation beschrieben).

Ansonsten verwenden Sie zum Laden von Objekten in die Systemdateien den Object Handler.

INPL	Wenn Sie das <code>INPL</code> -Kommando ohne Parameter eingeben, rufen Sie damit die <code>INPL</code> -Utility auf.
INPL R	<p>Ruft die <code>INPL</code>-Utility-Funktion <b>Natural Security Recover</b> auf. Diese Option ist nur verfügbar, wenn Natural Security installiert ist.</p> <p>Damit können Sie den Zugang zur Natural Security-Library <code>SYSSEC</code> zurücksetzen: Der User <code>DBA</code>, die Library <code>SYSSEC</code> sowie der Link zwischen beiden werden wieder so definiert, wie Sie nach der Installation ursprünglich definiert waren; gleichzeitig werden alle anderen Links nach <code>SYSSEC</code> gelöscht. Siehe auch <i>Inaccessible Security Profiles</i> im Abschnitt <i>Countersignatures</i> der <i>Natural Security</i>-Dokumentation.</p>

Weitere Informationen siehe *INPL Utility* in der *Utilities*-Dokumentation.

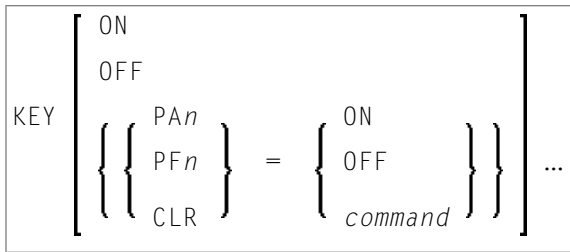
---

# 21

## KEY

---

■ Kommandos zuweisen .....	93
■ Aktivieren/Deaktivieren aller Tasten - KEY ON/OFF .....	93
■ Aktivieren/Deaktivieren einzelner Tasten — KEY key=ON/OFF .....	94



Mit dem Systemkommando **KEY** können Sie bestimmten Funktionstasten auf der Tastatur Ihres Video-Terminals eine Funktion zuweisen oder eine zugewiesene Funktion ändern. Außerdem können Sie die zugewiesenen Funktionen deaktivieren und später wieder aktivieren.

Dies ist für folgende Funktionstasten möglich:

- PA1 bis PA3,
- PF1 bis PF24
- CLEAR bzw. LÖSCH

Als Funktion können Sie einer Taste folgendes zuweisen:

- ein Natural-Systemkommando,
- ein Natural-Terminalkommando,
- ein benutzerdefiniertes Kommando.

Natural führt das zugewiesene Kommando immer dann aus, wenn Sie die betreffende Taste im Kommando-Modus (**NEXT-Modus**) drücken.



#### **Anmerkungen:**

1. Die mit dem Systemkommando **KEY** zugewiesenen Funktionen sind völlig unabhängig von den Funktionen, die den Tasten mittels eines **SET KEY**-Statements in einem Programm zugewiesen werden.
2. Der Natural-Administrator hat außerdem die Möglichkeit, mit dem Profilparameter **KEY** bestimmten Tasten Funktionen zuzuweisen.
3. Dieses Kommando kann nicht im Batch-Betrieb ausgeführt werden.

## Kommandos zuweisen

Wenn Sie nur das Kommando `KEY` (ohne Parameter) eingeben, erscheint der **Function-Key Assignments**-Schirm. Auf diesem Schirm können Sie den einzelnen Tasten Kommandos zuweisen, indem Sie sie in den Eingabefeldern eintragen.

Um einer Taste eine anderes Kommando zuzuweisen, überschreiben Sie den bestehenden Eintrag mit einem neuen Kommando.

Um eine Kommandozuweisung zu löschen, löschen Sie den Eintrag oder überschreiben ihn mit Leerzeichen.

Sie können einzelnen Tasten auch Kommandos zuweisen, indem Sie diese direkt mit dem `KEY`-Kommando angeben, und zwar in der Form `KEY key=command` (wobei *key* der Name der Taste und *command* der Name des Kommandos ist, das der Taste zugewiesen werden soll). Beispiel:

```
KEY PF1=CLEAR
```

Enthält das zugewiesene Kommando Leerzeichen, muß es in Apostrophen stehen. Beispiel:

```
PF13='UPDATE OFF'
```

## Aktivieren/Deaktivieren aller Tasten - KEY ON/OFF

Mit dem Kommando `KEY OFF/ON` aktivieren/deaktivieren Sie alle zugewiesenen Kommandos.

KEY OFF	Wenn Sie eine Funktionstaste drücken, erhalten Sie von Natural eine entsprechende Meldung, dass die Taste nicht aktiv ist.
KEY ON	Mit dem Kommando <code>KEY ON</code> aktivieren Sie die mit <code>KEY OFF</code> außer Kraft gesetzten Kommandos wieder.

Sie können die Tasten auch aktivieren/deaktivieren, indem Sie oben rechts auf dem **Function-Key Assignments**-Schirm den Eintrag `ON` bzw. `OFF` im Feld **Activate Keys** überschreiben.



**Anmerkung:** Die `CLEAR`- bzw. `LÖSCH`-Taste kann nicht aktiviert bzw. deaktiviert werden. Falls ihr keine andere Funktion zugewiesen wird, hat sie die gleiche Funktion wie das Terminalkommando `%%`. Das Kommando `KEY ON/OFF` und das **Activate Keys**-Feld haben keine Auswirkung auf die `CLEAR`-Taste.

## Aktivieren/Deaktivieren einzelner Tasten — KEY key=ON/OFF

---

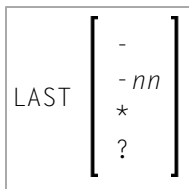
Mit dem Kommando `KEY key=OFF/ON` setzen Sie das einer bestimmten Taste zugewiesene Kommando außer Kraft.

KEY <i>key</i> =OFF	Damit setzen Sie das einer bestimmten Taste ( <i>key</i> ) zugewiesene Kommando außer Kraft.  Beispiel:  <code>KEY PF24=OFF</code>
KEY <i>key</i> =ON	Damit reaktivieren Sie eine zuvor außer Kraft gesetzte Kommandozuweisung.  Beispiel:  <code>KEY PF24=ON</code>



**Anmerkung:** Das Kommando `KEY CLR=ON/OFF` ist nicht möglich (vgl. [Anmerkung](#) oben).

# 22 LAST



Mit dem Systemkommando `LAST` können Sie sich die zuletzt ausgeführten Systemkommandos anzeigen lassen.

Sobald das Kommando in die Kommandozeile oder NEXT-Zeile gestellt worden ist, können Sie es durch Drücken von `EINGABE` erneut ausführen. Sie können es auch überschreiben, bevor Sie es ausführen.

Das `LAST`-Kommando zeigt nur die Systemkommandos an, die Sie tatsächlich eingegeben haben; Kommandos, die Natural aufgrund eines von Ihnen eingegebenen Kommandos intern ausgeführt hat, werden von `LAST` nicht erfaßt.

LAST	Das zuletzt ausgeführte Kommando wird in die Kommandozeile oder NEXT-Zeile gestellt und kann ausgeführt werden.
LAST -	<p>Das zuletzt ausgeführte Kommando wird in die Kommandozeile oder NEXT-Zeile gestellt und kann ausgeführt werden.</p> <p>Wenn Sie <code>LAST -</code> noch einmal eingeben, wird das vorletzte Kommando in die Kommandozeile oder NEXT-Zeile gestellt und kann ausgeführt werden.</p> <p>Durch wiederholte Eingabe von <code>LAST -</code> können Sie so Kommando für Kommando „zurückblättern“.</p> <p><b>Anmerkung:</b> Anstatt es wiederholt von Hand einzugeben, können Sie <code>LAST -</code> über das Systemkommando <code>KEY</code> auch einer PF-Taste zuweisen.</p>

LAST - <i>nn</i>	<p>Natural kann sich maximal die 20 letzten Kommandos „merken“; <i>nn</i> darf also nicht größer als 20 sein.</p> <p>Das <i>nn</i>tletzte ausgeführte Kommando wird in die Kommandozeile oder NEXT-Zeile gestellt und kann ausgeführt werden.</p>
LAST *	<p>Es erscheint ein Fenster, in dem die 20 zuletzt abgesetzten Kommandos angezeigt werden. Mit PF8 und PF7 können Sie in der Liste vor und zurück blättern, wenn mehr als 10 Kommandos angezeigt werden:</p> <ul style="list-style-type: none"> <li>■ Wenn Sie ein <i>einzelnes</i> Kommando erneut ausführen möchten, markieren Sie es mit dem Cursor und drücken Sie PF5 oder markieren Sie das Kommando mit einem Zeichen und drücken Sie ENTER.</li> <li>■ Wenn Sie <i>mehrere</i> Kommandos erneut ausführen möchten, markieren Sie sie mit Zahlen in der Reihenfolge, in der sie ausgeführt werden sollen, und drücken Sie ENTER, die Kommandos werden dann wie durch die Zahlen vorgegeben in aufsteigender Reihenfolge ausgeführt.</li> </ul>
LAST ?	Die Hilfe-Funktion für das LAST-Kommando wird aufgerufen.



## 23 LASTMSG

---

### LASTMSG

Mit dem Systemkommando `LASTMSG` können Sie sich zusätzliche Informationen zu der zuletzt aufgetretenen Fehlersituation anzeigen lassen.

Wenn Natural eine Fehlermeldung ausgibt, kann es in manchen Fällen sein, dass es sich bei dem betreffenden Fehler nicht um den tatsächlichen Fehler handelt, sondern um einen Folgefehler eines anderen Fehlers (welcher wiederum ein Folgefehler eines anderen Fehlers sein kann usw.) Mit dem `LASTMSG`-Kommando können Sie in solchen Fällen den ausgegebenen Fehler bis zu dem Fehler zurückverfolgen, der die Fehlersituation ursprünglich verursacht hat.

Wenn Sie das Kommando `LASTMSG` eingeben, erhalten Sie — jeweils zu der zuletzt aufgetretenen Fehlersituation — die ausgegebene Fehlermeldung sowie alle vorherigen (nicht ausgegebenen) Fehlermeldungen, die zu diesem Fehler geführt haben.

#### » Um Informationen zu dem entsprechenden Fehler anzuzeigen

- Markieren Sie eine dieser Meldungen mit dem Cursor und drücken Sie `ENTER`.

Sie erhalten folgende Informationen zu dem betreffenden Fehler:

- Fehlernummer;
- Nummer der Zeile, in der der Fehler auftrat;
- Name, Typ und Aufrufebene (Level) des Objekts, das den Fehler verursacht hat;
- Name, Datenbank-ID und Dateinummer der Library, in der das Objekt enthalten ist;
- Fehlerklasse (System = von Natural ausgegebener Fehler, User = von Benutzeranwendung ausgegebener Fehler);
- Fehlertyp (Laufzeit, Syntax, Kommandoausführung, Session-Beendigung, Program-Beendigung, Remote Procedure Call);

- Datum und Uhrzeit, wann der Fehler auftrat.



**Anmerkung:** Die Library SYSEXT enthält eine Programmierschnittstelle (API) USR2006N, über die Sie die von LASTMSG gelieferten Fehlerinformationen auch in Ihrer Natural-Anwendung erhalten können.

**Natural RPC (Remote Procedure Call):**

Bei einem Fehler auf dem Server werden folgende Fehlerinformationen nicht angezeigt: Datenbank-ID, Dateinummer, Datum und Uhrzeit.

# 24

## LIST

---

■ Syntax-Übersicht .....	100
■ Inhalt des Arbeitsbereichs auflisten .....	107
■ Sourcecode eines einzelnen Objekts anzeigen .....	108
■ Anzahl und Größe von Objekten anzeigen .....	108
■ Sourcecode mehrerer Objekte nacheinander anzeigen .....	109
■ Liste von Objekten anzeigen .....	109
■ Vorsortierte Liste von ausgewählten Objekten anzeigen .....	109
■ Langnamen katalogisierter Subroutinen und Klassen anzeigen .....	110
■ NOC-Optionen katalogisierter Objekte anzeigen .....	110
■ Compiler-Optionen katalogisierter Objekte anzeigen .....	110
■ Compiler Output zu einem katalogisierten Objekt anzeigen .....	110
■ Directory-Informationen anzeigen .....	112
■ DDMs (Views) anzeigen .....	113
■ Optionen .....	113
■ Objekt-Auswahlliste .....	118
■ Source-Liste .....	126
■ Individuelles List-Profil erstellen .....	131

Mit dem Systemkommando `LIST` können Sie sich den Sourcecode eines einzelnen Objekts anzeigen oder mehrere in Ihrer aktuellen Library gespeicherte Objekte auflisten lassen. Die zahlreichen Möglichkeiten des `LIST`-Kommandos sind im Folgenden beschrieben.

Siehe auch [LIST XREF](#) und die speziellen `LIST`-Kommandos für die DB2- und SQL-Datenbankverwaltungssysteme, die in separaten Dokumentationsabschnitten beschrieben werden.

Programmierschnittstellen (APIs): USR1054N, USR1055N, USR1056N, USR2018N, USR4216N. Siehe auch *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

## Syntax-Übersicht

```

LIST [ object-type object-name-range [settings]
      [ object-type object-name [options] [settings]
        object-name-range [range-clause] [settings]
        COUNT [object-name-range] [range-clause] [settings]
        SEQUENTIAL [object-type] object-name-range [options]
        DIRECTORY [ { object-name
                       object-name-range } ]
        EXTENDED [extended-type] object-name-range [settings]
        NOCOPT [object-type] object-name-range [settings]
        OPTIONS [object-type] object-name-range [settings]
        COMPOUT [compout-options] [settings]
      ]
      DDM [dgm-name]

```



### Anmerkungen:

1. Statt des Schlüsselworts `DDM` können Sie auch das Schlüsselwort `VIEW` (oder kurz `V`) verwenden.
2. Da das `LIST`-Kommando lange Zeilen mit bis zu 244 Zeichen anzeigen kann, sollten Sie die Zeilenlänge mit dem Profilparameter `LS` so groß wie möglich wählen. Wenn möglich, setzen Sie den Parameter `LS=250`.

**object-type**

Für *object-type* können Sie einen der nachfolgend aufgeführten Objekttypen angeben oder Stern-Notation (\*) benutzen.

*
{ CLASS }
4
COPYCODE
DATA-AREAS
GLOBAL
LOCAL
PARAMETER
{ DIALOG }
3
{ FUNCTION }
7
{ ADAPTER }
8
{ RESOURCE }
9
MAP
{ PROCESSOR }
CP
5
PROGRAM
RECORDING
ROUTINES
HELPROUTINE
{ SUBPROGRAM }
N
SUBROUTINE
TEXT

## object-name

Für *object-name* können Sie den Namen eines Objekts (maximal 8 Zeichen; bei [LIST EXTENDED](#) 32 Zeichen).

## object-name-range

Für *object-name-range* können Sie Stern-Notation (\*) und Wildcard-Notation (?) verwenden:

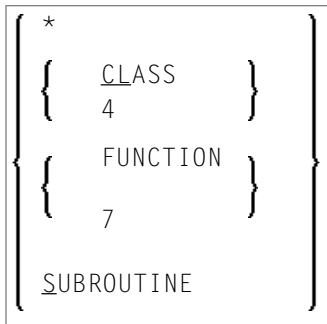
- Für eine Liste aller Objekte in der aktuellen Library geben Sie als *object-name-range* einen Stern (\*) an, aber keinen *object-type*.
- Für eine Liste aller Objekte eines bestimmten Objekttyps geben Sie den gewünschten *object-type* sowie als *object-name-range* einen Stern (\*) an.
- Für eine Liste eines bestimmten Bereichs von Objekten können Sie für den *object-name-range* Stern-Notation (\*) und Wildcard-Notation (?) verwenden:
  - Stern-Notation ist die Möglichkeit, innerhalb des *object-name-range* einen Stern (\*) anzugeben: der Stern steht für eine beliebige Zeichenkette beliebiger Länge.
  - Wildcard-Notation ist die Möglichkeit, innerhalb des *object-name-range* ein Fragezeichen (?) anzugeben: das Fragezeichen steht für ein beliebiges Zeichen.
- In einem *object-name-range* können Sie eine oder mehrere Stern- und Wildcard-Notationen miteinander kombinieren.
- Für eine vollständige Liste von Objekten ab einen bestimmten Startwert bzw. bis zu einem bestimmten Endwert können Sie die Größer-Zeichen-Notation (>) bzw. die Kleiner-Zeichen-Notation (<) verwenden.
- Die Größer-Zeichen-Notation (>) und die Kleiner-Zeichen-Notation (<) können nicht miteinander oder mit Stern-Notation (\*) oder Wildcard-Notation (?) kombiniert werden. Sie können nur zum Anzeigen einer Liste von Objekten verwendet werden (siehe [Liste von Objekten anzeigen](#)).

## options

Eine ausführliche Beschreibung der *options* ist im Abschnitt [Optionen](#) enthalten.

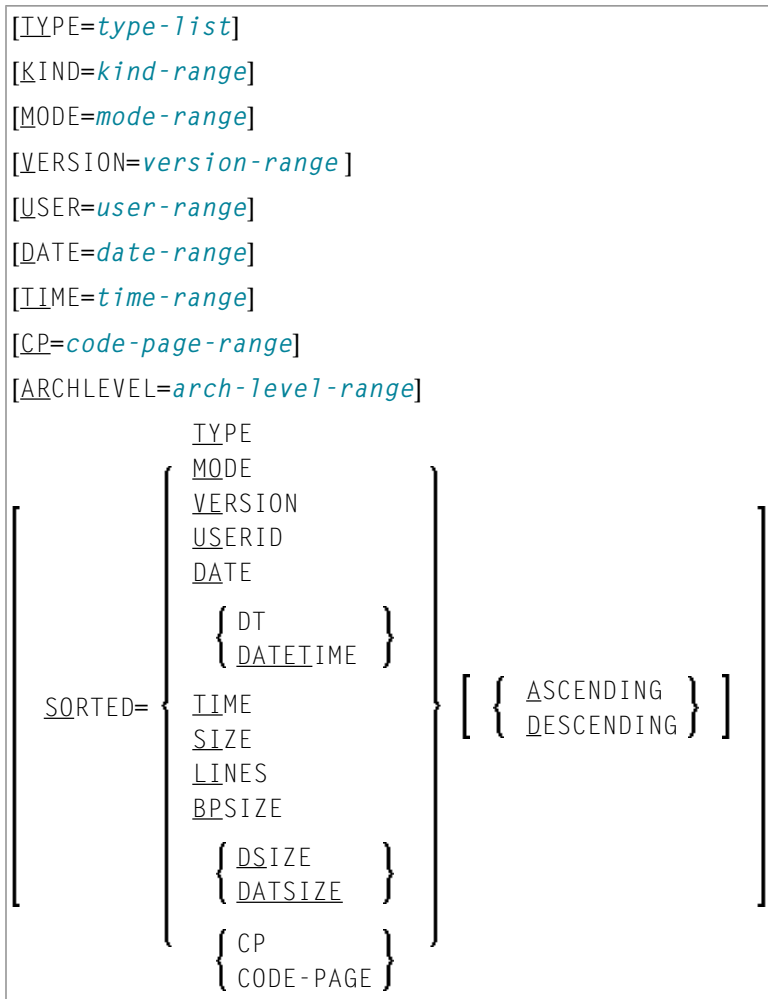
## extended-type

Für *extended-type* können Sie einen der nachfolgend aufgeführten Objekttypen oder einen Stern (\*) angeben.



Ausführliche Beschreibung siehe [LIST EXTENDED](#).

### range-clause



**Anmerkung:** SORTED kann beim LIST COUNT-Kommando nicht benutzt werden.

<i>type-list</i>	* (für alle Typen) oder eine Liste von 11 gültigen 1-Byte-Natural-Objektyp-Zeichen (z.B. P for Program, M für Map).	
<i>kind-range</i>	*	Listet alle Objekte.
	S	Listet nur Objekte in Sourceform.
	C	Listet nur katalogisierte Objekte.
	S/C	Listet nur Objekte, die in Sourceform und als katalogisiertes Objekt vorhanden sind.
	S/	Listet nur Objekte auf, die nur als Sourceobjekte vorhanden sind.
	/C	Listet nur Objekte, die als katalogisiertes Objekt vorhanden sind.
	W	Listet nur in Sourceform und Objektform gespeicherte Objekte.
<i>mode-range</i>	*	Listet alle Objekte.
	S	Listet nur Objekte, die im Structured Mode geschrieben wurden.
	R	Listet nur Objekte, die im Report Mode geschrieben wurden.
<i>version-range</i>	<p>Die Natural-Version der Natural-Objekte.</p> <p>Siehe auch Begriffsdefinition von Version im <i>Glossary</i>.</p> <p>Sie können einen Bereich von Versionsnummern angeben, siehe <i>range-notation</i>.</p>	
<i>user-range</i>	<p>Die User-ID des Benutzers, der ein Natural-Programmierobjekt gespeichert oder katalogisiert hat.</p> <p>Sie können eine einzelne User-ID oder einen Bereich von User-IDs angeben, siehe <i>range-notation</i>.</p>	
<i>date-range</i>	<p>Listet alle Objekte mit einem Speicher- oder Katalogisierungsdatum innerhalb des angegebenen Datumsbereichs. Sie können ein einzelnes Datum oder einen Datumsbereich angeben.</p> <p>Gültiges Datumsformat: <i>YYYY-MM-DD</i> (<i>YYYY</i>=Jahr, <i>MM</i>=Monat, <i>DD</i>=Tag)</p> <p>Gültige Datumsbereiche:</p> <ul style="list-style-type: none"> <li>■ Vorangestellte Zeichen (Beispiel: 2002*)</li> <li>■ Startwert (Beispiel: 2002-05&gt;)</li> <li>■ Endwert (Beispiel: 2003-02&lt;)</li> </ul> <p>Zulässige spezielle Datumsangaben:</p>	
	<i>TODAY</i> (+/- <i>nnnn</i> )	Alle Objekte mit aktuellem Tagesdatum.



		<p>Im Anschluß an den Tag kann +<i>nnnn</i> oder -<i>nnnn</i> angegeben werden. Der Wert <i>nnnn</i> darf maximal 4 Zeichen lang sein.</p> <p>Das resultierende Datum wird berechnet aus dem aktuellen Tagesdatum plus oder minus <i>nnnn</i> Tage.</p> <p>Kann mit der Startwert-Option (&gt;) oder der Endwert-Option (&lt;) kombiniert werden, z.B. listet T0-1&gt; alle Objekte, die in den letzten zwei Tage gespeichert oder katalogisiert wurden.</p>
	<u>Y</u> ESTERDAY	Alle Objekte mit dem Datum des Tages vor dem aktuellen Tag.
	<u>M</u> ONTH	Alle Objekte, deren Datum im Bereich des aktuellen Monats liegt.
	<u>Y</u> EAR	Alle Objekte, deren Datum im Bereich des aktuellen Jahres liegt.
<i>time-range</i>	<p>Listet alle Objekte mit einem Speicher- oder Katalogisierungszeitpunkt, der im angegebenen Zeitbereich liegt. Geben Sie einen einzelnen Zeitwert oder einen Zeitbereich an.</p> <p>Gültiges Zeitformat: <i>HH:II:SS</i> (<i>HH</i> = Stunden, <i>II</i> = Minuten, <i>SS</i> = Sekunden)</p> <p>Gültige Zeitbereiche:</p> <ul style="list-style-type: none"> <li>■ Vorangestellte Zeichen (Beispiel: 10:*)</li> <li>■ Startwert (Beispiel: 10:30&gt;)</li> <li>■ Endwert (Beispiel: 11:20&lt;)</li> </ul>	
<i>code-page-range</i>	Sie können eine einzelne Codepage oder einen Bereich von Codepages angeben, siehe <i>range-notation</i> .	
<i>arch-level-range</i>	<p>Gilt nur, wenn der Natural Optimizer Compiler installiert ist und wenn die Option ARCH gesetzt ist, um die IBM-Architekturebenen zu liefern (siehe <i>Natural Optimizer Compiler</i>-Dokumentation).</p> <p>Listet alle Natural-Objekte auf, die unter dem Natural Optimizer Compiler bei gesetzter Option ARCH katalogisiert wurden.</p> <p>Sie können, so wie unten aufgeführt, eine einzelne Ebene oder einen Bereich von Ebenen angeben. Dabei ist <i>n</i> eine ein- oder zweistellige Ebenennummer.</p>	
	<i>n</i> *	Listet alle Objekte mit Ebenen von <i>n</i> bis 9 auf.
	<i>n</i> >	Listet alle Objekte mit Ebenen auf, die gleich oder größer als <i>n</i> sind.
	<i>n</i> <	Listet alle Objekte mit Ebenen auf, die kleiner als <i>n</i> sind.
	<i>nn</i>	Listet alle Objekte mit Ebenen auf, die gleich <i>nn</i> sind.

	<i>nn</i> *	Listet alle Objekte mit Ebenen auf, die gleich <i>nn</i> sind.
	<i>nn</i> >	Listet alle Objekte mit Ebenen auf, die gleich oder größer als <i>nn</i> sind.
	<i>nn</i> <	Listet alle Objekte mit Ebenen auf, die gleich oder kleiner als <i>nn</i> sind.

### *range-notation*

- Für eine Liste aller Objekte in der aktuellen Library geben Sie einen Stern (\*) an.
- Für eine Liste eines bestimmten Bereichs von Objekten können Sie Stern-Notation (\*) und Wildcard-Notation (?) verwenden:
  - Stern-Notation ist die Möglichkeit, einen Stern (\*) anzugeben: der Stern steht für eine beliebige Zeichenkette beliebiger Länge.
  - Wildcard-Notation ist die Möglichkeit, ein Fragezeichen (?) anzugeben: das Fragezeichen steht für ein beliebiges Zeichen.
- Sie können eine oder mehrere Stern-Notationen (\*) und Wildcard-Notationen (?) miteinander kombinieren.
- Für eine vollständige Liste von Objekten ab einen bestimmten Startwert bzw. bis zu einem bestimmten Endwert können Sie die Größer-Zeichen-Notation (>) bzw. die Kleiner-Zeichen-Notation (<) verwenden.
- Die Größer-Zeichen-Notation (>) und die Kleiner-Zeichen-Notation (<) können nicht miteinander oder mit Stern-Notation (\*) oder Wildcard-Notation (?) kombiniert werden. Sie können nur zum Anzeigen einer Liste von Objekten verwendet werden (siehe [Liste von Objekten anzeigen](#)).

### Einstellungen (Settings)

SET	[	{	<a href="#">REUSE-LAST-LIST=YNvalue</a>	}	]
			<a href="#">RLL=YNvalue</a>		
	[	{	<a href="#">COUNT-SOURCE-LINES=YNvalue</a>	}	]
			<a href="#">CNTS=YNvalue</a>		
	[	{	<a href="#">SORT-TEXT-MEMBER-NAME=A8value</a>	}	]
			<a href="#">STMNA=A8value</a>		
	[	{	<a href="#">SORT-TEXT-MEMBER-LIBRARY=A8value</a>	}	]
			<a href="#">STMLI=A8value</a>		
	[	{	<a href="#">DELETE-SORT-TEXT-MEMBER=YNvalue</a>	}	]
			<a href="#">DELST=YNvalue</a>		
	[	{	<a href="#">PRINT-PAGE-SIZE=N3value</a>	}	]
			<a href="#">PPS=N3value</a>		
			<a href="#">[MARK-LONG-LINES=YNvalue]</a>		

[	{	<a href="#">DATA-AREA-DEFINE-DATA-FORMAT=YNvalue</a>	}	]
		<a href="#">DADDF=YNvalue</a>		
[	{	<a href="#">SOURCE-LIST-WITH-DBID-FNR=YNvalue</a>	}	]
		<a href="#">SLDF=YNvalue</a>		
[	{	<a href="#">PAGE-TITLE-IN-BATCH=AFvalue</a>	}	]
		<a href="#">PTIB=AFvalue</a>		
[	{	<a href="#">OUTPUT-DESTINATION-IN-BATCH=DPTWvalue</a>	}	]
		<a href="#">ODIB=DPTWvalue</a>		
[	{	<a href="#">OUTPUT-TEXT-NAME=A8value</a>	}	]
		<a href="#">OTN=A8value</a>		
[	{	<a href="#">OUTPUT-TEXT-LIBRARY=A8value</a>	}	]
		<a href="#">OTL=A8value</a>		
[	{	<a href="#">TOTAL-LINES-DISPLAY-BATCH=YNvalue</a>	}	]
		<a href="#">TLDB=YNvalue</a>		
<a href="#">[USEMAINPR=YNvalue]</a>				

Dabei ist:

*YNvalue* = Y oder N

*N3value* = numerischer Wert, 3-Byte

*A8value* = alphanumerischer Wert, 8-Byte

*AFvalue* = A oder F

*DPTWvalue* = D, P, T oder W



**Anmerkung:** Beschreibung der Optionen siehe [List-Profilparameter](#).

## Inhalt des Arbeitsbereichs auflisten

LIST	Wenn Sie nur das LIST-Kommando (ohne Parameter) angeben, wird der Inhalt des Editor-Arbeitsbereichs aufgelistet.
------	--

## Sourcecode eines einzelnen Objekts anzeigen

LIST <i>object-name</i> [ <i>options</i> ]	In beiden Fällen wird der Sourcecode des angegebenen Objekts gelistet.
LIST <i>object-type object-name</i> [ <i>options</i> ]	

Wenn Sie einen einzelnen *object-name* mit dem LIST-Kommando eingeben, brauchen Sie keinen *object-type* anzugeben; es wird der Sourcecode des angegebenen Objekts gelistet.

Wenn Sie einen *object-type* angeben, müssen Sie auch einen *object-name* angeben.

## Anzahl und Größe von Objekten anzeigen

LIST COUNT [ <i>range-clause</i> ] [ <i>settings</i> ]	<p>Zeigt einen zusammenfassenden Bericht, der die Anzahl und die Größe (in Bytes oder in KB, falls größer als 1 MB) der Objekte enthält, die in der aktuellen Library gespeichert sind.</p> <p>Die aufgelisteten Angaben zu Anzahl und Größe beziehen sich auf alle Objekte, die als Sourcecode oder nur als katalogisierte Objekte gespeichert worden sind, sowie auf alle Objekte, zu denen gemäß den Auswahlkriterien in der <i>range-clause</i> sowohl gespeicherte als auch katalogisierte Objekte existieren.</p> <p>Siehe auch <a href="#">Beispiele zur Benutzung der Objekt-Liste</a>.</p>
LIST COUNT <i>object-name-range</i> [ <i>range-clause</i> ] [ <i>settings</i> ]	<p>Zeigt einen erweiterten Bericht, in dem die Angaben zu Anzahl und Größe der gespeicherten/katalogisierten Objekte (so wie oben) nach gefundenen Objekttyp(en) aufgelistet werden, und zwar gemäß den Auswahlkriterien in <i>object-name-range</i> und <i>range-clause</i>.</p> <p>Objekttyp unknown (unbekannt) bedeutet, dass für diese Objekte keine Verzeichnisinformationen existieren.</p> <p>Siehe auch <a href="#">Beispiele zur Benutzung der Objekt-Liste</a>.</p>

## Sourcecode mehrerer Objekte nacheinander anzeigen

LIST SEQUENTIAL <i>object-name-range</i> [ <i>options</i> ]	In beiden Fällen müssen Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den <i>object-name-range</i> benutzen. Dann werden nacheinander die Sourcecodes aller Objekte angezeigt, die die angegebenen Auswahlkriterien erfüllen.
LIST SEQUENTIAL <i>object-type object-name-range</i> [ <i>options</i> ]	

## Liste von Objekten anzeigen

LIST <i>object-name-range</i>	In beiden Fällen müssen Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den <i>object-name-range</i> benutzen. Sie erhalten eine Liste aller Objekte, die die angegebenen Auswahlkriterien erfüllen. In der Liste können Sie dann Objekte zur Anzeige auswählen, indem Sie sie mit dem Funktionscode LI markieren (siehe <a href="#">Eine Funktion auf einem Objekt ausführen</a> ).
LIST <i>object-type object-name-range</i>	

## Vorsortierte Liste von ausgewählten Objekten anzeigen

LIST <i>object-name-range</i>	In beiden Fällen müssen Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den <i>object-name-range</i> benutzen. Sie erhalten eine Liste aller Objekte, die die angegebenen Auswahlkriterien erfüllen. In der Liste können Sie dann Objekte zur Anzeige auswählen, indem Sie sie mit dem Funktionscode LI markieren (siehe <a href="#">Eine Funktion auf einem Objekt ausführen</a> ).  Mit der <i>range-clause</i> können Sie zusätzliche Auswahl- und Sortierkriterien angeben, siehe auch das <a href="#">Beispiel</a> weiter unten.
LIST <i>object-name-range range-clause</i>	

## Langnamen katalogisierter Subroutinen und Klassen anzeigen

LIST EXTENDED <i>object-name-range</i>	Listet die Langnamen der katalogisierten Subroutinen, Klassen und Functions. Bezüglich der Namensoptionen siehe auch <i>object-name-range</i> weiter oben.
LIST EXTENDED <i>extended-type object-name-range</i>	

## NOC-Optionen katalogisierter Objekte anzeigen

LIST NOCOPT [ <i>object-type</i> ] <i>object-name-range</i>	Listet die katalogisierten Objekte, die mit dem Natural Optimizer Compiler (NOC) kompiliert wurden, und zeigt außerdem die zu Beginn beim CATALOG verwendeten NOC-Optionen an. Bezüglich der Namensoptionen siehe auch <i>object-name-range</i> weiter oben.
--	--

## Compiler-Optionen katalogisierter Objekte anzeigen

LIST OPTIONS [ <i>object-type</i> ] <i>object-name-range</i>	<p>Listet die katalogisierten Objekte auf und zeigt die beim CATALOG verwendeten Compiler-Optionen. Bezüglich der Namensoptionen siehe auch <i>object-name-range</i> weiter oben.</p> <p>Standardmäßig werden die endgültig verwendeten Compiler-Optionen angezeigt, das heißt, die Optionseinstellungen, die am Ende des CATALOG aktiv waren. Bei Objekten, die mit Natural Version 4.2.5 oder höher katalogisiert wurden, können auch die zu Beginn gesetzten Compiler-Optionen (das heißt, die Optionseinstellungen, die am Anfang des CATALOG aktiv waren) oder die geänderten Compiler-Optionen (das heißt, die Optionseinstellungen, die innerhalb des Sourcecode geändert wurden) angezeigt werden. Bezüglich der Bereichsfelder auf der Online-Maske siehe den entsprechenden Hilfe-Bildschirm.</p>
--	---

## Compiler Output zu einem katalogisierten Objekt anzeigen

Das Kommando LIST COMPOUT zeigt Compiler-Informationen zu einem katalogisierten Objekt in einem Report, der am Bildschirm angezeigt oder in Workfile 1 oder in ein Natural-Source-Objekt des Typs TEXT geschrieben werden kann. Wenn keine zusätzlichen Optionn angegeben werden oder wenn erforderliche Optionen weggelassen werden, erscheint eine Maske, in der die gewünschten Optionen und das Ausgabeziel angegeben werden können.

Kommando-Syntax:



Syntax-Element	Beschreibung
TEXT	Den Report in eine Natural-Source des Typs TEXT schreiben.
<i>text-name</i>	Name der Natural-Source des Typs TEXT. Der Standardwert ist REPCOUT.
<u>LIBRARY</u> <i>library-name</i>	Library-Name für die Natural-Source des Typs TEXT. Die Standard-Library ist die aktuelle Library.

Siehe auch [Beispiele zur Benutzung der Objekt-Liste](#).

## Directory-Informationen anzeigen

LIST DIRECTORY	<p>Zeigt die Verzeichnisinformationen zu dem gerade im Editor-Arbeitsbereich befindlichen Objekt an:</p> <ul style="list-style-type: none"> <li>■ <b>Sourcecode:</b> Datum und Uhrzeit, wann das Objekt katalogisiert worden ist, Library-Name, Benutzerkennung (User-ID), Programmiermodus (Reporting oder Structured Mode), TP-System, Terminalkennung (Terminal-ID), Betriebssystem, Transaktionscode, Natural-Version, Codepage-Informationen (falls verfügbar), Source-Größe</li> <li>■ <b>Objektcode:</b> Datum und -Uhrzeit, wann das Source-Objekt katalogisiert worden ist, Library-Name, Benutzerkennung (User-ID), Programmiermodus (Reporting oder Structured Mode), TP-System, Transaktionscode, Natural-Version, Codepage-Informationen (falls verfügbar), Betriebssystem, Größe im DATSIZE-Puffer, Größe im Buffer Pool, Größe des NOC-Code (vom Natural Optimizer Compiler erzeugter Maschinencode), vom Objekt verwendete GDA, Datum und Uhrzeit, wann die benutzte GDA katalogisiert worden ist, GDA-Zeitstempel von Datum und Uhrzeit, wann die GDA katalogisiert worden ist, GDA-Zeitstempel von Datum und Uhrzeit, wann die Source katalogisiert worden ist, Anzahl der von dem Objekt benutzen Subroutinen, Natural Optimizer Compiler (NOC) Optionen, die mit dem OPT-Profilparameter gesetzt wurden, als das Source-Objekt katalogisiert wurde.</li> </ul>
LIST DIRECTORY <i>object-name</i>	Dieses Kommando zeigt die Verzeichnisinformationen zu dem betreffenden Objekt an (wie bei LIST DIRECTORY beschrieben).
LIST DIRECTORY <i>object-name-range</i>	Wenn Sie Stern-Notation (*) oder Wildcard-Notation (?) für <i>object-name-range</i> verwenden, werden nacheinander die Directory-Informationen der entsprechenden Objekte angezeigt.
LIST <i>object-name</i> WITH DIRECTORY	Dieses Kommando zeigt zunächst die Verzeichnisinformationen zu dem betreffenden Objekt an (wie bei LIST DIRECTORY beschrieben) und listet anschließend den Sourcecode des Objekts.



## DDMs (Views) anzeigen

LIST DDM	Dieses Kommando zeigt eine Liste aller DDMs an.
LIST DDM <i>ddm-name</i>	<p>Wenn Sie einen einzelnen DDM-Namen angeben, wird das angegebene DDM angezeigt.</p> <p>Für <i>ddm-name</i> können Sie einen einzelnen DDM-Namen (bis zu 32 Zeichen lang) oder wie für <i>object-name-range</i> einen Bereich angeben, um eine Liste eines bestimmten Bereichs von DDMs anzuzeigen.</p>



**Anmerkung:** Statt des Schlüsselworts DDM können Sie auch das Schlüsselwort VIEW (oder kurz V) verwenden.

## Optionen

Statt *options* können Sie eine der nachfolgend aufgeführten Optionen angeben.

<pre> { [[WITH] DIRECTORY] [NUMBERS OFF] [<i>expand-option</i>]   <i>formatted-option</i>   CONVERTED   UNCONVERTED }</pre>	
---	--

DIRECTORY	Dieses Kommando zeigt zunächst die Directory-Informationen zu dem betreffenden Objekt an (wie bei <i>Directory Informationen anzeigen</i> beschrieben) und listet anschließend den Sourcecode des Objekts.
NUMBERS OFF	Standardmäßig wird der Sourcecode eines Objekts mit Zeilennummern gelistet. Wünschen Sie eine Anzeige ohne Zeilennummern, verwenden Sie die Option NUMBERS OFF (Vgl. <i>Unterkommandos für eine angezeigte Source</i> ).
UNCONVERTED	<p>Standardmäßig wird der Sourcecode in der Session-Codepage angezeigt (siehe Systemvariable *CODEPAGE).</p> <p>Geben Sie diese Option an, um den Sourcecode in der zur Systemdatei gespeicherten Codepage anzuzeigen.</p>
CONVERTED	Wenn der Sourcecode in der zur Systemdatei gespeicherten Codepage angezeigt wird (mit der Option oder dem Kommando UNCONVERTED, geben Sie diese Option an, um den Sourcecode in der Session-Codepage anzuzeigen (siehe Systemvariable *CODEPAGE).

expand-option

```
EXPAND [ FORMATTED ] [ { COMMENTS } ] [ expand-type [ { object-name } ] ]  
...10]
```

EXPAND <a href="#">object-name</a>	Die EXPAND-Option erlaubt es Ihnen, die Sourcen von Objekten, die in der gelisteten Source referenziert werden (Copycodes, Data Areas, Maps, Helprouninen, externe Subroutinen, Subprogramme, mit einem FETCH-Statement aufgerufene Programme, Fehlermeldungen), <i>innerhalb</i> der gelisteten Source anzuzeigen. Dies kann besonders im Batch-Betrieb sehr hilfreich sein.
EXPAND <a href="#">object-name-range</a>	<p>Wenn Sie beispielsweise ein Source-Programm listen, das ein INCLUDE-Statement enthält, können Sie den Sourcecode des betreffenden Copycodes innerhalb des gelisteten Source-Programms unmittelbar nach dem INCLUDE-Statement anzeigen.</p> <p>In den folgenden Erklärungen werden Objekte, die innerhalb einer Source gelistet werden, als „Expand-Objekte“ bezeichnet.</p> <p><b>Unterkommandos zur EXPAND-Option</b></p> <p>Innerhalb eines gelisteten Expand-Objekts sind nur die folgenden Unterkommandos verfügbar (die oben beschrieben sind):</p> <pre>PRINT + - - .</pre> <p>Siehe <a href="#">Beispiele zur Benutzung der Objekt-Liste</a>.</p>
EXPAND FORMATTED	<p>Die Option EXPAND FORMATTED ist nur für innerhalb einer Source gelistete Data Areas (mit STOW gespeichert, d.h. bei denen der Zeitstempel des Sourceobjekts und des katalogisierten Objekts identisch ist) und Maps relevant.</p> <p>Für Data Areas gilt:</p> <ul style="list-style-type: none"><li>■ Wenn Sie FORMATTED nicht angeben, ähnelt die Anzeige der Data Area der im Data-Area-Editor.</li><li>■ Wenn Sie FORMATTED angeben, ähnelt die Anzeige der Data Area einem DEFINE DATA-Statement. Dies gilt nur bei mit einem STOW-Kommando gespeicherte Data Areas (d.h. bei denen der Zeitstempel des Sourceobjekts und des katalogisierten Objekts identisch ist), siehe auch Unterkommando <a href="#">FORMAT</a>.</li></ul> <p>Für Maps gilt:</p> <ul style="list-style-type: none"><li>■ Wenn Sie FORMATTED nicht angeben, wird die <i>Source</i> der Map gelistet.</li><li>■ Wenn Sie FORMATTED angeben, wird das <i>Layout</i> der Map gelistet (d.h. die Map, wie Sie ein Benutzer zur Laufzeit sieht).</li></ul>

EXPAND COMMENTS	Wenn Sie die Option EXPAND COMMENTS verwenden, werden nur die		
EXPAND <i>n</i>	<p>Kommentarzeilen am Anfang des Expand-Objekts gelistet; d.h. das Expand-Objekt wird bis zu (aber nicht einschließlich) der ersten Sourcecode-Zeile, die keine Kommentarzeile ist, gelistet.</p> <p>Wenn Sie die Option EXPAND <i>n</i> verwenden, werden nur die ersten <i>n</i> Zeilen des Expand-Objekts gelistet.</p> <p>Wenn Sie keine dieser beiden Optionen verwenden, wird das Expand-Objekt vollständig gelistet.</p>		
<i>expand-type</i>	Als <i>expand-type</i> geben Sie den Objekttyp des Expand-Objekts an. Sie können die folgenden <i>expand-types</i> angeben:		
	P	Programme	Falls Sie mehr als einen <i>expand-type</i> angeben möchten, geben Sie sie in beliebiger Reihenfolge und ohne Leerzeichen an; um Maps, Copycodes und Subroutinen innerhalb einer gelisteten Source zu listen, geben Sie beispielsweise MCS als <i>expand-type</i> an.
	N	Subprogramme	
	S	Externen Subroutinen	
	H	Helproutinen	
	G	Global Data Areas	
	L	Local Data Areas	
	A	Parameter Data Areas	
	M	Maps	
	C	Copycodes	
	E	Fehlermeldungen	
	4	Klassen	
	*	Alle Objekttypen	
<i>object-name</i>	Als <i>object-name</i> oder <i>object-name-range</i> geben Sie den Namen bzw. die Namen der Expand-Objekte an, die innerhalb der gelisteten Source gelistet werden soll(en).		
<i>object-name-range</i>			
<p>Für den <i>object-name</i> bzw. <i>object-name-range</i> eines Expand-Objekts stehen die gleichen Optionen zur Verfügung wie für den <i>object-name</i> bzw. <i>object-name-range</i> des gelisteten Hauptobjekts in der primären LIST-Kommandosyntax. Ausnahmen: die Größer-Zeichen-Notation (&gt;) und die Kleiner-Zeichen-Notation.</p>			

### formatted-option

FORMATTED	[ 'c' ]	[ 'c' ]	[ SETTINGS ]	[ { FIELDS EXTFIELDS } ]	[ [ RULES INLINERULES FREERULES AUTORULES ] ]
-----------	---------	---------	--------------	-----------------------------	--

### FORMATTED-Option

Die FORMATTED-Option gilt für Data Areas (mit STOW gespeichert, d.h. bei denen der Zeitstempel des Sourceobjekts und des katalogisierten Objekts identisch ist) und Maps:

FORMATTED	<p><b>Mit STOW-Kommando gespeicherte Data Area:</b></p> <p>Wenn Sie diese Option für eine Data Area angeben, wird die Data Area formatiert angezeigt, d.h. die Anzeige ähnelt einem <code>DEFINE DATA</code>-Statement (Vgl. Unterkommando <a href="#">FORMAT</a>). Standardmäßig ist die Anzeige von Data Areas unformatiert, d.h. die Anzeige ähnelt der im Data Area Editor.</p> <p>Dies gilt nur für mit <code>STOW</code>-Kommando gespeicherte Data Area (das heißt, der Zeitstempel des Source-Objekts und des katalogisierten Objekts sind identisch). Standardmäßig ist die Anzeige von Data Areas unformatiert, das heißt, die Anzeige ist ähnlich wie im Dateneditor.</p> <p>Sie können die Standardeinstellung im List-Profil ändern, siehe <a href="#">Individuelles List-Profil erstellen</a> und siehe auch Unterkommando <a href="#">FORMAT</a></p> <p><b>Map:</b></p> <p>Wenn Sie die <code>FORMATTED</code>-Option für eine Map angeben, wird das <i>Layout</i> der Map angezeigt, d.h. die Map, wie Sie ein Benutzer zur Laufzeit sieht.</p>
-----------	---

Weitere **FORMATTED**-Optionen für Maps

Wenn Sie Maps mit dem `LIST`-Kommando anzeigen, können Sie zusätzlich zum Schlüsselwort `FORMATTED` weitere Optionen angeben:

<code>['c'] ['c']</code>	<p><b>Füllzeichen benutzen:</b></p> <p>Sie können Füllzeichen <code>c</code> für Eingabefelder (<code>AD=A</code> und <code>AD=M</code>) und Ausgabefelder (<code>AD=O</code>) angeben, um diese Felder sichtbar zu machen. Sie können ein beliebiges Zeichen als Füllzeichen angeben.</p> <p>Das folgende Beispiel bewirkt, dass alle Eingabefelder mit einem Unterstrich (<code>_</code>) und alle Ausgabefelder mit einer Raute (<code>#</code>) gefüllt angezeigt werden.</p> <pre>LIST MAP map-name FORMATTED  '_'  '#'</pre>
SETTINGS	<p><b>Map-Einstellungen:</b></p> <p>Diese Option bewirkt, dass die <b>Map Settings</b> der Map angezeigt werden.</p> <pre>LIST MAP map-name FORMATTED SETTINGS</pre>

FIELDS	<b>Feldübersicht:</b> Diese Option bewirkt, dass das <b>Field Summary</b> , d.h. die Liste der Map-Felder, angezeigt wird.  LIST MAP <i>map-name</i> FORMATTED FIELDS
EXTFIELDS	<b>Erweiterte Feldeditier-Informationen:</b> Diese Option bewirkt, dass die <b>Extended Field Editing</b> -Informationen für alle Map-Felder angezeigt werden.  LIST MAP <i>map-name</i> FORMATTED EXTFIELDS

### Anzeige der Verarbeitungsregeln für eine Map

Die folgenden Optionen bewirken, dass die von der Map verwendeten **Processing Rules** angezeigt werden. Diese Verarbeitungsregeln werden in der Reihenfolge der Felder, denen sie zugewiesen sind, angezeigt und je Feld in der Reihenfolge des Rangs (Rank).

RULES	<b>Alle Verarbeitungsregeln anzeigen:</b>  LIST MAP <i>map-name</i> FORMATTED RULES  Zeigt <i>alle</i> Verarbeitungsregeln für die angegebene Map an.
INLINERULES	<b>Nur die Inline-Verarbeitungsregeln anzeigen:</b>  LIST MAP <i>map-name</i> FORMATTED INLINERULES  Zeigt nur die Inline-Verarbeitungsregeln für die angegebene Map an.
FREERULES	<b>Nur die freien Verarbeitungsregeln anzeigen:</b>  LIST MAP <i>map-name</i> FORMATTED FREERULES  Zeigt nur die freien Verarbeitungsregeln für die angegebene Map an.
AUTORULES	<b>Nur die automatischen Verarbeitungsregeln anzeigen:</b>  LIST MAP <i>map-name</i> FORMATTED AUTORULES  Zeigt nur die automatischen Verarbeitungsregeln für die angegebene Map an.

Siehe auch Unterkommandos [LAYOUT](#) und [FORMAT](#) im Abschnitt [Source-Liste](#).

## Objekt-Auswahlliste

Wenn Sie Stern- oder Wildcard-Notation für den *object-name* verwenden, erhalten Sie eine Liste aller Objekte, die Ihr Auswahlkriterium erfüllen. Auf der Liste können Sie dann die Objekte, die Sie gelistet, gedruckt usw. haben möchten, mit einem Funktionscode markieren oder Sie können ein Natural-Systemkommando oder ein LIST-Unterkommando in der Kommandozeile absetzen.

Dieser Abschnitt beschreibt die Funktionen, Unterkommandos und Funktionscodes, die in der angezeigten Liste der Objekte verfügbar sind, z.B. nachdem Sie ein LIST \*-Kommando abgesetzt haben.

- Bedeutung der Spaltenüberschriften
- In der Objekt-Auswahlliste blättern
- Neue Kriterien für die Auswahlliste angeben
- Informationen in der Auswahlliste
- Intensivierte dargestellte Informationen in der Auswahlliste
- Unterkommandos für die Auswahlliste
- Eine Funktion auf einem Objekt ausführen
- Inhalt der Auswahlliste sortieren
- Beispiele zur Benutzung der Objekt-Liste

### Bedeutung der Spaltenüberschriften

Die Liste der Objekte enthält folgende Spalten:

Spalte	Bedeutung
Cmd	In diese Spalte können Sie einen Code eingeben, um eine Funktion auf einem Objekt auszuführen. Siehe <i>Eine Funktion auf einem Objekt ausführen</i> .
Name	Name des Objekts.
Type	Objekttyp.
S/C	Zeigt an, ob das Objekt als Source (S) und/oder als katalogisiertes Objekt (C) vorhanden ist.
SM	Der Natural-Programmiermodus, der beim Erstellen des Objekts verwendet wurde.  S = Structured Mode  R = Reporting Mode
Version	Natural-Produktversion, die zum Erstellen bzw. Katalogisieren des Objekts verwendet wurde.
User ID	User-ID des Benutzers, der das Objekt erstellt bzw. katalogisiert hat.
Date, Time	Zeitpunkt (Datum, Uhrzeit), zu dem das Objekt erstellt bzw. katalogisiert wurde.

## In der Objekt-Auswahlliste blättern

Wenn Ihnen eine Auswahlliste von Objekten angezeigt wird, können Sie darin wie folgt blättern:

- Um in der Liste eine Seite vor bzw. zurückzublättern, drücken Sie PF8 bzw. PF7.
- Um an den Anfang bzw. ans Ende der Liste zu blättern, drücken Sie PF6 bzw. PF9.

## Neue Kriterien für die Auswahlliste angeben

Wenn Ihnen eine Auswahlliste von Objekten angezeigt wird, befinden sich unmittelbar unterhalb der Spaltenüberschriften Felder, in denen die Auswahlkriterien für die derzeitige Liste angezeigt werden. Sie können die Auswahlkriterien ändern, indem Sie die Werte dieser Felder überschreiben. Wenn Sie ein Fragezeichen (?) in eins dieser Felder eingeben, erhalten Sie Informationen über die möglichen Werte für dieses Feld.

## Informationen in der Auswahlliste

Falls von einem Objekt sowohl eine Source als auch ein Objektmodul existiert (wie in der Spalte S/C angezeigt), beziehen sich die angezeigten Informationen auf die Source, nicht das Objektmodul.



**Anmerkung:** Wenn die SORT-Funktion aktiv ist, können die Source und das Objektmodul getrennt angezeigt werden, d.h. wenn die Liste nach Objektdatum sortiert wird und die Source und das Objektmodul unterschiedliche Datumswerte aufweisen.

### ➤ Um weitere Informationen zu Source- und katalogisierten Objekten anzuzeigen

- Drücken Sie PF11, um nach rechts zu blättern.

Oder:

Drücken Sie PF10, um nach links zu blättern.



**Anmerkung:** Standardmäßig wird die Anzahl der Source-Zeilen bei Sourceobjekten aus Durchsatzgründen nicht berechnet. Wenn Sie wünschen, dass die die Anzahl der Source-Zeilen bei Sourceobjekten angezeigt wird, können Sie entweder das Unterkommando **COUNTSOURCE ON** eingeben oder im List-Profil (siehe [Individuelles List-Profil erstellen](#)) den Parameter **COUNT-SOURCE-LINES** auf Y setzen.

## Intensivierte dargestellte Informationen in der Auswahlliste

Ist eine Informationseinheit in der Liste intensiviert dargestellt, bedeutet dies, dass dort eine Abweichung zwischen der Source eines Objekts und dem entsprechenden Objektmodul besteht. Um Näheres über die Abweichung zu erfahren, können Sie das Objekt mit Funktionscode **LD** markieren, um sich die betreffende Directory-Information anzeigen zu lassen. Zur Beseitigung der Abweichung genügt es in der Regel, das Objekt neu mit **STOW** zu speichern (Funktionscode **ST**; siehe unten).

## Unterkommandos für die Auswahlliste

In einer Objekt-Auswahlliste können Sie ein Natural-Systemkommando oder ein **LIST**-Unterkommando in der Kommandozeile eingeben. Gültige Unterkommandos sind:

Code	Funktion	
SC	Nur die Objekte werden aufgelistet, die einen Scan-Wert enthalten (kann nur bei langen Listen angewendet werden).	
SC OFF	Scan-Modus ausschalten.	
SHORT	Liste der Objekte in Kurzform anzeigen, d.h. es werden nur die Objektnamen angezeigt (kann nur bei SC OFF verwendet werden).	
LONG	Liste der Objekte in Langform (mit allen vorhandenen Feldern) anzeigen.	
PRINT	Liste der Objekte drucken.	
EXTENDED	Langnamen von Subroutinen/Klassen anzeigen; entspricht der Funktion <b>LIST EXTENDED *</b> .	
ALL $fx$	Den Funktionscode $fx$ für alle angezeigten Objekte eingeben. $fx$ = gültiger Funktionscode für ein gelistetes Objekt.	
SORT	Sort-Fenster anzeigen (siehe <b>Liste der Objekte sortieren</b> ).	
COUNTSOURCE	ON	Anzahl der Source-Zeilen für Sourceobjekte wird angezeigt.
	OFF	Anzahl der Source-Zeilen für Sourceobjekte wird nicht angezeigt.
MARK-LONG-LINES	ON	Lange Zeilen in der Auflistung eines Sourceobjekts werden in den ersten zwei Stellen mit einem L markiert.  Der Standardwert kann im List-Profil angegeben werden; siehe <b>Individuelles List-Profil erstellen</b> .
	OFF	Lange Zeilen in der Auflistung eines Sourceobjekts werden nicht markiert.
DEFINE-DATA	ON	Die aufgelistete Data-Area-Source wird standardmäßig im DEFINE DATA-Format angezeigt (wie bei LIST <i>dataarea</i> FORMATTED).



Code	Funktion	
		Der Standardwert kann im List-Profil angegeben werden; siehe <i>Individuelles List-Profil erstellen</i> .
	OFF	Die Anzeige einer Data Area Source erfolgt unformatiert.
<u>LISTCOUNT</u>	Liste mit Anzahl der Objekte und Angaben zur Objektgröße entsprechend den zurzeit benutzten Auswahlkriterien.	
<u>LISTPROFILE</u>	Aktuelle Werte der Parameter des List-Profiles anzeigen (siehe <i>Individuelles List-Profil erstellen</i> ).	
<u>NOCOPT</u>	Listet die katalogisierten Objekte, die mit dem Natural Optimizer Compiler (NOC) kompiliert wurden, und zeigt außerdem die zu Beginn beim <b>CATALOG</b> verwendeten NOC-Optionen an; entspricht <code>LIST NOCOPT *</code> ; siehe <i>NOC-Optionen katalogisierter Objekte anzeigen</i> .	
<u>OPTIONS</u>	Listet die katalogisierten Objekte auf und zeigt die beim <b>CATALOG</b> zu Beginn verwendeten Compiler-Optionen; entspricht <code>LIST OPTIONS *</code> ; siehe <i>Compiler-Optionen katalogisierter Objekte anzeigen</i> .	
<u>REUSE</u>	ON	<p>Wiederverwendungsmodus einschalten.</p> <p>Die zuletzt angezeigte Liste wird erneut verwendet, nachdem in der Spalte <b>Cmd</b> Kommandos ausgeführt wurden. Ausgenommen sind folgende Kommandos:</p> <p>E ED (Edit) CA (Catalog) UC (Uncat) S ST (Stow) D DE (Delete) RE (Rename)</p>
	OFF	<p>Wiederverwendungsmodus ausschalten.</p> <p>Nach der Ausführung von Kommandos, die in der Spalte <b>Cmd</b> eingegeben wurden, wird die Liste neu erstellt.</p>
<u>REFRESH</u>	<p>Die gerade angezeigte Liste aktualisieren.</p> <p>Dieses Unterkommando können Sie insbesondere bei eingeschaltetem Wiederverwendungsmodus benutzen.</p>	
+	Eine Seite vor blättern.	
-	Eine Seite zurück blättern.	
++	Ans Ende der Objektliste blättern.	
--	An den Anfang der Objektliste blättern.	
?	Hilfe zur Kommandozeile aufrufen.	

## Eine Funktion auf einem Objekt ausführen

Um ein Objekt von der Auswahlliste für eine bestimmte Funktion auszuwählen, markieren Sie es einfach in der linken Spalte (unter der Überschrift **Cmd**) mit dem entsprechenden Funktionscode.

Sie können mehrere Objekte in der Auswahlliste mit unterschiedlichen Funktionscodes markieren. Die Funktionen werden dann nacheinander ausgeführt.

Die folgenden Funktionscodes stehen Ihnen zur Verfügung (mögliche Abkürzungen sind unterstrichen).

Funktionscode	Funktion
?	Ein Fenster zeigt Ihnen, welche Funktionen für das markierte Objekt zur Verfügung stehen. Es werden jeweils nur die Funktionen aufgeführt, die für das betreffende Objekt auch tatsächlich möglich sind (zum Beispiel wird Ihnen für eine Subroutine nicht die Funktion RU(N) angeboten und für ein Objekt, das nur in Sourceform existiert, nicht die Funktion EX(ECUTE)).  In dem Fenster können Sie für das betreffende Objekt die gewünschte Funktion auswählen.
CA	Kompilieren und speichern in Objektform (entspricht dem Systemkommando CATALOG).
<u>DE</u>	Löschen eines Objekts (entspricht dem Systemkommando DELETE).
DL	Herunterladen (Download) des Objekts vom Großrechner auf einen Personal Computer (nur möglich, wenn Natural Connection installiert ist).
<u>ED</u>	Editieren des Sourcecodes (entspricht dem Systemkommando EDIT).
EX	Ausführen des Objektmoduls (entspricht dem Systemkommando EXECUTE).
LC	Anzeigen des in die Default-Codepage *CODEPAGE umgesetzten Sourcecodes des Objekts (entspricht dem Systemkommando LIST object-name CONVERTED).
LD	Anzeigen der Directory-Information zum Objekt (entspricht dem Systemkommando LIST DIRECTORY object-name).
LE	Anzeigen des Sourcecodes des Objekts in expandierter Form (entspricht LIST object-name EXPAND *).
LF	Formatierte Anzeige von Data Area oder Map (entspricht dem Systemkommando LIST object-name FORMATTED).
<u>LI</u>	Anzeigen (List) des Sourcecodes.
LN	Anzeigen des Langnamens der Subroutine oder der Klasse (nur möglich, wenn ein katalogisiertes Objekt vorhanden ist) oder der Resource.
LO	Aufrufen der Funktion LIST COMPOUT für das katalogisierte Objekt (gleichbedeutend mit dem Kommando LIST COMPOUT object-name); siehe <i>Compiler Output zu einem katalogisierten Objekt anzeigen</i> .
LU	Anzeigen des Sourcecodes des Objekts, konvertiert in die zur Systemdatei gespeicherte Codepage (gleichbedeutend mit LIST object-name UNCONVERTED).
NO	Zeigt die während des Katalogisierens mit CATALOG verwendeten Natural Optimizer (NOC) Optionen (funktioniert nur, wenn ein katalogisiertes Objekt existiert).

Funktionscode	Funktion
QP	Zeigt die zu Beginn, die am Ende verwendeten und die geänderten Compiler-Optionen, die beim Katalogisieren mit <b>CATALOG</b> verwendet wurden (funktioniert nur, wenn ein katalogisiertes Objekt existiert).  Die zu Beginn verwendeten und die geänderten Compiler-Optionen können nur bei Objekten angezeigt werden, die mit Natural Version 4.2.5 oder höher katalogisiert wurden.
PR	Drucken (Print) des Sourcecodes.
RE	Umbenennen des Objekts (entspricht dem Systemkommando <b>RENAME</b> ).
RU	Kompilieren und Ausführen der Source (entspricht dem Systemkommando <b>RUN</b> ).
ST	Speichern in Source- und Objektform (entspricht dem Systemkommando <b>STOW</b> ).
UC	Löschen des Objektmoduls (entspricht dem Systemkommando <b>UNCATALOG</b> ).
.	Ende (Verlassen des Auswahllistenfensters).

### Inhalt der Auswahlliste sortieren

Das LIST-Kommando bietet Ihnen die Möglichkeit, die Liste der angezeigten Objekte nach mehreren Sortierkriterien zu sortieren.



**Anmerkung:** Damit Sie diese Funktion benutzen können, muss die Größe des vom Sortierprogramm benutzten Work Buffers mit dem Schlüsselwortparameter **WRKSIZE** des Natural-Profilparameters **SORT** auf einen ausreichenden Wert gesetzt werden. Die maximale Größe der Liste, die sortiert werden kann, wird durch die Größe des Work Buffers begrenzt.

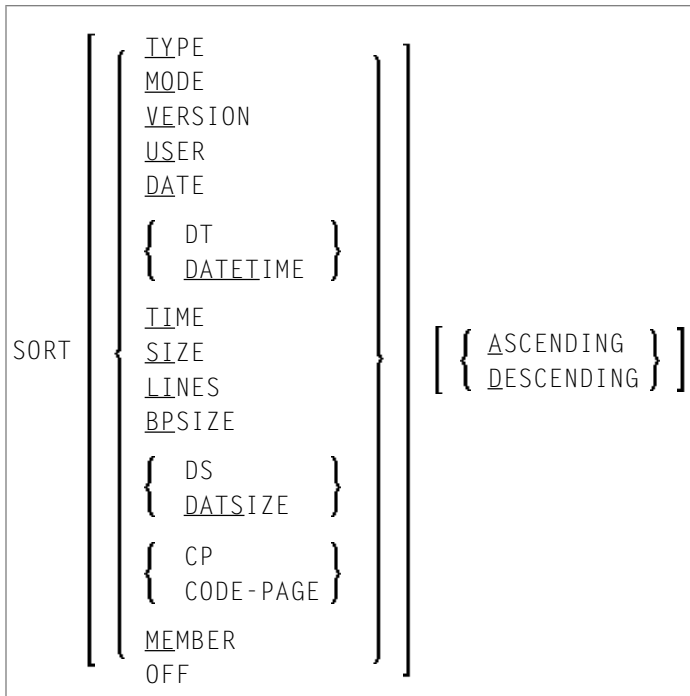
#### ➤ Um die Sort-Funktion aufzurufen

- Drücken Sie PF4.

Oder:

Setzen Sie in der Liste der Objekte ein SORT-Unterkommando ab.

### Syntax des Unterkommandos SORT



Wenn Sie PF4 drücken, können Sie in einem Fenster angeben, ob Sie die Liste sortieren wollen (Y/N), ein Sort-Feld markieren und in auf- oder absteigender Reihenfolge sortieren wollen.

Sort-Feld	Schlüsselwort in Sort-Syntax
Natural object type	TYPE
Programming mode (Reporting oder Structured mode)	MODE
Version	VERSION
User ID	USER
Date	DATE
Date and time	DATETIME
Time	TIME
Source size	SIZE
Number of source lines	LINES
Buffer pool size	BPSIZE
DATSIZE (Größe des lokalen Daten-Buffers)	DS/DATSIZE
Code page	CP/CODE - PAGE
Member names of subroutines or classes (nur in der erweiterten Auswahlliste verfügbar)	MEMBER

Nachdem eine Liste sortiert und angezeigt wurde, bewirkt eine Änderung in den Auswahlkriterien, dass eine neue Liste aus allen Objekten der Library entsprechend den geänderten Auswahlkriterien erzeugt wird, die dann nach den Auswahlkriterien sortiert wird.

## ➤ Um die Sortierfunktion auszuschalten

- Geben Sie das Unterkommando `SORT OFF` ein.

Oder:

Deaktivieren Sie die Sortierfunktion in dem mit `PF4` aufgerufenen Fenster.

Die sortierte Liste wird in einem Natural-Textobjekt in der Library `WORKPLAN` erstellt. Der Name des Objekts wird durch das `LIST`-Kommando erzeugt. Wenn Sie das List-Profil aufrufen (siehe [Individuelles List-Profil erstellen](#)), können Sie den Namen für das Textobjekt sowie die Library angeben.

## Beispiele zur Benutzung der Objekt-Liste

<code>LIST *</code>	Listet alle Objekte jeglichen Typs in der aktuellen Library.
<code>LIST S *</code>	Listet alle Subroutinen in der aktuellen Library.
<code>LIST SYS*</code>	Listet alle Objekte (egal welchen Typs), deren Namen mit <code>SYS</code> anfangen.
<code>LIST M SYS*</code>	Listet alle Maps, deren Namen mit <code>SYS</code> anfangen.
<code>LIST C *CODE</code>	Listet alle Copycodes, deren Namen mit <code>CODE</code> aufhören.
<code>LIST NAT*AL</code>	Listet alle Objekte, deren Namen mit <code>NAT</code> anfangen und mit <code>AL</code> aufhören, ganz gleich wieviele andere Zeichen zwischen <code>NAT</code> und <code>AL</code> stehen (dies würde die Namen <code>NATURAL</code> und <code>NATIONAL</code> sowie <code>NATAL</code> einschließen).
<code>LIST D00?</code>	Listet alle Objekte mit 4-stelligen Namen, die mit <code>D00</code> anfangen (dies würde die Namen <code>DOOR</code> und <code>DOOM</code> einschließen, aber nicht <code>D00</code> oder <code>DOODLE</code> ).
<code>LIST M NAT?AL</code>	Listet alle Maps, deren Namen mit <code>NAT</code> anfangen und mit <code>AL</code> aufhören, und zwar mit genau einem Zeichen zwischen <code>NAT</code> und <code>AL</code> (dies würde die Namen <code>NAT1AL</code> und <code>NAT2AL</code> einschließen, aber nicht <code>NATAL</code> oder <code>NATIONAL</code> ).
<code>LIST M *1*</code>	Listet alle Maps, deren Namen eine <code>1</code> enthalten.
<code>LIST M F&gt;</code>	Listet alle Maps, und zwar ab der ersten, deren Name mit <code>F</code> anfängt.
<code>LIST M MA&lt;</code>	Listet alle Maps, und zwar ab der ersten und bis zu der namens <code>MA</code> (falls vorhanden).
<code>LIST N?T*AL</code>	Listet Objekte wie zum Beispiel <code>NATAL</code> , <code>NATURAL</code> , <code>NATvrAL</code> (wobei <code>vr</code> für die Versions- und Release-Nummer steht).
<code>LIST E* TYPE=PM KIND=S DATE=YEAR SORTED=DATE ASCENDING</code>	Erstellt eine Liste aller Sourceobjekte vom Programmen und Maps, deren Namen mit <code>E</code> anfangen und die im aktuellen Jahr gespeichert wurden. Die Liste ist in aufsteigender Reihenfolge nach dem Objektdatum sortiert.
<code>LIST COUNT</code>	Listet die Gesamtangaben zu Anzahl und Größe aller Objekte in der aktuellen Library, für die gespeicherte und/oder katalogisierte Objekte existieren.

LIST COUNT * TYPE=PM KIND=S	Listet die Angaben zu Anzahl und Größe aller Programme und Maps (Masken), für die gespeicherte Objekte existieren.
LIST COMPOUT HUGO REPORT UNV EXTS WORKF	Schreibt einen Report des katalogisierten Objekts HUGO mit allen nicht benutzten Variablen und allen benutzten externen Subroutinen in die Arbeitsdatei 1.

## Source-Liste

- Unterkommandos für eine angezeigte Source
- Unterkommandos FIND, REF und SCAN
- Unterkommando FORMAT
- Cursor-sensitive Objektauswahl

### Unterkommandos für eine angezeigte Source

Wenn Ihnen der Sourcecode eines Objekts angezeigt wird, können Sie in der Kommandozeile eines der folgenden Kommandos eingeben.



**Anmerkung:** Standardmäßig werden die Datenbankkennung (DBID) und die Dateinummer (FNR) der Source-Library nicht in der Kopfzeile der aufgelisteten Source angezeigt. Wenn Sie wollen, dass die DBID und FNR der Source-Library angezeigt werden, können Sie entweder das Unterkommando **DBFNR ON** eingeben oder im List-Profil den Parameter **SOURCE-LIST-WITH-DBID-FNR** auf Y setzen (siehe *Individuelles List-Profil erstellen*).

Unterkommando	Funktion
+	Blättert ein Seite vor.
-	Blättert eine Seite zurück.
+H	Blättert ein halbe Seite vor.
-H	Blättert eine halbe Seite zurück.
++	Blättert ans Ende der Source.
BOTTOM	
--	Blättert an den Anfang der Source.
TOP	
+n	Blättert n Zeilen vor.
-n	Blättert n Zeilen zurück.
nnnn	Blättert zu Zeile Nummer nnnn.
UNCONVERTED	Siehe UNCONVERTED bei Optionen.
CONVERTED	Siehe CONVERTED bei Optionen.
DBFNR ON	Zeigt die DBID und FNR der Source-Library in der Kopfzeile des Sourcecodes.

Unterkommando	Funktion
DBFNR OFF	Zeigt die Kopfzeile des Sourcecodes ohne die DBID und FNR der Source-Library. Dies ist der Standardwert.
EXPAND	Siehe <i>expand-option</i> .
FIELDS	Gilt nur für Maps: Zeigt das <b>Field Summary</b> , d.h. die Liste der Felder in der Maske.
FIND	Zeigt nur die Sourcecode-Zeilen, in denen der angegebene Wert ( <i>value</i> ) vorkommt. Weitere Informationen siehe <i>Unterkommandos FIND, REF und SCAN</i> .
FIND <i>value</i>	
FIND ABSOLUTE <i>value</i>	
FIND ABSOLUTE <i>value</i>	
FIND CASESENSITIVE <i>value</i>	
FORMAT	Gilt nur für Data Areas und Maps: zeigt die <i>formatierte</i> Data Area bzw. Map, sowie zur Map gehörige Informationen.
LAYOUT	Gilt nur für Maps. Zeigt das Layout der Map, d.h. die Map, wie Sie ein Benutzer zur Laufzeit sieht.
NUMBERS ON	Zeigt die Source mit Sourcecode-Zeilennummern. Dies ist der Standardwert.
NUMBERS OFF	Zeigt die Source ohne Sourcecode-Zeilennummern.
PRINT	Druckt die Source.
REF	Zeigt die Zeilennummer der Sourcecode-Zeilen, in denen der angegebene Wert ( <i>value</i> ) vorkommt.
REF <i>value</i>	
REF ABSOLUTE <i>value</i>	
	<p>Wenn Sie nur das Kommando REF eingeben, erscheint ein Fenster, indem Sie den gewünschten Wert eingeben sowie bestimmen können, ob die Suche absolut sein soll oder nicht.</p> <p>Wenn Sie REF ohne ABSOLUTE eingeben, dann wird der Wert nur gefunden, wenn er als isoliertes Wort vorkommt. Dies ist der Standardwert.</p> <p>Wenn Sie REF mit ABSOLUTE eingeben, ist die Suche absolut, d.h. der Wert wird auch gefunden, wenn er als Teil einer längeren Zeichenkette vorkommt.</p>
RULES	Gilt nur für Maps. Zeigt die von der Map verwendeten Verarbeitungsregeln (Processing Rules). Die Rules werden in der Reihenfolge der Felder, denen sie zugewiesen sind, gezeigt und je Feld in der Reihenfolge des Rangs (Rank).
SCAN	Zeigt alle Zeilen intensiviert an, in denen der angegebene Wert ( <i>value</i> ) vorkommt. Die erste Zeile, in der der Wert vorkommt, wird an den Anfang der Seite geblättert.
SCAN <i>value</i>	
SCAN ABSOLUTE <i>value</i>	
	Wenn Sie nur das Kommando SCAN eingeben, erscheint ein Fenster, in dem Sie den gewünschten Wert eingeben sowie bestimmen können, ob die Suche absolut sein soll oder nicht.

Unterkommando	Funktion
	<p>Wenn Sie SCAN ohne ABSOLUTE eingeben, dann wird der Wert nur gefunden, wenn er als isoliertes Wort vorkommt. Dies ist der Standardwert.</p> <p>Wenn Sie SCAN mit ABSOLUTE eingeben, ist die Suche absolut, d.h. der Wert wird auch gefunden, wenn er als Teil einer längeren Zeichenkette vorkommt.</p>
SCAN= or SC=	Führt das letzte SCAN-Kommando erneut aus. Alternativ können Sie PF5 drücken.
SETTINGS	Gilt nur für Maps. Zeigt die Map Settings der Map.
ZOOM [ <i>expand-type</i> ...10] <i>object-name</i>	<p>Die Angabe eines einzelnen Namens (<i>object-name</i>) mit dem ZOOM-Kommando hat die gleiche Auswirkung wie das Markieren des Namens mit dem Cursor in der gelisteten Source (siehe <a href="#">Cursor-sensitive Objektauswahl</a>). Das ausgewählte Objekt wird in einem Fenster angezeigt.</p> <p>Wenn Sie Stern-Notation (*) bzw. Wildcard-Notation (?) im <i>object-name</i> verwenden, werden alle ausgewählten Objekte in dem Fenster gezeigt, und zwar in der Reihenfolge, in der sie in der gelisteten Source referenziert werden.</p> <p>Die Angabe eines <i>expand-type</i> entspricht der <i>expand-option</i>.</p> <p>Für ein Objekt, das in einem über ZOOM aufgerufenen Fenster gezeigt wird, stehen die gleichen Unterkommandos zur Verfügung wie für eine normal gelistete Source (außer PRINT, EXPAND und ZOOM). Wenn aufgrund von Stern- oder Wildcard-Notation mehrere Objekte angezeigt werden, können Sie darüber hinaus mit den Kommandos <u>N</u>EXT und <u>P</u>REV (bzw. PF4 und PF5) von einem Objekt im Fenster zum nächsten bzw. zum vorigen blättern.</p>
ZOOM [ <i>expand-type</i> ...10] <i>object-name-range</i>	
.	Beenden.

## Unterkommandos FIND, REF und SCAN

### Angabe des Wertes (*value*)

Der Wert (*value*) ist eine Suchzeichenkette mit bis zu 34 Zeichen, die beim Unterkommando FIND, REF oder SCAN anzugeben ist.

Falls der Wert (*value*) identisch ist mit der Kommando-Option CAS, CASESENSITIVE, ABS, ABSC oder ABSOLUTE, müssen Sie den Wert (*value*) durch einzelne Hochkommas begrenzen ( ' ' ).

Beispiel: FIND 'ABS'.

Wenn Sie ein Kommando ohne Wert (*value*) absetzen, erscheint ein Fenster, in dem Sie den gewünschten Wert eingeben sowie bestimmen können, ob die Suche absolut sein soll oder nicht (Y = Ja, N = Nein) und/oder ob sie **absolut** sein oder ob **Groß- und Kleinschreibung** beachtet werden soll.



### Absolute Suche

Wenn Sie eines der Kommandos mit `ABSOLUTE` absetzen, werden alle Treffer für den Wert (*value*) angezeigt, und zwar auch dann, wenn der Wert (*value*) als Teil eines Wortes vorkommt. Standardmäßig erfolgt die Suche nur nach ganzen Wörtern.

Sie können `ABSOLUTE` auch als Option im Eingabeaufforderungsfenster setzen. Siehe [Angabe des Wertes](#).

### Suche mit Unterscheidung nach Groß- und Kleinschreibung

Wenn Sie ein Kommando mit `CASESENSITIVE` absetzen, wird bei der Suche auf Groß- und Kleinschreibung geachtet. Die zu suchende Zeichenkette muss exakt mit der als Wert (*value*) angegebenen Zeichenkette mit Groß- und Kleinbuchstaben entsprechen. Standardmäßig wird bei der Suche nicht auf Groß- und Kleinschreibung geachtet. Es werden ohne Rücksicht auf Groß- und Kleinschreibung alle Treffer angezeigt.

Sie können `CASESENSITIVE` auch als Option im Eingabeaufforderungsfenster setzen. Siehe [Angabe des Wertes](#). Und Sie können die Standardeinstellung für `CASESENSITIVE` im LIST-Profil ändern. Siehe `SCAN-CASE-SENSITIVE` im Abschnitt [Individuelles List-Profil erstellen](#).

### Kombination aus absoluter Suche und Unterscheidung nach Groß- und Kleinschreibung

`ABSC` (oder `ABSOLUTECASESENSITIVE`) ist eine Kombination aus `ABSOLUTE` und `CASESENSITIVE`.

### Unterkommando FORMAT

Dieses Unterkommando gilt nur für Data Areas (die mit `STOW` gespeichert wurden und bei denen der Zeitstempel des Sourceobjekts und des katalogisierten Objekts identisch ist) und für Maps.

Bei Data Areas entspricht dieses Unterkommando der Option `FORMATTED`.

Im List-Profil können Sie angeben, wie die Data Areas standardmäßig angezeigt werden sollen:

- formatiert, d.h. die Anzeige ähnelt der in einem `DEFINE DATA`-Statement oder
- unformatiert, d.h. die Anzeige ähnelt der im Data-Area-Editor.

In der List der Objekte können Sie das Unterkommando `DEFINE-DATA ON/OFF` benutzen, um den Standardwert für die Zeit zu setzen, während der das `LIST`-Kommando ausgeführt wird.

Wenn Data Areas standardmäßig formatiert angezeigt werden und wenn es nicht möglich ist, den Sourcecode der Data Area in das `DEFINE DATA`-Format umzusetzen, erscheint eine entsprechende Meldung und die Data Area wird unformatiert angezeigt.

Wenn Sie das Unterkommando `FORMAT` für eine Map eingeben, erscheint ein Fenster, in dem Sie eine oder mehrere zusätzliche Informationen auswählen können, die mit der Map angezeigt werden:

- **Map Settings** (entspricht dem Unterkommando `SETTINGS`).

- **Map Layout** (entspricht dem Unterkommando `LAYOUT`). Zusätzlich können Sie Füllzeichen für Eingabefelder (`AD=A` und `AD=M`) und Ausgabefelder (`AD=0`) angeben, um diese Felder sichtbar zu machen. Sie können ein beliebiges Zeichen als Füllzeichen angeben.
- **Field Summary** (entspricht dem Unterkommando `FIELDS`).
- **Processing Rules** (entspricht dem Unterkommando `RULES`).

Die von Ihnen gewählten Optionen werden nacheinander in der im Auswahlfenster vorhandenen Reihenfolge angezeigt.

Im `FORMAT`-Modus stehen dieselben Unterkommandos zum Blättern (Ausnahme: `B`) und die Unterkommandos `FIELDS`, `LAYOUT`, `PRINT`, `RULES` und `SETTINGS` zur Verfügung wie bei einer normal mit `LIST` angezeigten Source (siehe oben). Zusätzlich sind zu jeder Option die nachfolgend beschriebenen Unterkommandos verfügbar.

- [Zusätzliche Unterkommandos für Map Layout](#)
- [Zusätzliche Unterkommandos für Field Summary List](#)
- [Zusätzliche Unterkommandos für Processing Rules](#)

#### Zusätzliche Unterkommandos für Map Layout

<code>S&gt;n</code>	Verschieben des Map Layouts um <i>n</i> Spalten nach rechts.
<code>S&lt;n</code>	Verschieben des Map Layouts um <i>n</i> Spalten nachlinks.

#### Zusätzliche Unterkommandos für Field Summary List

<code>EXTEND</code>	<p>Zeigt erweiterte Feld-Etitierinformationen zu allen Map-Feldern.</p> <p>Um die erweiterten Feld-Etitierinformationen zu einem einzelnen angezeigten Feld anzuzeigen, markieren Sie den Namen des Feldes in der Field Summary List mit dem Cursor und drücken Sie ENTER.</p>
<code>RULES nn</code>	<p>Zeigt die Verarbeitungsregeln, die mit dem Feld <i>nn</i> verknüpft sind (wobei <i>nn</i> für die fortlaufende Zeilennummer in der ersten Spalte der <b>Field Summary List</b> steht).</p> <p>Um die Verarbeitungsregeln zu einem einzelnen angezeigten Feld anzuzeigen, geben Sie ein <code>R</code> in der Kommandozeile ein, markieren Sie dann den Namen des Feldes in der Field Summary List mit dem Cursor und drücken Sie ENTER.</p>
<code>SCAN [ABSOLUTE] value</code>	Wie bei <a href="#">Unterkommandos für eine angezeigte Source</a> .
<code>SCAN =</code>	

## Zusätzliche Unterkommandos für Processing Rules

SCAN [ABSOLUTE] <i>value</i>	Wie bei <a href="#">Unterkommandos für eine angezeigte Source</a> .
SCAN =	

## Cursor-sensitive Objektauswahl

Innerhalb einer gelisteten Source können Sie mit dem Cursor den Namen eines Objektes, das in dieser Source referenziert wird, markieren; die Source des so ausgewählten Objektes wird dann in einem Fenster gezeigt.

Für die in dem Fenster gezeigte Source stehen Ihnen dieselben Unterkommandos — außer PRINT, EXPAND und ZOOM — zur Verfügung wie für eine „normal“ gelistete Source

## Individuelles List-Profil erstellen

Sie können sich ihr eigenes, individuelles Profil für das LIST-Kommando erstellen. Dazu gibt es beim Natural-LIST-Kommando das Unterkommando LISTPROF.

Mit diesem Kommando rufen Sie das Profilverwaltungs-Tool auf,

- das ein neues List-Profil mit den Standardwerten erstellt, wie sie intern benutzt werden, wenn das List-Profil nicht existiert;
- das es Ihnen gestattet, allgemeine Werte für einzelne Benutzer für die Profilparameter im List-Profil anzugeben (unterliegt der Kontrolle durch Natural Security);
- das es Ihnen gestattet, individuelle Werte für einzelne Benutzer für die Profilparameter im List-Profil anzugeben (unterliegt der Kontrolle durch Natural Security);
- das eine Beschreibung und Hilfe-Informationen zu jedem Profilparameter zur Verfügung stellt.

Das Profilverwaltungs-Tool zeigt auf einer Maske die allgemeinen Profilparameter und ihre aktuellen Werte an. Mit einer PF-Taste oder einem Kommando können Sie zu den Einstellungen Ihrer eigenen Profildaten wechseln.

Über Zeilenkommandos können Sie Parameter zum benutzerspezifischen Profil hinzufügen oder Parameter entfernen.

Weitere Informationen erhalten Sie, wenn Sie die Hilfe-Bildschirme zu den einzelnen Feldern aufrufen.

Wenn neue Parameter eingeführt werden, können Sie diese mit dem internen Kommando des Profilverwaltungs-Tools Änderungen in das List-Profil integrieren.

Das List-Profil hat den Namen `LIST-PRF` und befindet sich in der Library `SYSLIB`. Das für Aktualisierungen des List-Profiles verwendete Standardprofil hat den Namen `LIST-DEF` und befindet sich ebenfalls in der Library `SYSLIB`.

### List-Profilparameter

In der nachfolgenden Tabelle sind die Parameter aufgeführt, die im List-Profil enthalten sind, sowie die möglichen Werte, die zu diesen Parametern angegeben werden können. Außerdem enthält die Tabelle eine kurze Beschreibung der Parameter. Die Standardwerte der Parameter sind unterstrichen.

Parameter	Mögliche Werte / Format	Erläuterung
REUSE-LAST-LIST	<u>N</u> oder Y	N Die Standardeinstellung ist „No“. Y Die zuletzt angezeigte Objektliste wird nach der Ausführung von solchen Kommandos wiederverwendet, welche in der <b>Cmd</b> -Spalte eingegeben werden; eine Ausnahme bilden die Kommandos E, ED (Edit), CA (Catalog), UC (Uncat), S, ST (Stow), D, DE (Delete) oder RE (Rename).
SCAN-CASE-SENSITIVE	<u>N</u> oder Y	N Die Standardeinstellung ist N (No). Y Unterscheidung zwischen Groß- und Kleinschreibung beim Durchsuchen des Quellcodes nach einem mit dem Unterkommando FIND, REF oder SCAN angegebenen Suchwert. Siehe auch <i>Unterkommandos FIND, REF und SCAN</i> .
COUNT-SOURCE-LINES	<u>N</u> oder Y	N Die Standardeinstellung ist „No“. Y Anzeige der Anzahl an Sourcecode-Zeilen für Source-Objekte in der Maske mit den Source Directory-Daten (zur Anzeige der Source Directory-Daten benutzen Sie PF11).
SORT-TEXT-MEMBER-NAME	A08	Name des Natural Text Member, das zum Speichern der sortierten Liste benutzt wird.  Wird kein Name angegeben, wird ein generierter Name verwendet.

Parameter	Mögliche Werte / Format	Erläuterung
SORT-TEXT-MEMBER-LIBRARY	A08	Name der Library, in der das Natural Text Member, das zum Speichern der sortierten Liste benutzt wird, gespeichert wird.  Wird kein Name angegeben, wird die Library WORKPLAN verwendet.
DELETE-SORT-TEXT-MEMBER	N oder <u>Y</u>	N Die für die Sort-Funktion benutzte Natural Source des Typs Text soll <i>nicht</i> gelöscht werden, wenn das LIST-Kommando beendet wird.  Y Die für die Sort-Funktion benutzte Natural Source des Typs Text soll gelöscht werden, wenn das LIST-Kommando beendet wird. Dies ist die Standardeinstellung.
PRINT-PAGE-SIZE	N3	Standard-Seitengröße für das Drucken.  Die Standardeinstellung ist 60.
MARK-LONG-LINES	<u>N</u> oder Y	Y Lange Zeilen in einer angezeigten Source werden am Beginn der Sourcecode-Zeile mit einem L gekennzeichnet.  N Lange Zeilen in einer angezeigten Source werden nicht gekennzeichnet. Dies ist die Standardeinstellung.
DATA-AREA-DEFINE-DATA-FORMAT	<u>N</u> oder Y	Dient zur Angabe, ob Data Area Sources im DEFINE DATA-Format oder unformatiert angezeigt werden sollen.  Y Standardmäßig wird eine Data Area Source im DEFINE DATA-Format angezeigt (identisch mit dem Kommando LIST FORMATTED).  N Eine aufgelistete Data Area Source wird unformatiert angezeigt. Dies ist die Standardeinstellung.
SOURCE-LIST-WITH-DBID-FNR	<u>N</u> oder Y	Dient zur Angabe, ob die Datenbankkennung (DBID) und die Dateinummer (FNR) der Source Library in der Kopfzeile angezeigt werden sollen  N Die Standardeinstellung ist „No“.  Y Die Datenbankkennung (DBID) und die Dateinummer (FNR) der Source Library werden in der Kopfzeile angezeigt.

Parameter	Mögliche Werte / Format	Erläuterung
PAGE-TITLE-IN-BATCH	F oder A	Dient zur Angabe, ob die Kopfzeile oben auf jeder einzelnen Seite oder nur auf der ersten Seite ausgedruckt werden soll.  F Die Kopfzeile wird nur auf der ersten Seite ausgedruckt. A Die Kopfzeile wird auf jeder Seite ausgedruckt. Dies ist die Standardeinstellung.
OUTPUT-DESTINATION-BATCH	D, P, W oder T	Dient zur Angabe, ob die LIST-Ausgabe im Stapelverarbeitungsmodus geschrieben werden soll.  D Auf das Standard-Ausgabegerät schreiben (CMPRINT). P Auf das Standard-Druckergerät schreiben (CMPRT01). W In die Standard-Arbeitsdatei (Work File 1) schreiben. T In ein Natural Text Member schreiben.
OUTPUT-TEXT-NAME	A08	Name des Natural Text Member, das zum Speichern der LIST-Ausgabe benutzt wird, wenn <b>OUTPUT-DESTINATION-BATCH=T</b> ist.  Falls kein Name angegeben ist, wird L_##### verwendet; dabei ist sind ##### die ersten 6 Bytes der Benutzerkennung.
OUTPUT-TEXT-LIBRARY	A08	Name der Natural Library für das Natural Text Member, das zum Speichern der LIST-Ausgabe benutzt wird, wenn <b>OUTPUT-DESTINATION-BATCH=T</b> ist.  Falls kein Name angegeben ist, wird die Library WORKPLAN benutzt.
TOTAL-LINES-DISPLAY-BATCH	N oder Y	Dient zur Angabe, ob im Stapelverarbeitungsmodus die Gesamtzahl der Quellcodezeilen aller in der Liste der Objekte angezeigten Source-Objekte am Ende der Liste der Objekte angezeigt werden soll.  N Die Gesamtzahl der Quellcodezeilen wird nicht angezeigt (bessere Performance). Y Die Gesamtzahl der Quellcodezeilen wird angezeigt
USE-MAINPR	N oder Y	Dient zur Angabe, ob der Profilparameter MAINPR zur Steuerung der Ausgabe des LIST-Kommandos im Stapelverarbeitungsmodus benutzt werden soll.  N Ausgabe nach Report 0 schreiben.

Parameter	Mögliche Werte / Format	Erläuterung
		Y Ausgabe an den im Profilparameter MAINPR angegebenen Drucker leiten.





## 25 LIST XREF

---

LIST XREF

Das Kommando `LIST XREF` ist nur verfügbar, wenn Predict installiert ist.

Mit diesem Kommando können Sie sich alle aktiven Referenzdaten für die aktuelle Library anzeigen lassen.

Weitere Informationen siehe *List XREF for Natural* in der Predict-Dokumentation.



## 26 LISTDBRM

---



### Anmerkungen:

1. Dieses Kommando steht nur bei Natural for DB2 und Natural for SQL/DS zur Verfügung.
2. Bei Natural for DB2 muss das Kommando `LISTDBRM` aus der Natural System Library `SYSDB2` abgesetzt werden, d.h. zuerst ist ein Logon auf `SYSDB2` erforderlich und danach kann das Kommando `LISTDBRM` angegeben werden.
3. Bei SQL/DS muss das Kommando `LISTDBRM` aus der Library `SYSSQL` abgesetzt werden; d.h. zuerst ist ein Logon auf `SYSSQL` erforderlich und danach kann das Kommando `LISTDBRM` angegeben werden.

Das Kommando `LISTDBRM` dient zur Anzeige vorhandener DBRMs (Natural for DB2) oder vorhandener Pakete (Natural for SQL) mit Natural-Programmen, die eine gegebene DBRM referenzieren.

Bei Verwendung in Natural for DB2 erscheint folgendes Menü:

```

10:56:20          ***** NATURAL Tools for SQL *****          2006-03-17
                      - List DBRM -

Code Function

D   Display DBRMs of Programs
R   List Programs Referencing DBRM
?   Help
.   Exit

Code .. _   Library .. EXAMPLE_
           Member .. _____
           DBRM  ..... _____

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

```

Bei Verwendung in Natural for SQL/DS erscheint folgendes Menü:

```

15:35:20          * * * LISDBRM Command * * *          2006-05-25

Code Function
-----
D   Display DBRMs of Programs
R   List Programs Referencing DBRM
?   Help
.   Exit
-----

Code .. _   Library .. EXAMPLE_
           Member .. _____
           DBRM  ..... _____

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

```

Folgende Funktionen stehen zur Verfügung:

Code	Beschreibung
D	Anzeige von Programmen, die Zugriff auf DB2 (Natural for DB2) bzw. SQL/DS (Natural for SQL/DS) haben, und der entsprechende Pakete (DBRM).  Falls kein DBRM-Name angegeben wird, dann verwendet das entsprechende Programm dynamisches SQL.
R	Liste aller Programme, die ein gegebenes Paket (DBRM) verwenden. Falls kein DBRM-Name angegeben wird, dann werden alle Programme aufgelistet, die dynamisches SQL verwenden.

Folgende Parameter sind vorhanden:

Parameter	Beschreibung
Library	Der Name einer Natural Library.  Library-Namen, die mit SYS beginnen, sind nicht zulässig. Dieser Parameter muss angegeben werden.
Member	Der Name des anzuzeigenden Natural-Programms (Member).  Dieser Parameter ist optional und kann zur Begrenzung der Ausgabe verwendet werden.  Bei Angabe eines Werts mit nachfolgendem Stern (*) werden alle Member in der angegebenen Library, deren Namen mit diesem Wert beginnen, aufgelistet.  Falls das Feld <b>Member</b> leer gelassen wird oder falls nur ein Stern (*) eingegeben wird, werden alle Member in der angegebenen Library aufgelistet.
DBRM	Ein gültiger Paketname (DBRM).  Wird das Feld leer gelassen, werden Programme referenziert, die dynamisch laufen.  Dieser Parameter gilt nur bei Funktionscode R.

### Beispiel-Bildschirm für DBRM-Ergebnisse

11:15:45		***** LISTDBRM Command *****				2006-03-17	
Library	Name	Type	DBRM	User ID	Date	Time	
-----	-----	-----	-----	-----	-----	-----	
EXAMPLE	PROG1	Program	PACK1	SAG	2006-03-17	11:10:43	
EXAMPLE	PROG2	Program	PACK1	SAG	2006-03-17	11:10:48	
EXAMPLE	PROG3	Program	PACK2	SAG	2006-03-17	11:11:04	
EXAMPLE	PROG4	Program		SAG	2006-03-17	11:11:07	



# 27

## LISTSQL

Dieses Kommando ist mit geringfügig abweichender Funktionalität bei Natural for DB2, Natural SQL Gateway und Natural for SQL/DS verfügbar. Die produktabhängigen Unterschiede sind in der folgenden Beschreibung entsprechend gekennzeichnet.

```
LISTSQL [ { object-name } [ALL] ]
        <sa>
```

Das Kommando LISTSQL listet diejenigen Natural-Statements im Sourcecode eines Natural-Objekts auf, die in Verbindung mit einem Datenbankzugriff stehen, sowie die entsprechenden SQL-Kommandos, in die sie übersetzt wurden. Damit ist es möglich, den generierten SQLCODE zu sichten, bevor das Natural-Programm, welches auf eine SQL-Tabelle zugreift, ausgeführt wird.

Syntax-Element	Beschreibung
<i>object-name</i> <sa>	<p>Wenn Sie einen gültigen Objektnamen angeben, muss das anzuzeigende Objekt in der Library gespeichert sein, in die Sie zurzeit eingeloggt sind.</p> <p>Wenn Sie keinen Objektnamen angeben oder wenn Sie &lt;sa&gt; (source area) angeben, bezieht sich LISTSQL auf das Objekt, das sich zurzeit im Natural-Source-Bereich befindet.</p> <p>In jedem Fall benötigt LISTSQL ein katalogisiertes oder ein mit dem STOW-Kommando kompiliertes und gespeichertes Objekt.</p>
ALL	<p>Bei Angabe des Schlüsselworts ALL werden die generierten SQL-Statements des Objekts direkt nacheinander (ohne Blättern) angezeigt. Wenn Sie das Schlüsselwort weglassen, werden die im angegebenen Objekt enthaltenen generierten Statements jeweils einzeln auf einer Seite aufgeführt.</p> <p>Das Schlüsselwort ALL können Sie sowohl im Online- als auch im Batch-Modus angeben. Das Ausgabeformat ist in beiden Fällen gleich. Die Funktionen <b>Error</b> (PF2), <b>Explain</b> (PF4) bzw. <b>Parms</b> (PF6) stehen nicht zur Verfügung.</p> <p>Wenn Sie das Schlüsselwort ALL angeben, können Sie ein Fragezeichen (?) oder einen Stern (*) als Stellvertretersymbol benutze, zum Beispiel: LISTSQL PGM* ALL. Die Sonderzeichen</p>

Syntax-Element	Beschreibung
	> und < sind erlaubt, jedoch nur am Ende einer Zeichenkette; das bedeutet, dass z.B. die Angabe ABC<DEF ein ungültiger Ausdruck ist.

### Beispiel-Bildschirm LISTSQL:

```

14:50:23          ***** NATURAL TOOLS FOR SQL *****          2009-12-04
Member DEM2SEL          - LISTSQL -          Library SYSDB243
SQL Builder Version 4.10
Natural statement at line 0140          Stmt 1 / 1

SELECT *
      INTO VIEW NAT-DEMO
      FROM NAT-DEMO

Generated SQL statement   Mode : dynamic   DBRM :          Line 1 / 3
                                          Length 68

SELECT  NAME, ADDRESS, DATEOFBIRTH, SALARY
FROM    NAT.DEMO
FOR FETCH ONLY

Command ==>          Queryno for EXPLAIN 1_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Error Exit  Expl          Parms          Prev Next Canc  ↵

```

Falls ein statisches DBRM generiert wurde, dann wird sein Name im Feld **DBRM** des Bildschirms **LISTSQL** angezeigt; andernfalls bleibt dieses Feld leer. Ein statisches DBRM steht nur bei Natural for DB2 und Natural for SQL/DS zur Verfügung.

Die folgenden bildschirmspezifischen PF-Funktionstasten stehen zur Verfügung:

PF-Taste (Bezeichnung):	Funktion:
PF2 (Error)	Diese Taste dient zur Ausführung des Kommandos <b>SQLERR</b> . Wenn bei der EXPLAIN-Funktion ein Fehler auftritt, können Sie diese Taste drücken, um Informationen zu DB2- bzw. SQL/DS-Fehler zu erhalten.
PF4 (Expl)	Diese Taste dient zur Ausführung des Kommandos EXPLAIN für das zurzeit aufgelistete SQL-Statement. Die Abfragenummer für das EXPLAIN-Kommando (im Feld <b>Queryno for EXPLAIN</b> ) ist standardmäßig auf 1 gesetzt und kann überschrieben werden.  Das Kommando EXPLAIN wird nur bei Natural for DB2 und Natural for SQL/DS unterstützt.



PF-Taste (Bezeichnung):	Funktion:
PF6 (Parms)	Mit dieser Taste kann ein weiterer Bildschirm aufgerufen werden, auf dem alle Parameter aus dem SQLDA für das zurzeit aufgelistete SQL-Statement angezeigt werden; siehe Beispiel-Bildschirm weiter unten.
PF10 (Prev), PF11 (Next)	Durch Drücken dieser Tasten können Sie innerhalb der aufgelisteten Ergebnisse zwischen den SQL-Statements hin- und herblättern.

Beispiel-Bildschirm mit Parametern:

```

14:55:24          ***** NATURAL TOOLS FOR SQL *****          2009-12-04
Member DEM2SEL          LISTSQL          Library SYSDB243

      Mode : dynamic   DBRM :          Contoken :
                (3rd/pre)
      static parms : (1st)
                (2nd)

      SQLDA

                                DBID : 250  FNR :   1  CMD : S1 0140 08
Nr  Type      Length      CCSID
1.  CHAR           20      8001 0000 0014 01C4 0000 0000 0800 0000
2.  CHAR          100      8002 0000 0064 01C4 0000 0000 0800 0000
3.  CHAR           10      8003 0000 000A 01C4 0000 0000 0800 0000
4.  DECIMAL        6.2      8004 4000 0602 01E5 0000 0000 0800 0000

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


```

Im statischen Modus werden ausserdem noch statische Informationen angezeigt (statischer DBRM-Name, DB2- bzw. SQL/DS-Konsistenzmerkmal (Contoken) und einige intern verwendete statische Parameter).

Falls die Informationen nicht auf den Parameter-Bildschirm passen, stehen Ihnen zum Navigieren folgende Tasten zur Verfügung:

PF-Taste (Bezeichnung):	Funktion:
PF6 (top,--), PF9 (bottom,++)	Direkter Sprung an den Anfang (--) bzw. and das Ende (++) der Liste.
PF8, PF7 (-)	Vorwärtsblättern (+) bzw. Zurückblättern (-).

Verwendung des Kommandos EXPLAIN bei Natural for DB2

 **Wichtig:** Vor Verwendung des DB2-Kommandos EXPLAIN siehe *LISTSQL and Explain Functions* im Abschnitt *Installing Natural for DB2 on z/OS* in der *Installation for z/OS-Dokumentation*.

Das EXPLAIN-Kommando liefert Informationen darüber, welche Strategie der DB2 Optimizer bezüglich der Ausführung von SQL-Statements gewählt hat. Damit das Kommando EXPLAIN ausgeführt werden kann, muss eine PLAN\_TABLE vorhanden sein. In diese Tabelle werden die vom DB2 Optimizer bestimmten Informationen geschrieben. Die entsprechende Erläuterung wird aus der PLAN\_TABLE ausgelesen und auf dem Bildschirm **EXPLAIN Result** angezeigt.

Beispiel-Bildschirm Explain Result:

```
10:57:47          ***** NATURAL TOOLS FOR SQL *****          2009-12-03
Queryno 1          EXPLAIN Result          Row 1 / 1

          Estimated cost :   296.6   timerons

Qblock  Plan Mixop Acc. Match Index Pre-  Column- Access-
  No      No   seq type  cols only  fetch fn_eval Creator.Name
-----
      1      1      R              S

          Table-
          TabNo Creator.Name          Tslock          -- sortn --  -- sortc --
          mode   Method uq jo or gr  uq jo or gr
-----
      1 NAT.DEMO          IS          N  N  N  N  N  N  N  N

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

Falls eine Erläuterung nicht vollständig auf dem Bildschirm angezeigt werden kann, können Sie mit den Tasten PF7 (-) bzw. PF8 (+) rückwärt- bzw. vorwärts blättern.

Der Wert im Feld **Estimated cost** wird SQLERRD (4) in der SQLCA entnommen; es handelt sich um eine ungefähre Schätzung der benötigten Ressourcen.

Mit PF4 (Info) können die zusätzlichen, mit dem Kommando EXPLAINB gelieferten Informationen angezeigt werden.

## Verwendung des Kommandos EXPLAIN bei Natural SQL Gateway

Bei Natural SQL Gateway ist dieses Kommando nicht gültig.

## Verwendung des Kommandos EXPLAIN bei Natural for SQL/DS

Unter LISTSQL können Sie das SQL/DS-Kommando EXPLAIN benutzen, das Ihnen Informationen darüber liefert, welche Strategie der DB2 Optimizer bezüglich der Ausführung von SQL-Statements gewählt hat.

Natural führt das Kommando EXPLAIN für das auf dem LISTSQL-Bildschirm angezeigte SQL-Statement aus.

Die vom SQL/DS Optimizer bestimmten Informationen werden in Ihre PLAN\_TABLE geschrieben. Natural liest die Tabelle und zeigt den Inhalt an.

### Beispiel-Bildschirm EXPLAIN Result:

```

10:22:07          ***** NATURAL TOOLS FOR SQL *****          2009-12-03
Queryno 1          EXPLAIN Result          Row 1 / 1

          Estimated cost :      3.3 timerons

          Qblockno          Table
          Planno Method Tabno creator          Tablename
          ---
          1      1          1      NAT          DEMO

          Access Access
          type creator          Accessname          sort_new sort_comp
          ---
          R          N          N

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

```



# 28

## LISTSQLB

---



### Anmerkungen:

1. Diese Kommando steht nur bei Natural for DB2 zur Verfügung.
2. Vor Verwendung des Kommandos LISTSQLB siehe *LISTSQL and Explain Functions* im Abschnitt *Installing Natural for DB2 on z/OS* in der *Installation for z/OS*-Dokumentation.

Das Kommando LISTSQLB kann im Batch-Modus ausgeführt oder am Natural-Eingabeaufforderungszeichen NEXT abgesetzt werden

Bei Ausführung im Online-Modus erscheint folgender Bildschirm:

```

10:54:35          ***** NATURAL Tools for SQL *****          2006-03-17
                      - LISTSQLB -

                                Code Function

                                X   Explain all SQL Statements
                                .   Exit

Code .. _   Member ... _____
            Queryno .. 1_____

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

```

Durch Angabe eines gültigen Member-Namens kann die Erläuterung eines SQL-Statements auf bestimmte Member eingeschränkt werden; zur Einschränkung auf einen bestimmten Bereich kann Stern-Notation (\*) verwendet werden.

- Wenn Sie einen eindeutigen Member-Namen eingeben, werden alle in diesem Member enthaltenen SQL-Statements erläutert.
- Wenn Sie einen Wert, gefolgt von einem Stern (\*), eingeben, werden alle SQL-Statements angezeigt, die sich in allen Member befinden, deren Namen mit dem angegebenen Wert beginnen.
- Wenn Sie einen Stern (\*) eingeben (oder das Feld leer lassen), werden alle SQL-Statements aller vorhandenen SQL Member erläutert.

Die Eingabe einer Abfragenummer im Feld **Queryno** ist erforderlich, damit bei jedem erneuten Absetzen des EXPLAIN-Kommandos die neu erstellte Erläuterung zur entsprechenden Abfragenummer hinzugefügt wird. Standardmäßig ist das Feld mit der Nummer 1 vorbelegt.

Um das Kommando EXPLAIN abzusetzen, geben Sie den Funktionscode X und einen gültigen Member-Namen ein. Es werden alle in dem bzw. den angegebenen Member enthaltenen SQL-Statements erläutert.

Bei Ausführung von LISTSQLB im Online-Modus liefert der folgende Bildschirm Informationen über den Verarbeitungsstand des Kommandos und eventuell aufgetretene Fehler.

```
10:55:24          ***** NATURAL Tools for SQL *****          2006-03-17
                        - LISTSQLB -
Queryno : 1                      Member Stmtno Message

Current Object :
Library      TEST
Member      RTTB--IN

Statistics :
Members read  1
  with SQL    1
SQL statements 7

Member      Message

RTTB--IN OK

Press Enter to continue
Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
```

Bei Ausführung von LISTSQLB im Batch-Modus werden Fehlermeldungen in einen Dataset geschrieben, auf den mit dem DD-Namen CMPRINT (logischer Drucker 0) verwiesen wird.





# 29 LOGOFF

---

LOGOFF

Verwandtes Kommando: [LOGON](#)

Mit dem Systemkommando LOGOFF erreichen Sie, dass die Library-ID auf SYSTEM und das Adabas-Passwort auf Leerzeichen gesetzt wird. Der Source-Inhalt des Editor-Arbeitsbereichs bleibt erhalten.

Das LOGOFF-Kommando hat keinen Einfluß auf die Werte der globalen Parameter.

Informationen zur LOGOFF-Verarbeitung unter Natural Security siehe *How to End a Natural Session* im Abschnitt *Logging On* in der *Natural Security*-Dokumentation.



**Anmerkung:** Mit dem LOGOFF-Kommando beenden Sie nicht die Natural-Session. Um eine Session zu beenden, verwenden Sie das Systemkommando [FIN](#) oder führen ein Programm aus, das ein TERMINATE-Statement enthält.



# 30

## LOGON

---

`LOGON library-id [password]`

Verwandtes Kommando: [LOGOFF](#)

Mit dem Systemkommando `LOGON` wählen Sie eine bestimmte Library. In dieser Library werden dann alle von Ihnen während der Session in Source- und/oder Objektform erstellten Objekte gespeichert (es sei denn, Sie geben bei einem [SAVE](#)-, [CATALOG](#)- oder [STOW](#)-Kommando ausdrücklich eine andere Library an).

Der Source-Inhalt des Editor-Arbeitsbereichs bleibt bei einem `LOGON`-Kommando unverändert.

Das Kommando `LOGON` bewirkt, dass alle Natural Global Data Areas und anwendungsunabhängigen Variablen (AIVs), alle mit einem `SET KEY`-Statement vorgenommenen Zuordnungen und alle gehalten ISN-Listen freigegeben werde. Im DDM-Buffer-Bereich enthaltene DDMs werden ebenfalls freigegeben.

Siehe auch *Library Naming Conventions* in der *Using Natural*-Dokumentation.

LOGON library-id	Eine Library-ID darf 1 bis 8 Zeichen lang sein und keine Leerzeichen enthalten. Sie darf folgende Zeichen enthalten:	
	A - Z	Großbuchstaben
	0 - 9	Ziffern
	-	Bindestrich
	_	Unterstrich
	Das erste Zeichen einer Library-ID muß ein Großbuchstabe sein.	
LOGON library-idpassword	Das Adabas-Passwort; siehe <i>Session Parameters</i> im Abschnitt <i>Library Maintenance</i> der <i>Natural Security</i> -Dokumentation.	

Unter Natural Security gelten für das LOGON-Kommando bestimmte Regeln und Einschränkungen. Diese erfahren Sie von Ihrem Administrator oder finden Sie im Abschnitt *Logging On* in der *Natural Security*-Dokumentation.

# 31 MAIL

---

MAIL  $\left[ \begin{array}{c} * \\ ? \\ mailbox-id \end{array} \right]$

Dieses Kommando ist nur verfügbar, wenn Natural Security installiert ist.

Mit dem Systemkommando `MAIL` können Sie eine Mailbox aufrufen, um den Inhalt sowie das Ablaufdatum zu ändern.

Eine Mailbox ist eine Art „Schwarzes Brett“ zur Übermittlung von Informationen, das mit Natural Security definiert wird.

MAIL	Wenn Sie das <code>MAIL</code> -Kommando ohne Parameter eingeben, erscheint ein Fenster, in dem Sie aufgefordert werden eine Mailbox-ID anzugeben.
MAIL *	Es wird eine Liste der Mailboxen angezeigt, die Sie benutzen können. Sie können aus der Liste eine Mailbox auswählen.
MAIL ?	
MAIL <i>mailbox-id</i>	Wenn Sie eine <i>mailbox-id</i> angeben (bis zu 8 Zeichen lang), wird die entsprechende Mailbox direkt aufgerufen. Die <i>mailbox-id</i> , die Sie angeben, muß in Natural Security definiert sein.

Weitere Informationen siehe *Mailboxes* in der *Natural Security*-Dokumentation.



## 32 MAINMENU

---

MAINMENU	$\left[ \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{user-program} \end{array} \right\} \right]$
----------	---

Mit dem Systemkommando `MAINMENU` können Sie die Anzeige des Natural-Hauptmenüs ein- bzw. ausschalten oder ein Programm aufrufen, das einen eigenen Menüschild anzeigt.

Das Kommando ist in der Kommandozeile einer Remote-Entwicklungsumgebung verfügbar.

MAINMENU	Die Eingabe des MAINMENU-Kommandos ohne Parameter ist identisch mit MAINMENU ON. Dies ist der Standardwert.
MAINMENU ON	
MAINMENU OFF	Schaltet den Hauptmenü-Modus aus.
MAINMENU <i>user-program</i>	Statt des Natural-Hauptmenüs wird ein benutzerdefiniertes Programm aufgerufen, das seinerseits ein benutzerdefiniertes Menü aufruft.

Siehe auch Natural-Profilparameter `MENU`.





## 33 NATQVS

---

Mit diesem Kommando können Sie sich Informationen über die Eigenschaften der Maschine (z.B. die CPU-Kennung und logische Partitionen) anzeigen lassen, auf der zurzeit Natural ausgeführt wird. Anhand dieser Daten können Sie überprüfen, ob eine Übereinstimmung mit den Daten in der Produktlizenz gegeben ist.

Weitere Informationen siehe *Software AG Mainframe Product Licensing* in der *Installation-Dokumentation*.



# 34 NOCOPT

---

NOCOPT

Mit dem Kommando `NOCOPT` können Sie verschiedene beim Start von Natural gesetzte Optionen für den Natural Optimizer Compiler anzeigen und ändern.

Weitere Informationen zu `NOCOPT` finden Sie im Abschnitt *Activating the Optimizer Compiler* in der *Natural Optimizer Compiler-Dokumentation*.



# 35 NOCSHOW

---

NOCSHOW

Das Kommando NOCSHOW liefert Pufferspeicher-Informationen zur Ausgabe der PGEN-Option des Natural Optimizer Compiler.

Weitere Informationen zu NOCSHOW finden Sie im Dokument *Optimizer Options*, Abschnitt *Output of the PGEN Option*, in der *Natural Optimizer Compiler*-Dokumentation.



## 36 NOCSTAT

---

NOCSTAT

Das Kommando `NOCSTAT` liefert Statistikdaten zu Programmen, die für die Verarbeitung durch den Natural Optimizer Compiler geeignet sind.

Weitere Informationen zu `NOCSTAT` siehe *NOCSTAT Command* in der *Natural Optimizer Compiler-Dokumentation*.





## 37 PROFILE

---

Dieses Kommando ist nur verfügbar, wenn Natural Security installiert ist.

PROFILE

Mit dem Systemkommando `PROFILE` können Sie sich Ihr gegenwärtig gültiges Benutzerprofil anzeigen lassen. Dieses Profil informiert Sie über die Bedingungen und Voraussetzungen, die in Ihrer aktuellen Natural-Umgebung für Sie gelten.

Weitere Informationen siehe *PROFILE Command* in der *Natural Security*-Dokumentation.

---

# 38

## PROFILER

---

### PROFILER

Mit diesem Systemkommando können Sie die Profiler Utility aufrufen. Diese Utility dient dazu, die internen Abläufe und Prozesse in einer Natural-Anwendung zu überwachen und die Leistung der Anwendung zu analysieren.

Weitere Informationen siehe *Profiler Utility* in der *Utilities*-Dokumentation.



# 39 RDC

---

RDC { ON OFF }
-------------------

Dieses Kommando wird in Verbindung mit der `PROFILER`-Utility benutzt, um die Trace Recording-Funktion des Natural Data Collector zu aktivieren bzw. zu deaktivieren.

Weitere Informationen siehe *Activating and Controlling the Natural Data Collector* in *SYSRDC Utility* in der *Utilities*-Dokumentation.

RDC ON	Aktiviert die Trace Recording-Funktion des Natural Data Collector
RDC OFF	Deaktiviert die Trace Recording-Funktion des Natural Data Collector

Siehe auch Terminalkommandos `%<RDC+` und `%<RDC-` in der *Terminalkommandos*Dokumentation und Profilparameter `RDC` in der *Parameter-Referenz*-Dokumentation.

---

# 40

## READ

---

```
READ object-name [library-id]
```

Verwandtes Kommando: [EDIT](#)

Mit dem Kommando READ können Sie ein in Sourceform gespeichertes Objekt in den Arbeitsbereich des entsprechenden Editors einlesen.



**Vorsicht:** Ein bereits im Editor befindliches Objekt wird dadurch überschrieben.

Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural* benutzen.

READ <i>object-name</i>	Der Name des gewünschten Objekts.  Wenn Sie den <i>object-name</i> ohne Library-ID angeben, wird das gewünschte Objekt nur dann eingelesen, wenn es in Ihrer aktuellen Library gespeichert ist.
READ <i>object-name</i> <i>library-id</i>	Falls das gewünschte Objekt nicht in Ihrer aktuellen Library, sondern in einer anderen gespeichert ist, müssen Sie die Library-ID dieser Library angeben.  Wenn Sie sowohl den <i>object-name</i> und die <i>library-id</i> angeben, liest Natural das Objekt nur dann ein, wenn es in der angegebenen Library gespeichert ist.





# 41

## RENAME

---

RENAME [*old-name* [*new-name* [*new-type*]]

Mit dem Kommando `RENAME` können Sie ein Natural-Objekt umbenennen und außerdem seinen Objekttyp ändern.

Sie können jeweils nur ein Objekt zur Zeit umbenennen. Das Objekt muß in der Library, in der Sie sich gerade befinden, gespeichert sein. Um Inkonsistenzen zu vermeiden, benennt Natural jede bestehende Form (Sourcecode, Objektmodul oder beides) des Objekts um.

Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural benutzen*.

RENAME	Wenn Sie das Kommando ohne Parameter absetzen, erscheint ein <b>Rename Object</b> -Fenster, in dem Sie dieselben Parameter wie in der Kommandozeile angeben können.	
<i>old-name</i>	Als <i>old-name</i> geben Sie den derzeitigen Namen des Objekts an.	
<i>new-name</i>	Als <i>new-name</i> geben Sie den neuen Namen an, unter dem das Objekt von nun an gespeichert sein soll.	
<i>new-type</i>	Wenn Sie ein Objekt umbenennen, haben Sie außerdem die Möglichkeit, seinen Objekttyp zu ändern, indem Sie das entsprechende Zeichen für den <i>new-type</i> angeben. Dies ist allerdings nur bei Objekten möglich, die lediglich in Sourceform bestehen.	
	Mögliche Werte für <i>new-type</i> :	
	4	Klasse
	5	Processor
	8	Adapter
	9	Resource
	A	Parameter Data Area
	C	Copycode
	G	Global Data Area
	H	Helproutine

	L	Local Data Area
	M	Map
	N	Subprogramm
	O	Macro
	P	Programm
	S	Subroutine
	T	Text
	Y	Rule
	Z	Recording

# 42

## RENUMBER

---

`RENUMBER [(n)]`

Das Kommando `RENUMBER` bewirkt, dass die Quellcode-Zeilen des Objekts, das sich gerade im im Arbeitsbereich des Natural-Editors befindet, neu durchnummeriert werden.



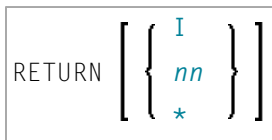
**Anmerkung:** Wenn auch alphanumerische Konstanten umnummeriert werden sollen, muss der Profilparameter `RNCONST` auf `ON` gesetzt werden.

RENUMBER	Wenn Sie das Kommando ohne Parameter eingeben, werden, je nach Gesamtzahl der Quellcode-Zeilen im Arbeitsbereich, die folgenden Standard-Schrittwerte für die Umnummerierung verwendet:	
	0001 bis 0999 Zeilen:	10er-Schritte
	1000 bis 1999 Zeilen:	5er-Schritte
	2000 bis 4999 Zeilen:	2er-Schritte
	5000 Zeilen und mehr:	1er-Schritte
	Ab Zeile 9999 wird immer 9999 als Zeilennummer zugewiesen. Beachten Sie, dass Sie keinen Quellcode kompilieren können, der mehr als 9999 Zeilen enthält.	
RENUMBER (n)	Mit <i>n</i> können Sie einen Schrittwert im Bereich von 1 bis 9999 angeben, um den eine Zeilennummer jeweils höher als die vorherige sein soll. Falls der gegebene Schrittwert bewirken würde, dass die Umnummerierung den Zeilennummern-Grenzwert von 9999 überschreitet, wird stattdessen der Standard-Schrittwert genommen.	

Siehe auch *Quellcode-Zeilennummern umnummerieren im Leitfaden zur Programmierung*.

---

# 43 RETURN



Mit dem Kommando `RETURN` können Sie zu einer bestimmten vorherigen Natural-Anwendung (oder der Ausgangsanwendung) zurückkehren.

Programmierschnittstelle (API): USR1026N. Siehe *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

RETURN	<p>Wenn Sie <code>RETURN</code> ohne Parameter eingeben, wird die Kontrolle an die vorherige Anwendung übergeben. Eine vorherige Anwendung wird mit dem Systemkommando <code>SETUP</code> als solche definiert. Alle (mit <code>SETUP</code> spezifizierten) Informationen über diese vorherige Anwendung werden gelöscht. Ist keine vorherige Anwendung definiert, erfolgt ein Rücksprung zur Ausgangsanwendung.</p> <p>Wenn Sie <code>RETURN</code> eingeben und kein Rückkehrpunkt definiert ist, wird das <code>RETURN</code>-Kommando ignoriert.</p> <p><b>Unter Natural Security:</b></p> <p>Ein <code>LOGOFF</code>-Kommando wird ausgeführt, wenn Sie <code>RETURN</code> eingeben und kein Rückkehrpunkt definiert ist.</p>
RETURN I	Mit diesem Kommando übergeben Sie die Kontrolle direkt an die Ausgangsanwendung (I = Initial Application). Alle Informationen über vorherige Anwendung (außer der Ausgangsanwendung) werden dabei gelöscht.
RETURN nn	Mit diesem Kommando übergeben Sie die Kontrolle an die <i>nn</i> -te vorherige Anwendung. Alle Informationen über die der <i>nn</i> -ten Anwendung nachfolgenden Anwendungen werden dabei gelöscht.
RETURN *	Dieses Kommando zeigt Ihnen eine Liste aller gegenwärtig definierten Rückkehrpunkte an. Auf der Liste können Sie dann den gewünschten Rückkehrpunkt auswählen.

Siehe Systemkommando [SETUP](#) für Beispiele und weitere Informationen.

# 44

## ROUTINES

---

### ROUTINES

Mit dem Kommando `ROUTINES` können Sie feststellen, welche katalogisierten Objekte in Ihrer aktuellen Library welche externen Subroutinen verwenden.

Sie erhalten eine Liste aller Objekte in der aktuellen Library, wobei für jedes Objekt die Namen der von ihm aufgerufenen externen Subroutinen angezeigt werden, sowie die Objektnamen der Subroutinen, in denen diese externen Subroutinen enthalten sind.

Handelt es sich bei dem Objekt selbst um eine Subroutine, Klasse oder Function, wird der Langname der in ihr enthaltenen Subroutine, Klasse oder Function angezeigt.





## 45 RPCERR

---

### RPCERR

Mit dem Kommando `RPCERR` können Sie sich die Nummer und die Meldung des zuletzt aufgetretenen Natural-Fehlers anzeigen lassen, falls er in Bezug zum RPC steht. Außerdem wird der Reason Code des Brokers und die zugehörige Meldung angezeigt. Darüber hinaus kann der Node- und Server-Name aus dem letzten Broker-Aufruf abgerufen werden.

Weitere Informationen siehe *Monitoring the Status of an RPC Session* im Abschnitt *Operating a Natural RPC Environment* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.



# 46

## RUN

---

`RUN [REPEAT] [program-name [library-id]]`

Das Kommando `RUN` dient dazu, ein Source-Programm zu kompilieren und auszuführen. Das auszuführende Programm kann sich entweder in der Natural-Systemdatei oder im Arbeitsbereich des Editors befinden.

Siehe auch:

*Natural-Compiler in Natural System-Architektur*  
*Namenskonventionen für Objekte in Natural benutzen*

<code>RUN</code>	Wenn Sie den Programmnamen ( <i>program name</i> ) nicht angeben, dann kompiliert Natural das Programm, das sich gegenwärtig im Editor-Arbeitsbereich befindet, und führt es aus.
<code>REPEAT</code>	Wenn das auszuführende Programm mehrere Ausgabeschirme erzeugt und Sie möchten, dass die Schirme unmittelbar nacheinander (d.h. ohne zwischengeschaltete Eingabezeilen) ausgegeben werden, müssen Sie zusätzlich zum Schlüsselwort <code>RUN</code> das Schlüsselwort <code>REPEAT</code> angeben. Nach Beendigung des Programms wechselt Natural in den Kommando-Modus.
<i>program-name</i>	Der Name des Programms, das kompiliert und ausgeführt werden soll.  Wenn Sie den <i>program-name</i> ohne Library-ID angeben, wird das Programm in den Arbeitsbereich gelesen (wobei ein bereits im Arbeitsbereich befindliches Objekt überschrieben wird), kompiliert und ausgeführt, vorausgesetzt es ist in Ihrer aktuellen Library gespeichert. Wenn es nicht unter der aktuellen Library-ID gespeichert ist, erscheint eine Fehlermeldung.
<i>library-id</i>	Die Library in der sich das gewünschte Programm befindet.  Falls Sie ein Programm kompilieren und ausführen wollen, das nicht in Ihrer aktuellen Library gespeichert ist, müssen Sie zusätzlich zum Programmnamen die Library-ID dieser Library angeben.

	<p>Wenn Sie sowohl den <i>program-name</i> und die <i>library-id</i> angeben, dann wird das angegebene Programm nur dann kompiliert und ausgeführt, wenn es unter der angegebenen Library-ID gespeichert ist. Wenn es nicht unter der aktuellen Library-ID gespeichert ist, erscheint eine Fehlermeldung.</p> <p>Eine Library-ID, die mit <code>SYS</code> beginnt, darf nicht angegeben werden (Ausnahme: <code>SYSTEM</code>).</p> <p>Wenn Natural Security aktiv ist, können Sie keine Library-ID angeben, d.h. Sie können nur Programme aus ihrer aktuellen Library kompilieren/ausführen.</p>
--	--

# 47

## SAVE

```
SAVE [object-name [library-id]]
```

Verwandte Kommandos: [STOW](#) | [CATALOG](#)

Das Kommando `SAVE` dient dazu, den zurzeit im Arbeitsbereich des Natural-Editors befindlichen Quellcode zu speichern und ihn in Source- und Objektform in der aktuellen Natural-Systemdatei abzulegen.

Siehe auch:

*Namenskonventionen für Objekte in Natural benutzen*

*Natural-Compiler in Natural System-Architektur* als Hintergrundinformation



**Vorsicht:** Wird das `SAVE`-Kommando zurückgewiesen, bedeutet dies, dass der Parameter `RECAT` auf `ON` gesetzt ist und das Programm, das Sie in Sourceform speichern wollen, bereits in kompilierter Form als Objektmodul gespeichert ist. In diesem Falle geben Sie das Systemkommando `STOW` ein, mit dem Sie das Programm gleichzeitig in Source- und Objektform speichern und so Diskrepanzen zwischen beiden Formen vermeiden.

SAVE	Wenn Sie das Kommando ohne <i>object-name</i> benutzen, wird das im Arbeitsbereich des Editors befindliche Objekt in der aktuellen Library gespeichert. Vorhandener Sourcecode wird ersetzt.
SAVE <i>object-name</i>	Ein neues Objekt wird erstellt. Als <i>object-name</i> geben Sie den Namen an, unter dem das neue Objekt gespeichert werden soll. Falls ein Objekt dieses Namens schon vorhanden ist, wird das Kommando zurückgewiesen.
SAVE <i>object-name</i> <i>library-id</i>	Wenn Sie keine Library-ID angeben, wird das Programm in der Library, in der Sie gerade arbeiten, gespeichert. Wollen Sie es in einer anderen Library speichern, müssen Sie die Library-ID dieser Library angeben. Es ist allerdings nur möglich, ein Programm in einer anderen Library zu speichern, falls dort noch kein Objekt gleichen Namens gespeichert ist.

	Wenn Sie ein Objekt unter einem anderen Namen oder ein neu erstelltes Objekt speichern, wird das Objekt standardmäßig in der aktuellen Library gespeichert. Wollen Sie es in einer anderen Library speichern, müssen Sie nach dem <i>object-name</i> die <i>library-id</i> dieser Library angeben. Es wird ein neues Objekt angelegt. Falls ein Objekt dieses Namens schon vorhanden ist, wird das Kommando zurückgewiesen.
--	---

# 48

## SCAN

---

■ Menü-Optionen .....	192
■ SCAN-Editierkommandos .....	195
■ SCAN mittels Direktkommando .....	195
■ Individuelles SCAN-Profil definieren .....	199
■ SCAN unter Natural Security .....	200

Mit dem Kommando `SCAN` können Sie den Sourcecode von Objekten nach einer bestimmten Zeichenkette absuchen. Darüber hinaus besteht die Möglichkeit, die gesuchte Zeichenkette durch eine andere Zeichenkette zu ersetzen.

Sie können ein oder mehrere oder alle Objekte einer Library absuchen; hierbei können Sie bestimmte Objekte mit einer Anfangseinstellung oder Objekte in einer Library auswählen und außerdem die Suche auf einen bestimmten Objekttyp einschränken.



**Wichtig:** Da das `SCAN`-Kommando den Sourcecode im Arbeitsbereich beeinflusst, sollten Sie vor der Verwendung des `SCAN`-Kommandos den Inhalt des Arbeitsbereichs gegebenenfalls mit einem `SAVE`- oder `STOW`-Kommando speichern.

## Menü-Optionen

Wenn Sie das Kommando `SCAN` eingeben, wird das `SCAN`-Menü angezeigt, das Ihnen folgende Funktionen bietet:

Feld	Eingabe	
Code	T	<b>Statistics</b>  Diese Funktion liefert Ihnen folgende Informationen: <ul style="list-style-type: none"> <li>■ die Anzahl der Objekte, die durchsucht wurden;</li> <li>■ die Anzahl der Objekte, in denen der gesuchte Wert gefunden wurde;</li> <li>■ die Anzahl der Sourcecode-Zeilen, in denen der gesuchte Wert gefunden wurde.</li> </ul>
	L	<b>List of Objects Containing Scan Value</b>  Diese Funktion liefert Ihnen eine Liste aller Objekte, in denen der gesuchte Wert gefunden wurde. In der Liste können Sie dann einzelne Objekte zur weiteren Verarbeitung auswählen.  Bei Bedarf können Sie die Zeilen mit den entsprechenden <b>SCAN-Unterkommandos</b> verändern.  Mit dem <code>SCAN</code> -Editor können Sie alle Objekttypen, außer Maps und Data Areas und gesperrte Objekte (siehe <i>Locking of Source Objects</i> ), verändern, und zwar im Full-Screen-Modus oder zeilenorientiert. Dadurch haben Sie die Möglichkeit, alle Zeilen zu editieren, nicht nur die mit dem <code>SCAN</code> -Kommando gefundenen. Sobald Sie ein Objekt fertig editiert haben, sollten Sie es speichern, bevor Sie vom



Feld	Eingabe	
		normalen Editieren zur Fortsetzung der SCAN-Operation zurückkehren. Bevor die SCAN-Verarbeitung einen anderen Editor aufruft, erhalten Sie eine Aufforderung, die bis dato mit dem SCAN-Editor gemachten Änderungen zu bestätigen.
	S	<p><b>Object Lines with Scan Value</b></p> <p>Diese Funktion zeigt Ihnen nacheinander die Sourcecode-Zeilen an, in denen der gesuchte Wert gefunden wurde.</p> <p>Bei Bedarf können Sie die Zeilen mit den entsprechenden <i>SCAN-Unterkommandos</i> verändern.</p> <p>Mit dem SCAN-Editor können Sie alle Objekttypen, außer Maps und Data Areas und gesperrte Objekte (siehe <i>Locking of Source Objects</i>), verändern, und zwar im Full-Screen-Modus oder zeilenorientiert. Dadurch haben Sie die Möglichkeit, alle Zeilen zu editieren, nicht nur die mit dem SCAN-Kommando gefundenen. Sobald Sie ein Objekt fertig editiert haben, sollten Sie es speichern, bevor Sie vom normalen Editieren zur Fortsetzung der SCAN-Operation zurückkehren. Bevor die SCAN-Verarbeitung einen anderen Editor aufruft, erhalten Sie eine Aufforderung, die bis dato mit dem SCAN-Editor gemachten Änderungen zu bestätigen.</p>
Scan value	<p>Der Wert, nach dem gesucht werden soll.</p> <p><b>Anmerkung:</b> Bevor Sie das SCAN-Kommando verwenden, achten Sie bitte darauf, dass die automatische Umsetzung von Klein- in Großbuchstaben je nach Bedarf ein- bzw. ausgeschaltet ist. Benutzen Sie dazu das Terminal-Kommando %L.</p>	
Replace value	<p>Der Wert, durch den der gesuchte Wert ersetzt werden soll.</p> <p>Bei Maps, Data Areas, Recordings, Dialogen und gesperrten Objekten (siehe <i>Locking of Source Objects</i>) können Sie den gesuchten Wert nicht durch einen anderen ersetzen.</p>	
Library	<p>Die ID der Library, in der gesucht werden soll; standardmäßig ist dies Ihre aktuelle Library.</p> <p>Wenn Sie als Library SYSTEM angeben, wird die Library in der FUSER-Datei abgesucht. Wenn Sie den Namen einer anderen Library, die mit SYS anfängt, angeben, wird die Library in der FNAT-Datei abgesucht.</p>	
Object name	Das (bzw. die) zu durchsuchende(en) Objekt(e).	
	<i>Leerzeichen</i>	Alle Objekte der Library.
	*	
	<i>object-name</i> >	Alle Objekte, deren Namen größer oder gleich <i>object-name</i> sind.
	<i>object-name</i> <	Alle Objekte, deren Namen kleiner oder gleich <i>object-name</i> sind.
	<p>Sie können den Bereich der zu durchsuchenden Objekte auch einschränken, indem Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den Objektnamen verwenden, und zwar in der gleichen Weise wie beim Systemkommando LIST beschrieben.</p> <p>Siehe auch <i>Namenskonventionen für Objekte</i> in der Dokumentation Natural benutzen.</p>	

Feld	Eingabe	
Object type(s)	<p>Sie können die Suche auf bestimmte Objekttypen beschränken. Eine Auswahlliste der möglichen Objekttypen erhalten Sie, wenn Sie ein Fragezeichen (?) in dieses Feld eingeben.</p> <p>Wenn Sie dieses Feld leer lassen oder einen Stern (*) eingeben, ist die Suche nicht auf bestimmte Objekttypen beschränkt.</p>	
Absolute scan	Y	Es wird eine „absolute“ Suche durchgeführt, d.h. der zu suchende Wert wird in jeder möglichen Form gesucht, also auch als Teil einer längeren Zeichenkette.
	N	Standardmäßig ist die Suche nicht „absolut“.
Selection list	Y	Liefert Ihnen gemäß Ihren Angaben von Library, Name, Typ(en) für Code T oder S (siehe oben) zunächst eine Liste von Objekten, von denen Sie einzelne (durch Markieren mit einem beliebigen Zeichen) zur SCAN-Verarbeitung auswählen können.
	N	Es wird keine Auswahlliste angezeigt. Dies ist die Standardeinstellung.
Trace	Y	Die Trace-Funktion ist aktiviert.
	N	Die Trace-Funktion ist nicht aktiviert. Dies ist die Standardeinstellung.
Predict set number	<p>Die Nummer des Predict Set, der durchsucht werden soll.</p> <p><b>Anmerkung:</b> Dazu muss Predict installiert sein.</p> <p>Eine ein- oder zweistellige Nummer, die den gespeicherten Set kennzeichnet, welcher für die Namen der zu verarbeitenden Objekte verwendet werden soll. Ein gespeicherter Set wird mit der Save-Option des Systemkommandos <code>LIST XREF</code> erstellt.</p> <p>Der für <b>Library</b> angegebene Wert wird als Predict Set Library verwendet.</p> <p>Wenn Sie die Predict Set-Nummer angeben, müssen Sie bei <b>Object name</b> statt eines Namens ein Stern (*) angeben.</p> <p>Ausführliche Informationen zu Predict Sets finden Sie in der <i>Predict-Dokumentation</i>.</p>	
Predict set user	<p>Die Benutzerkennung des Benutzers, der den Predict Set erstellt hat.</p> <p><b>Anmerkung:</b> Dazu muss Predict installiert sein.</p> <p>Falls Sie keine Benutzerkennung angeben, wird der Wert der Systemvariablen <code>*USER</code> verwendet (siehe <i>Systemvariablen-Dokumentation</i>).</p>	



**Anmerkung:** Informationen zu möglichen Einschränkungen siehe *SCAN unter Natural Security*.

## SCAN-Editierkommandos

In der (den) Kommandozeile(n) des Ergebnis-Schirms können Sie folgende Editierkommandos verwenden:

Kommando	Funktion
Leerzeichen	Fortsetzen der normalen SCAN-Verarbeitung.
Q	Beenden der SCAN-Verarbeitung.
.	
EDIT	Objekt editieren im Full-Screen-Editor.
LIST	Anzeigen des Objekts, so wie es sich gegenwärtig im Editor-Arbeitsbereich befindet.
LET	Ignorieren aller Zeilenveränderungen, die erfolgt sind, seit das letzte Mal EINGABE gedrückt wurde.
I	Ignorieren des gegenwärtig abgesuchten Objekts, etwaige Änderungen des Objekts werden nicht gespeichert, die SCAN-Verarbeitung fährt mit dem nächsten Objekt fort.
.D	(D = Delete) Löschen einer Zeile. Neben der Zeile wird durch ein D angezeigt, dass sie gelöscht wurde.
.L	Ignorieren aller Veränderungen, die erfolgt sind, seit das letzte Mal EINGABE gedrückt wurde. Alle vorher mit dem Zeilenkommando .D gelöschten Zeilen werden ebenfalls wieder eingefügt.

### Editier-Regeln

- Die Zeilenlänge eines Source-Objekts auf dem Ergebnis-Schirm darf 72 Stellen nicht überschreiten. Zeilen mit mehr als 72 Zeichen werden mit einem L markiert und können nicht geändert werden.
- Wenn die **Replace value**-Option verwendet und/oder ein Objekt auf dem Ergebnis-Schirm verändert wird, wird das Objekt stets automatisch gesichert, es sei denn, Sie geben eines der Unterkommandos I, Q oder einen Punkt (.) ein, bevor Sie das nächste Objekt mit SCAN bearbeiten.
- Zeilen, die PASSW, PASSWORD=, CIPHER= oder CIPH= enthalten, werden bei der SCAN-Verarbeitung ignoriert.

## SCAN mittels Direktkommando

Sie können die SCAN-Funktionen auch per Direktkommando aufrufen (entweder im Online- oder im Batch-Modus), und zwar durch die Angabe von **Schlüsselwörtern** oder **positionalen Parametern**.

Im Batch-Modus verarbeitet das SCAN-Kommando nur eine Funktion pro Aufruf, um eventuelle nachteilige Auswirkungen, die durch die Eingabe ungültiger Daten entstehen können, so gering wie möglich zu halten.



**Vorsicht:** Wenn Sie Such-/Ersetzungswerte benutzen, die Zeichen in Kleinschreibung oder eingebettete Leerzeichen enthalten, empfehlen wir Ihnen dringend, die nachfolgenden Anleitungen zu beachten, um unerwartete Ergebnisse zu vermeiden.

» **Um SCAN mit Zeichen in Kleinschreibung oder eingebetteten Leerzeichen im Online-Modus zu benutzen:**

- 1 Geben Sie das Kommando `SCAN` ein, um das `SCAN`-Menü zu aktivieren.
- 2 Geben Sie alle erforderlichen Werte in die Eingabefelder des `SCAN`-Menüs ein.

» **Um SCAN mit Zeichen in Kleinschreibung oder eingebetteten Leerzeichen im Batch-Modus zu benutzen:**

- 1 Geben Sie das Kommando `SCAN` ein, um das `SCAN`-Menü aufzurufen.
- 2 Geben Sie die Daten in einer separaten Datenzeile und exakt in der Reihenfolge der Maskenfelder (siehe [Menü-Optionen](#)) ein, zum Beispiel:

```
SCAN  
S,MOVE LEFT,MOVE RIGHT,SYSTEM,*,N,P,N,N,1,SAG
```

In dem obigen Beispiel eines Batchjobs wird der Wert `MOVE LEFT` durch `MOVE RIGHT` bei allen Programmen des Predict Set 1 für den Benutzer `SAG` in der Library `SYSTEM` ersetzt.

Dieser Abschnitt behandelt folgende Themen:

## **SCAN mit Schlüsselwörtern**

Das Direktkommando `SCAN` besteht aus dem Kommando `SCAN` mit nachfolgenden Schlüsselwörtern.

Schlüsselwörter können in beliebiger Reihenfolge angegeben oder auch weggelassen werden. Wenn ein Schlüsselwort nicht benutzt wird, dann wird dafür der Standardwert übernommen.



**Vorsicht:** Bitte vermeiden Sie die gemischte Verwendung von Schlüsselwörtern und **positionalen Parametern** in der `SCAN`-Kommandosyntax, da dies zu unerwünschten Ergebnissen führen kann.

### **Syntax des Direktkommandos SCAN mit Schlüsselwörtern:**

```
SCAN keyword=value,...
```

**Syntax-Beschreibung:**

Schlüsselwort	Wert	Erklärung
FUNC	T, L oder S	Funktionscode. Einzelheiten siehe Beschreibung des Menüfeldes <i>Code</i> .
LIB	max. 8 Zeichen	Name der zu durchsuchenden Library. Einzelheiten siehe Beschreibung des Menüfeldes <i>Library</i> .  Standardwert ist der Name der aktuellen Library.
OBJ	max. 9 Zeichen	Objektname. Die zu durchsuchenden Objekt-Sourcecodes. Einzelheiten siehe Beschreibung des Menüfeldes <i>Object name</i> .
TYPE	max. 14 Zeichen	Objekttyp. Der (oder die) für die Auswahl zu verwendende(n) Objekttyp(en). Einzelheiten siehe Beschreibung des Menüfeldes <i>Object type(s)</i> .
SVAL	max. 32 Zeichen	Suchwert. Die Zeichenkette ( <b>ohne eingebettete Leerzeichen</b> ), nach der in jedem Objekt-Sourcecode gesucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Scan value</i> .
RVAL	max. 32 Zeichen	Der Ersetzungswert. Die Zeichenkette ( <b>ohne eingebettete Leerzeichen</b> ), die durch den Suchwert ersetzt werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Replace value</i> .
ABSOL	<u>Y</u> oder <u>N</u>	Absolute Suche. Einzelheiten siehe Beschreibung des Menüfeldes <i>Absolute scan</i> .
SETNO	2 Stellen	Nummer des Predict Set, der durchsucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Predict set number</i> .
SETUSER	8 Zeichen	Benutzerkennung des Predict Set, der durchsucht werden soll, d.h., die Benutzerkennung des Benutzers, der den Predict Set erstellt hat. Einzelheiten siehe Beschreibung des Menüfeldes <i>Predict set user</i> .

**Beispiele für Direktkommando SCAN mit Schlüsselwörtern**

```
SCAN FUNC=S,LIB=SYSTEM,OBJ=PGM0*,TYPE=S,SVAL=FETCH ↵
```

```
SCAN FUNC=S,SVAL=WRITE,RVAL=PRINT,OBJ=PGM1
```

```
SCAN FUNC=S,SETNO=1,TYPE=P,SVAL=GLOBAL
```

**SCAN mit positionalen Parametern**

Das Direktkommando `SCAN` mit positionalen Parametern besteht aus dem Kommando `SCAN` und nachfolgenden Parametern, die jedes einzelne Kommandozeilenwort entsprechend seiner Position innerhalb des Kommandos referenzieren. Die Reihenfolge der positionalen Parameter muss exakt eingehalten werden.

Wenn ein positionaler Parameter nicht angegeben wird, dann wird der Standardwert übernommen. In diesem Fall muss ein Komma gesetzt werden, um anzugeben, dass ein positionaler Parameter weggelassen wurde. Siehe *Beispiele für Direktkommando SCAN mit positionalen Parametern* weiter unten.



**Vorsicht:** Bitte vermeiden Sie die gemischte Verwendung von **Schlüsselwörtern** und positionalen Parametern in der SCAN-Kommandosyntax, da dies zu unerwünschten Ergebnissen führen kann.

### Syntax des Direktkommandos SCAN mit positionalen Parametern:

```
SCAN
func,scan-value,replace-value,library,object-name,object-type,absolute,set-number,set-user
```

### Syntax-Beschreibung:

Positionaler Parameter	Wert	Erklärung
<i>func</i>	T, L oder S	Funktionscode. Einzelheiten siehe Beschreibung des Menüfeldes <i>Code</i> .
<i>scan-value</i>	max. 32 Zeichen	Suchwert. Die Zeichenkette ( <b>ohne eingebettete Leerzeichen</b> ), nach der in jeder Objekt-Source gesucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Scan value</i> .
<i>replace-value</i>	max. 32 Zeichen	Einzusetzender Wert. Die Zeichenkette ( <b>ohne eingebettete Leerzeichen</b> ), die durch den Suchwert ersetzt werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Replace value</i> .
<i>library</i>	max. 8 Zeichen	Name der zu durchsuchenden Library. Einzelheiten siehe Beschreibung des Menüfeldes <i>Library</i> .  Standardwert ist der Name der aktuellen Library.
<i>object-name</i>	max. 9 Zeichen	Objektname. Die zu durchsuchenden Objekt-Sourcecodes. Einzelheiten siehe Beschreibung des Menüfeldes <i>Object name</i> .
<i>object-type</i>	max. 14 Zeichen	Objekttyp. Der (oder die) für die Auswahl zu verwendende(n) Objekttyp(en). Einzelheiten siehe Beschreibung des Menüfeldes <i>Object type(s)</i> .
<i>absolute</i>	Y oder <u>N</u>	Absolute Suche. Einzelheiten siehe Beschreibung des Menüfeldes <i>Absolute scan</i> .
<i>set-number</i>	2 Stellen	Nummer des Predict Set, der durchsucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes <i>Predict set number</i> .
<i>set-user</i>	8 Zeichen	Benutzerkennung des Predict Set, der durchsucht werden soll, d.h., die Benutzerkennung des Benutzers, der den Predict Set erstellt hat. Einzelheiten siehe Beschreibung des Menüfeldes <i>Predict set number</i> .

### Beispiele für Direktkommando SCAN mit positionalen Parametern:

```
SCAN S,ABC,,SYSTEM,PGM0*,P,N
```

```
SCAN S,ABC,DEF,SYSTEM,PGM1,*,Y
```

```
SCAN T,ABC,,SYSTEM,*,7,N,1,SAG
```

## Individuelles SCAN-Profil definieren

---

Sie können für das `SCAN`-Kommando ein individuelles Profil definieren. Dazu bietet das Natural-Systemkommando `SCAN` das Unterkommando `SCANPROF` bzw. im `SCAN`-Menü die PF-Taste `PF10` (Prof).

Das Unterkommando ruft ein Tool auf, das Ihnen folgende Funktionen zur Profilverwaltung zur Verfügung stellt:

- Erstellen eines neuen `SCAN`-Profils mit intern verwendeten Standardwerten, falls das `SCAN`-Profil noch nicht existiert,
- Angabe allgemeiner Werte für einzelne Benutzer für die Profilparameter im `SCAN`-Profil (unterliegt der Kontrolle durch Natural Security),
- Angabe individueller Werte für einzelne Benutzer für die Profilparameter im `SCAN`-Profil (unterliegt der Kontrolle durch Natural Security),
- Anzeige einer Beschreibung sowie von Hilfe-Informationen zu jedem Profilparameter. Das Profilverwaltungs-Tool zeigt einen Bildschirm mit den allgemeinen Profilparametern und ihren aktuellen Werten.

Sie können eine PF-Taste oder ein Kommando benutzen, um zu den Einstellungen Ihrer eigenen Profildaten zu wechseln.

Durch Eingabe von Zeilenkommandos können Sie Parameter in dem benutzerspezifischen Profil hinzufügen oder entfernen.

Weitere Informationen erhalten Sie, wenn Sie die Hilfe-Bildschirme zu den einzelnen Feldern aufrufen.

Wenn von der Software AG neue Parameter hinzugefügt wurden, können Sie diese Änderungen mit dem internen Profilverwaltungs-Tool-Kommando `UPDATE` in das `SCAN`-Profil integrieren. Das `SCAN`-Profil selbst hat den Namen `SCANPROF` und befindet sich in der Library `SYSLIB`. Das Standard-Profil, das für Aktualisierungen des `SCAN`-Profils benutzt wird, hat den Namen `SCANDEFP` und befindet sich ebenfalls in der Library `SYSLIB`.

### SCAN-Profil-Parameter

Die im `SCAN`-Profil enthaltenen Parameter entsprechen den Feldern, die sich auf dem `SCAN`-Menü-Bildschirm befinden, siehe [Menü-Optionen](#).

## SCAN unter Natural Security

---

In einer Natural Security-Umgebung können Sie ein Library-Profil angeben, um die Benutzung des Systemkommandos `SCAN` für eine bestimmte Library zu unterbinden oder die Benutzung einzuschränken, um Source-Änderung zu vermeiden. Sie haben folgende Möglichkeiten:

- Sie können die Benutzung des Systemkommandos `SCAN` unterbinden.
- Sie können die Benutzung des Systemkommandos `SCAN` erlauben, aber die Benutzung der Systemkommandos `LIST` und `EDIT` unterbinden.
- Sie können die Benutzung der Ersetzungsoption („Replace“) des Systemkommandos `SCAN` unterbinden. Zusätzlich können Sie die Benutzung des Systemkommandos `SAVE` unterbinden, um Source-Änderungen im Full-Screen-Editor zu vermeiden.

Ausführliche Informationen zur Einschränkung der Benutzung des Systemkommandos `SCAN` siehe *Command Restrictions* im Abschnitt *Library Maintenance* in der *Natural Security*-Dokumentation.

Falls für die Library nur der Structured-Modus erlaubt ist, können Objekte im Reporting-Modus durchsucht, aber nicht geändert werden.



## 49 SCRATCH

---

Dieses Kommando wird nur aus Kompatibilitätsgründen unterstützt. Es empfiehlt sich, statt des SCRATCH-Kommandos das DELETE-Kommando zu verwenden, da DELETE mehr Flexibilität sowie Schutz gegen versehentliches Löschen bietet.

---

# 50

## SETUP

---

■ Syntax-Erklärung .....	204
■ Beispiel für SETUP/RETURN .....	205

SETUP [*application-name*] [*command-name*] [I]

Mit dem Systemkommando **SETUP** können Sie Rückkehrpunkte setzen, zu denen Sie dann später mit dem Systemkommando **RETURN** zurückkehren können. Das erlaubt es Ihnen, während einer Natural-Session problemlos von einer Anwendung in eine andere zu gelangen.

Programmierschnittstelle (API): USR1026N. Siehe *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

## Syntax-Erklärung

Dieser Abschnitt beschreibt die Kommando-Syntax und die Parameter, die Sie mit dem **SETUP**-Kommando absetzen können. Wenn Sie einen Parameter auslassen wollen, müssen Sie das Eingabe-Begrenzungszeichen benutzen, um den Anfang des nächsten Parameters oder der nächsten Parameter zu markieren.

SETUP	Falls Sie <b>SETUP</b> ohne Parameter eingeben, erhalten Sie ein Menü, auf dem Sie die entsprechenden Informationen über einen Rückkehrpunkt eingeben können.
<i>application-name</i>	<p>Der Name der Anwendung, an die die Kontrolle übergeben werden soll. Der Name darf bis zu 8 Stellen lang sein (Format/Länge A8).</p> <p>Falls Sie keinen <i>application-name</i> angeben, wird kein <b>LOGON</b>-Kommando ausgeführt. Dadurch haben Sie die Möglichkeit, mehrere Rückkehrpunkte innerhalb einer Anwendung zu setzen.</p> <p>Falls Sie als <i>application-name</i> einen Stern (*) eingeben, wird der Inhalt der Natural-Systemvariablen *LIBRARY-ID (zum Zeitpunkt des <b>SETUP</b>-Kommandos) verwendet, um damit ein <b>LOGON</b>-Kommando auszuführen, sobald das <b>RETURN</b>-Kommando benutzt wird.</p>
<i>command-name</i>	<p>Der Name des Kommandos, das ausgeführt werden soll, sobald die Kontrolle an die Anwendung übergeben wird. Der Name darf bis zu 60 Stellen lang sein (Format/Länge A60).</p> <p>Wenn Sie kein Kommando angeben, wird nach dem <b>LOGON</b>-Kommando kein weiteres ausgeführt. Dies empfiehlt sich, wenn zu einer Anwendung zurückgekehrt wird, für die in Natural Security eine Startup-Transaktion definiert ist.</p> <p>Falls Sie als Kommando einen Stern (*) eingeben, wird der Inhalt der Natural-Systemvariablen *STARTUP (zum Zeitpunkt des <b>SETUP</b>-Kommandos) verwendet und als Kommando ausgeführt, sobald das <b>RETURN</b>-Kommando benutzt wird.</p>

I	<p>Wenn Sie ein I eingeben, werden alle mit vorherigen <code>SETUP</code>-Kommandos definierten Rückkehrpunkte gelöscht und die mit diesem <code>SETUP I</code>-Kommando angegebene Anwendung als neue Ausgangsanwendung definiert.</p> <p>In einer Nicht-Security-Umgebung gilt: Wenn Sie aus der Library <code>SYSTEM</code> heraus in eine andere Library wechseln und es ist noch kein Rückkehrpunkt definiert, dann wird diese andere Library automatisch als Ausgangsanwendung (Initial Return Point) definiert.</p>
---	--

## Beispiel für `SETUP/RETURN`

1. Der Benutzer beginnt eine Natural-Session und gelangt in die als Standardanwendung definierte Anwendung `APPL1`.

Auf Stufe 1 wird `APPL1` als Rückkehrpunkt definiert.

2. Der Benutzer begibt sich mit dem Kommando `LOGON APPL2` in eine andere Anwendung.
3. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

`SETUP *,MENU` (Definieren eines Rückkehrpunktes)

`LOGON APPL3` (Wechsel zu einer anderen Anwendung)

Auf Stufe 2 wird `APPL2, STARTUP MENU` als Rückkehrpunkt definiert.

4. Der Benutzer wechselt mit dem Kommando `LOGON APPL4` in eine andere Anwendung.
5. Der Benutzer drückt die PF-Taste, die mit dem Kommando `RETURN` belegt ist. Natural führt folgende Kommandos aus:

`LOGON APPL2`

`MENU`

Der Benutzer kehrt zu `APPL2` zurück; der auf Stufe 2 gesetzte Rückkehrpunkt wird gelöscht.

6. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

`SETUP *,MENU`

`LOGON APPL5`

Auf Stufe 2 wird `APPL2, STARTUP MENU` als Rückkehrpunkt definiert.

7. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

`SETUP *,MENU`

`LOGON APPL6`

Auf Stufe 3 wird APPL5, STARTUP MENU als Rückkehrpunkt definiert.

8. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

```
SETUP *,MENU
```

```
LOGON APPL7
```

Auf Stufe 4 wird APPL6, STARTUP MENU als Rückkehrpunkt definiert.

9. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

```
SETUP *,MENU
```

```
LOGON APPL8
```

Auf Stufe 5 wird APPL7, STARTUP MENU als Rückkehrpunkt definiert.

10. Der Benutzer führt ein Programm aus, das zwei Kommandos ausführt:

```
SETUP *,MENU
```

```
LOGON APPL9
```

Auf Stufe 6 wird APPL8, STARTUP MENU als Rückkehrpunkt definiert.

11. Der Benutzer begibt sich mit dem Kommando RETURN 2 zwei Stufen zurück.

Natural kehrt zu APPL7 (Stufe 5) zurück, weil das die vorletzte Session war (alle Informationen zu APPL8 sind jetzt verloren). Stufe 6 (APPL8) wird gelöscht. Der Rückkehrpunkt der Stufe 5 (APPL7) wird aktiviert und die Stufe gelöscht.

12. Der Benutzer begibt sich mit dem Kommando RETURN eine Stufe zurück.

Der Rückkehrpunkt der Stufe 4 (APPL6) wird aktiviert, der Rückkehrpunkt der Stufe wird gelöscht. Natural kehrt zu APPL6 zurück, weil dies die Session vor APPL7 war.

13. Der Benutzer begibt sich mit dem Kommando RETURN eine Stufe zurück.

Der Rückkehrpunkt der Stufe 3 (APPL5) wird aktiviert, der Rückkehrpunkt der Stufe wird gelöscht. Natural kehrt zu APPL5 zurück, weil dies die Session vor APPL6 war.

14. Der Benutzer setzt das Kommando RETURN 1 ab.

Der Rückkehrpunkt der Stufe 2 (APPL2) wird gelöscht, der Rückkehrpunkt der Stufe 1 (APPL1) wird aktiviert.

# 51

## SHOWDBS

---

■ Entsprechendes API USR8211N .....	209
-------------------------------------	-----

SHOWDBS

Dieses Kommando dient dazu, Informationen über die in Ihrer Umgebungen zur Verfügung stehenden Datenbanken und die für sie definierten Optionen (z.B. Schreibschutz) anzuzeigen.

Beispiel:

+-----SHOWDBS - Defined Databases-----+

!

! DBID DB Type DB Options

!

!-----

!

! 0 ADABAS V82

!

! 10 ADABAS V82

!

! 111 ADAV7 READ

!

! 147 INCORE DB

!

! 148 PROCESS

!

! 170 VSAM

!

! 200 DB2

!

! 246 DLI

!

! 247 DLI

!

! 249 CONNECX

!

! 250 DB2

!

! Default DB type ..... ADABAS V82

!

!

!-----

Top of list

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

Exit + ++ Canc

Die angezeigten Informationen werden aus den Einstellungen des DB-Parameters (bzw. des entsprechenden NTDB-Makros) gewonnen.

DBID	Die Datenbankkennung.
DB Type	Der Datenbanktyp, siehe <i>Mögliche Datenbanktypen</i> .
Default DB type	Der Standard-Datenbanktyp.
DB Options	Eine oder mehrere für die Datenbank definierte Optionen, siehe <i>Mögliche Datenbank-Optionen</i> .

Mit den nachfolgend aufgeführten PF-Tasten können Sie (je nach vorhandender Seitenzahl) in dem **SHOWDBS**-Fenster nach unten bzw. nach oben blättern oder die Funktion beenden:



PF3	Funktion beenden und Fenster schließen.
PF6	Erste Seite.
PF7	Vorige Seite.
PF8	Nächste Seite.
PF9	Letzte Seite.
PF12	Funktion abbrechen und Fenster schließen.

## Entsprechendes API USR8211N

---

Die Anwendungsprogrammierungsschnittstelle `USR8211N` in der System-Library `SYSEXT` liefert die gleichen Informationen wie das Systemkommando `SHOWDBS`.



## SHOWSU

SHOWSU

Dieses Kommando dient zur Anzeige einer Liste aller wählbaren Merkmale („Selectable Units“), die in Ihrer Natural-Umgebung zur Verfügung stehen und zeigt an, welche Merkmale mit dem Profilparameter `SELUNIT` angegeben und aktiviert worden sind.

Weitere Informationen siehe *Selectable Units for New Natural Features* in der *Operations* -Dokumentation.

### Beispiel für eine Anzeige:

[illegible]



## 53 SQLDIAG

---



**Anmerkung:** Dieses Kommando steht nur bei Natural for DB2 zur Verfügung.

Das Kommando `SQLDIAG` liefert Diagnose-Informationen (andere als beim `GET DIAGNOSTICS`-Statement) zu dem zuletzt ausgeführten SQL-Statement. Diese Diagnose-Informationen werden gesammelt, während das vorangehende SQL-Statement ausgeführt wird. Einige der mit dem `GET DIAGNOSTICS`-Statement erzielbaren Informationen sind auch in der SQLCA vorhanden.

Ausführliche Informationen über die zurückgelieferten Diagnose-Informationen finden Sie in der IBM DB2-Dokumentation zum `GET DIAGNOSTICS`-Statement.

Felder, vor deren Namen ein Plus-Zeichen (+) steht, können mehr Daten enthalten, als auf dem Bildschirm angezeigt wird. Um den vollständigen Inhalt eines solchen Feldes anzuzeigen, stellen Sie den Cursor auf das Feld (Beschreibung bzw. Daten) und drücken `Enter` oder geben Sie in der Kommandozeile die Abkürzung des Feldes (die Großbuchstaben aus der Beschreibung) mit dem Plus-Zeichen (+) als Präfix ein. Beispiel: `+SN` ergibt die Anzeige des vollständigen Wertes des Feldes `SERVER_NAME`.

Das Kommando `SQLDIAG` können Sie am Natural-Eingabeaufforderungszeichen `NEXT` oder innerhalb eines Natural-Programms (mittels eines `FETCH`-Statements) absetzen.

Beispiel-Bildschirme **SQLDIAG Diagnostic Information:**

```

11:03:12          *** SQLDIAG Diagnostic Information ***          2006-04-15
                    - Statement Information -

DB2_Last_Row ..... 0
DB2_Number_Parameter_Markers ..... 0
DB2_Number_Result_Sets ..... 0
DB2_Return_Status ..... 0
DB2_SQL_Attr_Cursor_Hold ..... _Rowset .. _Scrollable ...
                                   _Type .. _Sensitivity ..
DB2_Number_Rows ..... 0
Row_Count ..... 0

More .....

Number ..... 1

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

```

```

11:09:49          *** SQLDIAG Diagnostic Information ***          2006-04-15
                    - Condition Information 1 -

+Server_Name ..... DAEFDB28
+Cursor_Name .....
  DB2_Error_Code1 ..... -500  DB2_Error_Code2 ... 0
    _Code3 ..... 0          _Code4 ... -1
  DB2_Internal_Error_Pointer .. -500 +DB2_Sqlerrd1(-6) .. -500
  DB2_Module_Detecting_Error .. DSNXOTL
+DB2_Ordinal_Token_1 ..... HGK.DEMO
  DB2_Row_Number ..... 0
  DB2_Line_Number ..... 0
  DB2_Returned_SQLCode ..... -204
  DB2_Reason_Code ..... 0
  Returned_SQLState ..... 42704
  DB2_Message_ID ..... DSN00204E
  Message_Octet_Length ..... 0
+Message_Text ..... HGK.DEMO IS AN UNDEFINED NAME

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

```







# 54

## SQLERR

---

SQLERR



### Anmerkungen:

1. Dieses Kommando steht nur bei Natural for DB2, Natural SQL Gateway und Natural for SQL/DS zur Verfügung.
2. Je nachdem ob das Kommando bei Natural for DB2, Natural SQL Gateway oder Natural for SQL/DS abgesetzt wird, können geringfügige Unterschiede vorhanden sein, die in der nachfolgenden Beschreibung entsprechend gekennzeichnet sind.

Das Kommando `SQLERR` dient dazu, Diagnose-Informationen zum zuletzt aufgetretenen SQL-Fehler zu erhalten.

Beim Auftreten eines SQL-Fehlers gibt Natural eine entsprechende Fehlermeldung aus. Wenn Sie das Kommando `SQLERR` eingeben, werden die folgenden Informationen zum zuletzt aufgetretenen Fehler angezeigt:

- die Nummer der Natural-Fehlermeldung,
- der entsprechende Reason Code (falls zutreffend),
- bei Natural for DB2: die von DB2 zurückgegebenen Variablen `SQLSTATE` und `SQLCODE`,
- bei Natural SQL Gateway: der von der ConnecX SQL Engine bzw. dem SQL-Datenbanksystem zurückgegebene `SQLCODE`,
- bei Natural for SQL/DS: die von SQL/DS zurückgegebenen Variablen `SQLSTATE` und `SQLCODE`,
- die entsprechende Fehlermeldung.

Das Kommando `SQLERR` können Sie am Natural-Eingabeaufforderungszeichen `NEXT` oder innerhalb eines Natural-Programms (mittels eines `FETCH`-Statements) absetzen.

**Beispiel-Bildschirm SQLERR Diagnostic Information (bei Natural for DB2)**

```

***** SQLERR Diagnostic Information *****
----- NATURAL SQL Interface Codes -----
Return Code: 3700      Reason Code: 0      SQLSTATE : 52003      SQLCODE: -206
----- SQLCA -----
SQLERP (DB2 Sub routine where error occurred)      : DSNXOGP
SQLERRD (DB2 Internal State)
  RDS Return Code      :      700
  DBSS Return Code     :      0
  Number of Rows Processed      :      0
  Estimated Cost        :      11.2
  Syntax error on PREPARE or EXECUTE IMMEDIATE     :      0
  Buffer Manager ERROR Code     :      0
SQLWARN (Warning Flags)
  Data truncated
  Null Values ignored (AVG,SUM,MAX,MIN)            :
  No. of columns greater than no. of host variables :
  UPDATE/DELETE without WHERE clause              :
  SQL Statement not valid in DB2                  :
  Adjustment to DATE/TIMESTAMP Variable made      :
DB2 Error Message :
DSNT4081 SQLCODE = -206, ERROR: THE OBJECT TABLE OR VIEW OF THE INSERT,
      DELETE, OR UPDATE STATEMENT IS ALSO IDENTIFIED IN A FROM CLAUSE

```

**Beispiel-Bildschirm SQLERR Diagnostic Information (bei Natural SQL Gateway)**

```

*** SQLERR Diagnostic Information ***
----- Natural SQL Interface Codes -----
Return Code: 3700      Reason Code: 0      ZZN01      SQLCODE: -4017
----- SQLCA -----
SQLERRD (Additional Information)
  Number of Rows Processed      :      0
SQLWARN (Warning Flags)
  Data truncated
  No. of columns greater than No. of host variables :
CNX Error Message :
4017(E): SERVER ERROR: ODBC:(HY000) NATIVE:(0) : Ambiguous table reference: (D
EMO) ? PERS_ID , NAME , ADDRESS , DATEOFBIRTH , SALARY FROM << Syntax Error >>
NSB.DEMO ?.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Error Exit  Expl      Parms  -      +      Prev  Next  Canc

```

### Beispiel-Bildschirm SQLERR Diagnostic Information (bei Natural for SQL/DS)

```

*** SQLERR Diagnostic Information ***
----- NATURAL SQL Interface Codes -----
Return Code: 3700          Reason Code: 0          SQLCODE: -204
----- SQLCA -----
SQLERRP (Adabas SQL Subroutine where error occurred) : ARIXOCA
SQLERRD (Adabas SQL Internal State)
  RDS Return Code          :          100
  DBSS Return Code         :           0
  Number of Rows Processed :           0
  Estimated Cost           :          1.0
  Syntax error on PREPARE or EXECUTE IMMEDIATE :           0
  Buffer Manager ERROR Code :           0
SQLWARN (Warning Flags)
  Data truncated
  Null Values ignored(AVG,SUM,MAX,MIN) :
  No. of columns greater than no. of host variables :
  UPDATE/DELETE without WHERE clause :
  SQL statement causes a performance degradation :
  Adjustment to DATE/TIMESTAMP Variable made :
SQL/DS Error Message :
SAG.SYSTABLES not found in system catalog

```



# 55 STOW

STOW [*object-name* [*library-id*]]

Verwandte Kommandos: [SAVE](#) | [CATALOG](#)

Das Kommando STOW dient dazu, ein Natural-Objekt (sowohl in Sourceform als auch in Objektform) in der Natural-Systemdatei zu katalogisieren (kompilieren) und zu speichern. Es hat die gleiche Wirkung wie ein CATALOG-Kommando mit anschließend abgesetztem SAVE-Kommando.

Siehe auch:

- *Natural-Compiler in Natural System-Architektur*
- *Namenskonventionen für Objekte in Natural benutzen*

STOW	Wenn Sie das Kommando ohne <i>object-name</i> benutzen, wird das im Arbeitsbereich des Editors vorhandene Sourceobjekt und der erzeugte Code unter demselben Namen in der aktuellen Library gespeichert. Vorhandener Source- und Objektcode werden ersetzt.
STOW <i>object-name</i>	Verwenden Sie diese Kommandosyntax, um ein neues Objekt (Source und generierter Code) namens <i>object-name</i> in der aktuellen Library zu speichern. Falls das Objekt schon vorhanden ist, wird das Kommando zurückgewiesen.
STOW <i>object-name</i> <i>library-id</i>	Wenn Sie sowohl <i>object-name</i> als auch <i>library-id</i> angeben, wird ein neues Objekt angelegt und unter diesem Namen in der angegebenen Library gespeichert. Falls das Objekt in Sourceform oder katalogisierter Form vorhanden ist, wird das Kommando zurückgewiesen.



**Anmerkung:** Falls im Parametermodul eine ungültige Systemdatei `FDIC` angegeben ist, erscheint eine entsprechende Natural-Fehlermeldung, wenn das STOW-Kommando abgesetzt wird.



# 56

## STRUCT

---

■ Generate Structured Source into Work Area .....	224
■ Display Structure of Source .....	227
■ Print Structure of Source .....	228
■ Write Structure of Source into Work Area .....	229

### STRUCT

Das Kommando `STRUCT` erfüllt zwei Aufgaben:

- Sie können es dazu verwenden, die Sourcecode-Zeilen eines Programms entsprechend der Programmstruktur einzurücken.
- Diverse zusätzliche Anzeigen verdeutlichen die Programmstruktur und erlauben es Ihnen so, etwaige Inkonsistenzen in der Struktur aufzuspüren.

Da das `STRUCT`-Kommando die Natural-Sourcen unabhängig davon verarbeitet, ob sie tatsächlich katalogisiert werden können oder nicht, wird die Source nicht auf syntaktische Korrektheit analysiert. Obwohl das `STRUCT`-Kommando sorgfältig strukturierte Source-Zeilen liefert, können Zeilen vorhanden sein, die unklar sind und die nicht wie erwartet strukturiert werden.

Folgende Arten von Statements sind vom `STRUCT`-Kommando betroffen:

- Verarbeitungsschleifen (`READ`, `FIND`, `FOR` usw.),
- Statement-Blöcke mit Bedingungen (`AT BREAK`, `IF`, `DECIDE FOR` usw.),
- `DO/DOEND`-Statement-Blöcke,
- `DEFINE DATA`-Statement-Blöcke,
- interne Subroutinen.

Wenn Sie das `STRUCT`-Kommando eingeben, erhalten Sie das `STRUCT`-Menü, von dem Sie folgende Funktionen auswählen können:

## Generate Structured Source into Work Area

---

Mit dieser Funktion können Sie ein Source-Programm einrücken, so dass die Einrückung der Sourcecode-Zeilen die Struktur des Programms widerspiegelt.

Diese Funktion entspricht der des Editorkommandos `STRUCT`.

Bei der Einrückung wird die Länge der Sourcecode-Zeilen berücksichtigt; d.h. eine eingerückte Zeile wird nicht über den rechten Rand hinausgeschoben; selbst wenn eine „korrekte“ Einrückung dies erfordern würde, wird eine Zeile nicht über den Rand hinausgeschoben, sondern nur soweit nach rechts wie möglich.

Die **Generate**-Funktion bietet Ihnen folgende Optionen:



Feld	Erklärung
Source Name	In dieses Feld geben Sie den Namen der Source ein, die Sie einrücken möchten. Die betreffende Source wird dann von der Systemdatei in den Editor-Arbeitsbereich gelesen und strukturell eingerückt.  Wenn sie keinen Namen angeben, wird das gerade im Editor-Arbeitsbereich befindliche Objekt eingerückt. Wenn der Arbeitsbereich leer ist, müssen Sie einen Namen angeben.
Shift setting	In diesem Feld können Sie angeben, um wieviel Stellen (1 bis 9) die Zeilen jeweils eingerückt werden sollen. Standardmäßig wird um 2 Stellen eingerückt.
Align Comments	Y Jede Kommentarzeile wird soweit eingerückt wie die darüberstehende Statement-Zeile; Ausnahme: Kommentarzeilen, die am Anfang einer Zeile anfangen, werden nicht eingerückt.
	N Kommentarzeilen werden nicht eingerückt.
	L Kommentarzeilen werden linksbündig ausgerichtet.
Display Messages	Y Sie erhalten eine Meldung, dass das strukturierte Programm in den Arbeitsbereich generiert wurde, sowie eine Liste der Sourcecode-Zeilen, die nicht „korrekt“ eingerückt (siehe oben) werden konnten.
	N Es wird keine derartige Meldung ausgegeben.
Return to STRUCT	Y Nach Ausführen der <b>Generate</b> -Funktion gelangen Sie zum STRUCT-Menü zurück.
	N Nach Ausführen der <b>Generate</b> -Funktion gelangen Sie zu dem Schirm zurück, auf dem Sie das STRUCT-Kommando eingegeben hatten.



**Anmerkung:** Ein im Reporting Mode geschriebenes Programm wird anders eingerückt als ein im Structured Mode geschriebenes.

### Teilweise Einrückung

Mit den Spezial-Statements `/*STRUCT OFF` und `/*STRUCT ON` können Sie bestimmte Abschnitte Ihres Source-Programms von der Einrückung ausschließen. Die beiden Statements müssen jeweils am Anfang einer Sourcecode-Zeile stehen. Wenn Sie die **Generate**-Funktion ausführen, werden die Zeilen zwischen diesen beiden Statements nicht davon betroffen; sie bleiben, wie sie waren.

### Beispiel für strukturelle Einrückung

Programm, bevor es eingerückt wird:

```
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
2 PERSONNEL-ID
2 FULL-NAME
3 FIRST-NAME
3 NAME
1 VEHI VIEW OF VEHICLES
2 PERSONNEL-ID
2 MAKE
END-DEFINE
FIND EMPL WITH NAME = 'ADKINSON'
IF NO RECORDS FOUND
WRITE 'NO RECORD FOUND'
END-NOREC
FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
END-FIND
END-FIND
END
```

Dasselbe Programm, nachdem die Funktion **Generate Structured Source** angewandt wurde:

```
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 FULL-NAME
    3 FIRST-NAME
    3 NAME
1 VEHI VIEW OF VEHICLES
  2 PERSONNEL-ID
  2 MAKE
END-DEFINE
FIND EMPL WITH NAME = 'ADKINSON'
  IF NO RECORDS FOUND
    WRITE 'NO RECORD FOUND'
  END-NOREC
  FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
    DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
  END-FIND
END-FIND
END
```

## Display Structure of Source

Mit dieser Funktion können Sie den Sourcecode eines Objekts zusammen mit verschiedenen Informationen, die die Struktur des Objekts verdeutlichen, anzeigen.

Die **Display**-Funktion bietet Ihnen die folgenden Optionen:

Feld	Erklärung	
Source Name	In dieses Feld geben Sie den Namen der Source ein, die Sie anzeigen möchten. Die betreffende Source wird dann von der Systemdatei in den Editor-Arbeitsbereich gelesen und angezeigt.  Wenn sie keinen Namen angeben, wird das gerade im Editor-Arbeitsbereich befindliche Objekt angezeigt. Wenn der Arbeitsbereich leer ist, müssen Sie einen Namen angeben.	
Display Compressed	Y	Sourcecode-Zeilen auf derselben strukturellen Ebene werden nicht angezeigt. Es werden nur die Zeilen gezeigt, die eine Änderung in der Strukturtafel am rechten Bildschirmrand bewirken. Anhand der Lücke in der Abfolge der Zeilennummern können Sie sehen, wieviele Zeilen zwischen zwei gezeigten Zeilen nicht gezeigt werden.
	N	Alle Sourcecode-Zeilen werden angezeigt.
Return to STRUCT	Y	Nach Ausführen der Display-Funktion gelangen Sie zum STRUCT-Menü zurück.
	N	Nach Ausführen der Display-Funktion gelangen Sie zu dem Schirm zurück, auf dem Sie das STRUCT-Kommando eingegeben hatten.

Folgende Informationen werden angezeigt:

Line Numbers	Bei jedem Statement, das einen Statement-Block beendet, wird links neben dem Sourcecode die Sourcecode-Zeilenummer des Statements, das den betreffenden Block initiiert, angezeigt.
Structure Table	Rechts neben dem Sourcecode wird eine Strukturtafel angezeigt, die Aufschluß über offene Statement-Blöcke gibt. Für jeden offenen Block wird ein einzelner Buchstabe angezeigt. Die verschiedenen Buchstaben beziehen sich auf verschiedene Arten von Statements (für eine Erklärung der einzelnen Buchstaben drücken Sie PF1). Etwaige Inkonsistenzen in der Struktur des Sourcecodes werden durch eine entsprechende Meldung in der Strukturtafel angezeigt.

**Beispiel für Anzeige mit Struktur-Informationen:**

```
14:17:47 - Structured Source ABC in Library XYZ - 2003-02-04
0010      DEFINE DATA LOCAL                      *0
0020      1 EMPL VIEW OF EMPLOYEES                *0
0030          2 PERSONNEL-ID                      *0
0040          2 FULL-NAME                         *0
0050              3 FIRST-NAME                   *0
0060              3 NAME                         *0
0070      1 VEHI VIEW OF VEHICLES                *0
0080          2 PERSONNEL-ID                      *0
0090          2 MAKE                             *0
0100 0010 END-DEFINE                             *0
0110      FIND EMPL WITH NAME = 'ADKINSON'        *F
0120          IF NO RECORDS FOUND                 *FJ
0130              WRITE 'NO RECORD FOUND'         *FJ
0140 0120 END-NOREC                             *FJ
0150      FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID *FF
0160          DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE *FF
0170 0150 END-FIND                             *FF
0180 0110 END-FIND                             *F
0190      END                                    *
PF1=Help, PF2=Menu, PF3=Exit, PF6=Top, PF12=Cancel.
```

Der aktuelle Inhalt des Arbeitsbereichs wird von der angezeigten Source nicht beeinflusst.

---

## Print Structure of Source

Mit dieser Funktion können Sie den Sourcecode eines Objekts zusammen mit seinen Struktur-Informationen ausdrucken.

Die **Print**-Funktion entspricht der Funktion **Display Structure of Source**, nur dass die Ausgabe nicht auf dem Bildschirm angezeigt, sondern auf einem Drucker ausgedruckt wird.

Mit der **Print**-Funktion haben Sie die gleichen Optionen wie bei der **Display**-Funktion.

## Write Structure of Source into Work Area

---

Mit dieser Funktion können Sie eine Source von der Systemdatei lesen und zusammen mit ihren Struktur-Informationen sowie mehreren Zeilen (mit Zeilennummer 0000) am Anfang der Source, die die Struktur-Informationen erklären, in den Editor-Arbeitsbereich laden.

Mit der **Write**-Funktion haben Sie die gleichen Optionen wie bei der **Display**-Funktion, mit dem Unterschied, dass Sie einen Source-Namen angeben *müssen*.

Die Source und ihre Struktur-Informationen werden als Text in den Arbeitsbereich geladen und können mit dem Systemkommando **EDIT** editiert werden.



# 57

## SYSADA

---

### SYSADA

Mit dem Kommando `SYSADA` rufen Sie die `ADACALL`-Utility auf, die sich in der Library `SYSADA` befindet.

Mittele der `ADACALL`-Utility können Sie von einem Großrechner-Natural aus Adabas-Direktkommandos (Native Commands) direkt an eine Adabas-Datenbank absetzen.

Die `ADACALL`-Utility kann für Übungszwecke oder zum Testen/Analysieren verschiedenster Probleme oder Szenarien verwendet werden.

Weitere Informationen siehe *ADACALL - Issuing Adabas Direct Calls* in der *Utilities*-Dokumentation.





# 58

## SYSAPI

---

### SYSAPI

Mit diesem Kommando rufen Sie die SYSAPI-Utility auf.

Diese Utility dient zum Auffinden von Programmierschnittstellen (APIs), die von Natural-Add-on-Produkten, z.B. Entire Output Management (NOM), zur Verfügung gestellt werden.

Zu jedem API liefert die Utility SYSAPI ein oder mehrere Beispielprogramme, die eine Funktionsbeschreibung des API enthalten und zum Testen der Auswirkung des API verwendet werden können.

Weitere Informationen siehe *SYSAPI - APIs of Natural Add-on Products* in der *Utilities*-Dokumentation.



# 59

## SYSBPM

---

SYSBPM

Mit diesem Kommando rufen Sie die SYSBPM-Utility auf.

Die SYSBPM-Utility liefert statistische Informationen über den aktuellen Status des Natural Buffer Pool, über den Buffer Pool Cache und über die Objekte, die sich gerade im Buffer Pool und im Buffer Pool Cache befinden.

Außerdem bietet SYSBPM Administrationsfunktionen.

Weitere Informationen siehe *SYSBPM Utility - Buffer Pool Management* in der *Utilities*-Dokumentation.



# 60 SYSCP

---

SYSCP

Mit diesem Kommando rufen Sie die SYSCP-Utility auf.

Die SYSCP-Utility kann benutzt werden, um Informationen über Codepages zu erhalten und um die Codepage-Zuordnung für eine Natural-Source zu prüfen oder zu ändern.

Weitere Informationen siehe *SYSCP Utility - Code Page Administration* in der *Utilities*-Dokumentation.

Siehe auch:

- *Unicode and Code Page Support*
- Systemkommando [CPINFO](#)



# 61

## SYSDB2

---

SYSDB2

Mit diesem Kommando rufen Sie die Natural Tools for DB2 auf, wenn Natural for DB2 installiert ist.

Weitere Informationen siehe *Using Natural Tools for DB2* im *Natural for DB2*-Teil der *Database Management System Interfaces*-Dokumentation.





## 62 SYSDDM

---

### SYSDDM

Mit diesem Kommando rufen Sie die `SYSDDM`-Utility auf. Diese bietet Funktionen, die zum Erstellen und zum Pflegen von Natural Data Definition Modules (DDMs) benötigt werden.

Weitere Informationen siehe *SYSDDM Utility* in der *Editors*-Dokumentation.

#### **Hinweis bezüglich Natural Single Point of Development:**

In einer Remote-Entwicklungsumgebung steht dieses Kommando nicht in der Kommandozeile von Natural Studio zur Verfügung, weil DDMs in der Tree View unter dem Node DDM aufgelistet werden und alle Funktionen der `SYSDDM`-Utility über das Kontextmenü oder die Menüleiste verfügbar sind.



## 63 SYSEDIT

---

### SYSEDIT

Mit diesem Kommando rufen Sie die SYSEDIT-Utility for Editor Buffer Pool Services auf. Die SYSEDIT-Utility ist nur für Natural-Administratoren gedacht. Sie dient zum

- Anzeigen von Parametern und Laufzeitinformationen über den Editor Buffer Pool,
- Ändern von Parametern,
- Löschen von logischen Arbeitsdateien (Work Files) und Wiederherstellungsdateien (Recovery Files).

Weitere Informationen siehe *SYSEDIT Utility - Editor Buffer Pool Administration* in der *Utilities-Dokumentation*.



# 64

## SYSERR

---

### SYSERR

Mit diesem Kommando rufen Sie die SYSERR-Utility auf.

Mit der SYSERR-Utility können Sie Ihre eigenen anwendungsspezifischen Meldungen schreiben.

- Sie können mit der SYSERR-Utility Fehler- oder Informationsmeldungen von Ihrem Natural-Code trennen und sie getrennt verwalten.
- Sie können Meldungen vereinheitlichen und Meldungsbereiche für verschiedene Meldungsarten festlegen sowie Meldungen in andere Sprachen übersetzen und einen Langtext zu einer Meldung hinzufügen.
- Mit der SYSERR-Utility können Sie darüber hinaus die Texte vorhandener Natural-Systemmeldungen ändern, obwohl dies nicht empfohlen wird, weil diese Änderungen bei neuen Natural-Releases verloren gehen.

Weitere Informationen siehe *SYSERR Utility* in der *Utilities*-Dokumentation.



# 65 SYSEXT

---

## SYSEXT

Mit diesem Kommando rufen Sie die SYSEXT-Anwendung auf.

Sie enthält verschiedene Natural-Programmierschnittstellen (API). Zu jedem API steht folgendes zur Verfügung:

- ein Subprogramm (in Objektform),
- ein Beispielprogramm (in Sourceform) für den Aufruf des Subprogramms,
- ein Text-Member, das die Funktion des API beschreibt.

Weitere Informationen siehe *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.





## 66 SYSEXV

---

SYSEXV

Mit diesem Kommando rufen Sie die SYSEXV-Utility auf.

Die SYSEXV-Utility ermöglicht Ihnen den Zugang zu Beispielen der neuen Merkmale der aktuellen Natural-Version und einiger früherer Versionen.

Weitere Informationen siehe *SYSEXV Utility* in der *Utilities*-Dokumentation.



## 67 SYSDFILE

---

```
SYSFILE [ { WORKFILE }  
          { PRINTER } ]
```

Mit diesem Kommando können Sie die SYSDFILE-Funktion der SYSTP-Utility aufrufen. Diese Funktion liefert Informationen zu den verfügbaren Arbeitsdateien und Druckdateien (Work und Print Files).

Die vom Systemkommando SYSDFILE gelieferten Informationen erhalten Sie auch mit Hilfe der Programmierschnittstelle (API) USR1007N. Siehe SYSEXT - *Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.

SYSFILE	Wenn Sie nur SYSDFILE eingeben, werden Arbeitsdatei- und Druckdatei-Zuweisungen nacheinander angezeigt.
SYSFILE WORKFILE	Nur die Arbeitsdatei-Zuweisungen werden angezeigt.
SYSFILE PRINTER	Nur die Druckdatei-Zuweisungen werden angezeigt.

Weitere Informationen siehe *Natural Print/Work Files - SYSDFILE* im Abschnitt *General SYSTP Functions* in der *Utilities*-Dokumentation sowie die plattformspezifischen Informationen zu Arbeitsdateien und Druckdateien in der *Operations*-Dokumentation.



# 68

## SYSMAIN

---

SYSMAIN

Mit diesem Kommando rufen Sie die SYSMAIN-Utility auf.

Mit dieser Utility können Sie Natural-Objekte kopieren, übertragen und löschen. Die SYSMAIN-Utility dient außerdem dazu, innerhalb des Natural-Systems Objekte mittels der Import-Funktion von einer Umgebung in eine andere Umgebung zu übertragen.

Weitere Informationen siehe *SYSMAIN Utility - Object Maintenance* in der *Utilities*-Dokumentation.



# 69

## SYSNCP

---

SYSNCP

Mit diesem Kommando rufen Sie die `SYSNCP`-Utility auf.

Weitere Informationen siehe *SYSNCP Utility* in der *Utilities*-Dokumentation.





# 70

## SYSOBJH

---

SYSOBJH

Mit diesem Kommando rufen Sie den Object Handler auf. Mit dem Object Handler können Sie Natural- und Nicht-Natural-Objekte zwecks Verteilung in Natural-Umgebungen handhaben.

Weitere Informationen siehe *Object Handler* in der *Utilities*-Dokumentation.



# 71 SYSPARM

---

## SYSPARM

Mit diesem Kommando rufen Sie die SYSPARM-Utility auf. Sie können diese Utility benutzen, um Zeichenketten mit Natural-Profilparametern zu erstellen und zu pflegen, die als Natural-Profile gespeichert werden.

Diese Natural-Profile können dann zu Beginn einer Natural-Session mit dem Profilparameter PROFILE aufgerufen werden.

Das SYSPARM-Kommando bietet verschiedene Parameter, die hauptsächlich im Batch-Betrieb verwendet werden (siehe *Direct Commands and Batch Processing* in der Beschreibung der SYSPARM-Utility).

Weitere Informationen siehe *SYSPARM Utility* in der *Utilities*-Dokumentation.



# 72

## SYSPROD

---

- Kommando SYSPROD im Online-Modus benutzen ..... 262
- Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen) ..... 263
- Anwendungsprogrammierschnittstellen (APIs) ..... 263

SYSPROD

Mit dem Kommando `SYSPROD` können Sie feststellen, welche Produkte in Ihrer Natural-Umgebung installiert sind. Sie erhalten Informationen zu Ihrer aktuellen Natural-Version, zu wählbaren Natural-Merkmalen („Selectable Units“) sowie zu Produkten, die mit bzw. unter Natural laufen.

## Kommando SYSPROD im Online-Modus benutzen

---

Wenn Sie das Kommando `SYSPROD` eingeben, erscheint ein Fenster, in dem zu jedem installierten Produkt die folgenden Informationen angezeigt werden:

- Produktname
- Produktversion: (siehe auch *Version* im *Glossary*)
- Installationszeitpunkt (Datum und Uhrzeit)
- Installationskomponente: I (INPL) oder N (Nukleus)

Zu einigen der aufgeführten Produkte können Sie weitere Informationen erhalten, indem Sie sie in der Spalte **Cmd** mit einem Direktkommando markieren.

Zeilenkommando	Beschreibung
EX	Zusätzliche Produktinformationen anzeigen.
HI	Historie der Produktinformationen anzeigen.
SC	Unterbestandteile des Produkts anzeigen.
SU	„Selectable Units“, d.h. wählbare Natural-Merkmale (falls für das angegebene Produkt verfügbar) anzeigen, die für das Produkt mit dem Profilparameter <code>SELUNIT</code> (siehe <i>Parameter-Referenz-Dokumentation</i> ) aktiviert worden sind.  Weitere Informationen siehe <i>Selectable Units for New Natural Features</i> in der <i>Operations-Dokumentation</i> .



**Anmerkung:** Bei einigen Produkten sind keine Zeilenkommandos verfügbar.

## Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen)

```
SYSPROD [ ALL
          <product-code> ] [ EX
                             SC
                             HI
                             SU ]
```

Für den Batch-Betrieb oder eine Online-Ausgabe ohne Dialogfunktionen wie zum Beispiel Eingabefelder für Zeilenkommandos können Sie SYSPROD mit zusätzlichen Schlüsselwörtern und/oder unter Angabe eines Produktschlüssel (z.B. NAT) aufrufen:

Schlüsselwort/Parameter	Beschreibung
ALL	Liste aller installierten Produkte anzeigen.
<product-code>	Liste zu einem bestimmten Produkt ohne Dialogfunktionen anzeigen.
EX	Zusätzliche Produktinformationen anzeigen.
HI	Historie der Produktinformationen anzeigen.
SC	Unterbestandteile des Produkts anzeigen.
SU	„Selectable Units“, d.h. wählbare Natural-Merkmale (falls für das angegebene Produkt verfügbar) anzeigen, die für das Produkt mit dem Profilparameter SELUNIT (siehe <i>Parameter-Referenz-Dokumentation</i> ) aktiviert worden sind.  Weitere Informationen siehe <i>Selectable Units for New Natural Features</i> in der <i>Operations-Dokumentation</i> .



**Anmerkung:** Wenn im Batch-Betrieb das Systemkommando SYSPROD ohne ein Schlüsselwort abgesetzt wird, werden dieselben Masken wie im Online-Betrieb verwendet; dann müssen Sie jedoch die Online-Eingaben simulieren, indem Sie z.B. das Terminalkommando %Knn benutzen. Diese Vorgehensweise ist fehleranfällig und wird daher nicht für den Batch-Betrieb empfohlen.

## Anwendungsprogrammierschnittstellen (APIs)

Sie können die von SYSPROD gelieferten Informationen auch erhalten, indem Sie eine der folgenden Anwendungsprogrammierschnittstellen (APIs) benutzen, die sich in der Library SYSEXT befinden.

- USR0050N
- USR2031N

Siehe auch *SYSEXT Utility - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.



# 73 SYSPROF

---

## SYSPROF

Mit dem Kommando SYSPROF können Sie sich die gegenwärtigen Definitionen der Natural-Systemdateien anzeigen lassen.

Zu jeder Systemdatei werden folgende Informationen angezeigt (entspricht dem Drücken von PF4 wenn gerade eine der nachfolgend aufgeführten Seiten angezeigt wird).

- Dateiname
- Datenbank-ID
- Dateinummer
- Datenbanktyp

Programmierschnittstellen (APIs): USR0010N, USR2013N, USR3013N. Siehe auch SYSEXT - *Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.



# 74

## SYSRPC

---

SYSRPC

CSMASS

SGMASS *generation-option*

SM REPLACE *replace-clause*

Dieses Systemkommando ruft die SYSRPC-Utility auf. Die SYSRPC-Utility bietet Funktionen zum Verwalten von Natural Remote Procedure Calls.

Die Parameter und Optionen, die beim Systemkommando SYSRPC zum Ausführen RPC-spezifischer Aufgaben zur Verfügung stehen, werden in den entsprechenden Abschnitten in der *SYSRPC Utility*-Dokumentation erläutert:

CSMASS	Dient zur Berechnung der Größe der Puffer, die für die RPC-Aufrufe erforderlich sind. Siehe <i>Calculating Size Requirements</i> und <i>Using the SYSRPC CSMASS Command</i> .
SGMASS <i>generation-option</i>	Dient zur Generierung mehrerer Schnittstellenobjekte.  Siehe <i>Generating Multiple Interface Objects</i> und <i>Using the SYSRPC SGMASS Command</i> .
SM REPLACE <i>replace-clause</i>	Dient zur Ersetzung einzelner oder mehrerer Einträge in einem Service Directory. Siehe <i>replace-clause</i> in <i>Commands for Service Directory Maintenance</i> .

Informationen, wie Sie die Funktionen der SYSRPC-Utility benutzen können, um ein Kommunikationsnetzwerk zwischen Server- und Client-Systemen zu erstellen, finden Sie in der *Natural RPC (Remote Procedure Call)*-Dokumentation.



# 75 SYSTP

---

SYSTP

Mit diesem Kommando rufen Sie die SYSTP-Utility auf, mit der Sie verschiedene TP-Monitor-spezifische Charakteristika von Natural überwachen und steuern können.

Weitere Informationen siehe *SYSTP Utility* in der *Utilities*-Dokumentation.



# 76 TECH

---

## TECH

Mit diesem Kommando können Sie sich technische und andere Informationen über Ihre Natural-Session anzeigen lassen.

Folgende Informationen werden angezeigt:

- User-ID
- Library-ID
- Natural-Version: Major Version, Minor Version, Service Pack, kumulativer Fix (siehe auch *Version* im *Glossary*)
- Startup-Transaktion
- Natural Security-Indikator
- Betriebssystemname und -version
- Maschinenklasse
- Hardware
- TP-Monitor (auf Großrechnern, bei Windows (\*TPSYS) nur in einer Remote-Entwicklungsumgebung)
- Gerätetyp
- Terminal-ID (auf Großrechnern, bei Windows nur in einer Remote-Entwicklungsumgebung)
- Codepage
- Locale
- zuletzt ausgeführtes Kommando
- Informationen zum zuletzt aufgetretenen Fehler
- Namen, Datenbank-IDs und Dateinummern aller gerade aktiven Steplib-Libraries

- Namen, Typen und Ebenen aller gerade aktiven Programmierobjekte und aller Objekte auf höheren Ebenen sowie die Zeilennummern der Statements, die untergeordnete Programmierobjekte (nur bei Großrechnern, UNIX und OpenVMS) aufrufen.

**Anmerkungen:**

1. Nur bei Anwendungen mit zeichenorientierter Benutzeroberfläche: Um diese Informationen von einem beliebigen Punkt in einer Anwendung aus aufzurufen, können Sie das Terminalkommando %<TECH verwenden.
2. Dieses Kommando steht auch in einer Remote-Session zur Verfügung. Alle Informationen können auch im Batch-Betrieb gelesen werden.

Programmierschnittstelle (API): USR2026N. Siehe auch *SYSEXT - Natural Application Programming Interfaces* in der *Utilities*-Dokumentation.



# 77 TEST

TEST  $\left[ \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} \right]$   
*debugger-commands*

Mit dem Kommando `TEST` rufen Sie die Debugging-Funktion auf.

TEST	Wenn Sie nur <code>TEST</code> eingeben, erhalten Sie das Hauptmenü der Debugging-Funktion. Mit <code>TEST ON/OFF</code> aktivieren/deaktivieren Sie den Test-Modus der Debugging-Funktion.
TEST ON	Aktiviert den Test-Modus der Debugging-Funktion.
TEST OFF	Deaktiviert den Test-Modus der Debugging-Funktion.
<i>debugger-commands</i>	Die Direktkommandos zum Ausführen von Debugging-Funktionen sind im Abschnitt <i>Command Summary and Syntax</i> in der <i>Debugger</i> -Dokumentation beschrieben.

Um die Debugging-Funktion von einem beliebigen Punkt in einer Anwendung aus aufzurufen, können Sie das Terminalkommando `%<TEST` verwenden.

Weitere Informationen finden Sie in der *Debugger*-Dokumentation.

Die *Utilities*-Dokumentation enthält außerdem Informationen über weitere Natural-Utilities, die Sie zum Testen und Überwachen im Online-Modus verwenden können.

## Anmerkung zu Natural Single Point of Development:

Dieses Kommando ist nur auf Großrechnern verfügbar. Falls ein Benutzer ein Programm namens `TEST` geschrieben hat, dann führt Natural in einer lokalen Windows-, UNIX - oder OpenVMS-Umgebung dieses Programm aus. Besteht eine aktive Verbindung zu einem Development-Server auf einem Großrechner, wird die Großrechner-Utility `TEST` aufgerufen, wenn dieses Kommando abgesetzt wird.



# 78

## TEST DBLOG

---

TEST DBLOG [*parameters*]

Mit diesem Kommando rufen Sie die DBLOG-Utility auf, mit der Sie Datenbankaufrufe protokollieren können.

TEST DBLOG	Aktiviert bzw. deaktiviert die DBLOG-Utility.
<i>parameters</i>	Die Parameter, die Sie beim TEST DBLOG-Kommando benutzen können, sind im Abschnitt <i>TEST DBLOG Command</i> der <i>Utilities</i> -Dokumentation beschrieben.

Weitere Informationen siehe *DBLOG Utility - Logging Database Calls* in der *Utilities*-Dokumentation.

Die *Utilities*-Dokumentation enthält Informationen über weitere Natural-Utilities, die zum Testen und Überwachen verwendet werden können.

### Anmerkung zu Natural Single Point of Development:

Besteht eine aktive Verbindung zu einem Development-Server auf einem Großrechner, wird die Großrechner-Utility DBLOG aufgerufen, wenn dieses Kommando unter Natural for Windows abgesetzt wird.



## 79 `UNCATALOG`

---

Dieses Kommando wird nur aus Kompatibilitätsgründen unterstützt. Es empfiehlt sich, statt des `UNCATALOG`-Kommandos das `DELETE`-Kommando zu verwenden, da `DELETE` mehr Flexibilität sowie Schutz gegen versehentliches Löschen bietet.



# 80

## UNLOCK

---

■ Natural-Objekte entsperren .....	280
■ UNLOCK-Parameter-Beschreibungen .....	281
■ Parameterverarbeitung und Anzeige der gefundenen Objekte .....	282
■ Batch-Verarbeitung .....	283

Mit dem UNLOCK-Kommando können Sie Natural-Sourceobjekte lokal in einer Natural-Großrechnerumgebung entsperren.

Das UNLOCK-Kommando dient dazu, Sourceobjekte anzuzeigen, die gesperrt sind, und diese zu entsperren. Es wird empfohlen, dass dieses Kommando nur vom Natural-Administrator benutzt wird. Der Administrator kann jedoch die Benutzung dieses Kommandos für jedes Benutzerprofil in Natural Security ermöglichen.

**Anmerkungen:**

1. Als Voraussetzung für die Benutzung des UNLOCK-Systemkommandos muss der Profilparameter SLOCK auf PRE gesetzt sein.
2. Wenn die Anzahl der gesperrten Objekte groß ist, kann es vorkommen, dass die angezeigte Liste nicht sortiert ist. Zur Abhilfe können Sie die Größe des vom Sort-Programm benutzten Work-Buffer erhöhen; siehe Schlüsselwortparameter WRKSIZE des Profilparameters SORT.

Weitere Informationen siehe *Locking of Source Objects* in der *Editors-Dokumentation* und Profilparameter SLOCK in der *Parameter Reference-Dokumentation*.

Siehe auch *Namenskonventionen für Objekte* in der *Dokumentation Natural benutzen*.

## Natural-Objekte entsperren

---

Wenn Sie das Systemkommando UNLOCK ohne Parameter benutzen, erscheint eine Maske, auf der Sie die Parameter eingeben können.

UNLOCK

Die folgende Übersicht zeigt die Kommandosyntax zum Entsperren von Natural-Objekten:

```
UNLOCK [NATURAL] [OBJECT] object-name
      [TYPE object-type]
      [LIBRARY library-name]
      [DBID dbid] [FNR fnr]
      [PASSWORD password] [CIPHER cipher]
      [USER locked-by]
      [DATE locked-on [locked-on2]]
```



## UNLOCK-Parameter-Beschreibungen

Sie müssen in jedem Fall den *object-name* angeben. Falls Sie einen der übrigen Parameter nicht angeben, wird der Standardwert verwendet.

Parameter	Format/Länge	Standardwert	Beschreibung	
<i>object-name</i>	A33	*	Der Name des zu entsperrenden Objekts. Sie können Stern-Notation (*) oder das Größerzeichen (>) verwenden.	
<i>object-type</i>	A1	*	<b>Natural-Objekttypen:</b>	
			Anstelle von <i>object-type</i> können Sie einen der unten aufgeführten Objekttypcodes oder Stern-Notation (*) benutzen.	
			P	Programm
			4	Klasse
			N	Subprogramm
			S	Subroutine
			7	Function
			8	Adapter
			C	Copycode
			H	Helproutine
			T	Text
			M	Map
			L	Local Data Area
			G	Global Data Area
			A	Parameter Data Area
V	DDM (View)			



**Anmerkung:** Das Sperren können Sie auch lokal auf einem Großrechner-Server einschalten, der auf Natural für Großrechner Version 4.2 oder höher basiert. In diesem Fall gelten folgende Einschränkungen: Der *application-name* kann nicht als Auswahlkriterium verwendet werden. Bei *dbid* und *fnr* werden die aktuellen Systemdateien FNAT und FUSER durchsucht, wenn Sie Stern-Notation (\*) benutzen.

## Parameterverarbeitung und Anzeige der gefundenen Objekte

---

Das Objekt sofort entsperrt, wenn die angegebenen Parameter gültig sind und ein vollständiger Objektname angegeben wurde und wenn das entsprechende Objekt gefunden und es vom aktuellen Benutzer gesperrt wurde. Es erscheint eine entsprechende Meldung.

Dies gilt unter der Bedingung, dass der Objektname direkt und ohne die Benutzung von Stern-Notation (\*) angegeben wird und der aktuelle Benutzer seine eigenen gesperrten Objekte zu entsperren versucht.

Falls einer der angegebenen Parameter ungültig ist oder falls keine gesperrten Objekte gefunden werden, erscheint der Unlock-Dialog mit einer Fehlermeldung.

In den folgenden Fällen werden die gefundenen gesperrten Objekte in einer Maske aufgelistet und können mittels dem Zeilenkommando U entsperrt werden (siehe unten):

- Wenn Sie Stern-Notation (\*) oder (wo zutreffend) das Größerzeichen (>) benutzt haben.
- Wenn Sie keinen spezifischen Objektnamen angegeben haben.



**Anmerkung:** Wenn Sie Stern-Notation (\*) für den Objekttyp und die Library benutzen, müssen Sie die aktuelle Systemdatei `FDIC` absuchen, um die gesperrten DDMs aufzulisten.

### Unlock-Liste

#### Funktionstasten

Die Unlock-Liste bietet folgende Funktionstasten:

PF1	Help	Hilfe aufrufen.
PF3	Exit	Rückkehr zur Unlock-Liste.
PF6	--	Zum Anfang der Liste springen.
PF7	-	Eine Seite zurück blättern.
PF8	+	Eine Seite vor blättern.
PF9	++	Zum Ende der Liste springen.
PF10	<	Nach links blättern zum ersten Teil der Informationen (Typ, Library, Datenbankkennung, Dateinummer).
PF11	>	Nach rechts blättern zum zweiten Teil der Informationen (gesperrt durch, gesperrt am).
PF12	Cancel	Anulliert das UNLOCK-Kommando.

## Zeilenkommando

U	In der Spalte <b>Cmd</b> der Unlock-Liste können Sie das Kommando U in einer einzelnen oder in mehreren Zeilen eingeben, um das entsprechende Objekt oder mehrere Objekte zu entsperren. Die erfolgreiche Entsperrung wird durch die Meldung <b>unlocked</b> in der Spalte <b>Message</b> angezeigt.
---	--

## Batch-Verarbeitung

---

Falls kein Fehler aufgetreten ist, werden alle gesperrten Objekte entsperrt, und es erscheint eine entsprechende Meldung.



# 81 UPDATE

---

UPDATE	$\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$
--------	---

Das Kommando `UPDATE` dient dazu, von einem Natural-Programm durchgeführte Datenänderungen zu gestatten bzw. nicht zu gestatten.

UPDATE ON	Datenbankänderungen sind erlaubt.
UPDATE OFF	<p>Datenbankänderungen sind nicht erlaubt.</p> <p>Wenn ein Programm, das ein datenbankveränderndes Statement (<code>UPDATE</code>, <code>STORE</code> oder <code>DELETE</code>) enthält, ausgeführt wird, erfolgt keine Datenbankänderung. Stattdessen wird bei der nächsten Bildschirm-Ein-/Ausgabe eine Warnmeldung NAT1010 ausgegeben.</p> <p>Des Weiteren stellt eine Datenbankschleife, die ein <code>UPDATE</code> oder <code>DELETE</code> Statement enthält, keine Datensätze in den Hold-Status (kein „Lesen mit Hold“).</p>



## 82 XREF

---

XREF	[
	ON
	OFF
	FORCE
	DOC
	?
	]

Dieses Kommando ist nur verfügbar, wenn Predict installiert ist.

Mit dem Kommando `XREF` können Sie die Verwendung der Predict-Funktion **Active Cross-References** steuern.

Mit Hilfe der aktiven Referenzen wird im Predict Data Dictionary automatisch dokumentiert, welche Objekte von einem Programm bzw. einer Data Area referenziert werden. Bei diesen Objekten kann es sich handeln um: Programme, Subprogramme, Subroutinen, Helprouinen, Maps, Data Areas, Datenbanksichten (Views), Datenbankfelder, Benutzervariablen, Verarbeitungsregeln (Processing Rules), Fehlernummern, Arbeitsdateien, Drucker, Klassen und gehaltene ISN-Listen.

Die automatische Dokumentation erfolgt, wenn ein Programm bzw. eine Data Area katalogisiert, d.h. in Objektform gespeichert wird.

Mit der `XREF`-Option des Systemkommandos `LIST` können Sie sich die dokumentierten Informationen anzeigen lassen. Weitere Informationen zu aktiven Referenzen finden Sie in der Predict-Dokumentation.

Das `XREF`-Kommando bietet Ihnen folgende Möglichkeiten:

XREF	Wenn Sie das XREF-Kommando ohne Parameter eingeben, erhalten Sie ein Menü, von dem Sie die gewünschte Option auswählen können.
XREF ON	Dieses Kommando schaltet die Dokumentation von aktiven Referenzen ein. Predict dokumentiert die betreffenden Informationen für jedes Natural-Programm bzw. jede Data Area, welche(s) katalogisiert wird.
XREF OFF	Dieses Kommando schaltet die Dokumentation von aktiven Referenzen aus. Es erfolgt keine Dokumentation in Predict. Bereits bestehende Referenzdaten des betreffenden Objekts werden gelöscht.
XREF FORCE	Das Objekt kann nur katalogisiert werden, wenn dafür ein entsprechender Predict-Eintrag vorhanden ist. Beim Katalogisieren werden die betreffenden Referenzdaten in Predict gespeichert.  Wenn kein Predict-Eintrag vorhanden ist, kann das Objekt nicht katalogisiert werden.
XREF DOC	Das Objekt kann nur katalogisiert werden, wenn dafür ein entsprechender Predict-Eintrag vorhanden ist. Allerdings werden beim Katalogisieren keine Referenzdaten in Predict gespeichert, und bereits bestehende Referenzdaten des betreffenden Objekts werden gelöscht.  Wenn kein Predict-Eintrag vorhanden ist, kann das Objekt nicht katalogisiert werden.
XREF ?	Ruft die Help-Funktion des XREF-Kommandos auf.

### **XREF unter Natural Security**

Unter Natural Security wird die Dokumentation von aktiven Referenzen über die Security-Profile der einzelnen Libraries gesteuert. Je nach Security-Profil kann es sein, dass Sie das XREF-Kommando nur eingeschränkt benutzen können.



# 83

## ZIIP

---

■ Syntax der ziip-subfunction-and-option .....	290
■ Syntax-Element-Beschreibung .....	290
■ zIIP-Berichte .....	292
■ ZIIP-Kommando-Menü für Berichtausgabe-Optionen .....	295
■ User Exit für ZIIP PRINT-Kommando-Optionen .....	295
■ Anwendungsprogrammierschnittstelle (API) für ZIIP-Kommando-Funktionen .....	295

ZIIP

ziip-subfunction-and-option

MENU

Dieses Systemkommando gilt nur in z/OS Batch- oder TSO-Umgebungen, welche die Verwendung des IBM System z Integrated Information Processor (zIIP) unterstützen. Falls das Kommando in einer anderen Umgebung und/oder in einer Umgebung ohne zIIP-Unterstützung abgesetzt wird, erfolgt eine entsprechende Fehlermeldung.

Das Systemkommando `ZIIP` liefert Status- und Verarbeitungsinformationen zu den in Ihrer Umgebung verwendeten zIIPs und zu der für die zIIP-Unterstützung erforderlichen Lizenzdatei. Darüber hinaus können Sie mit diesem Kommando statistische Informationen zu den Komponenten abrufen, die ein Umschalten auf den General Central Processor (GCP) verursachen, und Sie können die Zählung der Komponentenumschaltvorgänge verwalten.

Informationen zur zIIP-Unterstützung und Überwachung erhalten Sie in der *Natural zIIP Enabler*-Dokumentation.

### Syntax der `ziip-subfunction-and-option`

PRINT

STAT

LIC

INFO

STAT

ALL

ON

OFF

CLR

ASIS

XML

EFORMATTED

SORTDESC

SORTNAME

SORTSWITCHES

### Syntax-Element-Beschreibung

Subfunktion	Option	Erklärung
PRINT	INFO	Zeigt den Bericht <i>Informationen zur zIIP-Verarbeitung</i> an.

Subfunktion	Option	Erklärung
	STAT	Zeigt den Bericht <i>zIIP-Komponenten-Umschaltstatistik</i> an, wenn die Zählung der Komponentenumschaltvorgänge eingeschaltet worden ist (siehe <i>STAT</i> weiter unten).  Bezüglich der Festlegung der Sortierreihenfolge siehe auch <i>Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik</i> .
	ALL	Zeigt zuerst den Bericht <i>Informationen zur zIIP-Verarbeitung</i> an, und danach den Bericht <i>zIIP-Komponenten-Umschaltstatistik</i> .  Dies ist die Standardeinstellung.  Bezüglich der Festlegung der Sortierreihenfolge siehe <i>Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik</i> .
STAT	ON	Aktiviert die Zählung der Komponentenumschaltvorgänge.
	OFF	Deaktiviert die Zählung der Komponentenumschaltvorgänge.
	CLR	Löscht die Tabelle mit den Komponentenumschaltvorgängen.
LIC	ANIS	Bewirkt die linksbündige Anzeige der zIIP-Lizenzdatei im XML-Format.
	XML	Bewirkt die Anzeige der zIIP-Lizenzdatei im XML-Format.
	FORMATTED	Bewirkt die Anzeige der zIIP-Lizenzdatei im freien Format ohne Klammern.  Dies ist die Standardeinstellung.

\* Alternativ können Sie die Schlüsselwortsubparameter *STAT* und *PRINT* des Profilparameters *ZIIP* bzw. des Makros *NTZIIP* setzen. Siehe *Schlüsselwort-Subparameter* im Abschnitt *ZIIP - zIIP-Verarbeitung (nur z/OS)* in der *Parameter-Referenz-Dokumentation*.

### Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik

Schlüsselwort	Sortierreihenfolge
SORTDESC	Der Bericht wird aufsteigend nach Komponentenbeschreibungen sortiert.
SORTNAME	Der Bericht wird aufsteigend nach Komponentennamen sortiert.
SORTSWITCHES	Der Bericht wird absteigend nach TCB-Umschaltungen (TCB = Task Control Block) sortiert.  Dies ist die Standardeinstellung.

## zIIP-Berichte

Dieser Abschnitt beschreibt die vom Systemkommando `ZIIP` zur Verfügung gestellten Berichte und Statistikdaten.

- Informationen zur zIIP-Verarbeitung
- zIIP-Komponenten-Umschaltstatistik

### Informationen zur zIIP-Verarbeitung

Die folgende Maske zeigt ein Beispiel eines zIIP-Verarbeitungsberichts:

```
+----- zIIP Processing Information -----+
|
|           Advanced zIIP Support Enabled
|
| Number of GCPs                      3
| Number of zIIPs                     1
| zIIP normalization factor           3.48
|
| Number of switches into TCB mode     105
| Number of SRB starts                 1
|
| Total enclave CPU time                24
| Qualified zIIP CPU time               12
| Eligible zIIP CPU time on GCP         0
| Total enclave zIIP CPU time           12
| Total enclave zIIP CPU time (%)       50.00
|
| (All times in milliseconds; zIIP times normalized)
|
+-----+
```

Der Bericht enthält folgende Status- und Verarbeitungsinformationen:

Statistikpunkt	Erklärung
Advanced zIIP Support Enabled	Zeigt an, dass die aktuelle Natural-Session mit aktivierter zIIP-Unterstützung läuft.
Anzahl der GCPs	Anzahl der in z/OS verfügbaren General Central Processors (GCPs), die unter Ihrem z/OS-System laufen.
Anzahl der zIIPs	Anzahl der zIIPs, die unter Ihrem z/OS-System laufen.
zIIP-Normalisierungsfaktor	Der zIIP-Normalisierungsfaktor gibt das Verhältnis der zIIP-Geschwindigkeit zur GCP-Geschwindigkeit an.  Dieser Faktor gibt an, wie schnell Ihr zIIP im Vergleich zu einem gedrosselten GCP mit reduzierter Leistung läuft.

Statistikpunkt	Erklärung
	In dem obigen Beispiel bedeutet der Wert 3.48, dass der zIIP mehr als dreimal so schnell ist wie ein GCP.
Anzahl der Umschaltungen in den TCB-Modus	Anzahl der Umschaltungen vom SRB-Modus (SRB = Service Request Block) in den TCB-Verarbeitungsmodus aufgrund von Systemdienstaufrufen, die abgesetzt werden mussten. Die entsprechenden Umschaltungen zurück in den SRB-Modus werden nicht mitgezählt.
Anzahl der SRB-Starts	Gibt an, wie oft ein SRB-Vorgang gestartet wurde.
CPU-Gesamtzeit der Enklaven	Die von der aktuellen WLM-Enklave verbrauchte CPU-Gesamtzeit (GCP plus zIIP).
Qualifizierte zIIP-CPU-Zeit	Die für den zIIP-Gebrauch bei SRB qualifizierte CPU-Zeit, welche die Summe der in Frage kommenden und der gesamten zIIP-CPU-Zeit ist.
In Frage kommende zIIP-CPU-Zeit beim GCP	Die CPU-Zeit beim GCP, die für den zIIP-Gebrauch in Frage kommt, aber von ihm nicht genutzt wird.  Wenn der Wert nicht Null ist, bedeutet dies, dass für den zIIP-Gebrauch in Frage kommende Arbeitslast nicht abgegeben werden konnte, weil kein zIIP zur Verfügung stand.
zIIP-CPU-Gesamtzeit der Enklave	Die von der aktuellen WLM-Enklave verbrauchte zIIP-CPU-Gesamtzeit.
zIIP-CPU-Gesamtzeit der Enklave (%)	Die von der aktuellen WLM-Enklave prozentual verbrauchte zIIP-CPU-Gesamtzeit. Im obigen <b>Beispiel</b> konnte Natural 50 % der CPU-Gesamtlast an einen zIIP abgeben.

Ausführliche Informationen zu den Statistikangaben siehe *Natural zIIP Processing: TCBs, SRBs and Enclaves* und *Calculating the CPU Time* in der *Natural zIIP Enabler*-Dokumentation.

Hinweise, wie Sie die zIIP-Nutzung verbessern können, finden Sie im Abschnitt *Tuning zIIP Usage* in der *Natural zIIP Enabler*-Dokumentation.

### PF-Tasten zur Online-Anzeige von zIIP-Verarbeitungsinformationen

ENTER oder PF5	Aktualisieren der Berichtsdaten.
PF8	Umschalten auf die Anzeige der <b>zIIP-Komponenten-Umschaltstatistik</b> .

### zIIP-Komponenten-Umschaltstatistik

Der Statistikbericht **zIIP Component Switch Statistics** enthält die Auflistung aller TCB-Umschaltungen, die nicht den Wert Null haben, sowie der Komponenten, die diese Umschaltungen verursachen, wie in der folgenden Beispielmaste dargestellt:

+----- zIIP Component Switch Statistics -----+		
Component Description		TCB Switches
-----		
CMDBAS	Call Adabas	72
CMGETMP	Acquire working storage	14
CMGTIMER	Set CPU time limit	10
CMFREMP	Release working storage	3
CMWTERM	Primary I/O	3
ZPINIT	Initialization of zIIP support	1
CMWTL	Write message to system log	1
CMDLET	Release external module	1
-----		
Total		105
-----		

Komponenten sind Service-Routinen des Natural Nucleus, Natural-Add-on-Produkte oder sonstige Software AG-Produkte (z.B. Adabas), die zum Beispiel bei der Ausführung von Adabas-Aufrufen oder bei externen CALLNAT-Anforderungen TCB-Umschaltungen verursachen. Der Bericht enthält den Namen, die Beschreibung und die jeweilige Anzahl der TCB-Umschaltungen der einzelnen Komponenten sowie die Anzahl der TCB-Umschaltungen, die von den Komponenten insgesamt verursacht wurden.

**PF-Tasten zur Online-Anzeige von zIIP-Komponenten-Statistikinformationen**

ENTER oder PF5	Aktualisiert die Statistikdaten.
PF7	Umschalten auf den Bericht <b>zIIP Processing Information</b> , siehe <i>Informationen zur zIIP-Verarbeitung</i>
PF10	Umschalten zwischen den beiden Anzeigen, die bei SORTDESC und SORTNAME in <i>Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik</i> beschrieben sind.
PF11	Umschalten zwischen den beiden Anzeigen, die bei SORTDESC und SORTSWITCHES in <i>Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik</i> beschrieben sind.

---

## ZIIP-Kommando-Menü für Berichtsausgabe-Optionen

---

Im Online-Modus können Sie das Menü des Systemkommandos `ZIIP` benutzen, um Einstellungen für die Ausgabe des Berichts entsprechend Ihren Erfordernissen vorzunehmen.

### » Um das ZIIP-Menü anzuzeigen

- Geben Sie bei einer beliebigen Kommandoingabeaufforderung folgendes Kommando ein:

```
ZIIP MENU
```

Das Menü (**Main Menu**) erscheint. Hier können Sie die Unterfunktion und Option(en) auswählen. Beschreibung der Menüpunkte siehe [Syntax-Element-Beschreibung](#)

---

## User Exit für ZIIP PRINT-Kommando-Optionen

---

Zur Angabe der Standardeinstellungen für die `PRINT`-Optionen des Systemkommandos `ZIIP` können Sie den User Exit `ZIIPUX01` benutzen. Der Quellcode wird im Unterprogramm `ZIIPUS01` in der Natural-System-Library `SYSLIB`, `SYSLIBS` oder `SYSTEM` zur Verfügung gestellt. Einzelheiten hierzu siehe Unterprogramm `ZIIPUS01` in der System-Library `SYSEXT`.

---

## Anwendungsprogrammierschnittstelle (API) für ZIIP-Kommando-Funktionen

---

Sie können über die Anwendungsprogrammierschnittstelle `USR8204N` alle Funktionen des Systemkommandos `ZIIP` ausführen. Einzelheiten siehe Textobjekt `USR8204T` und das Programm `USR8204P` in der Natural-System-Library `SYSEXT`.





# Stichwortverzeichnis

---

## A

AIV  
Systemkommando, 17

## B

BUS  
Systemkommando, 19

## C

CATALL  
Systemkommando, 21  
CATALOG  
Systemkommando, 31  
CHECK  
Systemkommando, 33  
CLEAR  
Systemkommando, 35  
COMPOPT  
Systemkommando, 37

## D

DELETE  
Systemkommando, 61  
DUMP  
Systemkommando, 65

## E

EDIT  
Systemkommando, 67  
EDT  
Systemkommando, 73  
EXECUTE  
Systemkommando, 77

## F

FIN  
Systemkommando, 81

## G

GLOBALS  
Systemkommando, 83

## H

HELP  
Systemkommando, 87

## I

INPL  
Systemkommando, 89

## K

KEY  
Systemkommando, 91

## L

LAST  
Systemkommando, 95  
LASTMSG  
Systemkommando, 97  
LIST  
Systemkommando, 99  
LIST XREF  
Systemkommando, 137  
LISTDBRM  
system command, 139  
LISTSQL  
system command, 143  
LISTSQLB  
system command, 149  
LOGOFF  
Systemkommando, 153  
LOGON  
Systemkommando, 155

## M

MAIL  
Systemkommando, 157  
MAINMENU  
Systemkommando, 159

## N

NATQVS  
system command, 161  
NOCOPT

- Systemkommando, 163
- NOCSHOW
  - Systemkommando, 165
- NOCSTAT
  - Systemkommando, 167

## P

- PROFILE
  - Systemkommando, 169
- PROFILER
  - system command, 171

## R

- RDC
  - system command, 173
- READ
  - Systemkommando, 175
- RENAME
  - Systemkommando, 177
- RENUMBER
  - Systemkommando, 179
- RETURN
  - Systemkommando, 181
- ROUTINES
  - Systemkommando, 183
- RPCERR
  - Systemkommando, 185
- RUN
  - Systemkommando, 187

## S

- SAVE
  - Systemkommando, 189
- SCAN
  - Systemkommando, 191
- SCRATCH
  - Systemkommando, 201
- SETUP
  - Systemkommando, 203
- SHOWDBS
  - system command, 207
- SHOWSU
  - system command, 211
- SQLDIAG
  - system command, 213
- SQLERR
  - system command, 217
- STOW
  - Systemkommando, 221
- STRUCT
  - Systemkommando, 223
- SYSADA
  - Systemkommando, 231
- SYSAPI
  - Systemkommando, 233
- SYSBPM
  - Systemkommando, 235
- SYSCP
  - Systemkommando, 237
- SYSDB2
  - Systemkommando, 239

- SYSDDM
  - Systemkommando, 241
- SYSEDT
  - Systemkommando, 243
- SYSERR
  - Systemkommando, 245
- SYSEXT
  - Systemkommando, 247
- SYSEXV
  - Systemkommando, 249
- SYSFILE
  - Systemkommando, 251
- SYSMAIN
  - Systemkommando, 253
- SYSNCP
  - Systemkommando, 255
- SYSOBJH
  - Systemkommando, 257
- SYSPARM
  - Systemkommando, 259
- SYSPROD
  - system command, 261
- SYSPROF
  - Systemkommando, 265
- SYSRPC
  - Systemkommando, 267
- Systemkommandos, vii
- SYSTP
  - Systemkommando, 269

## T

- TECH
  - Systemkommando, 271
- TEST
  - Systemkommando, 273
- TEST DBLOG
  - Systemkommando, 275

## U

- UNCATALOG
  - Systemkommando, 277
- UNLOCK
  - Systemkommando, 279
- UPDATE
  - Systemkommando, 285

## X

- XREF
  - Systemkommando, 287

## Z

- ZIIP
  - system command, 289