Natural for z/OS

Systemkommandos

Version 9.2.5

Oktober 2026

ADABAS & NATURAL

This document applies to Natural for z/OS Version 9.2.5 and all subsequent releases.

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

Copyright © 1979-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

Document ID: NATMF-NNATSYSCOM-925-20261013DE

Inhaltsverzeichnis

Vorwort ix
1 Über diese Dokumentation 1
Dokumentationskonventionen 2
Online-Informationen und Support 2
Datenschutz 3
2 Kommandoausführung 5
Kommandoeingabe 6
Kommandozeile 6
NEXT-Zeile 7
MORE-Zeile 7
3 Systemkommando-Syntax 9
Syntax-Elemente 10
Kommandosyntax-Beispiel 11
4 Systemkommandos sortiert nach Kategorien 13
In Natural navigieren 14
Natural-Entwicklungsumgebung 14
Anwendungen mit Natural-Programmobjekten pflegen 15
Monitor-, Debug- und Trace-Funktionen 17
Natural mit anderen Produkten 18
5 AIV 21
6 BUS 23
7 CATALL 25
Objekte katalogisieren von/bis 26
Predict Set und Set Benutzer 27
Vorhandene Module neu katalogisieren oder alle Quellcode-Objekte katalogisieren 27
Objekttypen auswählen 28
Funktion auswählen 28
Optionen auswählen 28
Auswahlliste 29
Direktkommando-Syntax 30
8 CATALOG 33
9 CHECK 35
10 CLEAR 37
11 COMPOPT 39
Syntax-Erklärung 40
Compiler-Schlüsselwortparameter angeben 40
Compiler-Optionen 41
12 CPINFO 61
13 DELETE 65
Syntax-Erklärung 66
Auswahlliste 67
Schutz vor versehentlichem Löschen 67

Beispiele ... 67
14 DUMP ... 69
15 EDIT ... 71
Syntax 1 ... 72
Syntax 2 ... 74
Syntax 3 ... 74
16 EDT ... 77
Syntax-Erklärung ... 78
EDT-Kommandos ... 78
EDT-Funktionstasten ... 79
17 EXECUTE ... 81
Syntax-Erklärung ... 82
Beispiele für das EXECUTE-Kommando ... 83
18 FIN ... 85
19 GLOBALS ... 87
Syntax-Erklärung ... 88
Liste der Parameter ... 88
Zusammenhang zwischen GLOBALS, SET GLOBALS und anderen Statements ... 89
20 HELP ... 91
21 INPL ... 93
22 KEY ... 95
Kommandos zuweisen ... 97
Aktivieren/Deaktivieren aller Tasten - KEY ON/OFF ... 97
Aktivieren/Deaktivieren einzelner Tasten - KEY key=ON/OFF ... 98
23 LAST ... 99
24 LASTMSG ... 101
25 LIST ... 103
Syntax-Übersicht ... 104
Inhalt des Arbeitsbereichs auflisten ... 112
Quellcode eines einzelnen Objekts anzeigen ... 112
Anzahl und Größe von Objekten anzeigen ... 112
Quellcode mehrerer Objekte nacheinander anzeigen ... 113
Liste von Objekten anzeigen ... 113
Vorsortierte Liste von ausgewählten Objekten anzeigen ... 113
Langnamen katalogisierter Subroutinen und Klassen anzeigen ... 114
NOC-Optionen katalogisierter Objekte anzeigen ... 114
Compiler-Optionen katalogisierter Objekte anzeigen ... 114
Compiler Output zu einem katalogisierten Objekt anzeigen ... 114
Directory-Informationen anzeigen ... 116
DDMs (Views) anzeigen ... 118
Optionen ... 118
Objekt-Auswahlliste ... 123
Quellcode-Liste ... 131
Individuelles List-Profil erstellen ... 136

26 LIST XREF 143
27 LISTDBRM 145
28 LISTPROF 149
29 LISTSQL 153
30 LISTSQLB 159
31 LOGOFF 163
32 LOGON 165
33 MAIL 167
34 MAINMENU 169
35 NATQVS 171
36 NOCOPT 173
37 NOCSHOW 175
38 NOCSTAT 177
39 PROFILE 179
40 PROFILER 181
41 RDC 183
42 READ 185
43 RENAME 187
44 RENUMBER 189
45 RETURN 191
46 ROUTINES 193
47 RPCERR 195
48 RUN 197
49 SAVE 199
50 SCAN 201
Menü-Optionen 202
SCAN-Editierkommandos 205
SCAN mittels Direktkommando 206
Individuelles SCAN-Profil definieren 209
SCAN unter Natural Security 210
51 SCRATCH 211
52 SETUP 213
Syntax-Erklärung 214
Beispiel für SETUP/RETURN 215
53 SHOWDBS 217
Entsprechendes API USR8211N 219
54 SQLDIAG 221
55 SQLERR 225
56 STOW 227
57 STRUCT 229
Generate Structured Source into Work Area 230
Display Structure of Source 233
Print Structure of Source 234
Write Structure of Source into Work Area 235
58 SYSADA 237

59 SYSAPI 239
60 SYSBPM 241
61 SYSCP 243
62 SYSDB2 245
63 SYSDDM 247
64 SYSEDT 249
65 SYSERR 251
66 SYSEXT 253
67 SYSEXV 255
68 SYSFILE 257
69 SYSMAIN 259
70 SYSNCP 261
71 SYSOBJH 263
72 SYSPARM 265
73 SYSPROD 267
Kommando SYSPROD im Online-Modus benutzen 268
Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen) 268
Anwendungsprogrammierschnittstellen (APIs) 269
74 SYSPROF 271
75 SYSRPC 273
76 SYSTP 275
77 TECH 277
78 TEST 279
79 TEST DBLOG 281
80 UNCATALOG 283
81 UNLOCK 285
Natural-Objekte entsperren 286
UNLOCK-Parameter-Beschreibungen 287
Parameterverarbeitung und Anzeige der gefundenen Objekte 289
Batch-Verarbeitung 290
82 UPDATE 291
83 WHICH 293
Direktkommando-Syntax 294
Parameter-Beschreibung 294
Standardausgabe des WHICH-Kommandos 294
Vollständige Liste des WHICH-Kommandos 295
Batch-Verarbeitung 296
WHICH-Kommando unter Natural Security 297
84 XREF 299
85 ZIIP 301
Syntax der ziip-subfunction-and-option 302
Syntax-Element-Beschreibung 303
zIIP-Berichte 305
ZIIP-Kommando-Menü für Berichtausgabe-Optionen 310

User Exit für ZIIP PRINT-Kommando-Optionen 311
Anwendungsprogrammierschnittstelle (API) für ZIIP-Kommando-Funktionen 311
Stichwortverzeichnis 313

Vorwort

Diese Dokumentation beschreibt die Natural-Systemkommandos und 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.

1 Über diese Dokumentation

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

Dokumentationskonventionen

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

Online-Informationen und Support

Produktdokumentation

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

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

Produktschulungen

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

Tech Community

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

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

Produktsupport

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

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

Datenschutz

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

2 Kommandoausführung

- Kommandoeingabe 6
- Kommandozeile 6
- NEXT-Zeile 7
- MORE-Zeile 7

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

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*).

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.

3 Systemkommando-Syntax

- Syntax-Elemente 10
- Kommandosyntax-Beispiel 11

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.
ABCDEF	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.
ABCDEF	Ist ein Teil eines Wortes in Großbuchstaben unterstrichen (kein Hyperlink!), kann der unterstrichene Teil als Abkürzung für das jeweilige Wort verwendet werden.
abcdef	Bei Angaben, die in *kursiven* 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 diese 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 ein 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 ein 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 ein in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
Andere Symbole (außer [ ] { } ... ,... :...)	Alle anderen Symbole außer den in dieser Tabelle definierten müssen genau so eingegeben werden wie angegeben. **Ausnahme:** Der skalare SQL-Verkettungsoperator wird durch zwei senkrechte Striche dargestellt, die genauso eingegeben werden müssen, wie sie in der Syntax-Definition erscheinen.

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.

4 Systemkommandos sortiert nach Kategorien

- In Natural navigieren 14
- Natural-Entwicklungsumgebung 14
- Anwendungen mit Natural-Programmobjekten pflegen 15
- Monitor-, Debug- und Trace-Funktionen 17
- Natural mit anderen Produkten 18

Dieser Abschnitt 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 NATCONFG 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 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 Produkte solche enthalten.
SYSCP	Mit diesem Kommando rufen Sie die SYSCP-Utility auf, die Codepage-Informationen liefert und benutzt werden kann, um Codepages für Natural-Quellcode-Objekte zu verwalten.
SYSEXT	Mit diesem Kommando rufen Sie die SYSEXT-Anwendung auf. Sie enthält verschiedene Natural-Programmierschnittstellen (API).
SYSEXV	Mit diesem Kommando rufen Sie die SYSEXV-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 *alle* 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 RUN, CATALL, CATALOG und STOW.
CLEAR	Mit diesem Kommando löschen Sie den Inhalt des Editor-Arbeitsbereichs.
DELETE	Dieses Kommando dient dazu, Quellcode-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 *nur* 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 benutzen, 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
SYSNCP	Mit diesem Kommando rufen Sie die SYSNCP-Utility auf, mit der Sie die in Ihren Anwendungen zu verwendenden Kommando-Prozessoren erstellen und pflegen können.
SYSOBJH	Mit diesem Kommando rufen Sie den Object Handler auf. Dort können Sie Natural- und Nicht-Natural-Objekte zwecks Verteilung in Natural-Umgebungen handhaben.
SYSRPC	Mit diesem Kommando rufen Sie die SYSRPC-Utility auf. Diese Utility bietet Funktionen zum Verwalten von Natural Remote Procedure Calls (Natural RPC).
UNLOCK	Mit diesem Kommando können Sie gesperrte Natural-Objekte anzeigen und entsperren.
WHICH	Dieses Kommando können Sie benutzen, um ein Objekt in den zur Laufzeit verwendeten Libraries zu finden (in der so genannten "Library-Suchreihenfolge").

Monitor-, Debug- und Trace-Funktionen

Kommando	Funktion
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 wie viel Puffer-Platz tatsächlich benutzt wird. **Anmerkung:** Das Kommando BUS hat die gleiche Funktion wie das Kommando SYSBUS, das nicht mehr zur Verfügung steht.
DUMP	Liefert Informationen, die der Support benötigt, um einen Fehler zu finden, der zum Abbruch (Abend) des Natural-Systems geführt hat.
HELP	Ruft das Natural-Hilfesystem auf. Es liefert Informationen zu Natural-Statements, -Kommandos und Fehlermeldungen.
LASTMSG	Zeigt zusätzliche Informationen zu der zuletzt aufgetretenen Fehlersituation an.
PROFILER	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.
RDC	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.
RPCERR	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.
SYSADA	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. Dieses Kommando steht nur zur Verfügung, wenn Adabas installiert ist.

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

Natural mit anderen Produkten

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

Produkt	Systemkommando	Kurzbeschreibung
Adabas	SYSADA	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	LISTSQLB	Ruft die Funktion **Explain all SQL Statements** in den **Natural Tools for DB2** auf. Siehe *Using Natural Tools for DB2* in der *Database Management System Interfaces*-Dokumentation.
	SQLDIAG	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.
	SYSDB2	Ruft die **Natural Tools for Db2** mit Funktionen auf, die zur Pflege von DBb-basierten Anwendungen benötigt werden. Siehe *Using Natural Tools for DB2* in der *Database Management System Interfaces*-Dokumentation.
Natural for Db2	LISTDBRM	Dient bei Natural for Db2 zur Anzeige von vorhandenen Datenbankverwaltungssystemen (DBRMs) mit Natural-Programmen oder zur Anzeige von Natural-Programmen, welche ein gegebenes DBRM referenzieren.

Produkt	Systemkommando	Kurzbeschreibung
	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.
	LISTPROF	Zeigt entweder eine Liste der SQLJ-Profile (erstellt durch statische NDZ-Generierung) für ein bestimmtes Natural-Programm oder eine Liste der Natural-Programme an, die auf ein bestimmtes SQLJ-Profil verweisen.
	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.
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 Produkte solche enthalten.

5 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*

6 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 wie viel Puffer-Platz tatsächlich benutzt wird. Näheres zu dieser Funktion finden Sie unter *SYSTP Utility > Buffer Usage Statistics (BUS)* in der *Debugger und Dienstprogramme*-Dokumentation.

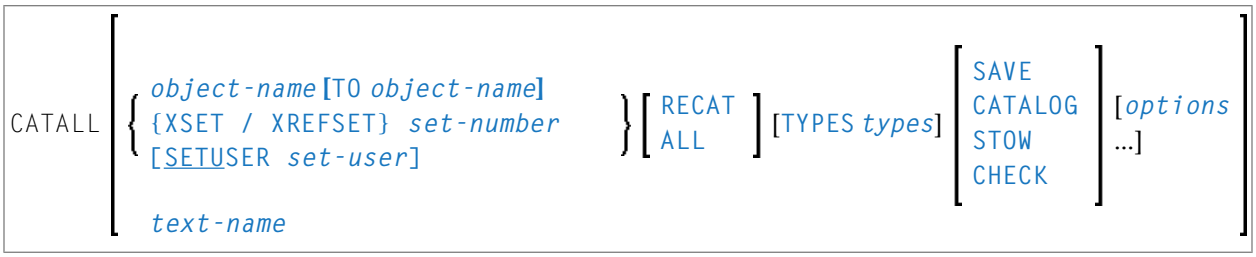


Anmerkung: Das Kommando BUS hat die gleiche Funktion wie das Kommando SYSBUS, das nicht mehr zur Verfügung steht.

Die vom Systemkommando BUS gelieferten Informationen erhalten Sie auch mit Hilfe der Programmierschnittstelle (API) USR1019N. Siehe *SYSEXT Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

7 CATALL

- Objekte katalogisieren von/bis 26
- Predict Set und Set Benutzer 27
- Vorhandene Module neu katalogisieren oder alle Quellcode-Objekte katalogisieren 27
- Objekttypen auswählen 28
- Funktion auswählen 28
- Optionen auswählen 28
- Auswahlliste 29
- Direktkommando-Syntax 30



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 **Objekte in Library katalogisieren**, 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 **Objekte in Library katalogisieren** aufgelistet sind (siehe auch Informationen zu TYPES im Abschnitt *Direktkommando-Syntax*.

Außerdem können Sie mit Hilfe des Subprogramms CATALLU2 standardmäßig Funktionen auf dem **Objekte in Library katalogisieren**-Schirm auswählen. Darüber hinaus können Sie auswählen, ob CATALLU2 im Batch Mode oder im Command Mode aufgerufen wird. Dieses Subprogramm wird in Quellcode-Form in der Library SYSTEM (FNAT) mitgeliefert. Um das Subprogramm zu aktivieren, müssen Sie es entsprechend den Angaben im Quellcode modifizieren, es anschließend katalogisieren und in die Library SYSLIB kopieren. Das Subprogramm wird aufgerufen, bevor der Schirm **Objekte in Library katalogisieren** 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 aus einer Auswahlliste übernehmen (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 Retained Set bezeichnet, welcher für die Namen der zu verarbeitenden Objekte verwendet werden soll.

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

Ein Retained 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 Quellcode-Objekte katalogisieren

Diese Option bewirkt Folgendes:

- Wenn Sie das Feld **Nur katalogisierte Objekte auswählen** markieren, werden nur die Quellcode-Objekte in der aktuellen Library katalogisiert, die bereits früher katalogisiert worden waren (ein entsprechendes katalogisiertes Objekt existiert).
- Wenn Sie das Feld **Alle Source-Objekte auswählen** markieren, werden *alle* Quellcode-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 CATALOG und STOW).
Quellcode-Zeilen neu numerieren	Standardmäßig werden mit den Funktionen SAVE und STOW die Quellcode-Zeilen von Quellcode-Objekten, die mit SAVE oder STOW bearbeitet wurden, auch neu nummeriert. Wenn Sie keine automatische Neunummierung 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.

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

Auswahlliste

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

Hierzu machen Sie zunächst die gewünschten Angaben unter **Funktion auswählen** und **Optionen auswählen** 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 Schirm **Objekte in Library katalogisieren** machen können, gibt es auch entsprechende Optionen, die Sie direkt mit dem Systemkommando CATALL angeben können:

object-name TO *object-name*	Entspricht den Feldern **Objekte katalogisieren von** und **bis** des Bildschirms **Objekte in Library katalogisieren**, siehe *Objekte katalogisieren von/bis*.
{XSET / XREFSET} *set-number*	Entspricht dem Feld **Predict Set** im Bildschirm **Objekte in Library katalogisieren**, siehe *Predict Set und Set User*.
[SETUSER *set-user*]	Entspricht dem Feld **Set Benutzer** im Bildschirm **Objekte in Library katalogisieren**, siehe *Predict Set und Set User*.
RECAT / ALL	Entspricht den Optionen **Nur katalogisierte Objekte auswählen** oder **Alle Source-Objekte auswählen** des Bildschirms **Objekte in Library katalogisieren**. RECAT ist der Standardwert, siehe *Vorhandene Module neu katalogisieren oder alle Quellecode-Objekte katalogisieren*.

<table>
<tr><td>TYPES types</td><td>Entspricht den markierten Objekttypen des Bildschirms Objekte in Library katalogisieren. Mögliche (in der unten aufgeführten Reihenfolge verarbeitete) types sind:

G Global Data Areas
L Local Data Areas
A Parameter Data Areas
C Copycodes
T Texte
7 Functions
N Subprogrammme
S Externe Subroutinen
H Helproutinen
M Maps
8 Adapter
P Programme
4 Classes
* Alle Typen (gilt standardmäßig)

Die types müssen als eine Zeichenkette angegeben werden (z.B. LAG für Local, Parameter und Global Data Areas). Standardmäßig gilt CATALL für alle Objektarten in der aktuellen Library.</td></tr>
<tr><td>SAVE / CATALOG / STOW / CHECK</td><td>Entspricht den gleichnamigen Funktionen des Bildschirms Objekte in Library katalogisieren, siehe Funktion auswählen. CATALOG ist der Standardwert.</td></tr>
<tr><td>options</td><td>Diese Optionen entsprechen den Optionen auswählen auf dem Bildschirm Objekte in Library katalogisieren, siehe Optionen auswählen. Mögliche options sind:
<table>
<tr><td>CC</td><td>Condition Code wird zurückgegeben.</td></tr>
<tr><td>NOREN</td><td>Keine automatische Neunummerierung von Quellcode-Zeilen.</td></tr>
<tr><td>KEEP</td><td>Ergebnisliste wird aufbewahrt.</td></tr>
<tr><td>NOSCROLL</td><td>Online: Keine laufende Anzeige von Verarbeitungsstatus-Informationen. Batch: Ausgabe derjenigen Objekte, die einen Fehler verursacht haben.</td></tr>
<tr><td>NOREPORT</td><td>Keine Fehlerliste.</td></tr>
<tr><td>FULL</td><td>Erweiterte Fehlerliste.</td></tr>
<tr><td colspan="2">EL <<i>text-object</i>> [R]</td></tr>
<tr><td>EL text-object</td><td>Ausgabe der Fehlerliste in ein Natural-Text-Objekt.</td></tr>
<tr><td>R</td><td>Wenn ein Objekt schon existiert, ist der EL-Parameter funktionslos, wenn kein R</td></tr>
</table></td></tr>
</table>

		(Replace/Ersetzen) nach *text-object* angegeben wird.
	Anmerkung: Bei Angabe von NOREPORT *und* NOSCROLL, gilt automatisch auch KEEP.	
text-name	Entspricht der Angabe eines Text-Namens im Feld **Objekte katalogisieren von** des Bildschirms **Objekte in Library katalogisieren**, siehe *Objekte katalogisieren von/bis*.	

Beispiele:

Um nur die Objekte in Quellcode- 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 Quellcode- 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.

8 CATALOG

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

Verwandte Kommandos: SAVE | STOW.

Mit dem Systemkommando CATALOG katalogisieren (kompilieren) Sie das im Arbeitsbereich eines Editors befindliche Quellcode-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 *object-name* angeben, wird das Programm in der aktuellen Library unter dem Namen des Quellcode-Objekts gespeichert, das zuletzt in den Arbeitsbereich gelesen wurde (z.B. mit einem EDIT-, READ- oder RUN-Kommando). Eventuell vorhandener Objektcode wird ersetzt.
CATALOG *object-name*	Ein neues Objekt wird angelegt. Als *object-name* 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 *object-name library-id*	Wollen Sie das neue Objekt in einer anderen Library katalogisieren, müssen Sie zusätzlich zum Objektnamen die *library-id* 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.

9 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.

10 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 Quellcode-Objekt im Arbeitsbereich, empfiehlt es sich, den Arbeitsbereich mit diesem Kommando zu löschen.

11 COMPOPT

- Syntax-Erklärung 40
- Compiler-Schlüsselwortparameter angeben 40
- Compiler-Optionen 41

```
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 **Compilations Options**-Schirm angezeigt, auf dem Sie die unten beschriebenen Optionen ein- bzw. ausschalten können. Die dort vorhandenen Schlüsselwörter werden nachfolgend beschrieben.
COMPOPT *option=value*	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-Parameter-modul 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 Kommandozuordnung (`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.

Compiler-Optionen

- CHKRULE - Validierung von INCDIR-Statements in Masken
- CPAGE - Codepage-Unterstützung bei alphanumerischen Konstanten
- DBSHORT - Interpretation von Datenbankfeld-Kurznamen
- DB2ARRY - 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

- MASKCME - MASK-Option kompatibel mit MOVE EDITED
- MAXPREC – Maximale Anzahl Nachkommastellen
- MEMOPT - Speicheroptimierung für lokal deklarierte Variablen
- NMOVE22 - Zuordnung von numerischen Variablen mit derselben Länge und Genauigkeit
- 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
- V82COMP - Auf Natural für Großrechner Version 8.2 nicht verfügbare Syntax unterbinden

Diese Compiler-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`	Die `INCDIR`-Validierung ist eingeschaltet. Wenn die bzw. das im `INCDIR`-Kontrollstatement referenzierte Datei (DDM) bzw. Feld nicht existiert, wird zur Kompilierungszeit ein Syntaxfehler (NAT0721) ausgegeben. 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 `INCDIR`-Statement in den Map-Statement-Rumpf ein, um damit das Laden und die Übernahme einer Predict-Regel auszulösen, wenn die Map (per `STOW`-Kommando) kompiliert wird. Die Funktionsweise ist ähnlich dem Vorgang bei der Verarbeitung eines `INCLUDE`-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 `INCDIR`-Statement angegeben. Eine während der Kompilierung angeforderte `INCDIR`-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. Bei Übernahme von Feldern aus einer DDM in eine Map, werden entsprechende `INCDIR`-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 `INCDIR`-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.

OFF	Die INCDIR-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 *Compiler-Option CPAGE* in der Dokumentation *Unicode- und Codepage-Unterstützung*.

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 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 (Identifier) 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	Die Verwendung eines Kurznamens zum Referenzieren eines Datenbankfeldes ist erlaubt, jedoch wird die Verwendung eines Datenbank-Kurznamens *nicht* generell gestattet (selbst wenn DBSHORT=ON) ■ für die Definition eines Feldes, wenn ein View erstellt wird; ■ wenn im Programmcode ein View verwendet wird; ■ wenn im Programmcode ein DEFINE DATA LOCAL-Statement zur Definition von Variablen verwendet wird; ■ wenn Natural Security aktiv ist. Dies ist der Standardwert.
OFF	Ein Datenbankfeld kann nur über seinen Langnamen referenziert werden. Jede Datenbankfeldkennung wird unabhängig von ihrer Länge als Langnamen-Referenz betrachtet. 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. 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. 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.

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*<= 253) oder VARCHAR (253<*n*<=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*<=255) oder VARBIN (255<*n*<=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	Der Compiler prüft das Vorhandensein eines Programmierobjekts, welches in einem objektaufrufenden Statement angegeben ist, zum Beispiel in FETCH [RETURN/REPEAT], RUN [REPEAT], CALLNAT, PERFORM, INPUT USING MAP, PROCESS PAGE USING, Function Call, Helproutine-Aufruf. Die Existenzprüfung basiert auf der Suche nach dem katalogisierten Objekt oder nach der Source des Objekts, wenn es von einem RUN [REPEAT]-Statement aufgerufen wird. Dazu ist erforderlich, dass der Name des aufzurufen bzw. auszuführenden (RUN) Objekts als alphanumerische Konstante (und nicht als alphanumerische Variable) definiert ist. Andernfalls hat die Einstellung ECHECK=ON keine Auswirkung. **Fehler-Behandlung bei ECHECK=ON** Die Existenzprüfung wird nur dann ausgeführt, wenn das Objekt keine syntaktischen Fehler enthält. Die Existenzprüfung wird bei jedem Statement ausgeführt, welches ein Objekt aufruft. Gesteuert wird die Existenzprüfung durch den Profilparameter PECK (siehe *Parameter-Referenz*-Dokumentation). **Probleme bei der Benutzung des CATALL-Kammandos mit ECHECK=ON** Wenn ein CATALL-Systemkommando zusammen mit der Einstellung ECHECK=ON benutzt wird, ist folgendes zu beachten: Wenn ein CATALL-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, Helproutinen, Maps, Adapter, Programme, Klassen. Bei Objekten desselben Typs bestimmt die alphabetische Reihenfolge der Namen die Abfolge, in der sie katalogisiert werden.

	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 ECHECK-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 CATALL erstellt worden. Lösung: ■ Setzen Sie die Compiler-Option ECHECK auf OFF. ■ Führen Sie mit dem CATALL-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. ■ Setzen Sie die Compiler-Option auf ECHECK=ON. ■ Führen Sie mit der Funktion CHECK des CATALL-Kommandos eine generelle Kompilierung in der gesamten Library durch.
OFF	Es erfolgt keine Existenzprüfung. Dies ist der Standardwert.

GDASC - GDA-Signaturprüfung

Diese Option dient zum Speichern der Strukturinformationen eines GDA (Global Data Area). Diese GDA-Informationen werden als „GDA-Signatur“ bezeichnet. Eine GDA-Signatur ist ein Hash-Wert, der aus den GDA-Datendefinitionen berechnet wird, darunter Variablenname, Format, Länge und Anfangswerte.

ON	Wenn ein GDA katalogisiert wird, wird die GDA-Signatur im katalogisierten GDA gespeichert.
OFF	Wenn ein GDA katalogisiert wird, wird der GDA ohne GDA-Signatur gespeichert. Dies ist der Standardwert.

Wenn ein Natural-Objekt, das eine GDA referenziert, katalogisiert wird und die GDA eine GDA-Signatur enthält, wird diese GDA-Signatur unabhängig von der aktuellen GDASC-Einstellung in das katalogisierte Objekt kopiert.

Zum Zeitpunkt der Ausführung wird die GDA-Signatur des ausgeführten Natural-Objekts mit der GDA-Signatur der referenzierten GDA verglichen. Wenn beide Signaturen vorhanden und identisch sind, wird die GDA akzeptiert, auch wenn die Zeitstempel nicht identisch sind.

Die folgende Tabelle beschreibt detailliert, was zum Zeitpunkt der Ausführung verglichen wird und welches Ergebnis zu erwarten ist.

GDA enthält Signatur	Natural-Objekt enthält Signatur	Vergleich	Ergebnis
ja	ja	Signaturen	Fehler, wenn nicht gleich
ja	nein	Zeitstempel	Fehler
nein	ja	Zeitstempel	Fehler
nein	nein	Zeitstempel	Fehler, wenn nicht gleich

Aktuelle Einstellungen der GDA-Signaturen ermitteln

Das Koammndo `LIST DIRECTORY` zeigt das Feld `GDA Signature` an, wenn ein GDA-Objekt oder ein Natural-Objekt eine GDA-Signatur enthält. Bei einem GDA wird geprüft, ob eine GDA-Signatur vorhanden ist. Bei einem Natural-Objekt wird die GDA-Signatur auf Übereinstimmung mit der Signatur des verwendeten GDA überprüft.

Aus der folgenden Tabelle können Sie die Einstellungen des Feldes `GDA Signature` entnehmen:

GDA enthält Signatur	Natural-Objekt enthält Signatur	GDA-Signatur in LIST DIRECTORY einer GDA	GDA-Signatur in LIST DIRECTORY eines Natural-Objekts
ja	ja	Generiert	Identisch / nicht identisch
ja	nein	Generiert	Nur im GDA
nein	ja	-*1	Nur im Objekt
nein	nein	-*1	-*2

Anmerkungen:

*1: Zeigt den GDA-Zeitstempel an, zu dem die GDA katalogisiert wurde.

*2: Zeigt die GDA-Zeitstempel an, zu denen die GDA und das Natural-Objekt katalogisiert wurden.

Vorteile der Verwendung von GDA-Signaturen

Die GDA-Signatur einer GDA bleibt bestehen, wenn eine GDA nur rekatalogisiert, aber nicht geändert wird oder wenn nur die Kommentare in der GDA geändert werden. In diesen Fällen haben Natural-Objekte, die auf die GDA verweisen, weiterhin die gleiche GDA-Signatur der GDA und brauchen deshalb nicht rekatalogisiert zu werden.

Aktivieren von GDA-Signaturen

Um GDA-Signaturen zu aktivieren, müssen Sie `GDASC` auf `ON` setzen.

Wenn Sie einen GDA zum ersten Mal mit `GDASC=ON` katalogisieren, müssen Sie auch alle referenzierenden Natural-Objekte katalogisieren. GDA-Signaturen werden dann bei der Ausführung verwendet.

Andere Bibliotheken, in denen der GDA (noch) nicht katalogisiert ist und `GDASC` auf `ON` gesetzt ist, verwenden weiterhin GDA-Zeitstempel für GDA-Validierungsprüfungen. Die Natural-Objekte bleiben ohne GDA-Signatur.

GFID - Generierung globaler Formatkennungen

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

`ON`	Global Format IDs werden für alle Datensichten (Views) generiert. Dies ist der Standardwert.
`VID`	Global Format IDs werden nur für in Local/Global Data Areas definierte Views generiert, aber nicht für innerhalb von Programmen definierte Views.
`OFF`	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=abccddee
```

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-Dienstprogramme (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, Variablennamen 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 Variablennamen 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 ↩
```

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	Der Bereich der gültigen Jahreswerte, die zu den YYYY-Maskenzeichen passen, ist 0000 - 2699. Dies ist der Standardwert. Wenn der Profilparameter MAXYEAR auf 9999 gesetzt ist, erstreckt sich der Bereich der gültigen Jahreswerte von 0000 bis 9999.

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	Dieser Wert gibt die maximale Anzahl der Nachkommastellen an, die der Compiler bei den Ergebnissen von arithmetischen Operationen erzeugt.
	
	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.
	
	Falls bei Zwischenergebnissen eine höhere Genauigkeit benötigt wird, sollte der Wert von MAXPREC erhöht werden.
	
	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.
	
	`DEFINE DATA LOCAL`
	`1 P (P1.15)`
	`END-DEFINE`
	`P := P + 0.1234567890123456`
	`END`
	
	Weitere Informationen siehe Genauigkeit von Ergebnissen bei arithmetischen Operationen im *Leitfaden zur Programmierung*.

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 ■ ein auf Programmebene 1 befindliches Feld, falls auf das Feld oder die Redefinition dieses Feldes zugegriffen wird; ■ eine Gruppe, wenn auf die Gruppe oder mindestens ein Gruppenfeld zugegriffen wird.
OFF	Datenspeicher wird für alle Gruppen und Felder zugeordnet, die lokal benannt wurden. Dies ist der Standardwert.

NMOVE22 - Zuordnung von numerischen Variablen mit derselben Länge und Genauigkeit

ON	Die Zuordnung von numerischen Variablen, bei denen Source und Target dieselbe Länge und Genauigkeit haben, erfolgt wie bei Natural Version 2.2.
OFF	Die Zuordnung von numerischen Variablen, bei denen Source und Target dieselbe Länge und Genauigkeit haben, erfolgt wie bei Natural Version 2.3 und höher, d.h., sie werden so verarbeitet, als ob Source und Target eine unterschiedliche Länge oder Genauigkeit hätten. Dies ist die Standardeinstellung.

PCHECK - Parameter-Prüfung bei CALLNAT-Statements

Prüfung der Parameter bei Objekt aufrufenden Statements.

ON	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. Die Parameterprüfung basiert auf dem Vergleich der Parameter des Objekt aufrufenden Statements mit den DEFINE DATA PARAMETER-Definitionen in dem aufzurufenden Subprogramm. Dazu ist folgendes erforderlich: ■ Der Name des aufzurufenden Subprogramms muss als alphanumerische Konstante (nicht als alphanumerische Variable) definiert sein. ■ Das aufzurufende Subprogramm muss als katalogisiertes Objekt zur Verfügung stehen. Andernfalls hat die Einstellung PCHECK=ON keine Auswirkung. **Fehler-Behandlung bei ECHECK=ON** Die Parameterprüfung wird nur dann ausgeführt, wenn das Objekt keine syntaktischen Fehler enthält. Die Parameterprüfung wird bei jedem Statement ausgeführt, welches ein Objekt aufruft. Gesteuert wird die Parameterprüfung durch den Profilparameter PECK (siehe *Parameter-Referenz*-Dokumentation). **Probleme bei der Benutzung des CATALL-Kammandos mit PCHECK=ON** Wenn ein CATALL-Kommando zusammen mit der Einstellung PCHECK=ON benutzt wird, ist folgendes zu beachten: Wenn ein CATALL-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, Helproutinen, Maps, Adapter, Programme, Klassen. Bei Objekten desselben Typs bestimmt die alphabetische Reihenfolge der Namen die Abfolge, in der sie katalogisiert werden. Wie zuvor erwähnt werden die Parameter 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. Das hat zur Folge, dass das *neue* Parameterlayout im Objekt aufrufenden Statement mit dem *alten* Parameterlayout im DEFINE DATA PARAMETER-Statement des aufgerufenen Subprogramms verglichen wird. Lösung:

	■ Setzen Sie die Compiler-Option auf PCHECK=OFF. ■ Führen Sie mit dem CATALL-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. ■ Setzen Sie die Compiler-Option auf PCHECK=ON. ■ Führen Sie mit der Funktion CHECK des CATALL-Kommandos eine generelle Kompilierung in der gesamten Library durch.
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 THSEPCH 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 ↩
```

TSENABL - 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 TSENABL=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.

V82COMP - Auf Natural für Großrechner Version 8.2 nicht verfügbare Syntax unterbinden

Die Option V82COMP prüft, ob die Syntax der seit Natural für Großrechner Version 9.2 zur Verfügung stehenden Funktionalität auch durch Natural für Großrechner Version 8.2 unterstützt wird. Falls Syntaxunverträglichkeiten zwischen diesen beiden Versionen festgestellt werden, wird die Kompilierung mit entsprechender Fehlermeldung und Reason Code abgebrochen.

Mögliche Werte:

ON	Mit Syntaxprüfung: Wenn ein Programm kompiliert wird, wird jeder Versuch, ein Syntaxkonstrukt zu verwenden, das von Natural für Großrechner Version 9.2, aber nicht von Natural für Großrechner Version 8.2 unterstützt wird, mit dem Syntaxfehler NAT0597 und einem entsprechenden **Reason Code** (siehe unten) zurückgewiesen.
OFF	Ohne Syntaxprüfung: Es erfolgt keine Syntaxprüfung. Dies ist die Standardeinstellung.

Reason Codes bei Syntaxfehlern

Die folgenden Reason Codes zeigen an, welche Syntaxteile nicht unter Natural für Großrechner Version 8.2 unterstützt werden:

Reason Code	Ungültige Syntax auf Natural für Großrechner Version 8.2
019	Es wurde die folgende neue Systemvariable benutzt: `*TIMESTMPX`

020	Das `COMPRESS`-Statement gestattet die Verwendung eines Feldes des Typs `LOGICAL`.
	
	Beispiel:
	
	`DEFINE DATA LOCAL`
	`1 #L (L)  INIT <TRUE>`
	`1 #RESULT (A50)`
	`END-DEFINE`
	`COMPRESS 'The value of #L is' #L INTO #RESULT`

021	Das `COMPRESS`-Statement gestattet es, jedes Feld mit einer Editiermaske zu versehen.
	
	Beispiel:
	
	`DEFINE DATA LOCAL`
	`1 #A (A10)  INIT <'ABCDEF'>`
	`1 #P (P5.3)  INIT <-12.345>`
	`1 #L (L)  INIT <TRUE>`
	`1 #RESULT (A50)`
	`END-DEFINE`
	`COMPRESS #A (EM=X_X)  #P(EM=-ZZZ9.9)  #L(EM=FALSE/TRUE) INTO #RESULT`

022	In einem `DEFINE WORK`-, `WRITE WORK`-, `READ WORK`- und `CLOSE WORK`-Statement kann die Work File-Nummer mit einer `CONST`-Variablen versehen werden.
	
	Beispiel:
	
	`DEFINE DATA LOCAL`
	`1 #WNBR (I2)  CONST <12>`
	`END-DEFINE`
	`WRITE WORK FILE #WNBR 'ABC'`
	`CLOSE WORK FILE #WNBR`

023	Das Statement `READ .. BY ISN` wurde verbessert, so dass rückwärts in absteigender Reihenfolge (`DESCENDING`) gelesen werden kann. Die Nutzung dieser Funktion setzt jedoch voraus, dass Adabas-Version 8.6 (oder höher) verwendet wird. Beispiel:

Reason Code	Ungültige Syntax auf Natural für Großrechner Version 8.2

	`DEFINE DATA LOCAL`
	`1 VIEW1 VIEW OF EMPLOYEES`
	`2 NAME`
	`END-DEFINE`
	`READ (10) VIEW1 DESCENDING BY ISN = 1000  /* Reads 10 records starting ↩`
	`with ISN=1000`
	`DISPLAY *ISN NAME  /* (or the next lower) in ↩`
	`descending order`
	`END-READ`

024	Das Statement `READ .. BY ISN` wurde erweitert, sodass nun ein Endwert an die Datenbank gesendet werden kann, indem die Option TO zwischen Start- und Endwert verwendet wird. Die Verwendung dieser Funktion erfordert Adabas Version 8.6 (oder höher).
	
	Beispiel:
	
	`DEFINE DATA LOCAL`
	`1 VIEW1 VIEW OF EMPLOYEES`
	`2 NAME`
	`END-DEFINE`
	`READ VIEW1 BY ISN = 11  TO  20  /* Reads all records between ISN=11 and 20.`
	`DISPLAY *ISN NAME  /* At loop end, the record ↩`
	`in VIEW1 is the last record found in this range.`
	`END-READ`

025	Das Statement `END-`*`XXX`* wurde erweitert, um die Verwendung von Label-Namen oder Zeilennummern zu ermöglichen.
	
	Beispiel 1:
	
	`DEFINE DATA LOCAL`
	`1 EMPLOY-VIEW VIEW OF EMPLOYEES`
	`  2 PERSONNEL-ID`
	`  2 NAME`
	`  2 JOB-TITLE`
	`  2 CITY`
	`END-DEFINE  ↩`
	
	`FD. FIND EMPLOY-VIEW WITH CITY = 'PARIS'`
	`  WHERE JOB-TITLE = 'INGENIEUR COMMERCIAL'`
	`  DISPLAY NOTITLE`
	`  CITY JOB-TITLE PERSONNEL-ID NAME`
	`END-FIND (FD.)  ↩`
	
	`END`

Reason Code	Ungültige Syntax auf Natural für Großrechner Version 8.2
	Beispiel 2:

12 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 NATCONFG definierten Codepages.

Codepage-Informationen

Auf dem ersten Bildschirm werden folgende Informationen angezeigt:

Feld	Erläuterung
*LOCALE	Zeigt die Sprache und das Land der aktuellen Region. Siehe auch Systemvariable *LOCALE.
*CODEPAGE	Zeigt den IANA-Namen der zurzeit für Konvertierungen in und von Unicode benutzten Codepage, und wird mit dem Natural-Profilparameter CP gesetzt. Siehe auch Systemvariable *CODEPAGE.
Source area CP	Zeigt die aktuelle Codepage des Source-Bereichs. Siehe auch Profilparameter CP.
CPOBJIN	Gibt die Codepage an, in der die Batch-Eingabedatei für Daten kodiert ist. Siehe auch Profilparameter CPOBJIN.
CPSYNIN	Gibt die Codepage an, in der die Batch-Eingabedatei für Kommandos kodiert ist. Siehe auch Profilparameter CPSYNIN.
CPPRINT	Gibt die Codepage an, in der die Batch-Ausgabedatei für Daten kodiert werden soll. Siehe auch Profilparameter CPPRINT.
CPCVERR	Gibt an, ob ein Konvertierungsfehler, der beim Konvertieren von Format U nach A oder von A nach U oder zwischen A-Formaten mit verschiedenen Codepages auftritt, zu einem Natural-Fehler führt oder nicht. Siehe auch Profilparameter CPCVERR.
SRETAIN	Gibt an, dass alle existierenden Quelldaten in ihrem ursprünglichen Codierformat gespeichert werden sollen. Siehe auch Profilparameter SRETAIN.
CFICU	Schaltet die Unicode-Unterstützung bei diversen Unicode-Einstellungen ein. Siehe auch Profilparameter CFICU.

Feld	Erläuterung
OPRB ACODE/WCODE	Diese Werte stellen den Schlüssel der gelieferten Kodierung für A-Felder und W-Felder während der Benutzer-Session für eine gegebene Datenbankkennung (DBID) dar. Um die DBID für OPRB ACODE/WCODE zu ändern, müssen Sie PF4 drücken. Es erscheint ein Fenster, in dem Sie die DBID für OPRB ACODE/WCODE ändern können.
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	Produktname: ICU for Adabas & Natural
Product code	Produktcode.
Product version	Version des Produkts ICS.
Revision	ICS Revisionsnummer.
Revision date	Erstellungsdatum der Revision.
z/OS driver	Erstellungsdatum des Treibers für z/OS.
z/VM driver	Erstellungsdatum des Treibers für z/VM.
ICU version	Aktuelle ICU-Version.
Unicode version	Aktuelle Unicode-Version.

In NATCONFG definierte Codepages anzeigen

Setzen Sie das Feld Display code pages defined in NATCONFG (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 NATCONFG 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 `NATCONFG` angezeigt. `E` Codepage ist in `NATCONFG` definiert und eingeschaltet. `D` Codepage ist in `NATCONFG` definiert, aber ausgeschaltet.

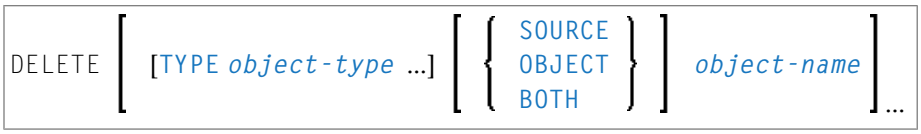
Weitere Informationen

Siehe auch:

- *Unicode- und Codepage-Unterstützung*
- *SYSCP Utility*

13 DELETE

- Syntax-Erklärung 66
- Auswahlliste 67
- Schutz vor versehentlichem Löschen 67
- Beispiele 67



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

object-name	Als *object-name* geben Sie den Namen des Objekts bzw. die Namen der Objekte an, das bzw. die Sie löschen wollen. Zusätzlich können Sie angeben, ob das Quellcode-Objekt, das katalogisierte Objekt oder beide gelöscht werden sollen: SOURCE – Quellcode-Objekt OBJECT – Katalogisiertes Objekt BOTH – Sowohl Quellcode-Objekt als auch katalogisiertes Objekt. Dies ist der Standardwert. 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 *object-name* Stern-Notation (*) verwenden.
object-type	In Verbindung mit der Stern-Notation (*) für *object-name* können Sie auch einen Objekttyp (*object-type*) angeben, wenn Sie nur Objekte eines bestimmten Typs löschen möchten. Die möglichen Angaben für *object-type* entsprechen denen des Systemkommandos EDIT. Außerdem können Sie X (= Global, Local und Parameter Data Areas) und U (= Subprogramme, Subroutinen und Helproutinen) angeben. **Anmerkung:** 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 Quellcode-Objekt und das katalogisierte Objekt `JOHN`, die Quellcode-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 Quellcode-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 TYPE GLA D*
```

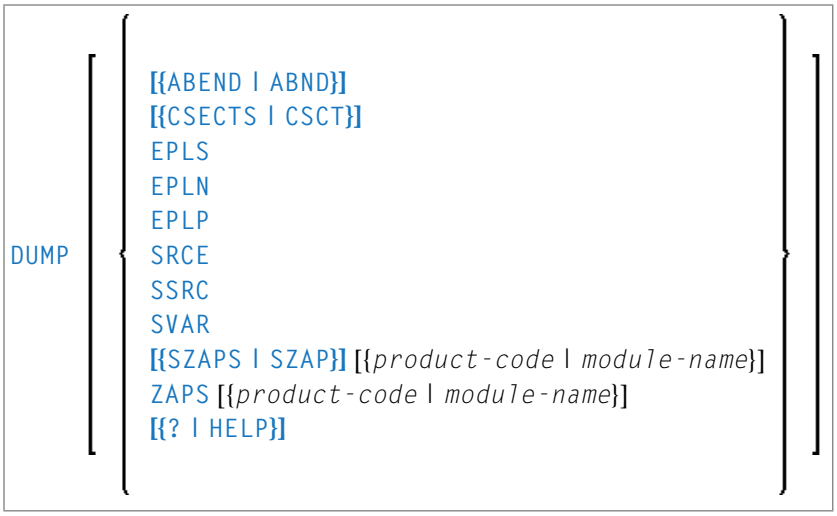
Mit diesem Kommando erhalten Sie eine Auswahlliste aller katalogisierten Objekte in der aktuellen Library, deren Namen mit `YYZ` anfangen:

```
DELETE OBJECT YYZ*
```

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

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

14 DUMP



Das Systemkommando `DUMP` liefert Informationen, die der Support benötigt, um einen Fehler zu finden, der zum Abbruch („Abend“) des Natural-Systems geführt hat. Übergeben Sie diese Informationen zwecks Fehlerdiagnose und -korrektur dem Support.

In der folgenden Tabelle bezieht sich der Begriff „Produkt“ auf das Natural-Basis-Produkt und Natural Add-On-Produkte. Die Listen der Quellcode-Änderungen oder Zaps, die von einem `DUMP`-Kommando zurückgegeben werden können, beziehen sich auf Quellcode-Änderungen und Zaps, die nach der Erstinstallation bei der aktuellen Produktversion eingespielt wurden.

`DUMP`	Zeigt Abbruchinformationen (Kernspeicher-Inhalt) an.

`DUMP ABEND`
 	
oder
`DUMP ABND`
	Zeigt nach einem Program Check (NAT0954) ganz übersichtlich auf einer Seite alle wichtigen Informationen an, wie z.B. Programm, Zeilennummer, Register, Abbruchstelle (`PSW`) usw.

`DUMP CSECTS` oder	Zeigt alle CSECTs im Nukleus an.

DUMP CSCT		
DUMP EPLS	Zeigt alle im Natural-Nukleus (z.B. QA92NRE) gelinkten 3-GL Routinen an, zusammen mit dem Aufrufzähler.	
DUMP EPLN	Wie DUMP EPLS, nur sortiert aufsteigend nach Namen.	
DUMP EPLP	Wie DUMP EPLN, wobei noch zusätzlich alle zum Frontend (z.B. QA92 oder NAT92) gelinkten Module mit aufgeführt sind.	
DUMP SRCE	Zeigt die Liste der Quellcode-Änderungen, die bei Produkten eingespielt wurden, die mit dem aktuellen Natural-Nukleus verlinkt sind.	
DUMP SSRC	Zeigt die Liste der speziellen Quellcode-Änderungen, die bei Produkten eingespielt wurden, die mit dem aktuellen Natural-Nukleus verlinkt sind.	
DUMP SVAR	Zeigt TP-Monitor- und betriebssystemabhängige Systemvariablen und zusätzliche Informationen.	
DUMP ZAPS oder DUMP SZAP	Zeigt die Liste der Zaps (DUMP ZAPS) bzw. nur der speziellen Zaps (DUMP SZAP), die bei Produkten eingespielt wurden, die mit dem aktuellen Natural-Nukleus verlinkt sind.	
	product-code	Zeigt die Liste der Zaps, die bei einem Einzelprodukt eingespielt wurden, das mit dem aktuellen Natural-Nukleus verlinkt ist. *product-code* ist der Buchstabenschlüssel, der dem erforderlichen Produkt entspricht., z.B. NAT für Basis-Natural. Eine Liste aller gültigen Produktschlüssel ist im Abschnitt *Übersicht über Natural-Add-On-Produkt-Versionen* in den aktuellen *Natural Version 9.2.1 Freigabemitteilung (Release Notes)* enthalten. Es gibt drei spezielle Produktcodes: ASM (für NATASM*vr*), RSM (für NATRSM*vr*) und DRV, der die Zaps für den aktuellen TP-Treiber anzeigt, auf dem Natural läuft.
	module-name	Zeigt die Liste aller Zaps, die bei einem Modul eingespielt wurden, das nicht mit dem aktuellen Natural-Nukleus verlinkt ist, jedoch mit der Natural Load Library in der Natural-Ausführungs-JCL oder in der entsprechenden Library des TP-Monitor-Programms verkettet ist. *module-name* ist der aus mindestens fünf Zeichen bestehende Name des Moduls. Den erforderlichen Modulnamen erfahren Sie vom Support.
DUMP ? oder DUMP HELP	Das DUMP-Kommando bietet noch eine Reihe weiterer Optionen (die auf den Hilfe-Bildschirmen erklärt sind, die Sie durch Eingabe eines Fragezeichens (?) oder der Option HELP im DUMP-Menü aufrufen können). Falls es für die Fehlerdiagnose erforderlich ist, erfahren Sie vom Support, wann und wie Sie diese Optionen einsetzen.	

15 EDIT

- Syntax 1 72
- Syntax 2 74
- Syntax 3 74

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

Anmerkung: Dieses Editierkommando steht nicht zur Verfügung, wenn Programm-Editor, Datenbereich-Editor (Data Area Editor) und Masken-Editor (Map Editor) in Ihrer Umgebung abgeschaltet (deaktiviert) sind. Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung* in der Editoren-Dokumentation.

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 Objektypen (`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 *Editoren*-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
```

16 EDT

- Syntax-Erklärung 78
- EDT-Kommandos 78
- EDT-Funktionstasten 79

Dieses Kommando ist hauptsächlich dazu gedacht, Natural-Codezeilen in den Quellcodebereich einzugeben, wenn Sie in einer Batch-Umgebung ohne Eingabeterminal arbeiten. Wenn die Natural-Sitzung online ausgeführt wird, empfehlen wir Ihnen, das Kommando EDIT anstelle des Kommandos EDT zu verwenden.

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

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

Mit dem Kommando .E können Sie das zeilenorientierte Editieren beenden.

Syntax-Erklärung

object-name	Als *object-name* geben Sie den Namen des Objekts an, das Sie editieren möchten (maximal 8 Zeichen lang). Wenn Sie *object-name* 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 SAVE-, CATALOG-, oder STOW-Kommando keinen Namen anzugeben. Wenn Sie keinen *object-name* 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.
library-id	Befindet sich das zu editierende Objekt in einer anderen Library als der, in der Sie gerade arbeiten, so müssen Sie die *library-id* dieser Library angeben. 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

EDT-Kommandos

Im EDT-Modus können Sie folgende Kommandos verwenden:

Kommando	Funktion
.B	Blättert zur letzten Zeile.
.C*nnnn*(*m*)	Kopiert *m* Zeilen, und zwar ab Zeile *nnnn*.
.C'*text*'(*m*)	Kopiert *m* Zeilen, und zwar ab der Zeile, die mit *text* beginnt.
.D	Löscht die Zeile.
.D(*n*)	Löscht die Zeile und die *n* minus 1 nachfolgenden Zeilen.
.E	Beendet das zeilenorientierte Editieren.
.I	Fügt eine Zeile ein.
.I(*program*)	Fügt das Objekt *program* ein.
.M*nnnn*	Verschiebt Zeile *nnnn*.
.M'*text*'(*m*)	Verschiebt *m* Zeilen, und zwar ab der Zeile, die mit *text* beginnt.
.R	Entspricht dem Systemkommando RENUMBER.
.S'*text*'	Sucht die Zeile mit der Zeichenkette *text*.
.T	Blättert zur ersten Zeile.
.*nnnn*	Blättert zu Zeile *nnnn*.
.+*n*	Blättert *n* Zeilen vor.
.-*n*	Blättert *n* Zeilen zurück.

EDT-Funktionstasten

Damit beim zeilenorientierten Editieren PF-Tasten benutzt werden können, müssen die PF-Tastenbelegungen mit Hilfe des Profilparameters KEY definiert worden sein. Siehe *KEY - PA-, PF- und CLEAR-Tastenbelegungen zuweisen* in der *Parameter-Referenz*-Dokumentation.

Die Zuweisung eines EDT-Unterkommandos zu einer PF- oder PA-Taste ist nicht möglich.

17 EXECUTE

- Syntax-Erklärung 82
- Beispiele für das EXECUTE-Kommando 83

```
{ 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`	Das Schlüsselwort `EXECUTE` ist nicht erforderlich; es genügt, *`program-name`*, d.h. den Namen des auszuführenden Programms anzugeben. **Vorsicht:** 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.
`REPEAT`	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`.
`program-name`	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.
`library-id`	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. Eine Library-ID, die mit `SYS` beginnt, darf nicht angegeben werden (Ausnahme: `SYSTEM`). Wenn Natural Security aktiv ist, ist es nicht möglich, eine Library-ID anzugeben, d.h. ein Objekt aus einer anderen Library auszuführen.
`parameter`	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. 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.

	Anmerkung: 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
```

18 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.

19 GLOBALS

- Syntax-Erklärung 88
- Liste der Parameter 88
- Zusammenhang zwischen GLOBALS, SET GLOBALS und anderen Statements 89

```
GLOBALS [parameter=value ...]
```

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

Syntax-Erklärung

GLOBALS	Falls Sie das GLOBALS-Kommando ohne Parameter eingeben, erscheint ein Bildschirm, der Ihnen die gegenwärtig gültigen Parameterwerte zeigt. Dort haben Sie auch die Möglichkeit, diese Werte zu ändern.
parameter	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 (Kurzbeschreibung*)
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
DO	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 (Kurzbeschreibung*)
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 Helproutinen
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

* Ausführliche Informationen zu den einzelnen Parametern sind in der *Parameter-Referenz*-Dokumentation enthalten.

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.

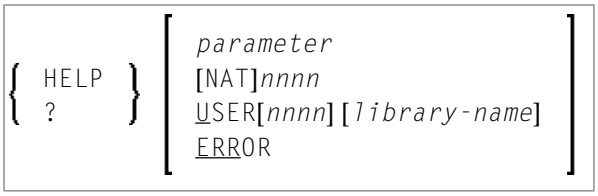
Die mit einem GLOBALS-Kommando angegebenen Parameterwerte gelten solange, bis Sie sie mit einem neuen GLOBALS-Kommando überschreiben, die Session beenden oder ein Logon in eine andere Library ausführen.

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.

20 HELP



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 parameter`	Zeigt Informationen zu dem betreffenden Parameter an. Mögliche Parameter sind:	
	Parameter	**Kommando-Beispiele**
	statement	`HELP FIND` `HELP DEFINE WORK FILE`
	system-variable	`HELP *CODEPAGE`
	system-function	`HELP SORTKEY`
	system-command	`HELP LIST`
	session-parameter	`HELP AD`
	terminal-command	`HELP %<TEST`
	Anmerkung: 1. Als *parameter* geben Sie nur die Grundform eines Statements, einer Systemvariablen, eines Systemkommandos usw. an. Gültiges Beispiel: `HELP AD`; ungültiges Beispiel: `HELP AD=O` 2. Bei einem ungültigen Kommando wird ein Parameterfehler angezeigt; zum Beispiel, wenn Sie Hilfe zu einem nicht unterstützten Hilfethema anfordern: `Parameter Error: SOSI`	

`HELP [NAT]nnnn`	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`*`nnnn`*.
`HELP NAT`	Zeigt Informationen zu allen Fehlermeldungen an.
`HELP USER`*`nnnn`*	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer *`nnnn`* aus der aktuellen Library an.
`HELP USER`*`nnnn`* `[`*`library-name`*`]`	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer *`nnnn`* aus der angegebenen Library an.
`HELP USER`	Zeigt den Langtext der library-spezifischen Fehlermeldung Nummer *`nnnn`* aus der aktuellen Library an.
`HELP USER [`*`library-name`*`]`	Zeigt eine Auswahlliste *aller* library-spezifischen Meldungen der aktuellen Library an.
`HELP ERROR`	Zeigt den Langtext des zuletzt aufgetretenen Fehlers an.

21 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).

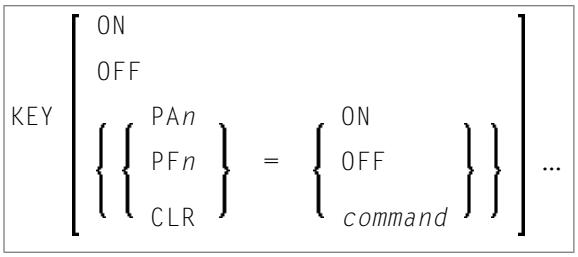
Ansonsten verwenden Sie zum Laden von Objekten in die Systemdateien den Object Handler.

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

Weitere Informationen siehe *INPL Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

22 KEY

- Kommandos zuweisen 97
- Aktivieren/Deaktivieren aller Tasten - KEY ON/OFF 97
- Aktivieren/Deaktivieren einzelner Tasten - KEY key=ON/OFF 98



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 KEY ON aktivieren Sie die mit KEY OFF 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

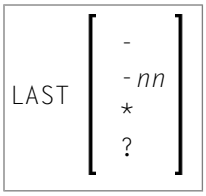
Mit dem Kommando `KEY key=OFF/ON` setzen Sie das einer bestimmten Taste zugewiesene Kommando außer Kraft.

`KEY key=OFF`	Damit setzen Sie das einer bestimmten Taste (*key*) zugewiesene Kommando außer Kraft. Beispiel: `KEY PF24=OFF`
`KEY key=ON`	Damit reaktivieren Sie eine zuvor außer Kraft gesetzte Kommandozuweisung. Beispiel: `KEY PF24=ON`



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

23 LAST



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

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

LAST -*nn*	Natural kann sich maximal die 20 letzten Kommandos „merken“; *nn* darf also nicht größer als 20 sein. Das *nn*tletzte ausgeführte Kommando wird in die Kommandozeile oder NEXT-Zeile gestellt und kann ausgeführt werden.
LAST *	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: ■ Wenn Sie ein *einzelnes* 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. ■ Wenn Sie *mehrere* 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.
LAST ?	Die Hilfe-Funktion für das LAST-Kommando wird aufgerufen.

24 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 oder PF5.

  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.

PF-Tasten

Im Ausgabebildschirm des Systemkommandos LASTMSG sind folgende PF-Tasten verfügbar:

PF-Taste	Name	Funktion
PF1	Help	Ruft die Hilfe auf. Liefert Informationen über die Verwendung von PF2 und PF5.
PF2	Clear	Löscht alle Fehlermeldungen aus dem Message Collector.
PF5	Info	Liefert detaillierte Informationen über einen Fehler, wenn der Cursor auf der Fehlermeldung steht und Sie PF5 drücken. Dasselbe geschieht, wenn Sie den Cursor auf eine Fehlermeldung setzen und ENTER drücken.



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.

25 LIST

- Syntax-Übersicht 104
- Inhalt des Arbeitsbereichs auflisten 112
- Quellcode eines einzelnen Objekts anzeigen 112
- Anzahl und Größe von Objekten anzeigen 112
- Quellcode mehrerer Objekte nacheinander anzeigen 113
- Liste von Objekten anzeigen 113
- Vorsortierte Liste von ausgewählten Objekten anzeigen 113
- Langnamen katalogisierter Subroutinen und Klassen anzeigen 114
- NOC-Optionen katalogisierter Objekte anzeigen 114
- Compiler-Optionen katalogisierter Objekte anzeigen 114
- Compiler Output zu einem katalogisierten Objekt anzeigen 114
- Directory-Informationen anzeigen 116
- DDMs (Views) anzeigen 118
- Optionen 118
- Objekt-Auswahlliste 123
- Quellcode-Liste 131
- Individuelles List-Profil erstellen 136

Mit dem Systemkommando LIST können Sie sich den Quellcode 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 Utility* in der *Debugger und Dienstprogramme*-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]

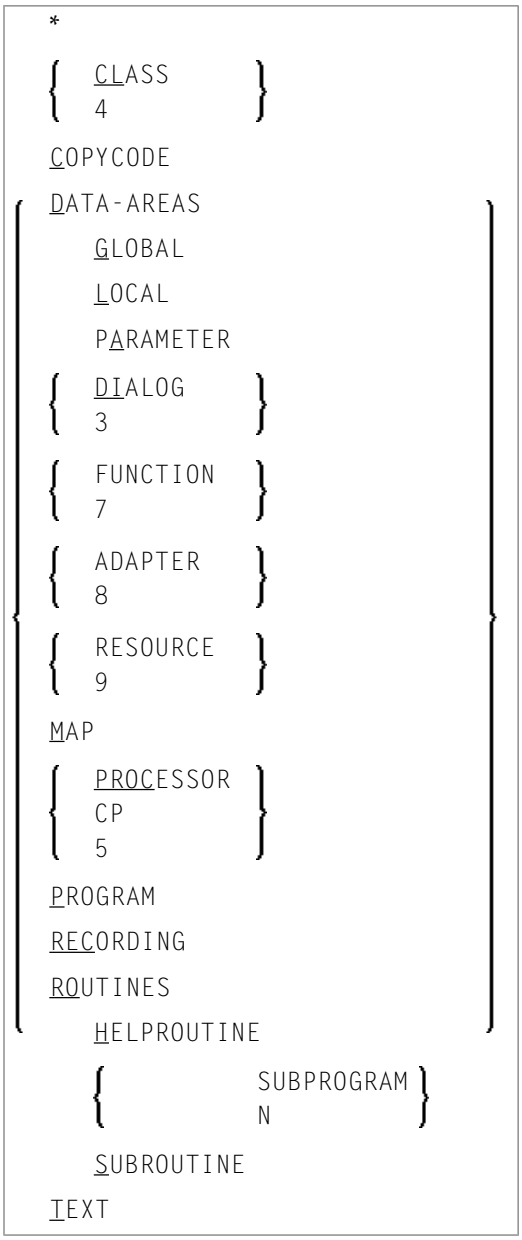
     DDM [ddm-name]
```

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.



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:

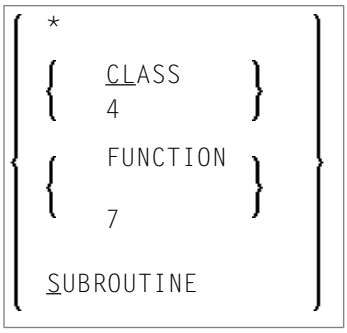
- Für eine Liste aller Objekte in der aktuellen Library geben Sie als *object-name-range* einen Stern (*) an, aber keinen *object-type*.
- Für eine Liste aller Objekte eines bestimmten Objekttyps geben Sie den gewünschten *object-type* sowie als *object-name-range* einen Stern (*) an.
- Für eine Liste 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.

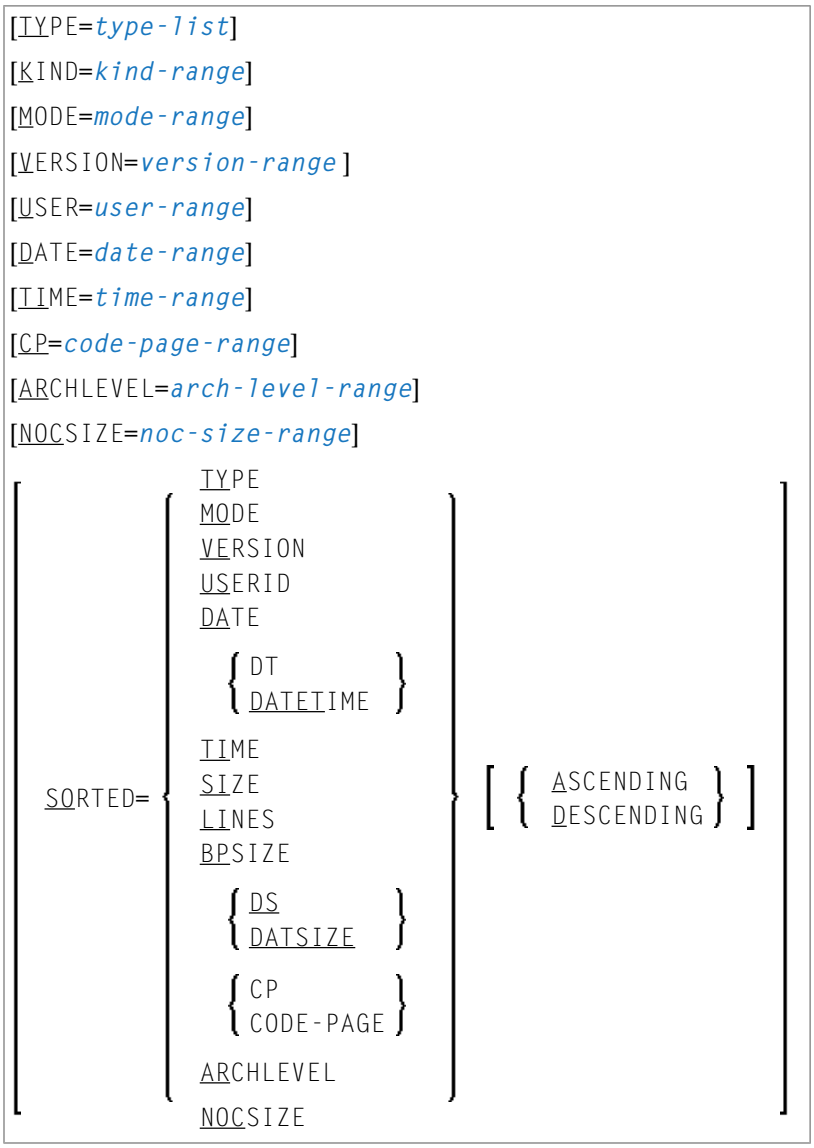
extended-type

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



Ausführliche Beschreibung siehe LIST EXTENDED.

range-clause



Anmerkungen:

1. `SORTED` kann beim `LIST COUNT`-Kommando nicht benutzt werden.
2. Wenn Sie bei `SORTED` nicht die Richtung angeben, wird die Sortierung in aufsteigender Reihenfolge ausgeführt.
3. Siehe auch Unterkommando `SORT` im Abschnitt ***Liste der Objekte sortieren***.

type-list	* (für alle Typen) oder eine Liste von 11 gültigen 1-Byte-Natural-Objekttyp-Zeichen (z.B. `P` für Program, `M` für Map).	
kind-range	*	Listet alle Objekte.
	S	Listet nur Objekte in Quellcodeform.
	C	Listet nur katalogisierte Objekte.
	S/C	Listet nur Objekte, die in Quellcodeform und als katalogisiertes Objekt vorhanden sind.
	S/	Listet nur Objekte auf, die nur als Quellcode-Objekte vorhanden sind.
	/C	Listet nur Objekte, die als katalogisierte Objekte vorhanden sind.
	W	Listet nur in Quellcodeform und Objektform gespeicherte Objekte.
mode-range	*	Listet alle Objekte.
	S	Listet nur Objekte, die im Structured Mode geschrieben wurden.
	R	Listet nur Objekte, die im Reporting Mode geschrieben wurden.
version-range	Die Natural-Version der Natural-Objekte. Siehe auch Begriffsdefinition von Version im *Glossary*. Sie können einen Bereich von Versionsnummern angeben, siehe *range-notation*.	
user-range	Die User-ID des Benutzers, der ein Natural-Programmierobjekt gespeichert oder katalogisiert hat. Sie können eine einzelne User-ID oder einen Bereich von User-IDs angeben, siehe *range-notation*.	
date-range	Listet alle Objekte mit einem Speicher- oder Katalogisierungsdatum innerhalb des angegebenen Datumsbereichs. Sie können ein einzelnes Datum oder einen Datumsbereich angeben. Gültiges Datumsformat: *YYYY-MM-DD* (*YYYY*=Jahr, *MM*=Monat, *DD*=Tag) Gültige Datumsbereiche: ■ Vorangestellte Zeichen (Beispiel: 2002*)	

	■ Startwert (Beispiel: 2002-05>) ■ Endwert (Beispiel: 2003-02<) Zulässige spezielle Datumsangaben:	
	TODAY (+/-*nnnn*)	Alle Objekte mit aktuellem Tagesdatum. Im Anschluss an den Tag kann +*nnnn* oder -*nnnn* angegeben werden. Der Wert *nnnn* darf maximal 4 Zeichen lang sein. Das resultierende Datum wird berechnet aus dem aktuellen Tagesdatum plus oder minus *nnnn* Tage. Kann mit der Startwert-Option (>) oder der Endwert-Option (<) kombiniert werden, z.B. listet TO-1> alle Objekte, die in den letzen zwei Tage gespeichert oder katalogisiert wurden.
	YESTERDAY	Alle Objekte mit dem Datum des Tages vor dem aktuellen Tag.
	MONTH	Alle Objekte, deren Datum im Bereich des aktuellen Monats liegt.
	YEAR	Alle Objekte, deren Datum im Bereich des aktuellen Jahres liegt.
time-range	Listet alle Objekte mit einem Speicher- oder Katalogisierungszeitpunkt, der im angegebenen Zeitbereich liegt. Geben Sie einen einzelnen Zeitwert oder einen Zeitbereich an. Gültiges Zeitformat: *HH:II:SS* (*HH* = Stunden, *II* = Minuten, *SS* = Sekunden) Gültige Zeitbereiche: ■ Vorangestellte Zeichen (Beispiel: 10:*) ■ Startwert (Beispiel: 10:30>) ■ Endwert (Beispiel: 11:20<)	
code-page-range	Sie können eine einzelne Codepage oder einen Bereich von Codepages angeben, siehe *range-notation*.	
arch-level-range	Gilt nur, wenn der Natural Optimizer Compiler installiert ist und wenn die Option ARCH gesetzt ist, um die IBM-Architekturebenen zu liefern (siehe *Natural Optimizer Compiler*-Dokumentation). Listet alle Natural-Objekte auf, die unter dem Natural Optimizer Compiler bei gesetzter Option ARCH katalogisiert wurden. Sie können, so wie unten aufgeführt, eine einzelne Ebene oder einen Bereich von Ebenen angeben. Dabei ist *n* eine numerische Zahl.	
	*n**	Listet alle Objekte mit Ebenen gleich *n* auf.

	n>	Listet alle Objekte mit Ebenen auf, die gleich oder größer als *n* sind.
	n<	Listet alle Objekte mit Ebenen auf, die gleich oder kleiner als *n* sind.
	*n**	Listet alle Objekte mit Ebenen von *n*0 bis *n*9 auf.
	nn	Listet alle Objekte mit Ebenen auf, die gleich *nn* sind.
	nn>	Listet alle Objekte mit Ebenen auf, die gleich oder größer als *nn* sind.
	nn<	Listet alle Objekte mit Ebenen auf, die gleich oder kleiner als *nn* sind.
noc-size-range	Gilt nur, wenn der Natural Optimizer Compiler installiert ist und NOC-Code generiert wurde (siehe *Natural Optimizer Compiler*). Listet alle Objekte auf, die mit dem Natural Optimizer Compiler katalogisiert wurden. Die NOC-Größe wird in Bytes angezeigt. Sie können eine einzelne Größe oder einen Bereich von Größen angeben (s.u.). Dabei ist *nnn* eine numerische Zahl mit maximal 7 Stellen.	
	nnn	Listet alle Objekte mit NOC-Größe gleich *nnn* auf.
	nnn>	Listet alle Objekte mit NOC-Größe gleich oder größer als *nnn* auf.
	nnn<	Listet alle Objekte mit NOC-Größe gleich oder kleiner als *nnn* auf.

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)



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.

Quellcode eines einzelnen Objekts anzeigen

LIST *object-name* [*options*] LIST *object-type object-name* [*options*]	In beiden Fällen wird der Quellcode des angegebenen Objekts gelistet. Wenn Sie einen einzelnen *object-name* mit dem LIST-Kommando eingeben, brauchen Sie keinen *object-type* anzugeben; es wird der Quellcode 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 [*range-clause*] [*settings*]	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. Die aufgelisteten Angaben zu Anzahl und Größe beziehen sich auf alle Objekte, die als Quellcode oder nur als katalogisierte Objekte gespeichert worden sind, sowie auf alle Objekte, zu denen gemäß den Auswahlkriterien in der *range-clause* sowohl gespeicherte als auch katalogisierte Objekte existieren. Siehe auch *Beispiele zur Benutzung der Objekt-Liste*.
LIST COUNT *object-name-range* [*range-clause*] [*settings*]	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 *object-name-range* und *range-clause*. Objektyp unknown (unbekannt) bedeutet, dass für diese Objekte keine Verzeichnisinformationen existieren.

	Siehe auch *Beispiele zur Benutzung der Objekt-Liste*.

Quellcode mehrerer Objekte nacheinander anzeigen

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

Liste von Objekten anzeigen

`LIST object-name-range` `LIST object-type object-name-range`	In beiden Fällen müssen Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den *object-name-range* 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 *Eine Funktion auf einem Objekt ausführen*).

Vorsortierte Liste von ausgewählten Objekten anzeigen

`LIST object-name-range` `LIST object-name-range range-clause`	In beiden Fällen müssen Sie Stern-Notation (*) und/oder Wildcard-Notation (?) für den *object-name-range* 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 *Eine Funktion auf einem Objekt ausführen*). Mit der *range-clause* können Sie zusätzliche Auswahl- und Sortierkriterien angeben, siehe auch das **Beispiel** weiter unten.

Langnamen katalogisierter Subroutinen und Klassen anzeigen

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

NOC-Optionen katalogisierter Objekte anzeigen

`LIST NOCOPT` [*object-type*] *object-name-range*	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 *object-name-range* weiter oben.

Compiler-Optionen katalogisierter Objekte anzeigen

`LIST OPTIONS` [*object-type*] *object-name-range*	Listet die katalogisierten Objekte auf und zeigt die beim `CATALOG` verwendeten Compiler-Optionen. Bezüglich der Namensoptionen siehe auch *object-name-range* weiter oben. 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 Quellcode geändert wurden) angezeigt werden. Bezüglich der Bereichsfelder auf der Online-Maske siehe den entsprechenden Hilfe-Bildschirm.

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-Quellcode-Objekt des Typs `TEXT` geschrieben werden kann. Wenn keine zusätzlichen Optionen 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
object-name	Der Name des katalogisierten Natural-Objekts.
REPORT	Füllwort zur verbesserten Lesbarkeit des Kommandos; kann weggelassen werden.
UNVARIABLES	Zum Report alle nicht benutzten Variablen hinzufügen; d.h., Variablen, die definiert, aber nicht nicht referenziert oder verändert werden.
USEDVARIABLES	Zum Report alle benutzten Variablen hinzufügen; d.h., Variablen, die referenziert oder verändert werden.
CONTSUBROUTINES	Zum Report die enthaltenen externen Subroutinen hinzufügen (nur bei Natural-Objekten des Typs SUBROUTINE).
EXTSUBROUTINES	Zum Report die benutzten externen Subroutinen hinzufügen.
STATPROFILE	Erstellt einen Report, der zusammenhängende Abfolgen von Statements, gruppiert nach Kategorien, in einem zur Optimierung geeigneten Quellcode-Programm anzeigt oder das für ein optimiertes Programm generierte NOC Coding auflistet; siehe auch *NOCSTAT Command , Code Profile*, in der *Natural Optimizer Compiler*-Dokumentation).
STATCATEGORY	Erstellt einen Report, der verschiedene Statement-Kategorien mit der entsprechenden Anzahl Ausprägungen auflistet; siehe auch *NOCSTAT Command , Statement Category*, in der *Natural Optimizer Compiler*-Dokumentation).
STATTYPE	Erstellt einen Report, der einzelne Statements mit der entsprechenden Anzahl Ausprägungen auflistet; siehe auch *NOCSTAT Command , Statement-Typ*, in der *Natural Optimizer Compiler*-Dokumentation).
DESTINATION	Füllwort zur verbesserten Lesbarkeit des Kommandos; kann weggelassen werden.
SCREEN	Anzeige des Reports am Bildschirm. Dies ist der Standardwert.
WORKFILE	Den Report in Workfile 1 schreiben.
TEXT	Den Report in ein Natural-Quellcode-Objekt des Typs Text schreiben.

Syntax-Element	Beschreibung
text-name	Name des Natural-Quellcode-Objekts des Typs Text. Der Standardwert ist REPCPOUT.
LIBRARY *library-name*	Library-Name für das Natural-Quellcode-Objekt des Typs Text. Die Standard-Library ist die aktuelle Library.

Siehe auch *Beispiele zur Benutzung der Objekt-Liste*.

Directory-Informationen anzeigen

LIST DIRECTORY	Zeigt die folgenden Directory-Informationen zum Quellcode und zum Objektcode des gerade im Editor-Arbeitsbereich befindlichen Objekts an: 1. **Quellcode:** ■ 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), Quellcode-Größe. 2. **Objektcode:** ■ Datum und -Uhrzeit, wann das Objekt katalogisiert worden ist, ■ Library-Name, ■ Benutzerkennung (User-ID), ■ DBID/FNR der Library, ■ Benutzerkennung (User-ID), ■ Programmiermodus (Reporting oder Structured Mode), ■ TP-System, ■ Terminalkennung (Terminal-ID), Transaktionscode, ■ Natural-Version,

	■ Codepage-Informationen (falls verfügbar), ■ Betriebssystem, ■ Größe im DATSIZE-Puffer, ■ Größe im Buffer Pool, ■ Größe der globalen Daten, Größe des NOC-Code (vom Natural Optimizer Compiler erzeugter Maschinencode), ■ Name der vom Objekt verwendeten GDA, ■ Datum und Uhrzeit, wann die benutzte GDA katalogisiert worden ist, ■ abhängig von der Verwendung der Kopilierungsoption GDASC beim Katalogisieren: - Vergleich der GDA-Signaturen des Objekts und der verwendeten GDA (z.B. identisch, nicht identisch) oder - GDA-Zeitstempel mit Datum und Uhrzeit, wann die GDA katalogisiert worden ist und wann das Objekt katalogisiert worden ist, ■ Anzahl der von dem Objekt benutzen Subroutinen, ■ beim Katalogisieren des Objekts mit dem Natural Optimizer Compiler (NOC) benutzte ARCH Level und die dabei gesetzte ARCH-Option, ■ Compiler-Optionen (ursprünglich, endgültig, geändert) zum Zeitpunkt der Katalogisierung des Objekts, ■ Langname der katalogisierten Subroutine, Function oder Class, ■ Liste der vom Objekt verwendeten externen Subroutinen (Langname und Name des katalogisierten Objekts).
LIST DIRECTORY *object-name*	Dieses Kommando zeigt die Verzeichnisinformationen zu dem betreffenden Objekt an (wie bei LIST DIRECTORY beschrieben). Wenn Sie Stern-Notation (*) oder Wildcard-Notation (?) für *object-name-range* verwenden, werden nacheinander die Directory-Informationen der entsprechenden Objekte angezeigt.
LIST DIRECTORY *object-name-range*	
LIST *object-name* WITH DIRECTORY	Dieses Kommando zeigt zunächst die Verzeichnisinformationen zu dem betreffenden Objekt an (wie bei LIST DIRECTORY beschrieben) und listet anschließend den Quellcode des Objekts.

DDMs (Views) anzeigen

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



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

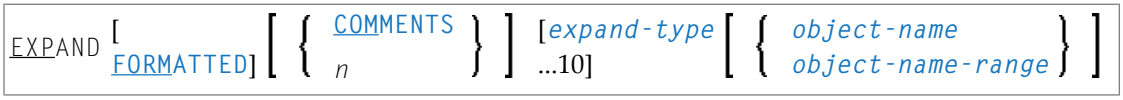
Optionen

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

```
{ [[WITH] DIRECTORY] [NUMBERS OFF] [expand-option]
  formatted-option
  CONVERTED
  UNCONVERTED                                      }
```

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

expand-option



EXPAND *object-name* EXPAND *object-name-range*	Die EXPAND-Option erlaubt es Ihnen, die Quellcodes von Objekten, die in dem gelisteten Quellcode referenziert werden (Copycodes, Data Areas, Maps, Helproutinen, externe Subroutinen, Subprogramme, mit einem FETCH-Statement aufgerufene Programme, Fehlermeldungen), *innerhalb* des gelisteten Quellcodes anzuzeigen. Dies kann besonders im Batch-Betrieb sehr hilfreich sein. Wenn Sie beispielsweise ein Quellcode-Programm listen, das ein INCLUDE-Statement enthält, können Sie den Quellcode des betreffenden Copycodes innerhalb des gelisteten Quellcode-Programms unmittelbar nach dem INCLUDE-Statement anzeigen. In den folgenden Erklärungen werden Objekte, die innerhalb eines Quellcodes gelistet werden, als „Expand-Objekte“ bezeichnet. **Unterkommandos zur EXPAND-Option** Innerhalb eines gelisteten Expand-Objekts sind nur die folgenden Unterkommandos verfügbar (die oben beschrieben sind): PRINT + - - . Siehe ***Beispiele zur Benutzung der Objekt-Liste***.
EXPAND FORMATTED	Die Option EXPAND FORMATTED ist nur für innerhalb eines Quellcodes gelistete Data Areas (mit STOW gespeichert, d.h. bei denen der Zeitstempel des Quellcode-Objekts und des katalogisierten Objekts identisch ist) und Maps relevant. Für Data Areas gilt: ■ Wenn Sie FORMATTED nicht angeben, ähnelt die Anzeige der Data Area der im Data-Area-Editor. ■ 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 Quellcode-Objekts und des katalogisierten Objekts identisch ist), siehe auch Unterkommando FORMAT. Für Maps gilt: ■ Wenn Sie FORMATTED nicht angeben, wird der *Quellcode* der Map gelistet. ■ Wenn Sie FORMATTED angeben, wird das *Layout* der Map gelistet (d.h. die Map, wie Sie ein Benutzer zur Laufzeit sieht).

<table>
<tr><td>EXPAND COMMENTS
EXPAND n</td><td>Wenn Sie die Option EXPAND COMMENTS verwenden, werden nur die Kommentarzeilen am Anfang des Expand-Objekts gelistet; d.h. das Expand-Objekt wird bis zu (aber nicht einschließlich) der ersten Quellcode-Zeile, die keine Kommentarzeile ist, gelistet.

Wenn Sie die Option EXPAND n verwenden, werden nur die ersten n Zeilen des Expand-Objekts gelistet.

Wenn Sie keine dieser beiden Optionen verwenden, wird das Expand-Objekt vollständig gelistet.</td></tr>
<tr><td>expand-type</td><td>Als expand-type geben Sie den Objekttyp des Expand-Objekts an. Sie können die folgenden expand-types angeben:
<table>
<tr><td>P</td><td>Programme</td><td rowspan="12">Falls Sie mehr als einen expand-type angeben möchten, geben Sie sie in beliebiger Reihenfolge und ohne Leerzeichen an; um Maps, Copycodes und Subroutinen innerhalb eines gelisteten Quellcodes zu listen, geben Sie beispielsweise MCS als expand-type an.</td></tr>
<tr><td>N</td><td>Subprogramme</td></tr>
<tr><td>S</td><td>Externen Subroutinen</td></tr>
<tr><td>H</td><td>Helproutinen</td></tr>
<tr><td>G</td><td>Global Data Areas</td></tr>
<tr><td>L</td><td>Local Data Areas</td></tr>
<tr><td>A</td><td>Parameter Data Areas</td></tr>
<tr><td>M</td><td>Maps</td></tr>
<tr><td>C</td><td>Copycodes</td></tr>
<tr><td>E</td><td>Fehlermeldungen</td></tr>
<tr><td>4</td><td>Klassen</td></tr>
<tr><td>*</td><td>Alle Objekttypen</td></tr>
</table></td></tr>
<tr><td>object-name
object-name-range</td><td>Als object-name oder object-name-range geben Sie den Namen bzw. die Namen der Expand-Objekte an, die innerhalb des gelisteten Quellcodes gelistet werden soll(en).

Für den object-name bzw. object-name-range eines Expand-Objekts stehen die gleichen Optionen zur Verfügung wie für den object-name bzw. object-name-range des gelisteten Hauptobjekts in der primären LIST-Kommandosyntax. Ausnahmen: die Größer-Zeichen-Notation (>) und die Kleiner-Zeichen-Notation.</td></tr>
</table>

formatted-option



FORMATTED-Option

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

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

Weitere FORMATTED-Optionen für Maps

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

['*c*'] ['*c*']	**Füllzeichen benutzen:** Sie können Füllzeichen *c* für Eingabefelder (AD=A und AD=M) und Ausgabefelder (AD=O) angeben, um diese Felder sichtbar zu machen. Sie können ein beliebiges Zeichen als Füllzeichen angeben. Das folgende Beispiel bewirkt, dass alle Eingabefelder mit einem Unterstrich (_) und alle Ausgabefelder mit einer Raute (#) gefüllt angezeigt werden. `LIST MAP map-name FORMATTED '_' '#'`

SETTINGS	**Map-Einstellungen:** Diese Option bewirkt, dass die **Map Settings** der Map angezeigt werden. `LIST MAP map-name FORMATTED SETTINGS`
FIELDS	**Feldübersicht:** Diese Option bewirkt, dass das **Field Summary**, d.h. die Liste der Map-Felder, angezeigt wird. `LIST MAP map-name FORMATTED FIELDS`
EXTFIELDS	**Erweiterte Feldeditier-Informationen:** Diese Option bewirkt, dass die **Extended Field Editing**-Informationen für alle Map-Felder angezeigt werden. `LIST MAP map-nameFORMATTED 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	**Alle Verarbeitungsregeln anzeigen:** `LIST MAP map-name FORMATTED RULES` Zeigt *alle* Verarbeitungsregeln für die angegebene Map an.
INLINERULES	**Nur die Inline-Verarbeitungsregeln anzeigen:** `LIST MAP map-name FORMATTED INLINERULES` Zeigt nur die Inline-Verarbeitungsregeln für die angegebene Map an.
FREERULES	**Nur die freien Verarbeitungsregeln anzeigen:** `LIST MAP map-name FORMATTED FREERULES` Zeigt nur die freien Verarbeitungsregeln für die angegebene Map an.

AUTORULES	**Nur die automatischen Verarbeitungsregeln anzeigen:** `LIST MAP map-name FORMATTED AUTORULES` Zeigt nur die automatischen Verarbeitungsregeln für die angegebene Map an.

Siehe auch Unterkommandos LAYOUT und FORMAT im Abschnitt ***Quellcode-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 ***Eine Funktion auf einem Objekt ausführen***.
Name	Name des Objekts.
Type	Objekttyp.
S/C	Zeigt an, ob das Objekt als Quellcode (S) und/oder als katalogisiertes Objekt (C) vorhanden ist.
SM	Der Natural-Programmiermodus, der beim Erstellen des Objekts verwendet wurde. S = Structured Mode

Spalte	Bedeutung
	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 ein Quellcode- als auch ein Objektmodul existiert (wie in der Spalte **S/C** angezeigt), beziehen sich die angezeigten Informationen auf das Quellcode-, nicht das Objektmodul.

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

Um weitere Informationen zu Quellcode- 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 Quellcode-Zeilen bei Quellcode-Objekten aus Durchsatzgründen nicht berechnet. Wenn Sie wünschen, dass die die Anzahl der Quellcode-Zeilen bei Quellcode-Objekten 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 dem Quellcode 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 `LIST EXTENDED *`.	
`ALL fx`	Den Funktionscode *fx* für alle angezeigten Objekte eingeben. *fx* = gültiger Funktionscode für ein gelistetes Objekt.	
`SORT`	Sort-Fenster anzeigen (siehe ***Liste der Objekte sortieren***).	
`COUNTSOURCE`	`ON`	Anzahl der Quellcode-Zeilen für Quellcode-Objekte wird angezeigt.
	`OFF`	Anzahl der Quellcode-Zeilen für Quellcode-Objekte wird nicht angezeigt.
`MARK-LONG-LINES`	`ON`	Lange Zeilen in der Auflistung eines Quellcode-Objekts werden in den ersten zwei Stellen mit einem `L` markiert. Der Standardwert kann im List-Profil angegeben werden; siehe ***Individuelles List-Profil erstellen***.
	`OFF`	Lange Zeilen in der Auflistung eines Quellcode-Objekts werden nicht markiert.
`DEFINE-DATA`	`ON`	Der aufgelistete Data-Area-Quellcode wird standardmäßig im `DEFINE DATA`-Format angezeigt (wie bei `LIST dataarea FORMATTED`).

Code	Funktion	
		Der Standardwert kann im List-Profil angegeben werden; siehe ***Individuelles List-Profil erstellen***.
	OFF	Die Anzeige eines Data-Area-Quellcodes erfolgt unformatiert.
LISTCOUNT	Liste mit Anzahl der Objekte und Angaben zur Objektgröße entsprechend den zurzeit benutzten Auswahlkritierien.	
LISTPROFILE	Aktuelle Werte der Parameter des List-Profils anzeigen (siehe ***Individuelles List-Profil erstellen***).	
NOCOPT	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; entspricht LIST NOCOPT *, siehe ***NOC-Optionen katalogisierter Objekte anzeigen***.	
OPTIONS	Listet die katalogisierten Objekte auf und zeigt die beim CATALOG zu Beginn verwendeten Compiler-Optionen; entspricht LIST OPTIONS *, siehe ***Compiler-Optionen katalogisierter Objekte anzeigen***.	
REUSE	ON	Wiederverwendungsmodus einschalten. Die zuletzt angezeigte Liste wird erneut verwendet, nachdem in der Spalte **Cmd** Kommandos ausgeführt wurden. Ausgenommen sind folgende Kommandos: E ED (Edit) CA (Catalog) UC (Uncat) S ST (Stow) D DE (Delete) RE (Rename)
	OFF	Wiederverwendungsmodus ausschalten. Nach der Ausführung von Kommandos, die in der Spalte **Cmd** eingegeben wurden, wird die Liste neu erstellt.
REFRESH	Die gerade angezeigte Liste aktualisieren. Dieses Unterkommando können Sie insbesondere bei eingeschaltetem Wiederverwendungsmodus benutzen.	
+	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 Quellcodeform 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).
DE	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).
ED	Editieren des Quellcodes (entspricht dem Systemkommando EDIT).
EX	Ausführen des Objektmoduls (entspricht dem Systemkommando EXECUTE).
LC	Anzeigen des in die Default-Codepage *CODEPAGE umgesetzten Quellcodes 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 Quellcodes 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).
LI	Anzeigen (List) des Quellcodes.
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 ***Compiler Output zu einem katalogisierten Objekt anzeigen***.
LU	Anzeigen des Quellcodes 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
OP	Zeigt die zu Beginn, die am Ende verwendeten und die geänderten Compiler-Optionen, die beim Katalogisieren mit CATALOG 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 Quellcodes.
RE	Umbenennen des Objekts (entspricht dem Systemkommando RENAME).
RU	Kompilieren und Ausführen des Quellcodes (entspricht dem Systemkommando RUN).
ST	Speichern in Quellcode- und Objektform (entspricht dem Systemkommando STOW).
UC	Löschen des Objektmoduls (entspricht dem Systemkommando UNCATALOG).
.	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.

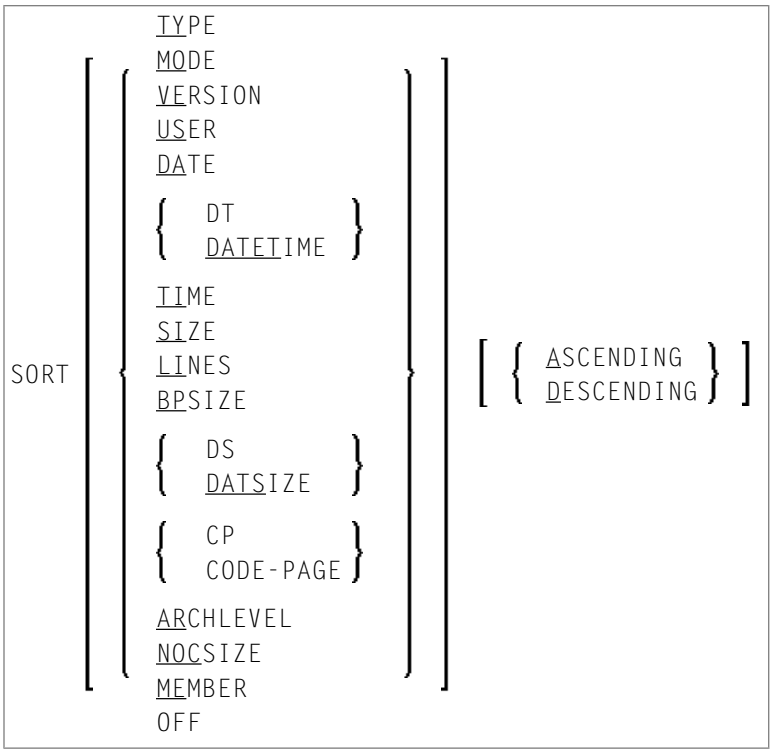
Um die Sort-Funktion aufzurufen

- 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 sotieren 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
ARCH level	ARCHLEVEL
NOC size	NOCSIZE
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

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

`LIST COUNT`	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.

Quellcode-Liste

- Unterkommandos für einen angezeigten Quellcode
- Unterkommandos FIND, REF und SCAN
- Unterkommando FORMAT
- Cursor-sensitive Objektauswahl

Unterkommandos für einen angezeigten Quellcode

Wenn Ihnen der Quellcode 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 Quellcode-Library nicht in der Kopfzeile des aufgelisteten Quellcodes angezeigt. Wenn Sie wollen, dass die `DBID` und `FNR` der Quellcode-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 des Quellcodes.
`BOTTOM`	
`--`	Blättert an den Anfang des Quellcodes.
`TOP`	
`+`*n*	Blättert *n* Zeilen vor.
`-`*n*	Blättert *n* Zeilen zurück.
nnnn	Blättert zu Zeile Nummer *nnnn*.
`UNCONVERTED`	Siehe `UNCONVERTED` bei *Optionen*.

Unterkommando	Funktion
CONVERTED	Siehe CONVERTED bei *Optionen*.
DBFNR ON	Zeigt die DBID und FNR der Quellcode-Library in der Kopfzeile des Quellcodes.
DBFNR OFF	Zeigt die Kopfzeile des Quellcodes ohne die DBID und FNR der Quellcode-Library. Dies ist der Standardwert.
EXPAND	Siehe *expand-option*.
FIELDS	Gilt nur für Maps: Zeigt das **Field Summary**, d.h. die Liste der Felder in der Maske.
FIND	Zeigt nur die Quellcode-Zeilen, in denen der angegebene Wert (*value*) vorkommt. Weitere Informationen siehe ***Unterkommandos FIND, REF und SCAN***.
FIND *value*	
FIND ABSOLUTE *value*	
FIND ABSOLUTE *value*	
FIND CASESENSITIVE *value*	
FORMAT	Gilt nur für Data Areas und Maps: zeigt die ***formatierte*** 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 den Quellcode mit Quellcode-Zeilennummern. Dies ist der Standardwert.
NUMBERS OFF	Zeigt den Quellcode ohne Quellcode-Zeilennummern.
PRINT	Druckt den Quellcode.
REF	Zeigt die Zeilennummer der Quellcode-Zeilen, in denen der angegebene Wert (*value*) vorkommt. 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. Wenn Sie REF ohne ABSOLUTE eingeben, dann wird der Wert wird nur gefunden, wenn er als isoliertes Wort vorkommt. Dies ist der Standardwert. 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.
REF *value*	
REF ABSOLUTE *value*	
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).

Unterkommando	Funktion
SCAN	Zeigt alle Zeilen intensiviert an, in denen der angegebene Wert (*value*) vorkommt. Die erste Zeile, in der der Wert vorkommt, wird an den Anfang der Seite geblättert. 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. Wenn Sie SCAN ohne ABSOLUTE eingeben, dann wird der Wert nur gefunden, wenn er als isoliertes Wort vorkommt. Dies ist der Standardwert. 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.
SCAN *value*	
SCAN ABSOLUTE *value*	
SCAN= oder 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 [*expand-type*...10] *object-name*	Die Angabe eines einzelnen Namens (*object-name*) mit dem ZOOM-Kommando hat die gleiche Auswirkung wie das Markieren des Namens mit dem Cursor in dem gelisteten Quellcode (siehe ***Cursor-sensitive Objektauswahl***). Das ausgewählte Objekt wird in einem Fenster angezeigt. Wenn Sie Stern-Notation (*) bzw. Wildcard-Notation (?) im *object-name* verwenden, werden alle ausgewählten Objekte in dem Fenster gezeigt, und zwar in der Reihenfolge, in der sie in dem gelisteten Quellcode referenziert werden. Die Angabe eines *expand-type* entspricht der *expand-option*. Für ein Objekt, das in einem über ZOOM aufgerufenen Fenster gezeigt wird, stehen die gleichen Unterkommandos zur Verfügung wie für einen normal gelisteten Quellcode (außer PRINT, EXPAND und ZOOM). Wenn aufgrund von Stern- oder Wildcard-Notation mehrere Objekte angezeigt werden, können Sie darüber hinaus mit den Kommandos NEXT und PREV (bzw. PF4 und PF5) von einem Objekt im Fenster zum nächsten bzw. zum vorigen blättern.
ZOOM [*expand-type*...10] *object-name-range*	
.	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*) angegeben 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 Quellcode-Objekts 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 Quellcode 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=O`) 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 eines normal mit `LIST` angezeigten Quellcodes (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

`S>`*n*	Verschieben des Map Layouts um *n* Spalten nach rechts.
`S<`*n*	Verschieben des Map Layouts um *n* Spalten nach links.

Zusätzliche Unterkommandos für Field Summary List

EXTEND	Zeigt erweiterte Feld-Editierinformationen zu allen Map-Feldern. Um die erweiterten Feld-Editierinformationen zu einem einzelnen angezeigten Feld anzuzeigen, markieren Sie den Namen des Feldes in der Field Summary List mit dem Cursor und drücken Sie ENTER.
RULES *nn*	Zeigt die Verarbeitungsregeln, die mit dem Feld *nn* verknüpft sind (wobei *nn* für die fortlaufende Zeilennummer in der ersten Spalte der **Field Summary List** steht). Um die Verarbeitungsregeln zu einem einzelnen angezeigten Feld anzuzeigen, geben Sie ein R in der Kommandozeile ein, markieren Sie dann den Namen des Feldes in der Field Summary List mit dem Cursor und drücken Sie ENTER.
SCAN [ABSOLUTE] *value*	Wie bei ***Unterkommandos für einen angezeigten Quellcode***.
SCAN =	

Zusätzliche Unterkommandos für Processing Rules

SCAN [ABSOLUTE] *value*	Wie bei ***Unterkommandos für einen angezeigten Quellcode***.
SCAN =	

Cursor-sensitive Objektauswahl

Innerhalb eines gelisteten Quellcodes können Sie mit dem Cursor den Namen eines Objektes, das in diesem Quellcode referenziert wird, markieren. Der Quellcode des so ausgewählten Objektes wird dann in einem Fenster gezeigt.

Für den in dem Fenster gezeigte Quellcode stehen Ihnen dieselben Unterkommandos - außer PRINT, EXPAND und ZOOM - zur Verfügung wie für einen „normal“ gelisteten Quellcode.

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-Profils 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`	`N` oder `Y`	`N` Die Standardeinstellung ist „No“. `Y` Die zuletzt angezeigte Objektliste wird nach der Ausführung von solchen Kommandos wiederverwendet, welche in der **Cmd**-Spalte eingeben 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`	`N` 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

Parameter	Mögliche Werte / Format	Erläuterung
		Suchwert. Siehe auch ***Unterkommandos FIND, REF und SCAN***.
COUNT-SOURCE-LINES	N oder Y	N Die Standardeinstellung ist „No“. Y Anzeige der Anzahl an Quellcode-Zeilen für Quellcode-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.
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 Y	N Der für die Sort-Funktion benutzte Natural-Quellcode des Typs Text soll *nicht* gelöscht werden, wenn das LIST-Kommando beendet wird. Y Der für die Sort-Funktion benutzte Natural-Quellcode 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	N oder Y	Y Lange Zeilen in einem angezeigten Quellcode werden am Beginn der Quellcode-Zeile mit einem L gekennzeichnet. N Lange Zeilen in einem angezeigten Quellcode werden nicht gekennzeichnet. Dies ist die Standardeinstellung.

Parameter	Mögliche Werte / Format	Erläuterung
DATA-AREA-DEFINE-DATA-FORMAT	N oder Y	Dient zur Angabe, ob Data-Area-Quellcodes im DEFINE DATA-Format oder unformatiert angezeigt werden sollen. Y Standardmäßig wird ein Data-Area-Quellcode im DEFINE DATA-Format angezeigt (identisch mit dem Kommando LIST FORMATTED). N Ein aufgelisteter Data-Area-Quellcode wird unformatiert angezeigt. Dies ist die Standardeinstellung.
SOURCE-LIST-WITH-DBID-FNR	N oder Y	Dient zur Angabe, ob die Datenbankkennung (DBID) und die Dateinummer (FNR) der Quellcode-Library in der Kopfzeile angezeigt werden sollen N Die Standardeinstellung ist „No“. Y Die Datenbankkennung (DBID) und die Dateinummer (FNR) der Quellcode-Library werden in der Kopfzeile angezeigt.
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 OUTPUT-DESTINATION-BATCH=T ist. Falls kein Name angegeben ist, wird L_*uuuuuu* verwendet; dabei ist sind *uuuuuu* die ersten 6 Bytes der Benutzerkennung.

Parameter	Mögliche Werte / Format	Erläuterung
OUTPUT-TEXT-LIBRARY	A08	Name der Natural Library für das Natural Text Member, das zum Speichern der LIST-Ausgabe benutzt wird, wenn OUTPUT-DESTINATION-BATCH=T ist. Falls kein Name angegeben ist, wird die Library WORKPLAN benutzt.
SHOW-CAT-TIMESTAMP-VALUES	N oder Y	Dient zur Angabe von Datum und Uhrzeit der Katalogisierung, die in der Objektauswahlliste und in einem Objektverzeichnis (LIST DIRECTORY) angezeigt werden: N Verwendung der Datums- und Uhrzeitwerte des Verzeichnisses des katalogisierten Objekts, die um die Werte korrigiert sind, die mit den Natural-Profilparametern DD und TD angegeben sind (siehe *Parameter-Referenz*-Dokumentation). Dies ist die Standardeinstellung. Y Verwendung des Zeitstempels (Store Clock-Wert) des katalogisierten Objekts.
SHOW-CAT-DBID-FNR-IN-LIST-DIR	N oder Y	Dient zur Angabe, ob in der Liste der katalogisierten Objekte die Datenbankkennung (DBID) und die Dateinummer (FNR) der Library angezeigt werden, in der das Objekt katalogisiert wurde (LIST DIRECTORY): N DBID und FNR der Library, in der das Objekt katalogisiert wurde, nicht anzeigen. Y DBID und FNR der Library, in der das Objekt katalogisiert wurde, anzeigen. Dies ist die Standardeinstellung.
TOTAL-LINES-DISPLAY-BATCH	N oder Y	Dient zur Angabe, ob im Stapelverarbeitungsmodus die Gesamtzahl der Quellcodezeilen aller in der Liste der Objekte angezeigten Quellcode-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.
`LIST-SHIFT-INCREMENT`	N3	Dient zur Angabe der Schrittweite, die das Kommando `LIST` beim Blättern mit PF10 bzw. PF11 nach rechts oder links in der Ergebnisliste verwendet.

26 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.

27 LISTDBRM

Anmerkungen:

1. Dieses Kommando steht bei Natural for DB2 zur Verfügung.
2. Bei Natural for DB2 muss das Kommando `LISTDBRM` aus der Natural System Library `SYSDB2` abgesetzt werden, d.h. zuerst ist eine Anmeldung bei der Library `SYSDB2` erforderlich und danach kann das Kommando `LISTDBRM` abgesetzt werden.

Das Kommando `LISTDBRM` dient zur Anzeige vorhandener Database Request Modules (DBRM) mit Natural-Programmen, die ein gegebenes DBRM referenzieren.

Es 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
```

Folgende Funktionen stehen zur Verfügung:

Code	Beschreibung
D	Anzeige von Programmen, die Zugriff auf DB2 (Natural for DB2) 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.

Parameter	Beschreibung
	Falls das Feld **Member** leer gelassen wird oder falls nur ein Stern (*) eingegeben wird, werden alle Member in der angegebenen Library aufgelistet.
DBRM	Ein gültiger Package-Name (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
```

28 LISTPROF

`LISTPROF`

Anmerkungen:

1. Dieses Kommando ist nur mit Natural for Db2 verfügbar.
2. `LISTPROF` muss aus der Natural-Systembibliothek `SYSDB2` heraus ausgeführt werden, d.h. Sie müssen sich zuerst bei `SYSDB2` anmelden und dann das Kommando `LISTPROF` eingeben.

Mit dem `LISTPROF` können Sie vorhandene SQLJ-Profile von Natural-Programmen oder Natural-Programmen, die ein bestimmtes SQLJ-Profil referenzieren, anzeigen.

Wenn das Kommando benutzt wird, wird das folgende Menü angezeigt:

```
 14:29:18               ***** NATURAL TOOLS FOR SQL *****            2023-02-16
                               - List SQLJPROF -




                     Code Function

                      D   Display SQLJ Profiles of Programs                    ↩

                      R   List Programs Referencing SQLJ Profiles              ↩

                      ?   Help
                      .   Exit

               Code .. _   Library .. _________
                           Member ... _________
                           SQLJPROF . _________




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

Die folgenden Funktionen sind verfügbar:

Code	Beschreibung
D	Zeigt die Programme an, die auf Db2 Zugriff haben und ihr entsprechendes SQLJ-Profil. Wenn kein SQLJ-Profilname angezeigt wird, verwendet das entsprechende Programm dynamisches SQL.
R	Listet alle Programme auf, die ein bestimmtes SQLJ-Profil verwenden. Wenn kein SQLJ-Profilname angegeben ist, werden alle Programme aufgelistet, die dynamisches SQL verwenden.

Es gelten die folgenden Parameter:

Parameter	Beschreibung
Library	Gibt den Namen einer Natural-Bibliothek an Bibliotheksnamen, die mit `SYS` beginnen, sind nicht zulässig. Dieser Parameter muss angegeben werden.

Parameter	Beschreibung
Member	Gibt den Namen des anzuzeigenden Natural-Programms (Member) an. Dieser Parameter ist optional und kann zur Einschränkung der Ausgabe verwendet werden. Wird ein Wert gefolgt von einem Stern (*) angegeben, werden alle Member in der angegebenen Bibliothek aufgelistet, deren Namen mit diesem Wert beginnen. Bleibt das Feld **Member** leer oder wird nur ein Stern angegeben, werden alle Member in der angegebenen Bibliothek aufgelistet.
SQLJPROF	Gibt einen gültigen SQLJ-Profilnamen an. Bleibt das Feld leer, wird auf Programme Bezug genommen, die dynamisch ausgeführt werden. Dieser Parameter gilt nur für den Funktionscode `R`.

Beispiel einer Ergebnisanzeige:

```
16:16:22              ***** NATURAL TOOLS FOR SQL *****           2023-02-16
                                          - List SQLJPROF -              ↩


 Library  Name      Type       SQLJPROF  User ID      Date       Time
 -------- --------  ---------   --------  --------   --------   --------
 HGK      P017523   Program                HGK      2011-02-23 16:16:25
 HGK      P017622   Program    P017622     HGK      2023-02-02 17:51:32
 HGK      P017622D  Program    P017622     HGK      2022-12-13 16:08:16
 HGK      P017622F  Program    P017622     HGK      2022-12-13 16:08:16
 HGK      P017622R  Program    P017622     HGK      2022-12-13 16:08:16
 HGK      P017622U  Program    P017622     HGK      2022-12-13 16:08:16
 HGK      P017948C  Program                HGK      2011-05-31 10:41:06
 HGK      P017948D  Program                HGK      2011-05-27 16:07:05
 HGK      P017948X  Program                HGK      2011-05-31 09:45:54
```

29 LISTSQL

Dieses Kommando ist bei Natural for DB2 verfügbar.



Das Kommando LISTSQL listet diejenigen Natural-Statements im Quellcode 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
object-name <sa>	Wenn Sie einen gültigen Objektnamen angeben, muss das anzuzeigende Objekt in der Library gespeichert sein, in die Sie zurzeit eingeloggt sind. Wenn Sie keinen Objektnamen angeben oder wenn Sie <sa> (source area) angeben, bezieht sich LISTSQL auf das Objekt, das sich zurzeit im Natural-Quellcode-Bereich befindet. In jedem Fall benötigt LISTSQL ein katalogisiertes oder ein mit dem STOW-Kommando kompiliertes und gespeichertes Objekt.
ALL	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. Das Schlüsselwort ALL können Sie sowohl im Online- als auch im Batch-Modus angeben. Das Ausgabeformat ist in beiden Fällen gleich. Die Funktionen **Error** (PF2), **Explain** (PF4) bzw. **Parms** (PF6) stehen nicht zur Verfügung. Wenn Sie das Schlüsselwort ALL angeben, können Sie ein Fragezeichen (?) oder einen Stern (*) als Stellvertretersymbol benutze, zum Beispiel: LISTSQL PGM* ALL. Die Sonderzeichen > 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 *****            2024-10-05
 Member DEM2SEL                    - LISTSQL -                   Library SYSDB243
 SQL Builder Version 4.30
 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
                           NDZ  : static       Prof : DEM2PROF
                                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
```

Wenn ein statisches Database Request Module (DBRM) generiert wurde, dann wird sein Name im Feld **DBRM** des Bildschirms **LISTSQL** angezeigt. Andernfalls bleibt dieses Feld leer.

Wenn eine statische NDZ-Generierung durchgeführt wurde, wird `static` im Feld **NDZ** angezeigt und der SQLJ-Profilname wird nach dem Literal **Prof** angezeigt.

Wenn keine statische NDZ-Generierung durchgeführt wurde, werden die Felder **NDZ** und **Prof** nicht angezeigt.

Die folgenden spezifischen PF-Funktionstasten stehen zur Verfügung:

PF-Taste (Bezeichnung):	Funktion:
PF2 (Error)	Diese Taste dient zur Ausführung des Kommandos `SQLERR`. Wenn bei der `EXPLAIN`-Funktion ein Fehler auftritt, können Sie diese Taste drücken, um Informationen zu DB2-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 **Queryno for EXPLAIN**) ist standardmäßig auf `1` gesetzt und kann überschrieben werden.

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 *****            2024-10-05
 Member DEM2SEL                      LISTSQL                     Library SYSDB243
        NDZ  : static       Prof : DEM2PROF  Sequence : 00008 Sequence2 : 00000  ↩

        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-Konsistenzmerkmal (Contoken) und einige intern verwendete statische Parameter).

Wenn eine statische NDZ-Generierung durchgeführt wurde, wird `static` im Feld **NDZ** angezeigt. In den Feldern **Sequence** und **Sequence2** wird die SQLJ-Sequenznummer des Statements und die Sequenznummer des sekundären Statements angezeigt, wenn die Option **File Server** für die statische Generierung angegeben wurde.

Wenn für das Programm keine statische NDZ-Generierung durchgeführt wurde, erscheinen die Felder **NDZ**, **Prof** und **Sequence** nicht auf dem Bildschirm

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. an das Ende (++) der Liste.
PF8, PF7 (-)	Vorwärtsblättern (+) bzw. Zurückblättern (-).

Verwendung des Kommandos EXPLAIN

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 *****            2024-10-05
 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-                              Tslock        -- sortn --  -- sortc --
 TabNo Creator.Name                        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 (+) rückwärts 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.

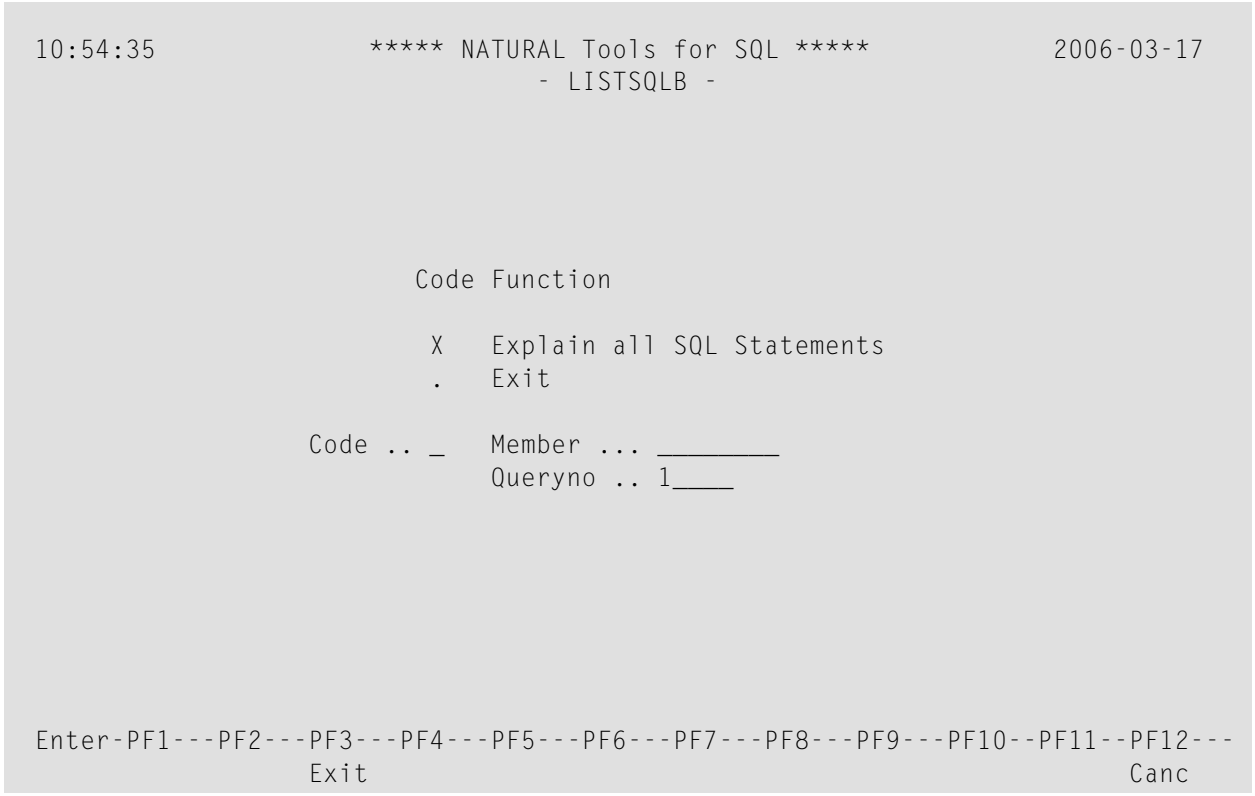
30 LISTSQLB

Anmerkungen:

1. Diese Kommando steht 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:



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 eine Wert, gefolgt von einem Stern (*), eingegeben, 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 angegeben 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.

31 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 Quellcode-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.

32 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 Quellcode- und/oder Objektform erstellten Objekte gespeichert (es sei denn, Sie geben bei einem SAVE-, CATALOG- oder STOW-Kommando ausdrücklich eine andere Library an).

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

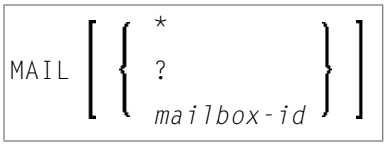
Das Kommando LOGON bewirkt, dass alle Natural Global Data Areas und anwendungsunabhängigen Variablen (AIVs), alle mit einem SET KEY-Statement vorgenommenen Zuordnungen und alle gehalten ISN-Listen freigegeben werde. Im DDM-Buffer-Bereich enthaltene DDMs werden ebenfalls freigegeben.

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

<table>
<tr><td>LOGON library-id</td><td colspan="2">Eine Library-ID darf 1 bis 8 Zeichen lang sein und keine Leerzeichen enthalten. Sie darf folgende Zeichen enthalten:<table><tr><td>A - Z</td><td>Großbuchstaben</td></tr><tr><td>0 - 9</td><td>Ziffern</td></tr><tr><td>-</td><td>Bindestrich</td></tr><tr><td>_</td><td>Unterstrich</td></tr></table>Das erste Zeichen einer Library-ID muss ein Großbuchstabe sein.</td></tr>
<tr><td>LOGON library-id password</td><td colspan="2">Das Adabas-Passwort; siehe Session Parameters im Abschnitt Library Maintenance der Natural Security-Dokumentation.</td></tr>
</table>

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.

33 MAIL



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

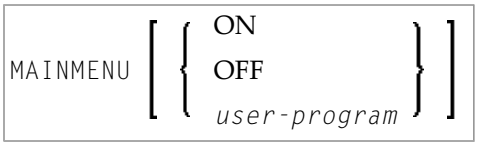
Mit dem Systemkommando `MAIL` können Sie eine Mailbox aufrufen, um den Inhalt sowie das Ablaufdatum zu ändern.

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

`MAIL`	Wenn Sie das `MAIL`-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` *`mailbox-id`*	Wenn Sie eine *`mailbox-id`* angeben (bis zu 8 Zeichen lang), wird die entsprechende Mailbox direkt aufgerufen. Die *`mailbox-id`*, die Sie angeben, muss in Natural Security definiert sein.

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

34 MAINMENU



Mit dem Systemkommando MAINMENU können Sie die Anzeige des Natural-Hauptmenüs ein- bzw. ausschalten oder ein Programm aufrufen, das einen eigenen Menüschirm 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 *user-program*	Statt des Natural-Hauptmenüs wird ein benutzerdefiniertes Programm aufgerufen, das seinerseits ein benutzerdefiniertes Menü aufruft.

Siehe auch Natural-Profilparameter MENU.

35 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 *Mainframe-Produktlizenzierung*-Dokumentation.

36 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 *Optimizer Compiler aktivieren* in der *Natural Optimizer Compiler*-Dokumentation.

37 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 *Ausgabe der PGEN-Option*, in der *Natural Optimizer Compiler*-Dokumentation.

38 NOCSTAT

NOCSTAT

Das Kommando NOCSTAT liefert Statistikdaten zu Programmen, die sich für die Verarbeitung durch den Natural Optimizer Compiler eignen.

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

39 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.

40 PROFILER

PROFILER

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

Weitere Informationen siehe *Natural Profiler* in der *Debugger und Dienstprogramme*-Dokumentation.

41 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 *Natural Data Collector aktivieren und steuern* in *SYSRDC Utility* in der *Debugger und Dienstprogramme*-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.

42 READ

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

Verwandtes Kommando: EDIT

Mit dem Kommando READ können Sie ein in Quellcodeform 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 *object-name*	Der Name des gewünschten Objekts. Wenn Sie den *object-name* ohne Library-ID angeben, wird das gewünschte Objekt nur dann eingelesen, wenn es in Ihrer aktuellen Library gespeichert ist.
READ *object-name library-id*	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 *object-name* und die *library-id* angeben, liest Natural das Objekt nur dann ein, wenn es in der angegebenen Library gespeichert ist.

43 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 (Quellcode, 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 **Rename Object**-Fenster, in dem Sie dieselben Parameter wie in der Kommandozeile angeben können.
old-name	Als *old-name* geben Sie den derzeitigen Namen des Objekts an.
new-name	Als *new-name* geben Sie den neuen Namen an, unter dem das Objekt von nun an gespeichert sein soll.
new-type	Wenn Sie ein Objekt umbenennen, haben Sie außerdem die Möglichkeit, seinen Objekttyp zu ändern, indem Sie das entsprechende Zeichen für den *new-type* angeben. Dies ist allerdings nur bei Objekten möglich, die lediglich in Quellcodeform bestehen. Mögliche Werte für *new-type*:

3	Dialog
4	Klasse
5	Processor
7	Function
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

44 RENUMBER

RENUMBER [*(n)*]

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

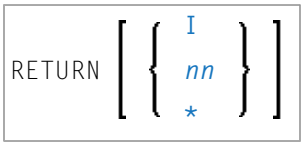


Anmerkung: Wenn auch alphanumerische Konstanten umnummeriert werden sollen, muss der Profilparameter RNCONST auf ON gesetzt werden.

<table>
<tr><td>RENUMBER</td><td>Wenn Sie das Kommando ohne Parameter eingeben, werden, je nach Gesamtzahl der Quellcode-Zeilen im Arbeitsbereich, die folgenden Standard-Schrittwerte für die Umnummerierung verwendet:
<table><tr><td></td><td></td></tr><tr><td>0001 bis 0999 Zeilen:</td><td>10er-Schritte</td></tr><tr><td>1000 bis 1999 Zeilen:</td><td>5er-Schritte</td></tr><tr><td>2000 bis 4999 Zeilen:</td><td>2er-Schritte</td></tr><tr><td>5000 Zeilen und mehr:</td><td>1er-Schritte</td></tr></table>
Ab Zeile 9999 wird immer 9999 als Zeilennummer zugewiesen. Beachten Sie, dass Sie keinen Quellcode kompilieren können, der mehr als 9999 Zeilen enthält.</td></tr>
<tr><td>RENUMBER (n)</td><td>Mit n 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.</td></tr>
</table>

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

45 RETURN



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

Programmierschnittstelle (API): USR1026N. Siehe *SYSEXT Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

RETURN	Wenn Sie RETURN ohne Parameter eingeben, wird die Kontrolle an die vorherige Anwendung übergeben. Eine vorherige Anwendung wird mit dem Systemkommando SETUP als solche definiert. Alle (mit SETUP spezifizierten) Informationen über diese vorherige Anwendung werden gelöscht. Ist keine vorherige Anwendung definiert, erfolgt ein Rücksprung zur Ausgangsanwendung. Wenn Sie RETURN eingeben und kein Rückkehrpunkt definiert ist, wird das RETURN-Kommando ignoriert. **Unter Natural Security:** Ein LOGOFF-Kommando wird ausgeführt, wenn Sie RETURN eingeben und kein Rückkehrpunkt definiert ist.
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 *nn*-te vorherige Anwendung. Alle Informationen über die der *nn*-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.

46 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.

47 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 *Den Status einer RPC-Sitzung überwachen* im Abschnitt *Betrieb einer Natural RPC-Umgebung* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

48 RUN

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

Das Kommando RUN dient dazu, ein Quellcode-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*

RUN	Wenn Sie den Programmnamen (*program name*) nicht angeben, dann kompiliert Natural das Programm, das sich gegenwärtig im Editor-Arbeitsbereich befindet, und führt es aus.
REPEAT	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 RUN das Schlüsselwort REPEAT angeben. Nach Beendigung des Programms wechselt Natural in den Kommando-Modus.
program-name	Der Name des Programms, das kompiliert und ausgeführt werden soll. Wenn Sie den *program-name* 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.
library-id	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.

	Wenn Sie sowohl den *program-name* und die *library-id* 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. 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 Programme aus ihrer aktuellen Library kompilieren/ausführen.

49 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 Quellcode- 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 Quellcodeform 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 Quellcode- und Objektform speichern und so Diskrepanzen zwischen beiden Formen vermeiden.

SAVE	Wenn Sie das Kommando ohne *object-name* benutzen, wird das im Arbeitsbereich des Editors befindliche Objekt in der aktuellen Library gespeichert. Vorhandener Quellcode wird ersetzt.
SAVE *object-name*	Ein neues Objekt wird erstellt. Als *object-name* 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 *object-name library-id*	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 *object-name* die *library-id* dieser Library angeben. Es wird ein neues Objekt angelegt. Falls ein Objekt dieses Namens schon vorhanden ist, wird das Kommando zurückgewiesen.

50 SCAN

- Menü-Optionen 202
- SCAN-Editierkommandos 205
- SCAN mittels Direktkommando 206
- Individuelles SCAN-Profil definieren 209
- SCAN unter Natural Security 210

SCAN

Mit dem Kommando SCAN können Sie den Quellcode 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 Quellcode 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	**Statistics** Diese Funktion liefert Ihnen folgende Informationen: ■ die Anzahl der Objekte, die durchsucht wurden (ohne die übersprungenen Objekte, siehe unten); ■ die Anzahl der Objekte, in denen der gesuchte Wert gefunden wurde; ■ die Anzahl der Quellcode-Zeilen, in denen der gesuchte Wert gefunden wurde. ■ die Anzahl der Objekte, die aufgrund eines Überlaufs des Quellcode-Arbeitsbereichs übersprungen wurden (Natural-Fehler NAT0886); erhöhen Sie die Größe des Profilparameters ESIZE, um alle erforderlichen Objekte zu scannen/ersetzen. **Anmerkung:** Durch Drücken der PF-Taste PF5 erhalten Sie eine Scan-Statistik der letzten Ergebnisse (falls verfügbar).
	L	**List of Objects Containing Scan Value** 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.

Feld	Eingabe	
		Bei Bedarf können Sie die Zeilen mit den entsprechenden ***SCAN-Unterkommandos*** verändern. Mit dem SCAN-Editor können Sie alle Objekttypen, außer Maps und Data Areas und gesperrte Objekte (siehe *Sperrmechanismen für Quellcode-Objekte*), 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.
	S	**Object Lines with Scan Value** Diese Funktion zeigt Ihnen nacheinander die Quellcode-Zeilen an, in denen der gesuchte Wert gefunden wurde. Bei Bedarf können Sie die Zeilen mit den entsprechenden ***SCAN-Unterkommandos*** verändern. Mit dem SCAN-Editor können Sie alle Objekttypen, außer Maps und Data Areas und gesperrte Objekte (siehe *Sperrmechanismen für Quellcode-Objekte*), 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.
Scan value	Der Wert, nach dem gesucht werden soll. **Anmerkung:** 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.	
Replace value	Der Wert, durch den der gesuchte Wert ersetzt werden soll. Bei Maps, Data Areas, Recordings, Dialogen und gesperrten Objekten (siehe *Sperrmechanismen für Quellcode-Objekte*) können Sie den gesuchten Wert nicht durch einen anderen ersetzen.	
Library	Die ID der Library, in der gesucht werden soll; standardmäßig ist dies Ihre aktuelle Library. 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.	
Object name	Das (bzw. die) zu durchsuchende(n) Objekt(e).	
	Leerzeichen	Alle Objekte der Library.

Feld	Eingabe	
	*	
	object-name>	Alle Objekte, deren Namen größer oder gleich *object-name* sind.
	object-name<	Alle Objekte, deren Namen kleiner oder gleich *object-name* sind.
	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. Siehe auch *Namenskonventionen für Objekte* in der Dokumentation *Natural benutzen*.	
Object type(s)	Sie können die Suche auf bestimmte Objekttypen beschränken. Eine Auswahlliste der möglichen Objekttypen erhalten Sie, wenn Sie ein Fragezeichen (?) in dieses Feld eingeben. Wenn Sie dieses Feld leer lassen oder einen Stern (*) eingeben, ist die Suche nicht auf bestimmte Objekttypen beschränkt.	
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		Die Nummer des Predict Set, der durchsucht werden soll. **Anmerkung:** Dazu muss Predict installiert sein. 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 LIST XREF erstellt. Der für **Library** angegebene Wert wird als Predict Set Library verwendet. Wenn Sie die Predict Set-Nummer angeben, müssen Sie bei **Object name** statt eines Namens ein Stern (*) angeben. Ausführliche Informationen zu Predict Sets finden Sie in der *Predict*-Dokumentation.

Feld	Eingabe	
Predict set user		Die Benutzerkennung des Benutzers, der den Predict Set erstellt hat. **Anmerkung:** Dazu muss Predict installiert sein. Falls Sie keine Benutzerkennung angeben, wird der Wert der Systemvariablen `*USER` verwendet (siehe *Systemvariablen*-Dokumentation).
Ignore Case	`Y`	Die Suche findet jede beliebige Kombination aus Groß- und Kleinbuchstaben.
	`N`	Standardmäßig wird nach dem exakten Wert gesucht.
Ignore Comments	`Y`	Innerhalb von Kommentaren gefundene Werte werden nicht angezeigt oder ersetzt.
	`N`	Standardmäßig werden alle gefundenen Werte angezeigt und ersetzt.

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

SCAN-Editierkommandos

In der (den) Kommandozeile(n) des Ergebnis-Schirms können Sie die in der folgenden Tabelle aufgeführten Editierkommandos verwenden.

Kommando	Funktion
Leerzeichen	Fortsetzen der normalen `SCAN`-Verarbeitung.
`Q`	Beenden der `SCAN`-Verarbeitung.
.	
`EDIT`	Objekt editieren im Full-Screen-Editor. **Anmerkung:** Dieses Editierkommando steht nicht zur Verfügung, wenn Programm-Editor, Datenbereich-Editor (*`Data Area Editor`*) und Masken-Editor (*`Map Editor`*) in Ihrer Umgebung abgeschaltet (deaktiviert) sind. Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung* in der *Editoren*-Dokumentation.
`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.

Kommando	Funktion
.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 Quellcode-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 **positionsgebundenen 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 **positionsgebundenen 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 ***Code***.
`LIB`	max. 8 Zeichen	Name der zu durchsuchenden Library. Einzelheiten siehe Beschreibung des Menüfeldes ***Library***. Standardwert ist der Name der aktuellen Library.
`OBJ`	max. 9 Zeichen	Objektname. Die zu durchsuchenden Objekt-Quellcodes. Einzelheiten siehe Beschreibung des Menüfeldes ***Object name***.
`TYPE`	max. 14 Zeichen	Objekttyp. Der (oder die) für die Auswahl zu verwendende(n) Objekttyp(en). Einzelheiten siehe Beschreibung des Menüfeldes ***Object type(s)***.
`SVAL`	max. 32 Zeichen	Suchwert. Die Zeichenkette (**ohne eingebettete Leerzeichen**), nach der in jedem Objekt-Quellcode gesucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes ***Scan value***.
`RVAL`	max. 32 Zeichen	Der Ersetzungswert. Die Zeichenkette (**ohne eingebettete Leerzeichen**), die durch den Suchwert ersetzt werden soll. Einzelheiten siehe Beschreibung des Menüfeldes ***Replace value***.
`ABSOL`	`Y` oder `N`	Absolute Suche. Einzelheiten siehe Beschreibung des Menüfeldes ***Absolute scan***.
`SETNO`	2 Stellen	Nummer des Predict Set, der durchsucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes ***Predict set number***.
`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 ***Predict set user***.

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 positionsgebundenen Parametern

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

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

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

Syntax des Direktkommandos `SCAN` mit positionsgebundenen Parametern:

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

Syntax-Beschreibung:

Positionsgebundener Parameter	Wert	Erklärung
func	`T`, `L` oder `S`	Funktionscode. Einzelheiten siehe Beschreibung des Menüfeldes *Code*.
scan-value	max. 32 Zeichen	Suchwert. Die Zeichenkette (**ohne eingebettete Leerzeichen**), nach der in jeder Objekt-Quellcode gesucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes *Scan value*.
replace-value	max. 32 Zeichen	Einzusetzender Wert. Die Zeichenkette (**ohne eingebettete Leerzeichen**), die durch den Suchwert ersetzt werden soll. Einzelheiten siehe Beschreibung des Menüfeldes *Replace value*.
library	max. 8 Zeichen	Name der zu durchsuchenden Library. Einzelheiten siehe Beschreibung des Menüfeldes *Library*. Standardwert ist der Name der aktuellen Library.

Positionsgebundener Parameter	Wert	Erklärung
object-name	max. 9 Zeichen	Objektname. Die zu durchsuchenden Objekt-Quellcodes. Einzelheiten siehe Beschreibung des Menüfeldes ***Object name***.
object-type	max. 14 Zeichen	Objekttyp. Der (oder die) für die Auswahl zu verwendende(n) Objekttyp(en). Einzelheiten siehe Beschreibung des Menüfeldes ***Object type(s)***.
absolute	Y oder N	Absolute Suche. Einzelheiten siehe Beschreibung des Menüfeldes ***Absolute scan***.
set-number	2 Stellen	Nummer des Predict Set, der durchsucht werden soll. Einzelheiten siehe Beschreibung des Menüfeldes ***Predict set number***.
set-user	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 ***Predict set number***.

Beispiele für Direktkommando SCAN mit positionsgebundenen 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 steht beim Natural-Systemkommando SCAN das Unterkommando SCANPROF oder die PF-Taste PF10 (Prof) zur Verfügung.

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 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 Quellcode-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 Quellcode-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.

51 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.

52 SETUP

- Syntax-Erklärung 214
- Beispiel für SETUP/RETURN 215

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

I	Wenn Sie ein I eingeben, werden alle mit vorherigen SETUP-Kommandos definierten Rückkehrpunkte gelöscht und die mit diesem SETUP I-Kommando angegebene Anwendung als neue Ausgangsanwendung definiert. In einer Nicht-Security-Umgebung gilt: Wenn Sie aus der Library SYSTEM 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.

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 I` ab.

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

53 SHOWDBS

- Entsprechendes API USR8211N 219

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                                                    !
      !   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 *Mögliche Datenbanktypen*.
Default DB type	Der Standard-Datenbanktyp.
DB Options	Eine oder mehrere für die Datenbank definierte Optionen, siehe *Mögliche Datenbank-Optionen*.

Mit den nachfolgend aufgeführten PF-Tasten können Sie (je nach vorhandener 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 Anwendungsprogrammierschnittstelle `USR8211N` in der System-Library `SYSEXT` liefert die gleichen Informationen wie das Systemkommando `SHOWDBS`.

54 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 DIAGOSTICS`-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
```

```
11:14:41           *** SQLDIAG Diagnostic Information ***           2006-04-15
                          - Connection Information -



DB2_Authentication_Type ..
DB2_Authentication_ID .... GGS

DB2_Connection_State .....           0
DB2_Connection_Status ....           0
DB2_Encryption_Type ......
DB2_Product_ID ........... DSN08010
DB2_Server_Class_Name .... QDB2 for DB2 UDB for z/OS









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

55 SQLERR

SQLERR

Anmerkungen:

1. Dieses Kommando steht bei Natural for DB2 zur Verfügung.

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),
- die von DB2 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

```
                ***** SQLERR Diagnostic Information *****
---------------------- NATURAL SQL Interface Codes -------------------------
Return Code: 3700    Reason Code: 0      SQLSTATE : 52003    SQLCODE: -206
--------------------------------- SQLCA-------------------------------------
SQLERRP (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
```

56 STOW

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

Verwandte Kommandos: SAVE | CATALOG

Das Kommando STOW dient dazu, ein Natural-Objekt (sowohl in Quellcodeform 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 *object-name* benutzen, wird das im Arbeitsbereich des Editors vorhandene Quellcode-Objekt und der erzeugte Code unter demselben Namen in der aktuellen Library gespeichert. Vorhandener Quellcode- und Objektcode werden ersetzt.
STOW *object-name*	Verwenden Sie diese Kommandosyntax, um ein neues Objekt (Quellcode und generierter Code) namens *object-name* in der aktuellen Library zu speichern. Falls das Objekt schon vorhanden ist, wird das Kommando zurückgewiesen.
STOW *object-name library-id*	Wenn Sie sowohl *object-name* als auch *library-id* angeben, wird ein neues Objekt angelegt und unter diesem Namen in der angegebenen Library gespeichert. Falls das Objekt in Quellcodeform 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.

57 STRUCT

- Generate Structured Source into Work Area 230
- Display Structure of Source 233
- Print Structure of Source 234
- Write Structure of Source into Work Area 235

```
STRUCT [(n)]
```

Das Kommando STRUCT erfüllt zwei Aufgaben:

- Sie können es dazu verwenden, die Quellcode-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-Quellcodes unabhängig davon verarbeitet, ob sie tatsächlich katalogisiert werden können oder nicht, wird der Quellcode nicht auf syntaktische Korrektheit analysiert. Obwohl das STRUCT-Kommando sorgfältig strukturierte Quellcode-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 Quellcode-Programm einrücken, so dass die Einrückung der Quellcode-Zeilen die Struktur des Programms widerspiegelt.

Diese Funktion entspricht der des Editorkommandos STRUCT.

Bei der Einrückung wird die Länge der Quellcode-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 des Quellcodes ein, den Sie einrücken möchten. Der betreffende Quellcode 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 wie viele 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 Quellcode-Zeilen, die nicht „korrekt“ eingerückt (siehe oben) werden konnten.
	N	Es wird keine derartige Meldung ausgegeben.
Return to STRUCT	Y	Nach Ausführen der **Generate**-Funktion gelangen Sie zum `STRUCT`-Menü zurück.
	N	Nach Ausführen der **Generate**-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 Quellcode-Programms von der Einrückung ausschließen. Die beiden Statements müssen jeweils am Anfang einer Quellcode-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 Quellcode 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 des Quellcodes ein, den Sie anzeigen möchten. Der betreffende Quellcode 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	Quellcode-Zeilen auf derselben strukturellen Ebene werden nicht angezeigt. Es werden nur die Zeilen gezeigt, die eine Änderung in der Strukturtabelle am rechten Bildschirmrand bewirken. Anhand der Lücke in der Abfolge der Zeilennummern können Sie sehen, wie viele Zeilen zwischen zwei gezeigten Zeilen nicht gezeigt werden.
	N	Alle Quellcode-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 Quellcode die Quellcode-Zeilennummer des Statements, das den betreffenden Block initiiert, angezeigt.
Structure Table	Rechts neben dem Quellcode wird eine Strukturtabelle 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 Quellcodes werden durch eine entsprechende Meldung in der Strukturtabelle 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-I   *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 dem angezeigten Quellcode nicht beeinflusst.

Print Structure of Source

Mit dieser Funktion können Sie den Quellcode 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 einen Quellcode von der Systemdatei lesen und zusammen mit ihren Struktur-Informationen sowie mehreren Zeilen (mit Zeilennummer `0000`) am Anfang des Quellcodes, 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 Quellcode-Namen angeben *müssen*.

Der Quellcode und seine Struktur-Informationen werden als Text in den Arbeitsbereich geladen und können mit dem Systemkommando `EDIT` editiert werden.

58 SYSADA

SYSADA

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

Mitteils 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 Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

59 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 der API verwendet werden können.

Weitere Informationen siehe *SYSAPI Utility* in der *Debugger und Dienstprogramme*-Dokumentation..

60 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* in der *Debugger und Dienstprogramme*-Dokumentation.

61 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 einen Natural-Quellcode zu prüfen oder zu ändern.

Weitere Informationen siehe *SYSCP Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

Siehe auch:

- *Unicode and Code Page Support*
- Systemkommando CPINFO

62 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 *Datenbankmanagementsystem-Schnittstellen*-Dokumentation.

63 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 *DDM-Editor (SYSDDM Utility)* in der *Editoren*-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.

64 SYSEDT

SYSEDT

Mit diesem Kommando rufen Sie die SYSEDT-Utility für Editor Buffer Pool Services auf. Die SYSEDT-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 *SYSEDT Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

65 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 *Debugger und Dienstprogramme*-Dokumentation..

66 SYSEXT

SYSEXT

Mit diesem Kommando rufen Sie das Dienstprogramm SYSEXT auf.

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

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

Weitere Informationen siehe *SYSEXT Utility* in der *Debugger und Dienstprogramme*-Dokumentation.



Anmerkung: Das Dienstprogramm SYSEXT steht nicht zur Verfügung, wenn Programm-Editor, Datenbereich-Editor (*Data Area Editor*) und Masken-Editor (*Map Editor*) in Ihrer Umgebung abgeschaltet (deaktiviert) sind. Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung* in der Editoren-Dokumentation.

67 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 *Debugger und Dienstprogramme*-Dokumentation..

68 SYSFILE



Mit diesem Kommando können Sie die `SYSFILE`-Funktion des Dienstprogramms `SYSTP` aufrufen. Diese Funktion liefert Informationen zu den verfügbaren Arbeitsdateien und Druckdateien (Work Files und Print Files).

Die vom Systemkommando `SYSFILE` gelieferten Informationen erhalten Sie auch mit Hilfe der Programmierschnittstelle (API) `USR1007N`. Siehe *SYSEXT Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

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

Weitere Informationen siehe *Natural Print/Work Files (SYSFILE)* im Abschnitt *Allgemeine SYSTP-Funktionen* in der *Debugger und Dienstprogramme*-Dokumentation sowie die plattformspezifischen Informationen zu Work Files (Arbeitsdateien) und Print Files (Druckdateien) in der *Operations*-Dokumentation.

69 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* in der *Debugger und Dienstprogramme*-Dokumentation.

70 SYSNCP

SYSNCP

Mit diesem Kommando rufen Sie die SYSNCP-Utility auf.

Weitere Informationen siehe *SYSNCP Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

71 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.

72 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 *Direktkommandos und Batch-Modus* in der Beschreibung der SYSPARM-Utility).

Weitere Informationen siehe *SYSPARM Utility* in der *Debugger und Dienstprogramme*-Dokumentation..

73 SYSPROD

- Kommando SYSPROD im Online-Modus benutzen 268
- Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen) 268
- Anwendungsprogrammierschnittstellen (APIs) 269

```
SYSPROD
```

Mit dem Kommando SYSPROD können Sie feststellen, welche Produkte in Ihrer Natural-Umgebung installiert sind. Sie erhalten Informationen zu Ihrer aktuellen Natural-Version und 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	Extended - Zusätzliche Produktinformationen anzeigen. Drücken Sie Enter, um die Anzeige des Fensters zu beenden.
HI	Historie der Produktinformationen anzeigen. Drücken Sie Enter, um in der Produkthistorie zu blättern oder die Anzeige des Fensters zu beenden.
SC	Subcomponents - Unterbestandteile des Produkts anzeigen.

Anmerkung: Bei einigen Produkten sind keine Zeilenkommandos verfügbar.

Kommando SYSPROD mit zusätzlichen Schlüsselwörtern benutzen (für Batch-Modus empfohlen)

```
SYSPROD [ ALL                ] [ EX ]
        [ <product-code>     ] [ 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.



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 %K*nn* 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* in der *Debugger und Dienstprogramme*-Dokumentation.

74 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 Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

75 SYSRPC

```
        ┌ CSMASS                          ┐
SYSRPC  │ 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 *Benötigte Puffergröße berechnen* und *Direktkommando SYSRPC CSMASS benutzen*.
`SGMASS` *`generation-option`*	Dient zur Generierung mehrerer Schnittstellenobjekte. Siehe *Mehrere Interface-Objekte generieren* und *Direktkommando SYSRPC SGMASS benutzen* .
`SM REPLACE` *`replace-clause`*	Dient zur Ersetzung einzelner oder mehrerer Einträge in einem Service Directory. Siehe *`replace-clause`* in *Kommandos zur Pflege des Service Directory* .

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.

76 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 *Debugger und Dienstprogramme*-Dokumentation.

77 TECH

TECH

Mit diesem Kommando können Sie sich technische und andere Informationen über Ihre Natural-Session anzeigen lassen.

Folgende Informationen werden angezeigt:

- Benutzerkennung (User ID)
- Bibliothekskennung (Library ID)
- Natural-Version: Major Version, Minor Version, Service Pack, kumulativer Fix (siehe auch *Version* im *Glossary*)
- Startup-Transaktion
- Natural Security-Indikator
- Natural-Editoren: Zeigt an, ob der Natural-Programm-Editor (*Program Editor*), der Datenbereich-Editor (*Data Area Editor*) und der Masken-Editor (*Map Editor*) ausgeschaltet (deaktiviert) oder eingeschaltet (aktiviert) sind (siehe auch Systemvariable `*EDITOR`).
- Betriebssystemname und -version
- Maschinenklasse
- Hardware
- IBM Architecture Level, der vom aktuellen IBM-Prozessor unterstützt wird (nur bei Großrechnern mit z/OS). `0` (Null) bedeutet, dass keine Architecture Levels unterstützt werden.
- TP-Monitor (Großrechner und Windows (Systemvariable `*TPSYS`) nur in einer Remote-Konfiguration)
- Gerätetyp
- Terminal-Kennung (auf Großrechnern, bei Windows nur in einer Remote-Konfiguration)
- Codepage

- Locale
- Zuletzt ausgeführtes Kommando
- Informationen zum zuletzt aufgetretenen Fehler
- Namen, Datenbankkennungen (DBID) und Dateinummern (FNR) aller zurzeit aktiven Steplib-Libraries
- Namen, Typen und Ebenen des zurzeit aktiven Natural-Objekts und aller Objekte auf höheren Ebenen sowie die Zeilennummern der Statements, die die untergeordneten Programmierobjekte (nur bei Großrechnern und UNIX) aufrufen.



Anmerkungen:

1. Nur bei Anwendungen mit zeichenbasierter Benutzeroberfläche (CUI): 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 Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

78 TEST



Mit dem Kommando TEST rufen Sie die Debugging-Funktion auf.

TEST	Wenn Sie nur TEST eingeben, erhalten Sie das Hauptmenü der Debugging–Funktion. Mit TEST ON/OFF 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.
debugger-commands	Die Direktkommandos zum Ausführen von Debugging-Funktionen sind im Abschnitt *Debug-Kommandoübersicht und -syntax* in der *Debugger und Dienstprogramme*-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.

Informationen über weitere Natural-Utilities, die Sie zum Testen und Überwachen im Online-Modus verwenden können, finden Sie in der *Debugger und Dienstprogramme*-Dokumentation. (deutsch).

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- oder UNIX-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.

79 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.
parameters	Die Parameter, die Sie beim TEST DBLOG-Kommando benutzen können, sind im Abschnitt *TEST DBLOG -Kommando* in der *Debugger und Dienstprogramme*-Dokumentation beschrieben.

Weitere Informationen siehe *DBLOG Utility* in der *Debugger und Dienstprogramme*-Dokumentation.

Die *Debugger und Dienstprogramme*-Dokumentation (englisch) enthält Informationen über weitere Natural-Dienstprogramme, 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.

80 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.

81 UNLOCK

- Natural-Objekte entsperren 286
- UNLOCK-Parameter-Beschreibungen 287
- Parameterverarbeitung und Anzeige der gefundenen Objekte 289
- Batch-Verarbeitung 290

Mit dem UNLOCK-Kommando können Sie Natural-Quellcode-Objekte lokal in einer Natural-Großrechnerumgebung entsperren.

Das UNLOCK-Kommando dient dazu, Quellcode-Objekte 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 *Sperrmechanismen für Quellcode-Objekte* in der *Editoren*-Dokumentation und Profilparameter SLOCK in der *Parameter Referenz*-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, in 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	
object-name	A33	*	Der Name des zu entsperrenden Objekts. Sie können Stern-Notation (*) oder das Größerzeichen (>) verwenden.	
object-type	A1	*	**Natural-Objekttypen:** Anstelle von *object-type* können Sie einen der unten aufgeführten Objekttypcodes oder Stern-Notation (*) benutzen.	
			P	Program
			4	Class
			N	Subprogram
			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)
library-name	A8	*	Name der Library, in der das gesperrte Objekt gespeichert ist. Sie können Stern-Notation (*) benutzen.	
dbid	A5	Aktuelle Datenbankkennung	Datenbankkennung der angegebenen Library. Sie können Stern-Notation (*) benutzen oder die ID im Format N5 angeben. Wenn Sie Stern-Notation (*) benutzen, werden nur die aktuellen Systemdateien `FNAT` und `FUSER` abgesucht.	

Parameter	Format/Länge	Standardwert	Beschreibung
fnr	A5	Aktuelle Dateinummer	Dateinummer der angegebenen Library. Sie können Stern-Notation (*) benutzen oder die Nummer im Format N5 angeben. Wenn Sie Stern-Notation (*) benutzen, werden nur die aktuellen Systemdateien FNAT und FUSER abgesucht.
password	A8	leer	Falls verwendet, das Passwort für die angegebene Systemdatei (*dbid* und *fnr*). Sie brauchen kein Passwort anzugeben, wenn Sie die *dbid* und *fnr* der aktuellen Datei FNAT oder FUSER benutzen.
cipher	A8	leer	Falls verwendet, der Chiffrierschlüssel für die angegebene Systemdatei (*dbid* und Dateinummer *fnr*). Braucht angegeben zu werden, wenn die *dbid* und *fnr* der aktuellen Systemdatei FNAT oder FUSER verwendet wird.
locked-by	A8	aktuelle Benutzerkennung oder Stern-Notation (*) bei Batch-Betrieb und mit Security-Flag "F"	Die Benutzerkennung des Benutzers, der die Sperrung des Objekts veranlasst hat. Sie können Stern-Notation (*) verwenden. Wenn Natural Security verwendet wird, kann der Parameter nur geändert werden, wenn im Natural Security-Benutzerprofil der Security-Unlock-Flag auf "F" (Forced Unlock) gesetzt ist.
locked-on	A10	leer	Die beiden Datumsparameter sind vorhanden, um die verschiedenen Datumsformate abzudecken: 2005-09-28 (Datumsformat gemäß DTFORM-Profilparameter) 2005-09-28 11:27:20 Heute Heute + *nnnn* Heute - *nnnn* Gestern
locked-on2	A8		

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 and 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 **Cmd** 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 **unlocked** in der Spalte **Message** angezeigt.

Batch-Verarbeitung

Falls kein Fehler aufgetreten ist, werden alle gesperrten Objekte entsperrt, und es erscheint eine entsprechende Meldung.

82 UPDATE

```
UPDATE { ON  }
       { OFF }
```

Das Kommando UPDATE dient dazu, von einem Natural-Programm durchgeführte Datenänderungen zu gestatten bzw. nicht zu gestatten.

UPDATE ON	Datenbankänderungen sind erlaubt.
UPDATE OFF	Datenbankänderungen sind nicht erlaubt. Wenn ein Programm, das ein datenbankveränderndes Statement (UPDATE, STORE oder DELETE) enthält, ausgeführt wird, erfolgt keine Datenbankänderung. Stattdessen wird bei der nächsten Bildschirm-Ein-/Ausgabe eine Warnmeldung NAT1010 ausgegeben. Des Weiteren stellt eine Datenbankschleife, die ein UPDATE oder DELETE Statement enthält, keine Datensätze in den Hold-Status (kein „Lesen mit Hold“).

83 WHICH

- Direktkommando-Syntax 294
- Parameter-Beschreibung 294
- Standardausgabe des WHICH-Kommandos 294
- Vollständige Liste des WHICH-Kommandos 295
- Batch-Verarbeitung 296
- WHICH-Kommando unter Natural Security 297

```
WHICH [ object-name [FULL] ]
      [       ?            ]
```

Dieses Kommando können Sie benutzen, um ein Objekt in den zur Laufzeit verwendeten Libraries zu finden (in der so genannten „Library-Suchreihenfolge“).

Direktkommando-Syntax

`WHICH`	Es erscheint ein Fenster, in dem Sie im Feld `Object name` den Namen des gesuchten Objekts eingeben müssen. Zusätzlich können Sie das Feld `Full` markieren, um die Ausgabe der vollständigen Liste der Fundstellen anzuzeigen.
`WHICH` *`object-name`*	Zeigt den Objekttyp, den Namen der ersten Library, in der das Objekt gefunden wurde, und die Datenbankkennung (DBID) und die Dateinummer (FNR) dieser Library an. Siehe ***Standardausgabe des WHICH-Kommandos***.
`WHICH` *`object-name`* `FULL`	Listet alle Libraries in der Library-Suchreihenfolge auf. Bei jeder Library wird angegeben, ob das katalogisierte Objekt oder das Quellcodeobjekt gefunden wurde. Sie können eine neue Suche starten, indem Sie PF5 (New) drücken. Siehe ***Vollständige Liste des WHICH-Kommandos***.
`WHICH ?`	Zeigt den Hilfetext für das Kommando `WHICH` an.

Parameter-Beschreibung

Parameter	Format/Länge	Standardwert	Beschreibung
`object-name`	A8	keiner	Der Name des Objekts, nach dem gesucht werden soll.

Standardausgabe des WHICH-Kommandos

Wenn Sie das Kommando `WHICH` ohne die Option `FULL` absetzen, sieht das Ergebnis wie in folgendem Beispiel aus:

```
Cataloged subprogram USR1002N found in library SYSEXT(12,124).
```

Weitere Vorgehensweise:

- Sie können eine neue Suche starten, indem Sie in der Eingabeaufforderungszeile `MORE` das Kommando `WHICH` erneut absetzen.
- Sie können Enter drücken, um das Systemkommando `WHICH` zu beenden.

Vollständige Liste des WHICH-Kommandos

Wenn Sie das Kommando `WHICH` mit der Option `FULL` absetzen, werden alle Libraries der Library-Suchreihenfolge aufgelistet. Die Ausgabe sieht dann zum Beispiel folgendermaßen aus:

```
12:00:49               ***** NATURAL WHICH COMMAND *****            2023-05-25
User XXXX                 - Search object USR1002N -           Library MYLIB

Library   DBID   FNR Type Save/Catalog Date   Message
-------- ----- ----- ---- ------------------ --------------------------------
MYLIB       12   123
MYLIB01     12   123
MYLIB02     12   123
PROTECT                                     Library access denied
SYSEXT      12   124  N   2022-04-20 07:59:36 Cataloged object found
                      N   2011-05-06 16:37:46 Saved object found
SYSTEM      12   123
SYSTEM      12   124









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

Spaltenüberschriften

Die Spaltenüberschriften haben folgende Bedeutung:

Spalte	Erläuterung
Library	Die Namen der Libraries in der Library-Suchreihenfolge.
DBID FNR	Die Datenbankkennung und die Dateinummer der Natural-Systemdatei, in der die Library enthalten ist.
Type	Der Natural-Objekttypcode des Objekts. Siehe auch Object Type Code (Objekttypcode) im Glossar.
Save/Catalog Date	Datum der Speicherung bzw. der Katalogisierung des Objekts.
Message	Anzeige des Suchergebnisses. Eine leere Zeile bedeutet, dass das gesuchte Objekt in der betreffenden Library nicht gefunden wurde.

Funktionstasten

Im Ausgabebildschirm des Systemkommandos `WHICH` mit der Option `FULL` stehen folgende Funktionstasten zur Verfügung:

PF-Taste	Belegung	Funktion
PF1	Help	Hilfe aufrufen.
PF3	Exit	Systemkommando `WHICH` beenden.
PF5	New	Neue Suche beginnen.
PF6	--	Zum Anfang der Liste springen.
PF7	-	Seitenweise rückwärts blättern.
PF8	+	Seitenweise vorwärts blättern.
PF9	++	Zum Ende der Liste springen.
PF12	Canc	Systemkommando `WHICH` abbrechen.

Batch-Verarbeitung

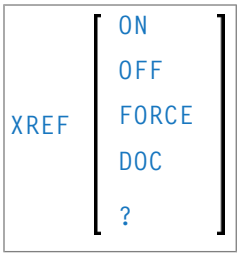
Im Batch-Modus ist die Ausgabe des `WHICH`-Kommandos ohne die Option `FULL` die gleiche wie im Online-Modus.

Wird das `WHICH`-Kommando mit der Option `FULL` abgesetzt, dann werden alle resultierenden Libraries in einer Liste ausgedruckt.

WHICH-Kommando unter Natural Security

Bezüglich möglicher Zugriffseinschränkungen siehe *Command Restriction* in der *Natural Security*-Dokumentation.

84 XREF



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, Helproutinen, 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:

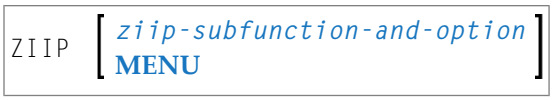
XREF	Wenn Sie das XREF-Kommando ohne Parameter eingeben, erscheint ein Menü, in dem Sie die gewünschte Option auswählen können.
XREF ON	Dieses Kommando schaltet die Dokumentation von aktiven Referenzen ein. Predict dokumentiert die betreffenden Informationen für jedes Natural-Programm bzw. jede Data 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.

85 ZIIP

- Syntax der ziip-subfunction-and-option 302
- Syntax-Element-Beschreibung 303
- zIIP-Berichte 305
- ZIIP-Kommando-Menü für Berichtausgabe-Optionen 310
- User Exit für ZIIP PRINT-Kommando-Optionen 311
- Anwendungsprogrammierschnittstelle (API) für ZIIP-Kommando-Funktionen 311

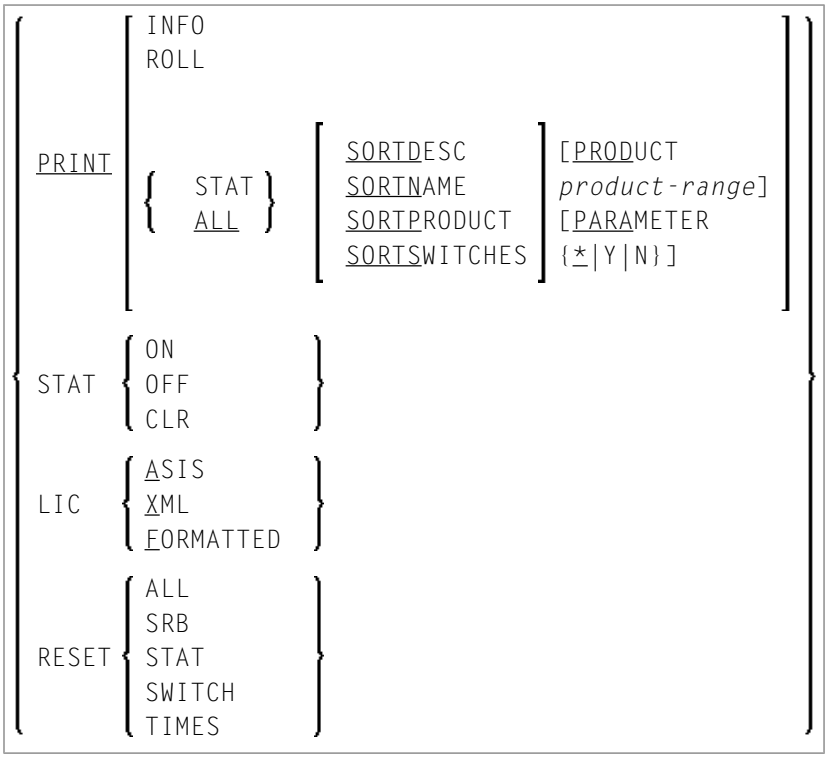


Dieses Systemkommando gilt nur in z/OS Batch- oder TSO-Umgebungen, welche die Verwendung des IBM System z Integrated Information Processor (zIIP) unterstützen. Falls das Kommando in einer anderen Umgebung und/oder in einer Umgebung ohne zIIP-Unterstützung abgesetzt wird, erfolgt eine entsprechende Fehlermeldung.

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 Komponenten-Umschaltvorgänge verwalten.

Informationen zur zIIP-Unterstützung und Überwachung erhalten Sie in der *Natural for zIIP*-Dokumentation.

Syntax der ziip-subfunction-and-option



Syntax-Element-Beschreibung

Subfunktion	Option	Erklärung
PRINT*	INFO	Zeigt den Bericht *Informationen zur zIIP-Verarbeitung* an.
	ROLL	Zeigt den Bericht *Roll Server-zIIP-Informationen* an.
	STAT	Zeigt den Bericht *zIIP-Komponenten-Umschaltstatistik* an, wenn die Zählung der Komponenten-Umschaltvorgänge eingeschaltet worden ist (siehe STAT weiter unten). Bezüglich der Festlegung der Sortierreihenfolge siehe auch *Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik*.
	ALL	Zeigt zuerst den Bericht *Informationen zur zIIP-Verarbeitung* an, danach den Bericht *Roll Server-zIIP-Informationen* und anschließend den Bericht *zIIP-Komponenten-Umschaltstatistik* (falls aktiviert). Dies ist die Standardeinstellung. Bezüglich der Festlegung der Sortierreihenfolge siehe *Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik*.
STAT*	ON	Aktiviert die Zählung der Komponenten-Umschaltvorgänge.
	OFF	Deaktiviert die Zählung der Komponenten-Umschaltvorgänge.
	CLR	Löscht die Tabelle mit den Komponenten-Umschaltvorgängen.
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.
RESET	ALL	Bewirkt das Zurücksetzen aller Zähler.
	SRB	Bewirkt das Zurücksetzen des SRB-Start-Zählers.
	STAT	Bewirkt das Zurücksetzen der Komponenten-Umschalttabelle (gleichbedeutend mit STAT CLR).
	SWITCH	Bewirkt das Zurücksetzen des SRB-Umschalt-Zählers.
	TIMES	Bewirkt das Zurücksetzen aller CPU-Zeitwerte.

* 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.

Die RESET-Funktion hat keinen Einfluss auf die Roll-Server-zIIP-Informationszähler und CPU-Zeiten.

Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik

Schlüsselwort	Sortierreihenfolge
SORTDESC	Der Bericht wird aufsteigend nach Komponentenbeschreibungen sortiert.
SORTNAME	Der Bericht wird aufsteigend nach Komponentennamen sortiert.
SORTPRODUCT	Der Bericht wird aufsteigend nach Produktnamen sortiert.
SORTSWITCHES	Der Bericht wird absteigend nach TCB-Umschaltungen (TCB = Task Control Block) sortiert. Dies ist die Standardeinstellung.

Auswahlkriterien für die zIIP-Komponenten-Umschaltstatistik

Schlüsselwort	Beschreibung	
PRODUCT	Listet die Komponenten eines einzelnen Produkts oder einer mit *product-range* angegebenen Produktreihe auf. Gültige Werte für *product-range*:	
	ppp	Ein gültiger Produktcode. Wählt das Produkt aus, das dem dreistelligen Produktcode *ppp* entspricht, wie in der Spalte **Prd** des Berichts **zIIP Component Switch Statistics** angegeben.
	*p** oder *pp**	Ein oder zwei führende Zeichen. Wählt alle Produkte aus, deren Produktcodes mit *p** oder *pp** beginnen.
	ppp>	Ein Startwert. Wählt alle Produkte mit Produktcodes aus, die größer oder gleich *ppp* sind.
	ppp<	Ein Endwert. Wählt alle Produkte mit Produktcodes aus, die kleiner oder gleich *ppp* sind.
PARAMETER	Gibt die aufzulistenden Komponenten an. Gültige Werte:	
	*	Wählt alle Komponenten aus (Standardeinstellung).
	Y	Wählt nur Komponenten mit einem P-Indikator aus, der angibt, dass die Komponentenschalter durch einen Parameter beeinflusst werden können.
	N	Wählt nur Komponenten (ohne P-Indikator) aus, deren Komponentenschalter nicht durch einen Parameter beeinflusst werden können.

zIIP-Berichte

Dieser Abschnitt beschreibt bei dem Systemkommando `ZIIP` zur Verfügung stehenden Berichte und Statistikdaten.

- Informationen zur zIIP-Verarbeitung
- Roll Server-zIIP-Informationen
- zIIP-Komponenten-Umschaltstatistik

Informationen zur zIIP-Verarbeitung

Der Bericht **zIIP Processing Information** sieht folgendermaßen aus (Beispiel):

```
+---------------- zIIP Processing Information ---------------+
|                                                            |
|            Advanced zIIP Support Enabled                   |
|                                                            |
|  General central processors (GPs)             6           |
|  Integrated Information Processors (zIIPs)    1           |
|   zIIP SMT threads                            2           |
|  zIIP normalization factor                 10.98          |
|                                                            |
|  Switches into TCB mode                 1590303            |
|  SRB starts                                  80            |
|                                                            |
|  Total enclave CPU time           01:00:56.411  (100.00%)  |
|   Enclave GP time                 00:12:38.305   (20.74%)  |
|   Enclave zIIP time               00:48:18.106   (79.26%)  |
|                                                            |
|  Eligible zIIP CPU time           00:48:22.619  (100.00%)  |
|   Enclave zIIP time               00:48:18.106   (99.84%)  |
|   Eligible zIIP time on GP        00:00:04.513    (0.16%)  |
|                                                            |
|  (zIIP times normalized)                                   |
|                                                            |
|                                                            |
+------------------------------------------------------------+
```

Wenn der z/OS-System-Parameter `PROJECTCPU=YES` gesetzt ist und für die aktuelle logische Partition keine zIIPs konfiguriert sind, sieht der Bericht folgendermaßen aus (Beispiel):

```
+--------------- zIIP Processing Information ---------------+
!                                                           !
!           Advanced zIIP Support Enabled (PROJECTCPU)      !
!                                                           !
! General central processors (GPs)             2            !
! Integrated Information Processors (zIIPs)    0            !
!                                                           !
! zIIP normalization factor                 0.00            !
!                                                           !
! Switches into TCB mode                     938            !
! SRB starts                                   6            !
!                                                           !
! Total enclave CPU time            00:00:06.091  (100.00%) !
!  Enclave TCB time                 00:00:04.640   (76.19%) !
!  Enclave SRB time (*)             00:00:01.452   (23.81%) !
!                                                           !
! (*) possible zIIP offload                                 !
!                                                           !
!                                                           !
!                                                           !
+-----------------------------------------------------------+
```

Der Bericht enthält folgende Status- und Verarbeitungsinformationen:

Eintrag	Bedeutung	Erklärung
Advanced zIIP Support Enabled	Erweiterte zIIP-Unterstützung eingeschaltet	Zeigt an, dass die aktuelle Natural-Session mit voller zIIP-Unterstützung läuft.
General central processors (GPs)	Generelle Zentralprozessoren	Die Anzahl der verfügbaren GPs, die unter Ihrem z/OS-System laufen.
Integrated Information Processors (zIIPs)	Integrierte Informationsprozessoren (zIIPs)	Die Anzahl der zIIPs, die unter Ihrem z/OS-System laufen.
zIIP SMT threads	zIIP Simultaneous Multi-Threading Threads	Die Anzahl der logischen zIIPs, die bei aktivem SMT unter Ihrem z/OS-System laufen.
zIIP normalization factor	zIIP-Normalisierungsfaktor	Der zIIP-Normalisierungsfaktor gibt das Verhältnis der zIIP-Geschwindigkeit zur GP-Geschwindigkeit an. Dieser Faktor gibt an, wie schnell Ihr zIIP im Vergleich zu einem gedrosselten GP mit reduzierter Leistung läuft. In dem obigen **Beispiel** bedeutet der Wert `10.98`, dass der zIIP circa elfmal schneller ist als ein GP.
Switches into TCB mode	Umschaltungen in den TCB-Modus	Anzahl der Umschaltungen vom SRB-Modus (SRB = Service Request Block) in den TCB-Verarbeitungsmodus (TCB = Task Control Block) aufgrund von Systemdienstaufrufen, die abgesetzt werden mussten.

Eintrag	Bedeutung	Erklärung
		Die entsprechenden Umschaltungen zurück in den SRB-Modus werden nicht mitgezählt.
SRB starts	Service Request Block Starts	Gibt an, wie oft ein SRB-Vorgang gestartet wurde. Der Zähler wird um 1 erhöht für den ersten SRB bei jedem Session-Start und für den SRB-Neustart nach einer abnormalen Beendigung im SRB-Modus.
Total enclave CPU time	CPU-Gesamtzeit der Enklave	Die von der aktuellen Natural-WLM-Enklave verbrauchte CPU-Gesamtzeit (GCP plus zIIP).
Enclave GP time	GP-Gesamtzeit der WLM-Enklave	Die innerhalb der aktuellen Natural-WLM-Enklave verbrauchte GP-Gesamtzeit.
Enclave zIIP time	zIIP-Gesamtzeit der WLM-Enklave	Die innerhalb der aktuellen Natural-WLM-Enklave verbrauchte zIIP-Gesamtzeit.
Eligible zIIP CPU time	In Frage kommende zIIP-CPU-Zeit	Die CPU-Gesamtzeit, die für einen zIIP in Frage kommt.
Eligible zIIP time on GP	In Frage kommende zIIP-CPU-Zeit beim GCP	Die CPU-Zeit innerhalb der Natural-WLM-Enklave, die für einen zIIP 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.

Ausführliche Informationen zu den Statistikangaben siehe *Natural zIIP-Verarbeitung: TCBs, SRBs und Enklaven* und *CPU-Zeit berechnen* in der *Natural for zIIP*-Dokumentation.

Hinweise, wie Sie die zIIP-Nutzung verbessern können, finden Sie im Abschnitt *Tuning der zIIP-Nutzung* in der *Natural for zIIP*-Dokumentation.

PF-Tasten auf dem Bildschirm zur Anzeige von zIIP-Verarbeitungsinformationen

Taste	Funktion
ENTER oder PF5	Aktualisieren der Berichtdaten.
PF7	Umschalten auf die Anzeige der ***zIIP-Komponenten-Umschaltstatistik***. (Gilt nur bei `PRINT ALL`.)
PF8	Umschalten auf die Anzeige der ***Roll Server-zIIP-Informationen***. (Gilt nur bei `PRINT ALL`.)

Roll Server-zIIP-Informationen

Der Bericht **Roll Server zIIP Information** sieht folgendermaßen aus (Beispiel):

```
+------------- Roll Server zIIP Information -------------+
!                                                        !
! Roll Server zIIP CPU Time Usage in Subsystem QA82      !
!                                  Job Name QAROLL82     !
!                                                        !
! Total WLM enclave CPU time ....               387524   !
! Qualified zIIP CPU time .......               387252   !
! Eligible zIIP CPU time on GCP                    133   !
! Total WLM enclave zIIP CPU time               387119   !
! Total WLM enclave zIIP CPU time (%)            99.89   !
!                                                        !
! (All times in milliseconds; zIIP times normalized.)    !
!                                                        !
+--------------------------------------------------------+
```

Der Bericht enthält folgende Status- und Verarbeitungsinformationen:

Eintrag	Bedeutung	Erklärung
Total WLM enclave CPU time	CPU-Gesamtzeit der WLM-Enklave	Die von der aktuellen WLM-Enklave verbrauchte CPU-Gesamtzeit (GCP plus zIIP).
Qualified zIIP CPU time	Qualifizierte zIIP-CPU-Zeit	Die für den zIIP-Gebrauch bei SRB qualifizierte CPU-Zeit, welche die Summe der in Frage kommenden plus der gesamten zIIP-CPU-Zeit ist.
Eligible zIIP CPU time on GCP	In Frage kommende zIIP-CPU-Zeit beim GCP	Die CPU-Zeit beim GCP, die für einen zIIP 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.
Total WLM enclave zIIP CPU time	zIIP-CPU-Gesamtzeit der WLM-Enklave	Die von der aktuellen WLM-Enklave verbrauchte zIIP-CPU-Gesamtzeit.
Total WLM enclave zIIP CPU time (%)	zIIP-CPU-Gesamtzeit der WLM-Enklave (%)	Die von der aktuellen WLM-Enklave prozentual verbrauchte zIIP-CPU-Gesamtzeit. Im obigen Beispiel konnte Natural 99.89 % der CPU-Gesamtlast an einen zIIP abgeben.

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 Beispielmaske 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 andere 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 auf dem Bildschirm zur Anzeige von zIIP-Komponenten-Statistikinformationen

ENTER oder PF5	Aktualisiert die Statistikdaten.
PF4	Öffnet ein Fenster, in dem Sie Auswahlkriterien (Produktbereich und Komponenten-Umschaltvorgänge, die durch einen Parameter beeinflusst werden können) für den Bericht **zIIP Component Switch Statistics** (zIIP-Komponenten-Umschaltstatistik) festlegen können.
PF7	Um eine Seite zurückblättern (in einer langen Liste) oder umschalten auf den Bericht **zIIP Processing Information**, siehe *Informationen zur zIIP-Verarbeitung*. (Gilt nur für die erste Seite einer Liste und wenn `PRINT ALL` verwendet wird.)
PF8	Um eine Seite weiterblättern (in einer langen Liste) oder umschalten auf den Bericht **zIIP Processing Information**, siehe *Informationen zur zIIP-Verarbeitung*. (Gilt nur für die letzte Seite einer Liste und wenn `PRINT ALL` verwendet wird.)
PF9	Umschalten zwischen den beiden Anzeigen, die bei `SORTPRODUCT` und `SORTSWITCHES` in *Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik* beschrieben sind.
PF10	Umschalten zwischen den beiden Anzeigen, die bei `SORTDESC` und `SORTNAME` in *Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik* beschrieben sind.
PF11	Umschalten zwischen den beiden Anzeigen, die bei `SORTDESC` und `SORTSWITCHES` in *Sortieroptionen für die zIIP-Komponenten-Umschaltstatistik* beschrieben sind.

ZIIP-Kommando-Menü für Berichtausgabe-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 Kommandoeingabeaufforderung 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 Subprogramm ZIIPUS01 in der Natural-System-Library SYSLIB, SYSLIBS oder SYSTEM zur Verfügung gestellt. Einzelheiten hierzu siehe Subprogramm ZIIPUS01 in der System-Library SYSEXT bzw. im Dienstprogramm (Utility) 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, 21

B

BUS
Systemkommando, 23

C

CATALL
Systemkommando, 25
CATALOG
Systemkommando, 33
CHECK
Systemkommando, 35
CLEAR
Systemkommando, 37
COMPOPT
Systemkommando, 39

D

DELETE
Systemkommando, 65
DUMP
Systemkommando, 69

E

EDIT
Systemkommando, 71
EDT
Systemkommando, 77
EXECUTE
Systemkommando, 81

F

FIN
Systemkommando, 85

G

GLOBALS
Systemkommando, 87

H

HELP
Systemkommando, 91

I

INPL
Systemkommando, 93

K

KEY
Systemkommando, 95

L

LAST
Systemkommando, 99
LASTMSG
Systemkommando, 101
LIST
Systemkommando, 103
LIST XREF
Systemkommando, 143
LISTDBRM
Systemkommando, 145
LISTPROF
system command, 149
LISTSQL
Systemkommando, 153
LISTSQLB
Systemkommando, 159
LOGOFF
Systemkommando, 163
LOGON
Systemkommando, 165

M

MAIL
Systemkommando, 167
MAINMENU
Systemkommando, 169

N

NATQVS

Systemkommando, 171
NOCOPT
Systemkommando, 173
NOCSHOW
Systemkommando, 175
NOCSTAT
Systemkommando, 177

P

PROFILE
Systemkommando, 179
PROFILER
Systemkommando, 181

R

RDC
Systemkommando, 183
READ
Systemkommando, 185
RENAME
Systemkommando, 187
RENUMBER
Systemkommando, 189
RETURN
Systemkommando, 191
ROUTINES
Systemkommando, 193
RPCERR
Systemkommando, 195
RUN
Systemkommando, 197

S

SAVE
Systemkommando, 199
SCAN
Systemkommando, 201
SCRATCH
Systemkommando, 211
SETUP
Systemkommando, 213
SHOWDBS
system command, 217
SQLDIAG
system command, 221
SQLERR
system command, 225
STOW
Systemkommando, 227
STRUCT
Systemkommando, 229
SYSADA
Systemkommando, 237
SYSAPI
Systemkommando, 239
SYSBPM
Systemkommando, 241
SYSCP
Systemkommando, 243
SYSDB2
Systemkommando, 245
SYSDDM
Systemkommando, 247
SYSEDT
Systemkommando, 249
SYSERR
Systemkommando, 251
SYSEXT
Systemkommando, 253
SYSEXV
Systemkommando, 255
SYSFILE
Systemkommando, 257
SYSMAIN
Systemkommando, 259
SYSNCP
Systemkommando, 261
SYSOBJH
Systemkommando, 263
SYSPARM
Systemkommando, 265
SYSPROD
system command, 267
SYSPROF
Systemkommando, 271
SYSRPC
Systemkommando, 273
Systemkommandos, ix
SYSTP
Systemkommando, 275

T

TECH
Systemkommando, 277
TEST
Systemkommando, 279
TEST DBLOG
Systemkommando, 281

U

UNCATALOG
Systemkommando, 283
UNLOCK
Systemkommando, 285
UPDATE
Systemkommando, 291

W

WHICH
Systemkommando, 293

X

XREF
Systemkommando, 299

Z

ZIIP
system command, 301