Natural for z/OS

Debugger und Dienstprogramme (Utilities)

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: NAT-DEBUG-925-20261013DE

Inhaltsverzeichnis

Debugger und Dienstprogramme (Utilities) xix
1 Über diese Dokumentation 1
Dokumentationskonventionen 2
Online-Informationen und Support 2
Datenschutz 3
I Debugger 5
2 Debugger-Tutorial 7
Voraussetzungen 8
Grundlagen des Debugging 8
Sitzung 1 - Analyse eines Natural-Fehlers 9
Sitzung 2 - Einen Breakpoint benutzen 14
Sitzung 3 - Einen Watchpoint benutzen 20
Sitzung 4 - Verfolgen des logischen Ablaufs von Programmen 26
Sitzung 5 - Verwendung von Statistiken über die Programmausführung 30
Weitere Hinweise zur Benutzung des Debuggers 33
Quellcode-Beispiele 37
3 Debugger-Konzept 41
Funktionen zur Kontrolle über die Natural-Sitzung 42
Debug-Einträge (Spies) 43
Debug Break-Fenster 45
4 Debugger starten 47
Debugger unter Natural Security 48
Voraussetzungen für den Betrieb 48
Debugger aufrufen 49
Standard-Objekt 50
5 Test-Modus ein- und ausschalten 53
6 Debug Environment Maintenance (Verwaltung der Debug-Umgebung) 55
Test-Modus für Debug-Umgebung ON/OFF setzen 57
Debug-Umgebung laden 57
Debug-Umgebung speichern 57
Debug-Umgebung zurücksetzen 58
Debug-Umgebung löschen 59
Debug-Umgebung in verschiedenen Libraries verwalten 59
7 Spy Maintenance (Verwaltung der Debug-Einträge) 61
Test-Modus für Debug-Einträge ON/OFF setzen 62
Spy-Funktion aktivieren 62
Spy deaktivieren 63
Debug-Einträge (Spies) löschen 63
Debug-Einträge (Spies) anzeigen 63
Debug-Einträge (Spies) ändern 64
8 Breakpoint Maintenance (Verwaltung der Breakpoints) 67
Verwendungsbedingungen 68

Test-Modus für Breakpoints ON/OFF setzen 69
Breakpoint aktivieren 70
Breakpoint deaktivieren 70
Breakpoint löschen 70
Breakpoint anzeigen 71
Breakpoint ändern 73
Breakpoint setzen 74
Felder und Spalten in Breakpoint-Bildschirmen 75
9 Watchpoint Maintenance (Verwaltung der Watchpoints) 77
Test-Modus für Watchpoints ON/OFF setzen 79
Watchpoint aktivieren 79
Watchpoint deaktivieren 79
Watchpoint löschen 80
Watchpoint anzeigen 80
Watchpoint ändern 83
Watchpoint setzen 84
Felder und Spalten in Watchpoint-Bildschirmen 86
10 Call Statistics Maintenance (Statistiken über gerufene Objekte) 91
Test-Modus für Call-Statistik ON/OFF setzen 92
Call-Statistik ON/OFF setzen 92
Alle Objekte anzeigen 93
Aufgerufene Objekte anzeigen 94
Nicht aufgerufene Objekte anzeigen 94
Objekte drucken 95
11 Statement Execution Statistics Maintenance (Statistiken über ausgeführte Statement-Zeilen) 97
Test-Modus für Statement-Ausführungsstatistiken ON/OFF setzen 98
Funktion Statement Execution Statistics auf ON/OFF/COUNT setzen 98
Statement-Ausführungsstatistiken löschen 101
Statement-Ausführungsstatistiken anzeigen 101
Statements drucken 105
12 Variable Maintenance (Anzeigen und Ändern von Variablen) 107
Benutzervariablen, globale Variablen und datenbankbezogene Systemvariablen anzeigen 108
Systemvariablen anzeigen 111
Variablen ändern 112
13 List Object Source (Objekt-Quellcode anzeigen) 113
Breakpoints verwalten - Maintain Breakpoints 115
14 Fehlerbehandlung 117
Fehler während der Anwendungsausführung 118
Fehler während der Debugger-Ausführung 119
15 Kommandos zur Ausführungssteuerung 121
ESCAPE BOTTOM 122
ESCAPE ROUTINE 122
EXIT 122

GO ... 123
NEXT ... 123
RUN ... 123
STEP ... 124
STEP SKIPSUBLEVEL ... 124
STEP SKIPSUBLEVEL n ... 124
STOP ... 124
16 Kommandos zum Navigieren und Anzeigen von Informationen ... 125
BREAK ... 126
FLIP ... 126
LAST ... 126
OBJCHAIN ... 126
ON/OFF ... 127
PROFILE ... 127
SCAN ... 128
SCREEN ... 128
SET OBJECT ... 128
STACK ... 129
SYSVARS ... 129
TEST ON/OFF ... 129
17 Debug-Kommandoübersicht und -syntax ... 131
Alle Debug-Kommandos ... 132
Syntax-Diagramme ... 138
18 Natural für Attached-Debugging vorbereiten ... 143
Einleitung ... 144
Voraussetzungen für Attached Debugging ... 144
Beispiel für z/OS Batch ... 145
II Dienstprogramme gruppiert nach Verwendungszweck ... 147
19 Dienstprogramme gruppiert nach Verwendungszweck ... 149
III Utility-Aktivierung ... 151
20 Utility-Aktivierung ... 153
IV ADACALL Utility ... 155
21 ADACALL Utility ... 157
ADACALL aufrufen ... 158
ADACALL-Parameter ... 159
ADACALL-Kommandos und PF-Tasten ... 161
User Exit ADAEXIT ... 164
V SYSBPM Utility ... 165
22 Allgemeines zum Dienstprogramm SYSBPM ... 167
23 SYSBPM aufrufen und benutzen ... 169
SYSBPM aufrufen und beenden ... 170
Online-Hilfe ... 172
SYSBPM Hauptmenü - Felder, Funktionen und Kommandos ... 173
SYSBPM in einer z/OS-Parallel-Sysplex-Umgebung ... 176
24 Objekte auflisten ... 179

Natural-Objekte auflisten 180
Meldungen auflisten 192
25 Objekte löschen 197
Objekte aus dem Buffer Pool und/oder dem BP Cache löschen 198
Objekte aus dem Message Pool löschen 198
26 Verzeichnisinformationen anzeigen 199
Felder für Buffer-Pool-Objekte 200
Felder für BP Cache-Objekte 202
PF-Tasten und Direktkommandos 203
27 Hexadezimale Anzeige 205
PF-Tasten und Direktkommandos 206
28 In Arbeitsdatei schreiben 209
29 Sortierten Auszug anzeigen 211
30 Buffer Pool-Statistiken anzeigen 213
Allgemeine Buffer Pool-Statistiken anzeigen 214
Buffer Pool Load/Locate-Statistiken 217
Buffer Pool-Fragmentierung 222
Nutzung der internen Funktionen 223
Buffer Pool Hash Table-Statistiken 224
Performance-Informationen anzeigen 226
PF-Tasten und Direktkommandos für die Buffer Pool-Statistiken 231
31 BP Cache-Statistiken anzeigen 233
Allgemeine BP Cache-Statistiken anzeigen 234
BP Cache Call-Statistiken 236
BP Cache Hash Table-Statistiken anzeigen 238
Performance-Hinweise 239
PF-Tasten und Direktkommandos für die BP Cache-Statistiken 239
32 Message Buffer Pool-Statistiken anzeigen 241
33 Buffer Pool auswählen 243
Funktion Select Buffer Pool aufrufen 244
Buffer Pools anzeigen 245
Buffer Pool zurücksetzen 245
34 Message Buffer Pool auswählen 247
Funktion Select Message Pool aufrufen 248
Message Pools anzeigen 249
Message Pool zurücksetzen 249
35 Sperrliste verwalten 251
Funktion Maintain Blacklist aufrufen 252
Objektsets auflisten 256
Objektset bearbeiten 257
Objektset zur Sperrliste hinzufügen 259
Objektset aus Sperrliste löschen 260
Quellobjekt des Objektsets löschen 261
Zusätzliche Objektset-Verwaltung mittels Dienstprogrammen (Utilities) 261

Sperrliste im Batch-Modus verwalten ... 262
36 Preload-Liste verwalten ... 265
Preload-Listen auflisten ... 266
Preload-Liste bearbeiten ... 267
Preload-Liste generieren ... 271
Preload-Liste löschen ... 273
Zusätzliche Verwaltungsfunktionen mit Dienstprogrammen ... 274
37 Performance-Aspekte ... 275
Interne Fast-Locate-Tabelle ... 276
Suche in Steplibs ... 277
Wiederverwendung und Beibehaltung von Objekten ... 278
Lokaler oder globaler Buffer Pool? ... 279
38 SYSBPM-Direktkommandos ... 281
39 Batch-Verarbeitung ... 289
Verwandte Themen ... 290
40 Anwendungsprogrammierschnittstellen ... 293
VI DBLOG Utility - Datenbankaufrufe protokollieren ... 295
41 DBLOG ausführen ... 297
Grundprinzipien der Datenbankprotokollierung ... 298
Verarbeitung und Speicherung der Daten ... 299
DBLOG aktivieren und deaktivieren ... 300
DBLOG mit Auswahlkriterien benutzen ... 302
42 DBLOG-Menü ... 305
DBLOG-Menü-Funktionen ... 306
Einschränkungen für das Protokollieren angeben ... 308
Adabas-Puffer angeben ... 308
43 DBLOG Trace-Bildschirm ... 311
DBLOG Trace-Bildschirm für Adabas-Kommandos ... 312
DBLOG Trace-Bildschirm für SQL-Statements ... 319
DBLOG Trace-Bildschirm für VSAM-Aufrufe ... 323
44 DBLOG Snapshot-Funktion ... 327
Snapshot-Funktion für Adabas-Kommandos ... 328
Snapshot-Funktion für SQL-Statements ... 330
Snapshot-Funktion für VSAM-Aufrufe ... 332
45 TEST DBLOG -Kommando ... 335
Syntax-Diagramme ... 336
Erklärung der Schlüsselwörter ... 337
VII INPL Utility ... 339
46 INPL Utility ... 341
Einführung in das Dienstprogramm INPL ... 342
Load Libraries Only - Nur Libraries laden ... 348
Load DDMs Only - Nur DDMs laden ... 349
Load Error Messages Only - Nur Fehlermeldungen laden ... 349
Load All Objects - Alle Objekte laden ... 350
Replace Product Installation - Produktinstallation ersetzen ... 350

Scan INPL File - INPL-Datei scannen 351
Natural Security Recover - Initialisierung der Natural Security-Umgebung erzwingen 352
User-Exit-Routinen 353
VIII NATPAGE Utility 355
47 NATPAGE Utility 357
IX NATRJE Utility 359
48 NATRJE Utility 361
Allgemeine Informationen zu NATRJE 362
NATRJE aus einem Natural-Programm aufrufen 363
NATRJE-Rückgabecodes 365
X Natural Profiler und Code Coverage 367
49 Profil-Erstellung für Natural-Anwendungen (Profiling) 369
Einführung in die Profil-Erstellung (Profiling) 370
Schnellstart für das Profiling 370
Plattformspezifische Profil-Erstellung 373
Werkzeuge zur Profil-Erstellung (Natural Profiling Tools) 373
Natural Profiler-Auswertungen 376
50 Codeabdeckung von Natural-Anwendungen (Code Coverage) 379
Einführung in die Codeabdeckung 380
Schnellstart für die Codeabdeckung 381
Grundlegende Konzepte für die Codeabdeckung 383
Plattformspezifische Codeabdeckung 384
Tools zur Messung der Codeabdeckung 385
Natural Code Coverage-Auswertungen 388
51 Grundlegendes Konzept der Profiler Utility 393
Profiler Utility - Übersicht 394
Datensammlung (Data Collection) im Batch-Modus 398
Datenkonsolidierung, Codeabdeckung und Datenverarbeitung 399
Stichprobentechnik (Natural Profiler Sampling) 403
Profiler-Performance im Batch-Modus 406
Langlaufende Anwendungen profilieren 407
Verwandte Themen 411
52 Profiler Utility im Online-Modus benutzen 413
Voraussetzungen 414
Profiler Utility im Online-Modus aufrufen und beenden 414
Ereignisse (Events) 415
Funktionen 416
53 Profiler Utility im Batch-Modus benutzen 429
Voraussetzungen für Profiler Utility im Batch-Modus 430
Profiler Utility aufrufen und beenden 432
Syntax und Schlüsselwörter 433
Gesammelte Ereignisse und Daten 437
Profilierung initialisieren 442
Initialisierung der Codeabdeckung (Code Coverage) 447

Datensammlung starten und anhalten ... 450
Filter zur Begrenzung der gesammelten Daten benutzen ... 453
Stichprobennahme (Sampling) aktivieren ... 459
Benutzerdefinierte Ereignisse schreiben ... 460
Monitor-Session CMPRMIN ... 461
Monitor-Session CMSYNIN ... 463
Batch-Natural RPC-Server profilieren ... 464
Profil-Erstellung einer Großrechner-Session aus Natural Studio ... 466
Ereignisdaten konsolidieren ... 467
Ereignisdaten auswerten ... 471
Profiler-Ressourcendateien verwalten ... 492
Profiler-Eingaben aus Natural-Textobjekten einfügen ... 496
Ereignisverfolgung (Event-Trace) ... 497
Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung ... 499
Interner Trace ... 502
Profiler-Statistikdaten ... 505
54 Natural Profiler Rich GUI-Schnittstelle ... 515
XI Recording Utility ... 519
55 Recording Utility ... 521
Verwendungszweck ... 522
Aufgezeichnete Daten und Funktionen ... 523
Aufzeichnen einer Session ... 523
Wiedergabe einer Aufzeichnung ... 524
Manipulieren einer Aufnahme ... 527
XII SYSAPI Utility ... 529
56 SYSAPI Utility ... 531
Voraussetzungen ... 532
SYSAPI Utility aufrufen und beenden ... 532
Reservierte Schlüsselwörter ... 533
SYSAPI Utility benutzen ... 533
XIII SYSCP Utility ... 537
57 SYSCP Utility ... 539
SYSCP Utility aufrufen und beenden ... 540
Codepages verwalten - Menü: Code Page Maintenance ... 542
Alle Codepages anzeigen - Funktion: All Code Pages ... 557
Geladene Codepages - Funktion: Loaded Code Pages ... 563
Unicode-Eigenschaften - Funktion: Unicode Properties ... 564
ICU Information ... 565
XIV SYSEDT Utility ... 567
58 SYSEDT Utility ... 569
Library-Profil für Natural Security definieren ... 570
SYSEDT Utility aufrufen und eine Funktion ausführen ... 570
General Information - Allgemeine Informationen ... 572
Generation Parameters - Generierungs-Parameter ... 574
Users - Benutzer ... 575

Logical Files - Logische Dateien ... 576
Recovery Files - Wiederherstellungsdateien ... 577
Administration Facilities - Verwaltungsfunktionen ... 577
Hilfe zu Direktkommandos und Menüfunktionen ... 578
XV SYSERR Utility ... 581
59 Allgemeine Informationen zu Meldungen ... 583
Meldungstypen ... 585
Meldungssprachen ... 586
Meldungen und Codepage-Unterstützung ... 586
Ausgabe von Meldungen ... 586
Abrufen von Natural-System-Kurzmeldungen ... 587
Abrufen von benutzerdefinierten Kurzmeldungen ... 588
Informationen zur Meldung erhalten ... 588
60 SYSERR aufrufen ... 591
SYSERR ausführen ... 592
61 Funktionen ... 595
Meldungen anlegen ... 596
Meldungen löschen ... 600
Meldungen anzeigen ... 601
Meldungen ändern ... 602
Meldungen drucken ... 605
Meldungen durchsuchen ... 607
Meldungen aus Liste auswählen ... 610
Meldungen in andere Sprachen übersetzen ... 613
62 Parameter ... 617
Meldungstyp ... 618
Library ... 618
Meldungsnummer ... 618
Sprachcodes ... 619
63 Direktkommandos ... 621
64 Umwandlung in Großbuchstaben - ERRUPPER ... 623
65 Zeichen ersetzen - ERRCHAR ... 625
66 Meldungen in verschiedenen Libraries verwalten ... 627
Meldungen entladen - ERRULDUS ... 628
Meldungen laden - ERRLODUS ... 630
XVI SYSEXT Utility ... 631
67 SYSEXT Utility ... 633
Einleitung zu SYSEXT ... 634
SYSEXT Utility aufrufen und beenden ... 635
SYSEXT Utility benutzen ... 638
Interface-Versionen ... 643
Reservierte Schlüsselwörter ... 644
Eine Natural API verwenden ... 644
XVII SYSEXV Utility ... 647
68 SYSEXV Utility ... 649

SYSEXV starten und beenden ... 650
Untermenüs und Beispielprogramme aufrufen und beenden ... 652
Suchbegriff eingeben ... 652
SYSEXV-PF-Tasten ... 653
SYSEXV-Zeilenkommandos ... 653
Kategorie auswählen ... 654
Version auswählen ... 655
Member auswählen ... 657
XVIII SYSMAIN Utility ... 659
69 Allgemeine Informationen zu SYSMAIN ... 661
Grundsätzliches zu den SYSMAIN-Funktionen ... 662
Objekttypen und Speicherort ... 663
Übersicht über die Funktionen ... 663
70 SYSMAIN aufrufen und beenden ... 665
SYSMAIN online oder im Batch aufrufen ... 666
SYSMAIN mittels Anwendungsprogrammierschnittstelle aufrufen ... 667
SYSMAIN beenden ... 668
71 Menüfunktionen und Kommandos benutzen ... 671
Menüfunktionen ausführen ... 672
Kommandos ausführen ... 675
Beschreibung der Funktionen ... 677
Funktionsverarbeitung und Reporting ... 685
SYSMAIN-Online-Hilfe ... 690
72 Programmierobjekte bearbeiten ... 693
Felder in den Menüs für Programmierobjekte ... 694
Profilparameter RECAT verwenden ... 697
Auswahllisten für Programmierobjekte ... 697
Besonderheiten bei XRef ... 702
Angabe zusätzlicher Kriterien ... 704
Direktkommando-Syntax für Programmierobjekte ... 706
73 Debug-Umgebungen bearbeiten ... 709
Felder im Menü Debug Environments ... 710
Auswahllisten für Debug-Umgebungen ... 712
Direktkommando-Syntax für Debug-Umgebungen ... 713
74 Fehlermeldungen bearbeiten ... 717
Felder in den Menüs für Fehlermeldungen ... 718
Auswahllisten für Fehlermeldungen ... 720
Fehlermeldungen neu nummerieren ... 722
Sprachen angeben ... 723
Direktkommando-Syntax für Fehlermeldungen ... 723
75 Profile bearbeiten ... 727
Felder im Menü Profiles ... 728
Auswahllisten für Profile ... 729
Direktkommando-Syntax für Profile ... 731
76 Rules bearbeiten ... 735

Felder im Menü Rules ... 736
Auswahllisten für Rules ... 738
Direktkommando-Syntax für Rules ... 739
77 DDMs bearbeiten ... 743
Felder im Menü DDMs ... 744
Auswahllisten für DDMs ... 746
Direktkommando-Syntax für DDMs ... 747
78 Predict Sets bearbeiten ... 751
Felder im Menü Predict Sets ... 752
Auswahllisten für Predict Sets ... 753
Direktkommando-Syntax für Predict Sets ... 754
79 Schlüsselwörter und Variablen in Direktkommandos ... 759
Beschreibung der Schlüsselwörter ... 760
Angabe eines Namensbereichs ... 772
80 An SYSMAIN absetzbare spezielle Kommandos ... 775
81 Statusmeldungen und Fehlerbenachrichtigungen bearbeiten ... 779
Ablehnung von Objekten und Ursache ... 780
Status-Meldungen ... 782
SYSMAIN-Fehlerbenachrichtigung ... 786
82 Besondere Hinweise für Administratoren ... 791
File Security ... 792
Natural Security ... 793
User-Exit-Routinen ... 794
XIX SYSNCP Utility ... 801
83 SYSNCP Utility ... 803
Einführung in das Dienstprogramm SYSNCP ... 804
Dienstprogramm SYSNCP aufrufen ... 812
Auswahl des Prozessors ... 813
Header-Daten ... 814
Verwaltung der Schlüsselwörter ... 826
Verwaltung der Funktionen ... 832
Laufzeit-Aktionen ... 838
Kommandoprozessor katalogisieren ... 843
Administrative SYSNCP-Funktionen ... 844
Sitzungsprofil ... 851
XX Object Handler ... 855
84 Allgemeine Informationen zum Object Handler ... 857
Grundsätzliches zur Objektübertragung ... 858
Object Handler aufrufen ... 860
Batch- oder Direktkommandoaufrufe ... 862
Object Handler-Kommandos aus einem Natural-Programm heraus absetzen ... 863
Text-Objekte für Report-, Restart- und Trace-Daten ... 863
Natural Security ... 864
Standard-PF-Tasten ... 864

85 Funktionen 867
86 Assistenten (Wizards) 869
Schritt 1 - Vorgang starten 870
Schritt 2 - Objekte in/aus Arbeitsdateien entladen/laden/durchsuchen 871
Step 3 - Parameter festlegen 873
Schritt 4 - Objekte auswählen 874
Schritt 5 - Verarbeitung ausführen 876
Schritt 6 - Verarbeitung fortsetzen 876
87 Expertenmodus (Advanced User) 879
Objekte verarbeiten 880
88 Kompaktmodus (Compact Mode) 883
Wie Sie den Kompaktmodus wählen 884
Wie Anweisungen im Kompaktmodus verarbeitet werden 884
89 Ladefunktion erneut starten (Restart Load) 889
90 Objekte in der Natural-Umgebung sichten (View) 891
Natural-Library-Objekte 892
Natural-Systemfehlermeldungen 893
Natural-Kommandoprozessor-Quellcodes 894
FDTs 895
Natural-zugehörige Objekte (Natural Related-Objects) 896
DDMs 897
91 Objekte suchen (Find) 899
Suchfunktion aufrufen 900
92 Administration (Arbeitspläne verwalten) 901
Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek 903
Neuen Arbeitsplan anlegen (Create a new Workplan) 905
Arbeitsplan-Bibliothek ändern (Change Workplan Library) 907
Report-Bibliothek ändern (Change Report Library) 908
93 Systemdatei auswählen (Select System File) 911
94 Bibliothek auswählen (Select Library) 913
95 Systemfehlermeldungen auswählen (Select System Error Messages) 915
Spalten und Kommandos 916
96 Objekte auswählen (Select Objects) 919
Spalten und Kommandos in List-Bildschirmen 920
97 Objekte angeben 925
98 Objektangabe - Alle Objekte in der Arbeitsdatei laden/durchsuchen 927
99 Objektangabe - Natural-Library-Objekte entladen/laden/durchsuchen 929
Natural-Library-Objekte 930
Natural-Library-Objekte - weitere Auswahlkriterien angeben 931
Natural-Library-Objekte - Eigenschaften angeben 934
Natural-Library-Objekte - Ausnahmen angeben 935
Natural-Library-Objekte - Eigenschaften der Ausnahmen für Natural-Library-Objekte 936
100 Objektangabe - Natural Systemfehlermeldungen 939
Natural-Systemfehlermeldungen 940

Natural-Systemfehlermeldung, weitere Auswahlkriterien angeben 940
Natural-Systemfehlermeldungen, Ausnahmen angeben 941
101 Objektangabe - Natural-Kommandoprozessoren 943
Natural-Kommandoprozessoren entladen/laden/durchsuchen 944
Natural-Kommandoprozessor-Quellcodes, Ausnahmen angeben 945
102 Objektangabe - Natural-zugehörige Objekte 947
Natural-Profile 948
Natural-Debug-Umgebungen 949
103 Objektangabe - Datendefinitionsmodule (DDMs) 953
DDMs entladen/laden/durchsuchen 954
DDM-Eigenschaften angeben 955
DDM-Ausnahmen angeben 956
104 Objektangabe - Felddefinitionstabellen (FDTs) 957
105 Arbeitsplan-Typ SELECTION oder LIST verwenden 959
106 Einstellungen (Settings) 961
Felder auf dem Settings-Bildschirm 962
Zusätzliche Optionen festlegen (Set Additional Options) 964
Globale Parameter festlegen 974
107 Arbeitspläne (Workplans) 979
Arbeitspläne erstellen, auswählen und ändern 980
Inhalt von Arbeitsplänen 980
Beispiele für Arbeitspläne 981
Arbeitspläne referenzieren 982
108 Angabe von Name, Datum und Zeit 985
Name 986
Datum 987
Zeit 989
109 Arbeitsdateien (Work Files) 991
Zuweisung von Arbeitsdateien 992
Arbeitsdateiformat 992
110 Direktkommandos 995
111 Grundlegende Kommandosyntax 997
112 select-clause 1001
Syntax der select-clause 1002
Arbeitsplan vom Typ SELECTION oder LIST 1002
Auswahl von Natural-Library-Objekten und DDMs 1003
Auswahl von Natural-Related-Debug-Umgebungen 1010
Auswahl von Natural-Related-Profilen 1012
Auswahl von Natural-Systemfehlermeldungen 1013
Auswahl von Natural-Kommandoprozessoren 1015
Auswahl von FDTs 1017
Auswahl von Anwendungen (Applications) 1018
Auswahl von Löschanweisungen für Objekte 1021
Auswahl der Hilfetextverarbeitung 1023
113 Objektliste - Arbeitsplantyp LIST 1025

Syntax von object-type-and-location 1026
Syntax von object-name-description 1027
Beispiel für eine Objektliste 1029
114 parameter-setting 1031
Syntax von parameter-clause 1032
Erläuterung der Schlüsselwörter für parameter-clause 1033
115 option-setting 1037
Syntax der option-setting-Klausel 1038
Erläuterung der Schlüsselwörter für option-setting 1040
116 Beispiele für die Verwendung von Object Handler-Direktkommandos 1049
Entladen von Objekten für dieselbe Plattform 1050
Entladen von Objekten für verschiedene Plattformen 1051
Laden von Objekten im internen Format 1052
Laden von Objekten im Transferformat 1052
117 Kommandos für die Navigation und für spezielle Funktionen 1053
118 Batch Condition Codes und User Exit Routinen 1059
Im Batch-Modus zurückgegebene Condition Codes 1060
User Exit Routinen anwenden 1060
Verfügbare User Exit Routinen 1061
119 Profileinstellungen (Profile Settings) 1063
PF-Tasten 1065
Zeilenkommandos 1065
Profilparameter 1066
120 Werkzeuge (Tools) 1073
Status 1074
Letztes Ergebnis (Last Result) 1074
Aufzeichnungen (Traces) 1074
Reports 1075
121 Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler 1077
Konvertierung einzelner Kommandos 1078
Verarbeitung von Kommandos mit einer User Exit Routine 1080
Verarbeitung von SYSTRANS-Kommandos mit OBJHAPI 1080
Nicht unterstützte SYSTRANS-Optionen 1081
XXI SYSPARM Utility 1083
122 SYSPARM Utility 1085
Utility SYSPARM aufrufen 1086
Profile auflisten - Funktion List Profiles 1087
Profil anzeigen - Funktion Display Profile 1089
Neues Profil anlegen - Funktion Add New Profile 1090
Profil ändern - Funktion Modify Profile 1090
Profile bearbeiten - Editier-Funktion 1091
Profil kopieren - Funktion Copy Profile 1095
Profil löschen - Funktion Delete Profile 1095
Direktkommandos und Batch-Modus 1096

Parameterprofile in unterschiedlichen Umgebungen verwalten 1106
Hilfe zu Parametern aus der Kommandozeile aufrufen 1106
XXII SYSRDC Utility 1107
123 SYSRDC Utility 1109
Funktionsbausteine von SYSRDC 1110
Datenerfassung auslösende Ereignisse 1110
Gesammelte Daten 1112
Natural Data Collector aktivieren und konfigurieren 1114
Trace Recording-Funktion 1115
User Exits für externes Monitoring/Accounting 1116
Aufruf des CMRDC-Interface 1118
Beispiel-Programme 1122
XXIII SYSRPC Utility 1127
124 SYSRPC Utility aufrufen und beenden 1129
SYSRPC Utility aufrufen 1130
SYSRPC Utility beenden 1131
Online-Hilfe aufrufen 1131
125 Funktion Service Directory Maintenance benutzen 1133
Service Directory-Konzept 1134
Funktion Service Directory Maintenance aufrufen 1135
Felder auf dem Service Directory-Bildschirm 1137
Kommandos zur Pflege des Service Directory 1140
API USR8216N zur Pflege des Service Directory 1144
126 Elemente im Service Directory ersetzen 1145
Syntax des Direktkommandos SYSRPC SM REPLACE 1146
127 Interface-Objekte generieren - Allgemeine Aspekte 1149
128 Einzelne Interface-Objekte mit Parameterangabe generieren 1151
Funktion Interface Object Generation benutzen 1152
Parameter angeben 1154
Beispiele für die Generierung von Interface-Objekten 1157
129 Mehrere Interface-Objekte generieren 1161
Direktkommando SYSRPC SGMASS benutzen 1162
Name und Komprimierungsart angeben 1164
130 Benötigte Puffergröße berechnen 1167
Direktkommando SYSRPC CSMASS benutzen 1168
Namensangabe für Größenberechnung und Komprimierung 1170
131 Parameter verwalten (Parameter Maintenance) 1171
Funktion Parameter Maintenance aufrufen 1172
NTRPC/RPC-Schlüsselwort-Subparameter angeben 1172
132 Server-Kommandos ausführen (Server Command Execution) 1175
Funktion Server Command Execution benutzen 1176
RPC Server anpingen 1179
RPC-Server beenden 1181
Kommandos für RPC-Server-Tasks ausführen 1183
133 Beim EntireX Broker registrierte Server auflisten 1187

Beispiel für ein SYSRPC SRVLIST-Direktkommando ... 1189
Server-Liste anzeigen ... 1189
Zusätzliche Server-Informationen anzeigen ... 1190
Server-Listen anpassen ... 1191
134 Übersicht über die SYSRPC-Direktkommandos und -Batch-Kommandos ... 1193
XXIV SYSTP Utility ... 1195
135 SYSTP aufrufen und Funktionen ausführen ... 1197
136 SYSTP-Bildschirme benutzen ... 1199
137 Allgemeine SYSTP-Funktionen ... 1201
Natural Monitoring (SYSMON) ... 1202
Natural Print/Work Files (SYSFILE) ... 1206
Buffer Usage Statistics (BUS) ... 1207
Permanente Speichersegmente ... 1209
Natural-Subsysteme und Roll Server-Informationen anzeigen ... 1210
Natural Thread-Nutzungsstatistik ... 1211
Natural-Lizenzinformationen ... 1214
138 SYSTP-Funktionen unter CICS ... 1217
Natural-Benutzersitzungen ... 1218
Natural Roll Facilities ... 1223
Natural-Thread-Gruppen ... 1225
Natural-Speicher-Threads ... 1226
Globale Systeminformationen zum Natural CICS Interface anzeigen ... 1227
Natural/CICS-Generierungsoptionen ... 1229
Natural Thread-Gruppendefinitionen ... 1229
Eigene Natural-Benutzersitzung ... 1230
CICS Task Information ... 1231
System Administration Facilities ... 1231
139 SYSTP-Funktionen unter IMS TM ... 1233
Nachricht an eine Benutzergruppe verwalten ... 1234
Umgebungsdaten anzeigen ... 1234
Monitoring ... 1235
Angewandte NII-Zaps ... 1235
140 SYSTP im Batch-Modus für CICS-Sitzungen ... 1237
SYSTP im Batch-Modus aufrufen ... 1238
Protokolldatei auswerten ... 1238
Stichwortverzeichnis ... 1241

Debugger und Dienstprogramme (Utilities)

Dieses Dokument enthält die Beschreibungen für den Natural-Debugger und die Natural-Dienstprogramme (Utilities).

Natural-Debugger	
Natural Debugger	Mit dem Natural Debugger können Sie Programmfehler erkennen, auffinden und korrigieren, die Programmausführung testen oder optimieren und einen Natural-Fehler analysieren, der die Programmausführung unterbricht.
Natural-Dienstprogramme (Utilities)	
Dienstprogramme gruppiert nach Verwendungszweck	Liste *aller* Dienstprogramme/Utilities sortiert nach Verwendungszweck.
Utility-Aktivierung	Allgemeine Informationen zur Utility-Aktivierung durch Natural und zur Benutzung von Utilities mit/ohne Natural Security.
ADACALL	Mit dem Dienstprogramm ADACALL können Sie Adabas-Direktaufrufe (native Kommandos) direkt an eine Adabas-Datenbank absetzen.
DBLOG	Mit dem Dienstprogramm DBLOG können Sie Adabas-Kommandos, SYNC/ROLB-Aufrufe, SQL-Statements oder VSAM-Aufrufe protokollieren.
INPL	Mit dem Dienstprogramm INPL können Sie Natural-Objekte aus von uns gelieferten Datasets (Dateien) laden und scannen.
NATPAGE	Mit dem Dienstprogramm NATPAGE können Sie während einer Natural-Session Bildschirmausgabedaten (Masken und Reports) aufzeichnen.
NATRJE	Das Dienstprogramm NATRJE (Natural Remote Job Entry) zur Jobfernverarbeitung kann verwendet werden, um JCL-Karten aus einem Natural-Programm an das Betriebssystem zur Zeitplanung und Ausführung zu übergeben.
Natural Profiler und Code Coverage	Mit diesem Dienstprogramm können Sie den internen Programmablauf einer Natural-Anwendung überwachen und die Performance und Codeabdeckung der Anwendung analysieren.
Object Handler	Verarbeitet Natural- und Nicht-Natural-Objekte zur Verteilung in Natural-Umgebungen. Dies geschieht, indem die Objekte in der Quellumgebung in Arbeitsdateien entladen und aus den Arbeitsdateien in die Zielumgebung geladen werden.
Recording Utility	Mit der Recording Utility können Sie während einer Natural-Session eingegebene Kommandos und Eingabedaten aufzeichnen und eine aufgezeichnete Sitzung wieder ausführen.
SYSAPI	Mit dem Dienstprogramm SYSAPI können Sie Anwendungsprogrammierschnittstellen (Application Programming Interfaces/APIs), die von Natural Add-on-Produkten zur Verfügung gestellt werden, finden und testen.

SYSBPM	Das Dienstprogramm SYSBPM dient zur Verwaltung lokaler und globaler Buffer Pools des Typs Natural oder SORT sowie von Message Pools.
SYSCP	Mit dem Dienstprogramm SYSCP können Sie sich Informationen zu Codepages anzeigen lassen und Codepages für Natural-Source-Objekte verwalten.
SYSDDM	Siehe *DDM-Editor (SYSDDM Utility)* in der *Editoren*-Dokumentation.
SYSEDT	Das Dienstprogramm SYSEDT bietet Verwaltungsfunktionen für den Editor Buffer Pool, der als Datencontainer für den Software AG Editor dient.
SYSERR	Mit dem Dienstprogramm SYSERR können Sie anwendungsspezifische Meldungen anlegen und verwalten. Außerdem können Sie damit die Texte der existierenden Natural-Systemmeldungen ändern (nicht empfohlen).
SYSEXT	Mit dem Dienstprogramm SYSEXT können Sie Natural-Anwendungsprogrammierschnittstellen (Application Programming Interfaces/APIs), die in der aktuellen System Library SYSEXT enthalten sind, finden und testen.
SYSEXV	Das Dienstprogramm SYSEXV stellt Beispiel-Programme zur Verfügung, die die Verwendung von Natural-Funktionalität und Merkmalen veranschaulichen, die mit der aktuellen oder einer vorangegangenen Version von Natural eingeführt wurden.
SYSMAIN	Mit dem Dienstprogramm SYSMAIN können Sie Funktionen zur Pflege von Objekten ausführen, z.B. Kopieren, Verschieben, Ersetzen und Löschen.
SYSNCP	Mit dem Dienstprogramm SYSNCP können Sie kommandogesteuerte Navigationssysteme für Natural-Anwendungen definieren.
SYSPARM	Mit dem Dienstprogramm SYSPARM können Sie einen Satz Natural-Profilparameter als individuelles Parameterprofil anlegen und pflegen. Ein solches Parameterprofil können Sie für jede Natural-Session verwenden.
SYSRDC	Mit dem Dienstprogramm SYSRDC können Sie Monitoring- und Abrechnungsdaten über den internen Prozessablauf innerhalb einer Natural-Anwendung aufzeichnen und für die Auswertung von Natural-Session-Aktivitäten in externen oder Natural-Programmen verwenden.
SYSRPC	Das Dienstprogramm SYSRPC dient zum Einrichten und Pflegen von Natural RPC-Umgebungen (Remote Procedure Call).
SYSTP	Mit dem Dienstprogramm SYSTP können Sie TP-Monitor-spezifische Natural-Eigenschaften überwachen.

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.

I Debugger

Mit dem Debugger können Sie Programmfehler erkennen, finden und korrigieren, die Programmausführung testen oder optimieren und einen Natural-Fehler analysieren, der die Programmausführung unterbricht.

Debugger-Tutorial	Erste Schritte mit dem Debugger.
Debugger-Konzept	Grundkonzeption des Debuggers.
Debugger starten	Betriebsanforderungen und Anweisungen zum Aufrufen des Debuggers.
Test-Modus ein- und ausschalten	Einstellung des Test-Modus zum Aktivieren und Deaktivieren des Debuggers.
Debug Environment Maintenance	Verwaltung einer Debug-Umgebung: Speichern und Benutzen einer vordefinierten Debug-Umgebung.
Spy Maintenance	Verwaltung der Debug-Einträge: Setzen, Ändern, Löschen und Aktivieren von Haltepunkten (sowohl Breakpoints als auch Watchpoints).
Breakpoint Maintenance	Verwaltung von Breakpoints: Setzen, Ändern, Löschen und Aktivieren von Breakpoints. Erklärung der Breakpoint-Bildschirminhalte.
Watchpoint Maintenance	Verwaltung von Watchpoints: Setzen, Ändern, Löschen und Aktivieren von Watchpoints. Erklärung der Watchpoint-Bildschirminhalte.
Call Statistics Maintenance	Abrufen von Statistiken über aufgerufene Objekte.
Statement Execution Statistics Maintenance	Abrufen von Statistiken über ausgeführte Statement-Zeilen.
Variable Maintenance	Anzeigen und Ändern von Variablen.
List Object Source	Anzeigen eines Objekt-Quellcode.
Fehlerbehandlung	Behandlung von Fehlern, die während der Ausführung der Anwendung oder des Debuggers auftreten können.
Kommandos zur Ausführungssteuerung	Debugger-Kommandos für die Programmablaufsteuerung.

Kommandos zum Navigieren und Anzeigen von Informationen	Debugger-Kommandos für Bildschirmnavigation, Objektinformationen und Debugger-Profileinstellungen.
Kommandoübersicht und -syntax	Alle Debugger-Kommandos und die entsprechende Kommando-Syntax.
Natural für Attached-Debugging vorbereiten	Verwendung eines Debug-Attach-Servers, der unter NaturalONE läuft.

Anmerkungen:

1. Die Sprache der Debugger-Benutzungsoberfläche ist Englisch. Daher sind im folgenden Dokument alle zur Orientierung erforderlichen englischen Literale beibehalten und ggf. nur in Klammern übersetzt worden.
2. Terminologie: **„Breakpoints“** und **„Watchpoints“** sind Debug-Haltepunkte. **"Spies"** ist der Sammelbegriff für diese Haltepunkte. Die Funktion „Spy“ dient zur Verwaltung der verschiedenen Debug-Einträge.
3. Die in diesem Dokument verwendete Notation *vrs* bzw. *vr* steht als Platzhalter für die betreffende Produktversion (siehe auch Version im *Glossar*).

2 Debugger-Tutorial

- Voraussetzungen 8
- Grundlagen des Debugging 8
- Sitzung 1 - Analyse eines Natural-Fehlers 9
- Sitzung 2 - Einen Breakpoint benutzen 14
- Sitzung 3 - Einen Watchpoint benutzen 20
- Sitzung 4 - Verfolgen des logischen Ablaufs von Programmen 26
- Sitzung 5 - Verwendung von Statistiken über die Programmausführung 30
- Weitere Hinweise zur Benutzung des Debuggers 33
- Quellcode-Beispiele 37

In diesem Tutorial werden die grundlegenden Funktionen des Debuggers vorgestellt und verschiedene Debugging-Methoden besprochen. Das Tutorial führt Sie durch ein einfaches Szenario, das demonstriert, wie der Debugger verwendet werden kann, um Laufzeitfehler zu analysieren und die Programmausführung zu kontrollieren.

Es ist wichtig, dass Sie die Sitzungen 1 bis 5 nacheinander durcharbeiten.

Anmerkungen:

1. Der Einfachheit halber zitiert das Tutorial hauptsächlich Direktkommandos zur Demonstration der Debugger-Funktionen und nicht die alternativen Menü-Funktionen.
2. Eine vollständige Beschreibung aller Debugger-Funktionen, die in diesem Tutorial erwähnt werden, finden Sie in den anschließenden Kapiteln der *Debugger*-Dokumentation.

Voraussetzungen

- Sie sollten mit der Programmierung in Natural vertraut sein
- Bevor Sie mit Sitzung 1 beginnen, müssen Sie alle Beispielprogramme (`DEBUG1P` und `DEBUG2P`) und Subprogramme (`DEBUG1N`, `DEBUG2N`, `DEBUG3N` und `DEBUG4N`) erstellen, die im Abschnitt ***Beispiel-Quellcode*** weiter unten in diesem Tutorial vorhanden sind. Speichern und katalogisieren Sie diese Objekte mit dem Systemkommando `STOW`.

Grundlagen des Debugging

Mit dem Debugger können Sie den Ausführungsfluss eines Natural-Objekts bei einem bestimmten Debug-Event (Ereignis) unterbrechen und Informationen zum aktuellen Status des unterbrochenen Objekts, z.B. das nächste auszuführende Statement, den Wert einer Variablen oder die Hierarchie (Programmebenen) aufgerufener Objekte erhalten.

Grundsätzlich müssen Sie die folgenden zwei Hauptschritte ausführen, um die Kontrolle an den Debugger für die Programmunterbrechung zu übergeben:

1. Aktivieren Sie den Debugger mit dem Systemkommando `TEST ON`.

   Dadurch kann der Debugger die Steuerung für jedes auszuführende Statement erhalten.
2. Setzen Sie einen oder mehrere Debug-Einträge (**Breakpoints** und **Watchpoints**) für die Natural-Objekte, die ausgeführt werden sollen.

   Dadurch kann der Debugger entscheiden, wann er die Kontrolle vom Natural-Laufzeitsystem übernimmt und die Programmausführung stoppt.

Ein Natural-Fehler führt immer zur Unterbrechung des Programms.

Es ist dann keine Debug-Eingabe erforderlich, der Debugger greift automatisch ein.

Im Folgenden finden Sie eine Übersicht über alle möglichen Programmunterbrechungen:

Programmunterbrechung	Erklärung
Breakpoint	Haltepunkt: Bewirkt eine Programmunterbrechung für eine Statement-Zeile in einem Natural-Objekt. Der Debugger unterbricht das Programm, sobald die Statement-Zeile, für die ein Breakpoint gesetzt ist, ausgeführt werden soll, d.h., *bevor* das in dieser Zeile enthaltene Statement verarbeitet wird.
Watchpoint	Haltepunkt: Bewirkt eine Programmunterbrechung für eine Variable in einem Natural-Objekt. Der Debugger unterbricht das Programm, sobald sich der Inhalt der Variablen, für die ein Watchpoint gesetzt ist, geändert hat, d.h., *nachdem* das Statement, das diese Variable referenziert, verarbeitet wird.
Step-Modus	Einzelschritt-Modus: Das Objekt wird schrittweise während der Programmausführung durchlaufen. Der Step-Modus wird durch ein Debugger-Kommando eingeleitet. Dazu muss der Debugger zuvor wegen eines Breakpoint- oder Watchpoint-Haltepunkts die Kontrolle erhalten haben. Im Step-Modus unterbricht der Debugger die Programmausführung, *bevor* ein ausführbares Statement, das in diesem Objekt enthalten ist, verarbeitet wird.
Natural-Fehler	Bewirkt eine automatische Programmunterbrechung.

Sitzung 1 - Analyse eines Natural-Fehlers

Diese Sitzung beschreibt Untersuchungsmethoden für einen Natural-Fehler, der während der Programmausführung auftritt.

Um einen Natural-Fehler zu simulieren:

- Führen Sie an der Eingabeaufforderung `NEXT` das Beispielprogramm `DEBUG1P` aus.

  Es erscheint folgende Natural-Meldung: `DEBUG1N 0180 NAT0954 Abnormal termination S0C7 during program execution.` (Abnormale Beendigung S0C7 während der Programmausführung).

  Die Meldung verweist auf die Zeile 180 im Subprogramm `DEBUG1N`: `BONUS := SALARY * PERCENT / 100`. Dies zeigt an, dass von einer oder mehreren der referenzierten Variablen falsche Werte zurückgegeben werden. Dies ist jedoch noch kein eindeutiger Hinweis darauf,

was das Problem tatsächlich verursacht, und es könnte schwierig sein, die Ursache zu ermitteln, wenn die Variablenwerte aus einer Datenbank abgerufen wurden, so wie es typisch ist für Mitarbeiter-Datensätze (Employees Records).

Um den Debugger für die weitere Untersuchung des Problems zu aktivieren:

1 Geben Sie an der Eingabeaufforderung `NEXT` Folgendes ein:

```
TEST ON
```

Die Meldung `Test mode started.` (Test-Modus gestartet) zeigt an, dass der Debugger aktiviert ist.

Anmerkung: `TEST ON` bleibt aktiv für die Dauer der aktuellen Sitzung oder solange, bis Sie `TEST OFF` eingeben, um den Debugger zu deaktivieren.

2 Führen Sie an der Eingabeaufforderung `NEXT` wieder das Beispielprogramm `DEBUG1P` aus.

Das Fenster **Debug Break** erscheint (Beispiel):

```
+------------------- Debug Break --------------------+
| Break by ABEND SOC7 at NATARI2+2A4-4 (NAT0954)     |
| at line  180 in subprogram DEBUG1N (level 2)       |
| in library DEBUG    in system file (10,32).        |
|                                                    |
|          G   Go                                    |
|          L   List break                            |
|          M   Debug Main Menu                       |
|          N   Next break command                    |
|          R   Run (set test mode OFF)               |
|          S   Step mode                             |
|          V   Variable maintenance                  |
|                                                    |
| Code .. G                                          |
|                                                    |
| Abnormal termination SOC7 during program execution|
| PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS    |
+----------------------------------------------------+
```

Weil ein Natural-Fehler auftritt, greift der Debugger automatisch ein und zeigt das Fenster **Debug Break**.

Am oberen Rand des Fensters werden zusätzliche Informationen dazu angezeigt, wo der Fehler auftritt: Das Modul (`NATARI2`) im Natural-Nucleus (nützlich für unseren Support), der Objekttyp (`subprogram`), die Library (`DEBUG`) und die Datenbankkennung und Dateinummer (`10,32`) der Systemdatei.

Außerdem sind im Fenster **Debug Break** Debugger-Funktionen verfügbar, die Sie benutzen können, um z.B. die Programmausführung fortzusetzen (**Go** oder **Run**), das Debugger-

Hauptmenü (**Debug Main Menu**) aufzurufen oder den Step-Modus zu aktivieren. Sie können dazu entweder den entsprechenden Kennbuchstaben im Feld **Code** eingeben und Enter drücken oder die entsprechende PF-Taste drücken.

Um die fehlerhafte Statement-Zeile zu untersuchen:

- Ersetzen Sie im Feld **Code** den Standardeintrag `G` durch `L`, um die Funktion **List break** auszuführen.

  Der Quellcode des Beispielprogramms `DEBUG1N` wird angezeigt:

```
13:48:54              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                 - List Object Source -              Object DEBUG1N
                                                                  Bottom of data
Co Line Source                                                      Message
__ 0070   2 NUMCHILD  (N2)                                       |
__ 0080   2 ENTRYDATE (D)                                        |
__ 0090   2 SALARY    (P7.2)                                     |
__ 0100   2 BONUS     (P7.2)                                     |
__ 0110 LOCAL                                                    |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>                |
__ 0130 1 DIFFERENCE  (P3.2)                                     |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                          |
__ 0150 END-DEFINE                                               |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365             |
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | last line
__ 0180   BONUS := SALARY * PERCENT / 100                        | * NAT0954 *
__ 0190 END-IF                                                   |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA |
__ 0210 END                                                      |

Command ===>



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

  Die Meldung `last line` (letzte Zeile) zeigt an, dass das in Zeile 170 enthaltene Statement das letzte Statement ist, das erfolgreich ausgeführt wurde.

  Das Statement in Zeile 180, welches das Problem verursacht, ist hervorgehoben und mit der Anmerkung `* NAT0954 *` markiert.

  Dies zeigt an, dass der Fehler durch den Inhalt der Variablen `SALARY` (Gehalt) oder `PERCENT` (Prozent) verursacht wird. Höchstwahrscheinlich handelt es sich um `SALARY`, da `PERCENT` korrekt initialisiert ist.

Um den Inhalt der Variablen SALARY **zu prüfen:**

1 Geben Sie Folgendes in der Kommandozeile ein:

```
DIS VAR SALARY
```

Der Bildschirm **Display Variable** wird für die Variable SALARY angezeigt (Beispiel):

```
18:59:51              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON           - Display Variable (Alphanumeric) -       Object DEBUG1N


Name ...... EMPLOYEE.SALARY
Fmt/Len ... P 7.2
Type ...... parameter
Index .....
Range .....

Position ..
Contents ..


Command ===>


Variable contains invalid data.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Mod   Flip             Li Br Alpha Hex   Canc
```

Die Meldung `Variable contains invalid data.` (Variable enthält ungültige Daten) zeigt an, dass der Inhalt der Variablen, die scheinbar leer ist, nicht dem Format der Variablen entspricht. Dies wird deutlich, wenn Sie, wie im folgenden Schritt beschrieben, die hexadezimale Darstellung des Variableninhalts betrachten.

2 Drücken Sie PF11 (Hex), um den hexadezimalen Inhalt der Variablen anzuzeigen.

Der Bildschirm sieht nun ähnlich aus wie im folgenden Beispiel:

```
11:13:33              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON           - Display Variable (Hexadecimal) -        Object DEBUG1N


Name ..... EMPLOYEE.SALARY
Fmt/Len ... P 7.2
Type ..... parameter
Index .....
Range .....

Position ..
Contents .. 4040404040


Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Mod   Flip             Li Br Alpha Hex   Canc
```

Der hexadezimale Wert zeigt, dass die Variable nicht in gepacktem numerischem Format vorliegt, was folglich bei der Programmausführung zu einem Rechenfehler führt. Das Programm `DEBUG1P` übergibt dem Subprogramm `DEBUG1N` einen inkorrekten Wert für die Variable `SALARY`.

Tipp: Sie können `PF10` (Alpha) drücken, um wieder zur alphanumerischen Darstellung zurück zu wechseln.

3 Geben Sie in der Kommandozeile Folgendes ein:

```
GO
```

Das Kommando `GO` bewirkt, dass die Kontrolle vom Debugger an das Natural-Laufzeitsystem zurückgegeben und dass die Ausführung des Programms bis zum Programmende oder bis zum nächsten Debug-Ereignis (Event) fortgesetzt wird. Im vorliegenden Fall gibt es kein zusätzliches Debug-Ereignis. Die Eingabeaufforderung `NEXT` erscheint mit der bekannten Natural-Fehlermeldung (**s.o.**).

Um die Variable `SALARY` im Objekt-Quellcode zu korrigieren:

1 Öffnen Sie das Programm `DEBUG1P` mit dem Programm-Editor und entfernen Sie das für `SALARY := 99000` eingegebene Kommentarzeichen (*).

2 Benutzen Sie das Systemkommando `STOW`, um das geänderte Programm zu speichern und zu katalogisieren.

3 Führen Sie das Programm `DEBUG1P` aus.

Der Debugger unterbricht das Programm nicht, obwohl immer noch `TEST ON` gesetzt ist. Das Programm wird erfolgreich ausgeführt und gibt einen Bericht aus:

```
Page      1                                                  07-09-06  15:28:06

EMPLOYEE RECEIVES:   100800.00
    PLUS BONUS OF:     3465.00




NEXT                                                                 LIB=DEBUG
```

Sitzung 2 - Einen Breakpoint benutzen

Sie können die Programmausführung an einer bestimmten Statement-Zeile unterbrechen, indem Sie einen Breakpoint (Haltepunkt) für diese Zeile setzen.

Um für eine Zeile im Subprogramm `DEBUG1N` einen Breakpoint zu setzen:

1 Geben Sie Folgendes an der Eingabeaufforderung `NEXT` ein:

```
TEST SET BP DEBUG1N 170
```

Die Meldung `Breakpoint DEBUG1N0170 set at line 170 of object DEBUG1N.` erscheint. Sie bestätigt, dass für die Statement-Zeile 170 im Subprogramm `DEBUG1N` ein Breakpoint mit dem Namen `DEBUG1N0170` gesetzt ist.

Anmerkungen:

1. Ein Breakpoint kann nur für ein ausführbares Statement gesetzt werden. Falls Sie versuchen, für ein nicht ausführbares Statement einen Breakpoint zu setzen, erscheint eine entsprechende Fehlermeldung.
2. Normalerweise ist ein Breakpoint nur während der aktuellen Natural-Sitzung gültig. Falls erforderlich, können Sie einen Breakpoint für zukünftige Sitzungen speichern. Weitere Informationen siehe ***Breakpoints und Watchpoints speichern*** in *Weitere Hinweise zur Benutzung des Debuggers*.

2 Führen Sie das Programm `DEBUG1P` aus.

Der Debugger unterbricht nun die Programmausführung an der Statement-Zeile, bei der der neue Breakpoint gesetzt ist. Das Fenster **Debug Break** erscheint:

```
+------------------- Debug Break --------------------+
| Break by breakpoint DEBUG1N0170                    |
| at line  170 in subprogram DEBUG1N (level 2)       |
| in library DEBUG    in system file (10,32).        |
|                                                    |
|         G   Go                                     |
|         L   List break                             |
|         M   Debug Main Menu                        |
|         N   Next break command                     |
|         R   Run (set test mode OFF)                |
|         S   Step mode                              |
|         V   Variable maintenance                   |
|                                                    |
| Code .. G                                          |
|                                                    |
|                                                    |
| PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS    |
+----------------------------------------------------+
```

Das Fenster zeigt den Namen des Breakpoints, die entsprechende Statement-Zeile, das Objekt und die Library, in der das Objekt enthalten ist. Es zeigt auch die Bearbeitungsebene (Level) des Subprogramms `DEBUG1N`.

Um das im Fenster Debug Break gezeigte Statement zu betrachten:

- Führen Sie die Funktion **List break** aus.

  Der Quellcode des Subprogramms `DEBUG1N` wird im Bildschirm **List Object Source** angezeigt:

```
11:36:45              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                  - List Object Source -              Object DEBUG1N
                                                                  Bottom of data
Co Line Source                                                      Message
__ 0070   2 NUMCHILD  (N2)                                       |
__ 0080   2 ENTRYDATE (D)                                        |
__ 0090   2 SALARY    (P7.2)                                     |
__ 0100   2 BONUS     (P7.2)                                     |
__ 0110 LOCAL                                                    |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>                |
__ 0130 1 DIFFERENCE  (P3.2)                                     |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                          |
__ 0150 END-DEFINE                                               |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365             | last line
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | DEBUG1N0170
__ 0180   BONUS := SALARY * PERCENT / 100                        |
__ 0190 END-IF                                                   |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA |
__ 0210 END                                                      |

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

Die im Fenster **Debug Break** angezeigte Statement-Zeile 170 ist hervorgehoben. In der Spalte **Message** wird der Name des für dieses Statement gesetzten Breakpoints (`DEBUG1N0170`) sowie die zuletzt ausgeführte Statement-Zeile angezeigt (die Zeile 160 ist als `last line` kommentiert).

Zur Erinnerung: Ein Breakpoint unterbricht die Programmausführung *vor* dem Statement, für das der gesetzte Breakpoint verarbeitet wird.

Es gibt mehrere Direktkommandos, die Sie auf dem Bildschirm **List Object Source** eingeben können, um weitere Informationen über das aktuelle Objekt zu erhalten. Sie können sich z. B. alle Variablen wie im folgenden Schritt beschrieben anzeigen lassen.

Um eine Liste der in `DEBUG1N` enthaltenen Variablen zu erhalten:

- Geben Sie in der Kommandozeile Folgendes ein:

```
DIS VAR
```

Der Bildschirm **Display Variables** erscheint (Beipiel):

```
11:06:13              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON         - Display Variables (Alphanumeric) -       Object DEBUG1N
                                                                          All
Co Le Variable Name                    F       Leng Contents              Msg.
    1 EMPLOYEE
__  2 NAME                             A         20 MEIER
__  2 ENTRYDATE                        D            1989-01-01
__  2 SALARY                           P        7.2 99000.00
__  2 BONUS                            P        7.2 *** invalid data ***
__  1 TARGETDATE                       D            2009-01-01
__  1 DIFFERENCE                       P        3.2 20.00
__  1 PERCENT                          P        2.2 3.50



Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Zoom  Flip  -     +     Li Br Alpha Hex   Canc
```

Es werden alle in DEBUG1N definierten Variablen aufgelistet. Die Bemerkung invalid data (ungültige Daten) bei der Variablen BONUS können Sie unbeachtet lassen. Im vorliegenden Fall ist es nicht von Bedeutung, ob die Variable BONUS korrekt initialisiert wird, weil sie nur als Ziel-Operand benutzt wird.

Um jedoch ein anderes Debugger-Kommando auszuführen, ändern Sie den Inhalt von BONUS im folgenden Schritt.

Um den Inhalt von BONUS zu prüfen und zu ändern:

1 Geben Sie in der Spalte **Co** neben BONUS folgendes Zeilenkommando ein:

```
MO
```

Oder:

Geben Sie in der Kommandozeile Folgendes ein:

```
MOD VAR BONUS
```

Der Bildschirm **Modify Variable** wird angezeigt (Beispiel):

```
11:29:50              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON           - Modify Variable (Alphanumeric) -        Object DEBUG1N


Name ...... EMPLOYEE.BONUS
Fmt/Len ... P 7.2
Type ...... parameter
Index .....
Range .....

Position .. 1
Contents .. ____________



Command ===>


Variable contains invalid data.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Save  Flip             Li Br Alpha Hex   Canc
```

2 Sie können die hexadezimale Anzeige benutzen, um sicherzustellen, dass die Variable nicht in Format P (gepackt numerisch) ist. Drücken Sie dazu PF10 (Alpha), um auf die alphanumerische Anzeige zurückzuschalten.

3 Geben Sie im Feld **Contents** einen Wert im Format P (numerisch gepackt) ein, z.B. `12345.00` und drücken Sie PF5 (Save).

Der Bildschirm sieht nun folgendermaßen aus (Beispiel):

```
11:50:00              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON           - Display Variable (Alphanumeric) -       Object DEBUG1N


Name ...... EMPLOYEE.BONUS
Fmt/Len ... P 7.2
Type ...... parameter
Index .....
Range .....

Position ..
Contents .. 12345.00



Command ===>


Variable BONUS modified.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Mod   Flip             Li Br Alpha Hex   Canc
```

Die Änderung von **Contents** wird durch die Meldung `Variable BONUS modified.` (Variable `BONUS` geändert.) angezeigt.

4 Drücken Sie PF9 (Li Br) oder PF3 (Exit).

Der Bildschirm **List Object Source** erscheint.

5 Geben Sie in der Kommandozeile Folgendes ein:

```
GO
```

Der Debugger gibt die Kontrolle an das Natural-Laufzeitsystem zurück, welches die Ausführung des Programms `DEBUG1P` beendet, weil kein weiteres Debug-Ereignis (Event) auftritt. Der vom Programm erstellte Bericht wird ausgegeben:

```
Page      1                                                  07-09-06  10:02:51

EMPLOYEE RECEIVES:   100800.00
    PLUS BONUS OF:     3465.00




NEXT                                                                LIB=DEBUG
```

6 Bevor Sie fortfahren und mit der nächsten Sitzung beginnen, löschen Sie bitte alle aktuellen Breakpoints, indem Sie Folgendes an der Eingabeaufforderung `NEXT` eingeben:

```
TEST DEL BP * *
```

Es erscheint eine Meldung, die bestätigt, dass alle Breakpoints (im konkreten Fall nur ein Breakpoint) gelöscht sind.

Sitzung 3 - Einen Watchpoint benutzen

Das Programm `DEBUG1P` und das Subprogramm `DEBUG1N` führen eine Berechnung für die Bonus- und Gehaltszahlung eines einzelnen Mitarbeiters durch. Wenn mehrere Mitarbeiterdatensätze verarbeitet würden, würden Sie wahrscheinlich prüfen, ob die Variable `BONUS` nun korrekt aktualisiert wird. Dies geschieht durch Setzen eines Watchpoints für diese Variable. Ein Watchpoint ermöglicht es dem Debugger, die Programmausführung zu unterbrechen, wenn sich der Inhalt der angegebenen Variablen ändert.

Um einen Watchpoint für die Variable `BONUS` zu setzen:

1 Geben Sie Folgendes an der Eingabeaufforderung `NEXT` ein:

```
TEST SET WP DEBUG1N BONUS
```

Die Meldung `Watchpoint BONUS set for variable EMPLOYEE.BONUS.` erscheint. Sie bestätigt, dass ein Watchpoint für die Variable `BONUS` in Beispiel-Subprogramm `DEBUG1N` gesetzt ist.

Anmerkungen:

1. Wenn Sie in der Kommandozeile eines Debugger-Bildschirms ein Debugger-Direktkommando eingeben, müssen Sie das Schlüsselwort `TEST` weglassen. Beispiel: Anstelle von `TEST SET WP DEBUG1N BONUS` geben Sie dann nur `SET WP DEBUG1N BONUS` ein.
2. Normalerweise ist ein Watchpoint nur während der aktuellen Natural-Sitzung gültig. Falls erforderlich, können Sie einen Watchpoint für zukünftige Sitzungen speichern. Weitere Informationen siehe ***Breakpoints und Watchpoints speichern*** in *Weitere Hinweise zur Benutzung des Debuggers*.

2 Führen Sie das Programm `DEBUG1P` an der Eingabeaufforderung `NEXT` aus.

Der Debugger unterbricht nun die Programmausführung an dem neuen Watchpoint. Das Fenster **Debug Break** erscheint:

```
+------------------- Debug Break --------------------+
| Break by watchpoint BONUS                          |
| at line  180 in subprogram DEBUG1N (level 2)       |
| in library DEBUG    in system file (10,32).        |
|                                                    |
|         G   Go                                     |
|         L   List break                             |
|         M   Debug Main Menu                        |
|         N   Next break command                     |
|         R   Run (set test mode OFF)                |
|         S   Step mode                              |
|         V   Variable maintenance                   |
|                                                    |
| Code .. G                                          |
|                                                    |
|                                                    |
| PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS    |
+----------------------------------------------------+
```

In dem Fenster wird angezeigt, dass ein Watchpoint in Zeile 180 festgestellt wurde. Diese Zeile enthält das Statement, das die Variable `BONUS` verarbeitet.

Der Debugger unterbrach die Programmausführung, *nachdem* das Statement für `BONUS` verarbeitet war. Erst dann konnte der Debugger erkennen, dass sich der Inhalt der Variablen geändert hat.

3 Führen Sie die Funktion **List break** aus.

Der Bildschirm **List Object Source** erscheint (Beispiel):

```
16:24:46              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                  - List Object Source -              Object DEBUG1N
                                                                  Bottom of data
Co Line Source                                                      Message
__ 0070   2 NUMCHILD  (N2)                                       |
__ 0080   2 ENTRYDATE (D)                                        |
__ 0090   2 SALARY    (P7.2)                                     |
__ 0100   2 BONUS     (P7.2)                                     |
__ 0110 LOCAL                                                    |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>                |
__ 0130 1 DIFFERENCE  (P3.2)                                     |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                          |
__ 0150 END-DEFINE                                               |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365             |
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | DEBUG1N0170
__ 0180   BONUS := SALARY * PERCENT / 100                        | BONUS
__ 0190 END-IF                                                   |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA |
__ 0210 END                                                      |

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

Das Statement, welches die Variable BONUS referenziert, ist hervorgehoben. Die Meldung in der Spalte **Message** zeigt den Namen des für die Variable gesetzten Watchpoint.

Um auf Änderungen in BONUS zu prüfen:

1 Geben Sie in der Kommandozeile Folgendes ein:

```
DIS VAR BONUS
```

Der Bildschirm **Display Variable** erscheint. Im Feld **Contents** (Inhalt) wird der Wert 3465.00 angezeigt. Das zeigt, dass sich der Inhalt der Variablen BONUS geändert hat.

2 Drücken Sie PF3 (Exit), um zum Bildschirm **List Object Source** zurückzukehren.

Um auf Änderungen in SALARY zu prüfen:

1 Um den Inhalt der Variablen SALARY in einem späteren Schritt zu prüfen, setzen Sie einen Breakpoint für SALARY. Dazu geben Sie in der Spalte **Co** vor der Zeile 200 Folgendes ein:

```
SE
```

Vom Bildschirm **List Object Source** aus ist die Eingabe eines Zeilenkommandos wie SE eine bequeme Alternative zur Verwendung des Direktkommandos SET BP.

In der Spalte **Message** wird angezeigt, dass ein Breakpoint (BP) für die Zeile 200 gesetzt ist:

```
17:55:58             ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                - List Object Source -              Object DEBUG1N
                                                                  Bottom of data
Co Line Source                                                      Message
__ 0070   2 NUMCHILD  (N2)                                       |
__ 0080   2 ENTRYDATE (D)                                        |
__ 0090   2 SALARY    (P7.2)                                     |
__ 0100   2 BONUS     (P7.2)                                     |
__ 0110 LOCAL                                                    |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>                |
__ 0130 1 DIFFERENCE  (P3.2)                                     |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                          |
__ 0150 END-DEFINE                                               |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365             |
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | DEBUG1N0170
__ 0180   BONUS := SALARY * PERCENT / 100                        | BONUS
__ 0190 END-IF                                                   |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA | BP set
__ 0210 END                                                      |

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

2 Geben Sie in der Kommandozeile Folgendes ein:

```
GO
```

Das Fenster **Debug Break** erscheint:

```
+------------------- Debug Break -------------------+
| Break by breakpoint DEBUG1N0200                   |
| at line  200 in subprogram DEBUG1N (level 2)      |
| in library DEBUG    in system file (10,32).       |
|                                                   |
|         G   Go                                    |
|         L   List break                            |
|         M   Debug Main Menu                       |
|         N   Next break command                    |
|         R   Run (set test mode OFF)               |
|         S   Step mode                             |
|         V   Variable maintenance                  |
|                                                   |
| Code .. G                                         |
|                                                   |
|                                                   |
| PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS   |
+---------------------------------------------------+
```

3 Führen Sie die Funktion **List break** aus.

Der Bildschirm **List Object Source** sieht jetzt wie im folgenden Beispiel aus:

```
10:49:31             ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                - List Object Source -              Object DEBUG1N
                                                                  Bottom of data
Co Line Source                                                      Message
__ 0070   2 NUMCHILD  (N2)                                       |
__ 0080   2 ENTRYDATE (D)                                        |
__ 0090   2 SALARY    (P7.2)                                     |
__ 0100   2 BONUS     (P7.2)                                     |
__ 0110 LOCAL                                                    |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>                |
__ 0130 1 DIFFERENCE  (P3.2)                                     |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                          |
__ 0150 END-DEFINE                                               |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365             |
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | DEBUG1N0170
__ 0180   BONUS := SALARY * PERCENT / 100                        | last line
__ 0190 END-IF                                                   |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA | DEBUG1N0200
__ 0210 END                                                      |

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

Da es sich hier um einen Breakpoint handelt, ist das Statement, das die Variable SALARY referenziert (und ändert), noch nicht ausgeführt worden. Folglich hat sich der Inhalt der Variablen nicht geändert.

4 Geben Sie DIS VAR SALARY in der Kommandozeile ein, um zu überprüfen, dass der er Inhalt der Variablen unverändert ist.

Der Bildschirm **Display Variable** belegt, dass die Variable SALARY immer noch den Wert 99000 enthält. Dies ist der Anfangswert, der ihr im Programm DEBUG1P zugewiesen wurde.

5 Springen Sie nun zum nächsten Statement, um die Änderung des Variableninhalts zu betrachten. Wählen Sie eine der folgenden Methoden:

Geben Sie in der Kommandozeile Folgendes ein:

```
STEP
```

Oder:

Drücken Sie PF2 (Step).

Der neue Bildschirm sieht ähnlich wie im folgenden Beispiel aus:

```
13:38:24             ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON               - List Object Source -              Object DEBUG1N
                                                               Bottom of data
Co Line Source                                                   Message
__ 0070   2 NUMCHILD  (N2)                                    |
__ 0080   2 ENTRYDATE (D)                                     |
__ 0090   2 SALARY    (P7.2)                                  |
__ 0100   2 BONUS     (P7.2)                                  |
__ 0110 LOCAL                                                 |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>             |
__ 0130 1 DIFFERENCE  (P3.2)                                  |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                       |
__ 0150 END-DEFINE                                            |
__ 0160 DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365          |
__ 0170 IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPAN | DEBUG1N0170
__ 0180   BONUS := SALARY * PERCENT / 100                     |
__ 0190 END-IF                                                |
__ 0200 SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREA | last line
__ 0210 END                                                   | step mode

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

Sie haben eine Zeile übersprungen und das nächste ausführbare Statement in Zeile 200 ausgeführt, welches die Variable SALARY ändert. In der Spalte **Message** wird angezeigt, dass Step-

Modus gesetzt ist. Im Step-Modus setzt der Debugger die Programmausführung am nächsten ausführbaren Statement fort.

6 Geben Sie `DIS VAR SALARY` in der Kommandozeile ein, um den Variableninhalt zu prüfen.

Der Bildschirm **Display Variable** erscheint und zeigt im Feld **Contents** einen Wert von `100800.00` an. Das belegt, dass sich der Inhalt der Variablen `SALARY` geändert hat.

7 Geben Sie in der Kommandozeile Folgendes ein:

```
GO
```

Der Debugger gibt die Kontrolle an das Natural-Laufzeitsystem zurück, welches die Ausführung des Programms `DEBUG1P` beendet, weil kein weiteres Debug-Ereignis (Event) auftritt. Der vom Programm erstellte Bericht wird ausgegeben.

Sitzung 4 - Verfolgen des logischen Ablaufs von Programmen

Diese Sitzung beschreibt Debugging-Methoden, die Sie anwenden können, um eine komplexe Natural-Anwendung mit zahlreichen Objekten besser zu verstehen, zu überblicken und zu steuern.

Die Sitzung beginnt mit einer Anleitung zur Analyse des logischen Ablaufs einer Anwendung auf Statement-Ebene. Anschließend wird demonstriert, wie Breakpoints verwendet werden können, um die Reihenfolge der Programmausführung herauszufinden.

Die Anweisungen in dieser Sitzung basieren auf einer einfachen (aber zur Demonstration ausreichenden) Beispielanwendung. Diese besteht aus einem Programm (`DEBUG2P`) und drei Subprogrammen (`DEBUG2N`, `DEBUG3N` und `DEBUG4N`).

Um einen Breakpoint am Programmanfang oder -ende zu setzen:

1 Setzen Sie einen Breakpoint für `DEBUG2P`, indem Sie an der Eingabeaufforderung `NEXT` Folgendes eingeben:

```
TEST SET BP DEBUG2P BEG
```

Die Meldung `Breakpoint DEBUG2P-BEG set at line BEG of object DEBUG2P.` bestätigt, dass ein Breakpoint in `DEBUG2N` gesetzt ist.

Die Verwendung des Schlüsselworts `BEG` anstelle einer bestimmten Zeilennummer bewirkt, dass der Breakpoint am Anfang des Programms, d. h. für das erste auszuführende Statement, gesetzt wird. Dies kann sogar das `DEFINE DATA`-Statement sein, wenn z. B. eine `INIT`-Klausel verwendet wird, die beim Katalogisieren des Programms ein ausführbares Statement erzeugt.

Tipp: Sie können auch das Schlüsselwort `END` angeben, um einen Breakpoint für das letzte auszuführende Statement zu setzen. Dies kann das `END`-Statement sein, aber auch das `FETCH`- oder `CALLNAT`-Statement.

2 Führen Sie das Programm `DEBUG2P` aus.

Das Fenster **Debug Break** erscheint:

```
+------------------- Debug Break -------------------+
| Break by breakpoint DEBUG2P-BEG                   |
| at line  130 in program DEBUG2P (level 1)         |
| in library DEBUG    in system file (10,32).       |
|                                                   |
|         G   Go                                    |
|         L   List break                            |
|         M   Debug Main Menu                       |
|         N   Next break command                    |
|         R   Run (set test mode OFF)               |
|         S   Step mode                             |
|         V   Variable maintenance                  |
|                                                   |
| Code .. G                                         |
|                                                   |
|                                                   |
| PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS   |
+---------------------------------------------------+
```

Der Debugger greift nun am ersten für das Programm gesetzten Breakpoint ein.

3 Führen Sie die Funktion **List break** aus, um den Quellcode zu prüfen und zu sehen, dass der Debugger jetzt am ersten ausführbaren Statement `NAME := 'MEIER'` eingreift.

Um eine Anwendung schrittweise auszuführen:

1 Setzen Sie im Bildschirm **List Object Source** den Step-Mode, indem Sie entweder PF2 (Step) drücken oder `STEP` in der Kommandozeile eingeben.

Das zuletzt ausgeführte Statement ist mit `last line` (letzte Zeile) kommentiert. Das nächste auszuführende Statement ist hervorgehoben und mit `step mode` kommentiert.

Tipp: Wenn Sie nicht möchten, dass der Debugger bei jedem einzelnen Statement pausiert, sondern eine Anwendung schneller durchläuft, geben Sie im `STEP`-Kommando die Anzahl der Statements an, die Sie überspringen möchten, z.B.: `STEP 2` oder `STEP 10`.

2 Drücken Sie PF2 (Step) mehrmals, bis das `CALLNAT`-Statement mit `step mode` kommentiert ist.

3 Fahren Sie mit PF2 (Step) und führen Sie das `CALLNAT`-Statement aus.

Das aufgerufene Subprogramm DEBUG2N wird angezeigt, wobei das nächste auszuführende Statement hervorgehoben ist:

```
11:59:19             ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                  - List Object Source -              Object DEBUG2N
                                                                     Top of data
Co Line Source                                                      Message
__ 0010 ** SUBPROGRAM DEBUG2N: CALLS 'DEBUG3N' AND 'DEBUG4N'FOR |
__ 0020 ************************************************** |
__ 0030 DEFINE DATA                                          | step mode
__ 0040 PARAMETER                                            |
__ 0050 1 EMPLOYEE                                           |
__ 0060   2 NAME      (A20)                                  |
__ 0070   2 NUMCHILD  (N2)                                   |
__ 0080   2 ENTRYDATE (D)                                    |
__ 0090   2 SALARY    (P7.2)                                 |
__ 0100   2 BONUS     (P7.2)                                 |
__ 0110 LOCAL                                                |
__ 0120 1 TARGETDATE  (D)    INIT <D'2009-01-01'>            |
__ 0130 1 DIFFERENCE  (P3.2)                                 |
__ 0140 1 PERCENT     (P2.2) INIT <3.5>                      |
__ 0150 END-DEFINE                                           |

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Step  Exit  Last  Scan  Flip  -     +     Li Br <     >     Canc
```

Alternativ könnten Sie das CALLNAT-Statement überspringen, indem Sie STEP SKIP in der Kommandozeile eingeben.

Sie würden dann nur die Statements in dem aufrufenden Programm DEBUG2 schrittweise ausführen, jedoch nicht die Statements in einem aufgerufenen Programm.

Um die Ebenen zu betrachten, auf denen die Objekte ausgeführt werden:

1 Geben Sie im Bildschirm **List Object Source** für DEBUG2N Folgendes in der Kommandozeile ein:

```
OBJCHAIN
```

Der Bildschirm **Break Information** erscheint (Beispiel):

```
13:45:34              ***** NATURAL TEST UTILITIES *****            2007-09-06
                             - Break Information -

No GDA active for the current program.

Break by step mode
at line   30 in subprogram DEBUG2N (level 2)
in library DEBUG    in system file (10,32).
```

Zusätzlich zu den bereits bekannten Objektinformationen zeigt dieser Bildschirm an, ob das Programm einen globalen Datenbereich (GDA, Global Data Area) referenziert.

2 Drücken Sie ENTER, um eine Seite weiter nach unten zu blättern.

Der Bildschirm sieht jetzt aus wie im folgenden Beispiel:

```
13:46:34              ***** NATURAL TEST UTILITIES *****            2007-09-06
                           - Current Object Chain -

Level Name     Type        Line Library   DBID   FNR
  2   DEBUG2N  Subprogram     0 DEBUG       10    32
  1   DEBUG2P  Program      170 DEBUG       10    32
```

Dieser Bildschirm zeigt die Ebenen (Level) an, auf denen die Objekte ausgeführt werden: Subprogramm `DEBUG2N` wird auf Ebene 2 ausgeführt und das Programm `DEBUG2P` (welches das Subprogramm aufruft) wird auf der übergeordneten Ebene 1 ausgeführt.

3 Drücken Sie ENTER.

Der Bildschirm **List Object Source** erscheint.

4 Geben Sie in der Kommandozeile Folgendes ein:

```
GO
```

Der Debugger gibt die Kontrolle zurück an das Natural-Laufzeitsystem. Dieses beendet die Ausführung von `DEBUG2P`, weil kein weiteres Debug-Ereignis auftritt. Der vom Programm erzeugte Bericht wird ausgegeben:

```
Page      1                                                  07-09-06  10:04:21

EMPLOYEE RECEIVES:    99300.00
    PLUS BONUS OF:     3565.00




NEXT                                                                LIB=DEBUG
```

5 Löschen Sie alle zurzeit gesetzten Breakpoints, indem Sie Folgendes an der Eingabeaufforderung `NEXT` eingeben:

```
TEST DEL BP * *
```

Es erscheint eine Meldung, die bestätigt, dass alle Breakpoints gelöscht sind.

Um Breakpoints zur Verfolgung der Programmausführung zu setzen:

1 Geben Sie Folgendes an der Eingabeaufforderung `NEXT` ein:

```
TEST SET BP ALL BEG
```

Die Meldung `Breakpoint ALL-BEG set at line BEG of object ALL.` erscheint.

Sie zeigt an, dass Sie einen Breakpoint für das erste ausführbare Statement jedes auszuführenden Objekts gesetzt haben.

2 Führen Sie das Programm `DEBUG2P` aus.

Es erscheint ein **Debug Break**-Fenster für `DEBUG2P`.

3 Führen Sie die Funktion **Go** wiederholt aus.

Jedes Mal, wenn Sie die Funktion **Go** ausführen, wird das nächste aufgerufene Objekt im **Debug Break**-Fenster angezeigt (`DEBUG2N` zuerst und dann `DEBUG3N` und `DEBUG4N`). So können Sie leicht feststellen, welche Objekte an welcher Stelle während der Programmausführung aufgerufen werden. Zusätzlich können Sie für jedes Objekt die Menüfunktionen des **Debug Break**-Fensters anwenden.

4 Wenn die Eingabeaufforderung `NEXT` erscheint, löschen Sie alle zurzeit gesetzten Breakpoints, indem Sie Folgendes eingeben:

```
TEST DEL BP * *
```

Es erscheint eine Meldung, die bestätigt, dass alle Breakpoints gelöscht sind.

Sitzung 5 - Verwendung von Statistiken über die Programmausführung

Mit dem Debugger können Sie statistische Informationen darüber anzeigen, welche Objekte wie oft aufgerufen werden. Zusätzlich können Sie herausfinden, welche Statements ausgeführt werden und wie oft.

Um zu prüfen, welche Objekte während der Programmausführung aufgerufen werden:

1 Geben Sie Folgendes an der Eingabeaufforderung `NEXT` ein:

```
TEST SET CALL ON
```

Die Meldung `Call statistics started.` wird angezeigt. Sie bestätigt, das die Statistik-Funktion aktiviert ist.

2 Führen Sie das Programm DEBUG2P aus.

Der Debugger protokolliert alle ausgeführten Objektaufrufe. Der vom Programm erstellte Bericht wird ausgegeben.

3 Geben Sie Folgendes an der Eingabeaufforderung NEXT ein:

```
TEST DIS CALL
```

Der Bildschirm **Display Called Objects** erscheint (Beispiel):

```
10:43:47              ***** NATURAL TEST UTILITIES *****             2007-09-06
Test Mode ON               - Aufgerufene Objekte anzeigen-            Object
                                                                          All
Object   Library  Type         DBID   FNR S/C Ver Cat Date   Time       Calls
*_______ DEBUG___
DEBUG2P  DEBUG    Program        10    32 S/C 4.2 2007-08-30 13:48          1
DEBUG2N  DEBUG    Subprogram     10    32 S/C 4.2 2007-08-30 13:48          1
DEBUG3N  DEBUG    Subprogram     10    32 S/C 4.2 2007-08-30 13:48          1
DEBUG4N  DEBUG    Subprogram     10    32 S/C 4.2 2007-08-30 13:48          1



Command ===>


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

Der Bildschirm zeigt eine Liste aller ausgeführten Objekte: Das aufrufende Programm (DEBUG2P) und alle anderen aufgerufenen Objekte (DEBUG2N, DEBUG3N und DEBUG4N). Außerdem wird angezeigt, wie oft ein Objekt aufgerufen wird (CALLS), der Typ des aufgerufenen Objekts, wo und unter welcher Natural-Version das Objekt gespeichert ist, ob Source-Objekte und katalogisierte Objekte existieren und wann das Objekt katalogisiert wurde.

4 Drücken Sie PF3 (Exit) oder PF12 (Canc) mehrmals, bis die Eingabeaufforderung NEXT erscheint.

Um zu prüfen, welche Statements während der Programmausführung ausgeführt werden:

1 Geben Sie Folgendes an der Eingabeaufforderung NEXT ein:

```
TEST SET XSTAT COUNT
```

Die Meldung `Statement execution counting started for library/object */*.` bestätigt, dass die Statistik-Funktion für alle Objekte aktiviert ist, die in der aktuellen Library und in allen, mit dieser Library verketteten Steplibs enthalten sind.

2 Führen Sie das Programm DEBUG2P aus.

Der Debugger protokolliert alle von dem Programm verarbeiteten Statements. Anschließend wird der vom Programm erstellte Bericht ausgegeben.

3 Geben Sie Folgendes am Eingabeaufforderungszeichen `NEXT` ein:

```
TEST DIS XSTAT
```

Der Bildschirm **List Statement Execution Statistics** erscheint (Beispiel):

```
11:39:10              ***** NATURAL TEST UTILITIES *****             2007-09-06
Test Mode ON        - List Statement Execution Statistics -     Object
                                                                          All
Co Object   Library  Type        DBID   FNR Obj.Called Exec Exec   %  Total No.
   *_______ *_______                          n Times able uted     Executions
__ DEBUG2P  DEBUG    Program       10    32          1    8    8 100          8
__ DEBUG2N  DEBUG    Subprogram    10    32          1    8    8 100          8
__ DEBUG3N  DEBUG    Subprogram    10    32          1    2    2 100          2
__ DEBUG4N  DEBUG    Subprogram    10    32          1   10    7  70          7



Command ===>


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

Der Bildschirm zeigt eine Liste. Diese enthält die Anzahl der Aufrufe (`Obj. Called n Times`), die Anzahl der ausführbaren Statements (`Exec(ecut)able`), die Anzahl der ausgeführten Statements (`Executed`), den Prozentsatz der ausgeführten Statements im Verhältnis zur Gesamtzahl der ausführbaren Statements (`%`) und die Gesamtzahl der ausgeführten Statements (`Total No. Executions`).

4 Geben Sie in der Spalte **Co** neben `DEBUG4N` Folgendes ein:

```
DS
```

Folgender Statistik-Bildschirm wird angezeigt (Beispiel):

```
12:11:19              ***** NATURAL TEST UTILITIES *****             2007-09-06
Test Mode ON              - Display Statement Lines -             Object DEBUG4N

Line Source                                                                Count
0010 ** SUBPROGRAM 'DEBUG4N': CALCULATES SPECIAL SALARY INCREASE
0020 ************************************************************
0030 DEFINE DATA
0040 PARAMETER
0050 1 SALARY (P7.2)
0060 END-DEFINE
0070 DECIDE FOR FIRST CONDITION                                               1
0080   WHEN SALARY < 50000                                                    1
0090     SALARY := SALARY + 1800                                  not executed
0100   WHEN SALARY < 70000                                                    1
0110     SALARY := SALARY + 1200                                  not executed
0120   WHEN SALARY < 90000                                                    1
0130     SALARY := SALARY + 600                                   not executed
0140   WHEN NONE                                                              1
0150     SALARY := SALARY + 300                                               1

Command ===>


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

Der Bildschirm zeigt an, wie oft ein Statement ausgeführt wurde und welche ausführbaren Statements nicht bearbeitet wurden.

Weitere Hinweise zur Benutzung des Debuggers

Dieser Abschnitt enthält zusätzliche Hinweise zur Benutzung des Debuggers.

- Zeitstempel von Objekten
- Breakpoints und Watchpoints speichern
- Debug-Hauptmenü (Main Menu) für Verwaltungsfunktionen
- Hilfe zu Kommandos in Maintenance-Bildschirmen
- Während der Programmunterbrechung verfügbare Funktionen
- Next-Option für zusätzliche Kommandos während der Programmunterbrechung
- Große Variablen und Arrays anzeigen
- Debugger-Statistikberichte drucken

- Debugger im Batch-Modus benutzen

Zeitstempel von Objekten

Ein katalogisiertes Objekt, das nicht genau mit dem Source-Objekt übereinstimmt, kann zu Fehlern bei der Fehlersuche führen. Wenn Sie sicherstellen wollen, dass Source-Objekt und katalogisiertes Objekt übereinstimmen, speichern und katalogisieren Sie sie mit dem Systemkommando `STOW`.

Weitere Informationen siehe *Voraussetzungen für den Betrieb*.

Breakpoints und Watchpoints speichern

Sie können die in der aktuellen Sitzung gesetzten Breakpoints und Watchpoints als Debug-Umgebung speichern und diese Umgebung zur Verwendung in einer zukünftigen Sitzung laden. Dies ist hilfreich, wenn Sie eine Anwendung wiederholt mit denselben Debug-Einträgen testen wollen.

Weitere Informationen siehe Abschnitt *Debug Environment Maintenance (Verwaltung der Debug-Umgebung)*.

Debug-Hauptmenü (Main Menu) für Verwaltungsfunktionen

Alle Verwaltungsfunktionen des Debuggers, wie z. B. das Setzen eines Breakpoint oder das Erstellen einer Statistik, können entweder mit einem Direktkommando oder mit den Verwaltungsfunktionen im Hauptmenü (**Debug Main Menu**) ausgeführt werden. Um das Hauptmenü aufzurufen, haben Sie folgende Eingabemöglichkeiten:

- `TEST` an einer Eingabeaufforderung.
- `MENU` in der Kommandozeile eines Debugger-Bildschirms.
- `M` im Feld **Code** im Fenster **Debug Break**.

Hilfe zu Kommandos in Maintenance-Bildschirmen

Sie können sich eine Liste der in einem Debugger-Maintenance-Bildschirm zur Verfügung stehenden Direktkommandos anzeigen lassen, indem Sie `PF1` (Help) drücken oder in der Kommandozeile ein Fragezeichen (?) eingeben.

In einem Debugger-Maintenance-Bildschirm mit Listeneinträgen stehen außerdem Zeilenkommandos zur Verfügung, die Sie benutzen können, um einen Eintrag zu bearbeiten. Die Eingabe eines Zeilenkommandos erfolgt neben dem betreffenden Eintrag in der Spalte **Co**. Sie können in dieser Spalte ein Fragezeichen (?) eingeben, um eine Liste der gültigen Zeilenkommandos angezeigt zu bekommen.

Während der Programmunterbrechung verfügbare Funktionen

Dieser Abschnitt enthält eine Liste der wichtigsten, während der Programmunterbrechung verfügbaren Funktionen. Sie können sie entweder im Fenster **Debug Break** oder in der Kommandozeile eines Debugger-Maintenance-Bildschirms ausführen.

Code im Debug Break-Fenster	Alternatives Direktkommando	Funktion
G	GO	Fortsetzung der Programmausführung, bis das nächste Debug-Ereignis auftritt.
L	LIST BREAK	Auflistung der Objekt-Source bei der Statement-Zeile, in der das Debug-Ereignis auftritt.
N	NEXT	Ausführung des nächsten Break-Kommandos, wenn es für einen Breakpoint oder Watchpoint angegeben wurde. Siehe auch *Next-Option für zusätzliche Kommandos während der Programmunterbrechung*.
R	RUN	Ausschalten des Test-Modus und Fortsetzen der Programmausführung.
S	STEP	Zeilenweise Verarbeitung der ausführbaren Statements.
V	DIS VAR	Anzeige einer Liste der Variablen, die für das unterbrochene Objekt definiert sind.

Next-Option für zusätzliche Kommandos während der Programmunterbrechung

Wenn Sie einen Breakpoint oder Watchpoint anzeigen oder ändern, werden Sie feststellen, dass an jeden von ihnen das Debugger-Kommando BREAK angehängt ist. Dieses Kommando ruft das **Debug-Break**-Fenster auf und darf nicht entfernt werden. Sie können jedoch zusätzliche Debugger-Kommandos angeben, die während der Programmunterbrechung nach dem BREAK-Kommando ausgeführt werden sollen. Ein zusätzlicher Befehl wird ausgeführt, wenn Sie entweder das Kommando NEXT in der Kommandozeile oder den Funktionscode N im **Debug Break**-Fenster eingeben.

Sie können die Debugger-Kommandos, wie im folgenden Beispiel gezeigt, im Feld **Commands** des entsprechenden Breakpoint- oder Watchpoint-Maintenance-Bildschirm eingeben:

```
11:38:55              ***** NATURAL TEST UTILITIES *****            2007-09-06
Test Mode ON                 - Modify Breakpoint -              Object

Spy number ..............   1
Initial state ........... A (A = Active, I = Inactive)
Breakpoint name ......... DEBUG1P0170_   DBID/FNR ....... 10/32
Object name ............. DEBUG1P_       Library ........ DEBUG
Line number ............. 0170
Label ................... ________________________________
Skips before execution .. ____0
Max number executions ... ____0


Commands ... BREAK_____________________________________________________________
             STACK_____________________________________________________________
             DIS VAR BONUS_____________________________________________________
             _____________________________________________________________
             _____________________________________________________________
             _____________________________________________________________

Command ===>


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

Im obigen Beispiel weist das Kommando `STACK` den Debugger an, den Natural Stack zu untersuchen. Das Kommando `DIS VAR BONUS` weist den Debugger an, die angegebene Variable anzuzeigen. Dies ist z. B. hilfreich, wenn Sie in einer Schleife einen Breakpoint setzen und immer nur den Wert einer bestimmten Variablen sehen wollen. Sie müssen dann nicht wiederholt das Kommando `DIS VAR` eingeben.

Weitere Informationen siehe Beschreibung des Feldes **Commands** in den Abschnitten *Felder und Spalten in Breakpoint-Bildschirmen* und *Felder und Spalten in Watchpoint-Bildschirmen*.

Große Variablen und Arrays anzeigen

Der Bildschirm **Display Variable** zeigt alle Definitionen einer Variablen und zeigt ihren Inhalt in alphanumerischem oder hexadezimalem Format. Informationen zu Anzeigemöglichkeiten bei großen Variablen, deren Inhalt über den aktuellen Bildschirm hinausgeht, oder bei Variablen mit Array-Definitionen siehe Abschnitt *Variable anzeigen - einzeln*.

Debugger-Statistikberichte drucken

Sie können die vom Debugger erstellten Statistikberichte drucken oder auf einen PC herunterladen.

Weitere Informationen siehe ***Objekte drucken*** im Abschnitt *Call Statistics Maintenance (Statistiken über gerufene Objekte)* und ***Statements drucken*** im Abschnitt *Statement Execution Statistics Maintenance*.

Debugger im Batch-Modus benutzen

Der Debugger ist hauptsächlich für die interaktive Bedienung im Online-Modus ausgelegt. Zwar können Sie prinzipiell alle Debugger-Funktionen im Batch-Modus ausführen, die Verarbeitung von Online-Vorgängen im Batch-Modus (z.B. der Gebrauch von PF-Tasten) kann aber eine komplexe Batch-Programmierung erfordern. Es gibt jedoch Debugger-Funktionen, bei denen Batch-Verarbeitung eine komfortable Alternative darstellt. Eine Möglichkeit ist zum Beispiel das Sammeln und Drucken von Statistikdaten über eine Anwendung, siehe Abschnitt ***Beispiel für das Erstellen und Drucken von Statistiken im Batch-Modus*** im Abschnitt *Batch-Verarbeitung*.

Quellcode-Beispiele

Dieser Abschnitt enthält den Quellcode der Beispiel-Programme und -Subprogramme, die in den Sitzungen 1 bis 5 benötigt werden.

Programm `DEBUG1P`

```
** PROGRAM 'DEBUG1P: CALLS 'DEBUG1N' FOR SALARY AND BONUS CALCULATION
************************************************************************
DEFINE DATA
LOCAL
1 EMPLOYEE    (A42)
1 REDEFINE EMPLOYEE
  2 NAME      (A20)
  2 NUMCHILD  (N2)
  2 ENTRYDATE (D)
  2 SALARY    (P7.2)
  2 BONUS     (P7.2)
END-DEFINE
NAME      := 'MEIER'
NUMCHILD  := 2
ENTRYDATE := D'1989-01-01'
* SALARY  := 99000
CALLNAT 'DEBUG1N' NAME NUMCHILD ENTRYDATE SALARY BONUS
WRITE 'EMPLOYEE RECEIVES:'   SALARY
WRITE '    PLUS BONUS OF:'   BONUS
END
```

Subprogramm DEBUG1N

```
** SUBPROGRAM 'DEBUG1N': CALCULATES BONUS AND SALARY INCREASE
************************************************************************
DEFINE DATA
PARAMETER
1 EMPLOYEE
  2 NAME      (A20)
  2 NUMCHILD  (N2)
  2 ENTRYDATE (D)
  2 SALARY    (P7.2)
  2 BONUS     (P7.2)
LOCAL
1 TARGETDATE  (D)    INIT <D'2009-01-01'>
1 DIFFERENCE  (P3.2)
1 PERCENT     (P2.2) INIT <3.5>
END-DEFINE
DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365
IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPANY
  BONUS := SALARY * PERCENT / 100
END-IF
SALARY := SALARY + 1800    /* SALARY PLUS ANNUAL INCREASE
END
```

Programm DEBUG2P

```
** PROGRAM 'DEBUG2P': CALLS 'DEBUG2N'FOR SALARY AND BONUS CALCULATION
************************************************************************
DEFINE DATA
LOCAL
1 EMPLOYEE    (A42)
1 REDEFINE EMPLOYEE
  2 NAME      (A20)
  2 NUMCHILD  (N2)
  2 ENTRYDATE (D)
  2 SALARY    (P7.2)
  2 BONUS     (P7.2)
END-DEFINE
NAME      := 'MEIER'
NUMCHILD  := 2
ENTRYDATE := D'1989-01-01'
SALARY    := 99000
CALLNAT 'DEBUG2N' NAME NUMCHILD ENTRYDATE SALARY BONUS
WRITE 'EMPLOYEE RECEIVES:'   SALARY
WRITE '    PLUS BONUS OF:'   BONUS
END
```

Subprogramm DEBUG2N

```
** SUBPROGRAM DEBUG2N: CALLS 'DEBUG3N' AND 'DEBUG4N'FOR SPECIAL RATES
************************************************************************
DEFINE DATA
PARAMETER
1 EMPLOYEE
  2 NAME      (A20)
  2 NUMCHILD  (N2)
  2 ENTRYDATE (D)
  2 SALARY    (P7.2)
  2 BONUS     (P7.2)
LOCAL
1 TARGETDATE  (D)    INIT <D'2009-01-01'>
1 DIFFERENCE  (P3.2)
1 PERCENT     (P2.2) INIT <3.5>
END-DEFINE
DIFFERENCE := (TARGETDATE - ENTRYDATE) / 365
IF DIFFERENCE GE 10        /* BONUS FOR YEARS IN COMPANY
  BONUS := SALARY * PERCENT / 100
END-IF
IF NUMCHILD > 0
  CALLNAT 'DEBUG3N' NUMCHILD BONUS     /* SPECIAL BONUS
END-IF
CALLNAT 'DEBUG4N' SALARY               /* SPECIAL SALARY INCREASE
END                                                                  ↩
```

Subprogramm DEBUG3N

```
** SUBPROGRAM 'DEBUG3N': CALCULATES SPECIAL BONUS
************************************************************************
DEFINE DATA
PARAMETER
1 NUMCHILD (N2)
1 BONUS    (P7.2)
END-DEFINE
BONUS := BONUS + NUMCHILD * 50
END
```

Subprogramm DEBUG4N

```
** SUBPROGRAM 'DEBUG4N': CALCULATES SPECIAL SALARY INCREASE
************************************************************************
DEFINE DATA
PARAMETER
1 SALARY (P7.2)
END-DEFINE
DECIDE FOR FIRST CONDITION
  WHEN SALARY < 50000
    SALARY := SALARY + 1800
  WHEN SALARY < 70000
```

```
    SALARY := SALARY + 1200
  WHEN SALARY < 90000
    SALARY := SALARY + 600
  WHEN NONE
    SALARY := SALARY + 300
END-DECIDE
END
```

3 Debugger-Konzept

- Funktionen zur Kontrolle über die Natural-Sitzung 42
- Debug-Einträge (Spies) 43
- Debug Break-Fenster 45

Der Debugger übernimmt die Kontrolle über eine Natural-Sitzung zu Debugging-Zwecken, während ein Natural-Objekt ausgeführt wird. Dies ermöglicht es Ihnen, den Verarbeitungsablauf eines Programms zu verfolgen und verschiedene Programmuntersuchungen durchzuführen.

Sie können in einem Programm Stellen angeben, an denen der Debugger anhalten soll, indem Sie für das betreffende Programm Debug-Einträge (Breakpoints oder Watchpoints) setzen.

Wenn die Programmausführung pausiert, können Sie den Inhalt der im Programm verwendeten Variablen oder Parameter überprüfen, um die Programmlogik zu analysieren, oder Sie können den Grund für einen Natural-Fehler ermitteln.

In den folgenden Abschnitten finden Sie Informationen über die Funktionen, die im Debugger zur Verfügung stehen.

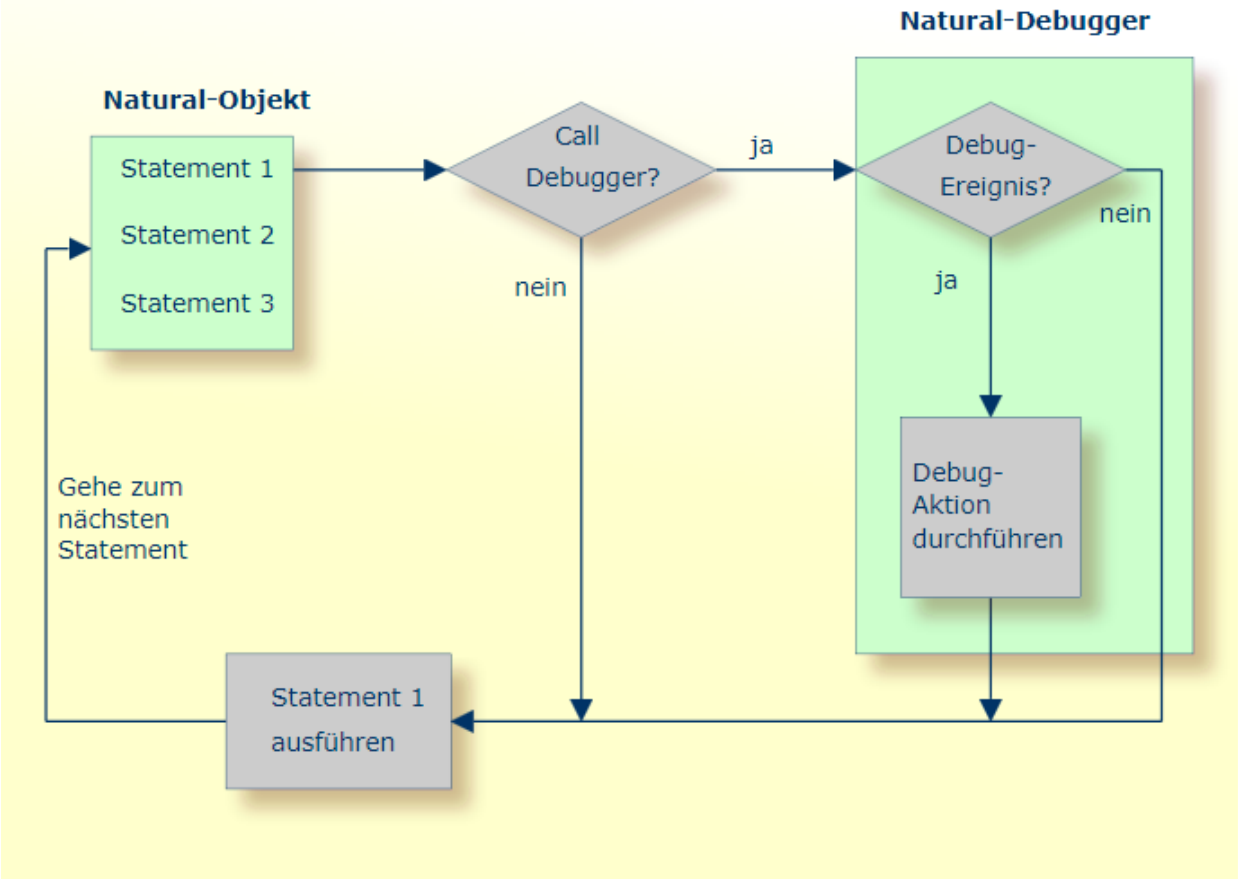
Funktionen zur Kontrolle über die Natural-Sitzung

Der Debugger enthält die Kontrolle über eine Natural-Sitzung, wenn der Test-Modus auf `ON` gesetzt ist (siehe ***Test-Modus ein- und ausschalten***). Ist der Natural-Profilparameter `DBGERR` auf `ON` gesetzt, wird der Debugger aufgerufen, wenn ein Natural-Fehler auftritt, und zwar unabhängig davon, ob Debug-Einträge vorhanden sind oder ob der Test-Modus auf `ON` oder `OFF` gesetzt ist

Wenn der Debugger die Kontrolle über eine Sitzung hat, führt er eine oder mehrere der folgenden Funktionen aus:

- Er prüft die Debug-Einträge.
- Er unterbricht ein Natural-Objekt an der Statement-Zeile, für die ein Breakpoint gesetzt wurde.
- Er unterbricht ein Natural-Objekt, wenn sich der Wert einer Variablen, für die ein Watchpoint gesetzt wurde, geändert hat.
- Er zeigt Informationen zu den gefundenen Debug-Einträgen (Watchpoint und/oder Breakpoint) an.
- Er liefert Statistiken zu den aufgerufenen Natural-Objekten.
- Er liefert Statistiken zu den in einem Natural-Objekt aufgerufenen Statements.
- Er unterbricht ein Natural-Objekt, wenn ein Natural-Fehler auftritt. Siehe auch Abschnitt ***Fehlerbehandlung***.

Die folgende Grafik zeigt ein Beispiel für den Ablauf bei der Ausführung eines Objekts mit dem Debugger:



Debug-Einträge (Spies)

Debug-Einträge werden in der Debugger-Umgebung auch als **Spies** bezeichnet. Es stehen zwei Arten von Spies zur Verfügung: Breakpoints und Watchpoints.

Folgende Themen werden behandelt:

- Verwaltung und Validierung der Debug-Einträge
- Namen von Debug-Einträgen
- Anfangszustand oder aktueller Zustand
- Zähler für Debug-Ereignisse

- Kommandos für Debug-Einträge

Verwaltung und Validierung der Debug-Einträge

Debug-Einträge für die aktuelle Debug-Sitzung können mit den entsprechenden Debugger-Maintenance-Funktionen, die in den entsprechenden Abschnitten der Debugger-Dokumentation beschrieben sind, gesetzt, geändert, aufgelistet, angezeigt, aktiviert, deaktiviert und gelöscht werden. Debug-Einträge können auch zur späteren Verwendung gespeichert werden, siehe Abschnitt *Debug Environment Maintenance (Verwaltung der Debug-Umgebung)*.

Die Validierung von Debug-Einträgen erfolgt entweder sofort bei der Definition eines Breakpoints oder Watchpoints auf dem entsprechenden Maintenance-Bildschirm oder während der Programmausführung.

Falls während der Programmausführung eine Validierung fehlschlägt, erscheint im Fenster **Debug Break** die Anmerkung `Check for invalid spy definition` (Auf ungültige Spy-Definition prüfen), siehe ***Debug Break-Fenster***. Außerdem wird der ungültige Breakpoint oder Watchpoint in den relevanten Bildschirmen zur Verwaltung der Breakpoints oder Watchpoints markiert.

Wird ein Debug-Eintrag gesetzt oder geändert, speichert Natural intern die Library, die Datenbankkennung und die Dateinummer, wo sich das Objekt befindet. Das Objekt kann sich in der aktuellen Library oder in einer ihrer Steplibs befinden. Wird später ein Objekt mit demselben Namen aus einer anderen Library ausgeführt, dann wird der entsprechende Debug-Eintrag nicht ausgeführt.

Namen von Debug-Einträgen

Der Debugger vergibt für jeden Debug-Eintrag einen Namen und eine eindeutige Nummer (Spy-Nummer). Der einem Debug-Eintrag zugewiesene Name (auch als Spy-Name bezeichnet) kann entweder ein vom Benutzer angegebener Name oder ein Standardname sein, der vom Debugger erstellt wird. Ein Debug-Eintrag kann über seine Nummer mit den entsprechenden Debugger-Kommandos ausgewählt werden. Wenn mehr als ein Debug-Eintrag an einer bestimmten Statement-Zeile ausgeführt werden muss, werden diese in aufsteigender Reihenfolge ihrer Nummern ausgeführt.

Anfangszustand oder aktueller Zustand

Jeder Debug-Eintrag hat einen Anfangszustand und einen aktuellen Zustand. Mögliche Werte sind `A` (aktiv) und `I` (inaktiv). Der Anfangswert wird angegeben, wenn Sie den Breakpoint oder Watchpoint setzen oder ändern. Er bestimmt den Zustand des Debug-Eintrags beim Start der Umgebung oder nach einem Reset. Während der Debug-Sitzung kann der Zustand mit den Debug-Kommandos `ACTIVATE` und `DEACTIVATE` geändert werden (siehe auch die Syntax-Diagramme in ***Debug-Kommandoübersicht und -syntax***).

Zähler für Debug-Ereignisse

Jeder Debug-Eintrag hat einen Ereigniszähler, der jedes Mal erhöht wird, wenn der Debug-Eintrag ausgeführt wird. Ein Debug-Eintrag wird nicht ausgeführt, wenn der aktuelle Zustand inaktiv ist. Die Ereignisanzahl des Breakpoint oder Watchpoint wird ebenfalls nicht erhöht.

Die Anzahl der Ausführungen eines Debug-Eintrags kann auf zwei Arten eingeschränkt werden:

- Es kann eine Anzahl von Auslassungen angegeben werden, bevor der Debug-Eintrag ausgeführt wird. Der Debug-Eintrag wird dann so lange ignoriert, bis die Ereignisanzahl höher ist als die angegebene Anzahl der Auslassungen.
- Es kann eine maximale Anzahl von Ausführungen angegeben werden, so dass der Debug-Eintrag ignoriert wird, sobald die Ereignisanzahl die angegebene Anzahl der Ausführungen überschreitet.

Kommandos für Debug-Einträge

Zu jedem Debug-Eintrag (Breakpoint oder Watchpoint) können bis zu sechs Debug-Kommandos angegeben werden. Diese Kommandos werden zur Ausführungszeit des Breakpoint oder Watchpoint ausgeführt. Sie können alle Debugger-Kommandos verwenden, die während einer Debug-Unterbrechung angewendet werden können. Das Standardkommando ist das `BREAK`-Kommando, mit dem das **Debug Break**-Fenster angezeigt wird. Siehe folgenden Abschnitt *Debug Break-Fenster*.

Vorsicht: Wenn Sie beim Setzen eines Debug-Eintrags das Kommando `BREAK` löschen und kein Kommando eingeben, das einen Dialog auslöst, gibt es keine Möglichkeit, die Steuerung während einer Programmunterbrechung zu übernehmen.

Debug Break-Fenster

Wenn der Debugger die Kontrolle über eine Sitzung übernimmt, wird ein **Debug Break**-Fenster angezeigt, zum Beispiel:

```
+------------------- Debug Break --------------------+
! Break by breakpoint DEBPGM-ALL                     !
! at line  180 in program DEBPGM (level 1)           !
! in library SAG      in system file (10,32).        !
!                                                    !
!         G   Go                                     !
!         L   List break                             !
!         M   Debug Main Menu                        !
!         N   Next break command                     !
!         R   Run (set test mode OFF)                !
!         S   Step mode                              !
!         V   Variable maintenance                   !
!                                                    !
! Code .. G                                          !
! Note: Check for invalid spy definition.            !
!                                                    !
! PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS    !
+----------------------------------------------------+
```

Das Fenster **Debug Break** zeigt den Typ und den Namen des Debug-Eintrags, der die Unterbrechung verursacht hat (d. h. den Namen des entsprechenden Breakpoint oder Watchpoint), seine Quellcode-Zeilennummer und den Namen des unterbrochenen Natural-Objekts.

Zusätzlich können unten im **Debug Break**-Fenster Meldungen erscheinen, die entweder einen Natural-Fehler anzeigen (siehe auch *Fehler während der Anwendungsausführung* in *Fehlerbehandlung*) und/oder auf die Möglichkeit eines ungültigen Debug-Eintrags hinweisen.

Die Funktionen, die im Fenster Debug Break zur Verfügung stehen, sind in der folgenden Tabelle beschrieben. Weitere Informationen siehe *Kommandos zur Ausführungssteuerung*.

Funktion	Code	Beschreibung
Go	**G**	Fortsetzung der Programmausführung, bis zum nächsten angegebenen Debug-Eintrag.
List break	**L**	Auflistung des Quellcode des zurzeit aktiven Natural-Objekts. Das zuletzt ausgeführte Statement ist hervorgehoben.
Debug Main Menu	**M**	Aufruf des **Debug Main Menu**, in dem Sie alle Funktionen benutzen können, die Sie zur Verwaltung von Debug-Einträgen benötigen, bei denen die Steuerung übernommen werden soll.
Next break command	**N**	Ausführung des nächsten Kommandos, das für den aktuellen Breakpoint oder Watchpoint angegeben ist.
Run (set test mode OFF)	**R**	Fortsetzung der Ausführung des Natural-Objekts mit ausgeschaltetem Test-Modus.
Step mode	**S**	Fortsetzung der Ausführung des Natural-Objekts im Step-Modus (Einzelschritt-Modus).
Variable maintenance	**V**	Anzeige der Variablen in dem zurzeit aktiven Natural-Objekt und Änderung des Inhalts dieser Variablen.

4 Debugger starten

- Debugger unter Natural Security 48
- Voraussetzungen für den Betrieb 48
- Debugger aufrufen 49
- Standard-Objekt 50

Dieser Abschnitt beschreibt Grundvoraussetzungen für den Betrieb und enthält eine grobe Richtlinie für die Vorgehensweise bei der Planung der Anwendung des Debuggers.

Debugger unter Natural Security

Die Benutzung des Debuggers kann durch Natural Security kontrolliert werden:

- Sie können den Debugger gegen nicht autorisierte Benutzung schützen, indem Sie den Gebrauch des Systemkommandos `TEST`, das den Debugger aufruft, unterbinden, siehe *Command Restrictions* im Abschnitt *Library Maintenance* in der *Natural Security*-Dokumentation.
- Sie können die Benutzung des Debuggers unterbinden oder einschränken, siehe Abschnitt *Components of an Environment Profile* in der *Natural Security*-Dokumentation.

Voraussetzungen für den Betrieb

Der Debugger wird nur aufgerufen, wenn Sie ein katalogisiertes Objekt ausführen, das in der aktuellen Library in der aktuellen Natural-Systemdatei befindet. Der Debugger wird nicht aufgerufen, wenn Sie im Arbeitsbereich enthaltenen Quellcode mit dem Kommando `RUN` ausführen.

Ein effizientes und korrektes Debugging setzt voraus, dass der Quellcode im Source-Objekt mit dem kompilierten Quellcode im katalogisierten Objekt übereinstimmt, was mit dem Systembefehl `STOW` gewährleistet werden kann. Wenn Sie ein Source-Objekt ändern, nachdem Sie es katalogisiert haben, besteht die Möglichkeit, dass ein Debug-Eintrag (Breakpoint oder Watchpoint) nicht richtig funktioniert, weil sich das referenzierte Statement oder die referenzierte Variable geändert hat oder nicht mehr existiert. Wenn der Debugger feststellt, dass ein Source-Objekt einen früheren Zeitstempel hat als das entsprechende katalogisierte Objekt erscheint die Warnung `Time stamps of source and cataloged object do not match.` (Zeitstempel des Source-Objekts und des katalogisierten Objekts passen nicht zueinander).

Der Debugger untersucht alle Natural-Objekte, die in der aktuellen Library oder in einer ihrer Steplibs enthalten sind. Der Debugger untersucht keine Natural-Objekte, die in der Natural-System-Library `SYSLIB` oder `SYSLIBS` gespeichert sind.

Für die Verwendung des Debuggers gilt folgende Einschränkung:

- Der Debugger kann nur bei Objekten der Natural-Version 2.3 und höher, jedoch nicht bei Natural-Objekten, die mit einer früheren Version katalogisiert wurden, angewendet werden.
- Der Debugger unterstützt nur Debug-Umgebungen, die mit Natural-Version 2.3 und höher erstellt wurden. Debug-Umgebungen, die mit einer früheren Version erstellt wurden, werden ignoriert. Weitere Informationen zu Debug-Umgebungen siehe *Debug Environment Maintenance (Verwaltung der Debug-Umgebung)*.

Batch-Verarbeitung

Obwohl der Debugger hauptsächlich für den interaktiven Gebrauch im Online-Modus ausgelegt ist, können die Debugger-Kommandos auch zur Batch-Ausführung benutzt werden, z.B. zum Setzen von Breakpoints oder Watchpoints.

Anmerkung: Bei der Batch-Verarbeitung gelten Einschränkungen, die zur Folge haben können, dass ein Debugger-Kommando zurückgewiesen wird. Beispielsweise unterstützt der Debugger nicht die Kommandos `++` und `+4`.

Beispiel für das Erstellen und Drucken von Statistiken im Batch-Modus

Das folgende Beispiel demonstriert den Gebrauch von Debugger-Direktkommandos im Batch-Modus, um einen Bericht über alle Aufrufstatistiken zu erstellen und zu drucken:

```
//NATBATCH EXEC PGM=NATBAT42,
//   PARM=('INTENS=1,IM=D,CF=$,PRINT=((1-2),AM=STD)')
//STEPLIB  DD DISP=SHR,DSN=NATURAL.V2.TEST.NUCLEUS
//CMPRINT  DD SYSOUT=X
//SYSOUT   DD SYSOUT=X
//CMPRT01  DD SYSOUT=X
//CMSYNIN  DD *
LOGON DEBUGLIB
TEST PROFILE
,,,,CMPRT01
,,,,,,$K3
,,$K3
TEST ON
TEST SET XSTAT COUNT
DEBUG2P
TEST PRINT XSTAT
FIN
/*
```

Debugger aufrufen

Um den Debugger aufzurufen:

1 Erstellen Sie eine Debug-Umgebung für ein Natural-Objekt oder eine Natural-Anwendung:

- Rufen Sie das Hauptmenü **Debug Main Menu** auf, indem Sie das Natural-Systemkommando `TEST` eingeben.

  Oder: Setzen Sie aus einer laufenden Anwendung das Terminalkommando `%<TEST` ab.
- Benutzen Sie die Funktionen des Hauptmenüs **Debug Main Menu**, um für ein Natural-Objekt oder eine Natural-Anwendung anzugeben:

Debug Environment Maintenance (Verwaltung der Debug-Umgebung)
Spy Maintenance (Verwaltung der Debug-Einträge)
Breakpoint Maintenance (Verwaltung der Breakpoints)
Watchpoint Maintenance (Verwaltung der Watchpoints)
Call Statistics Maintenance (Statistiken über gerufene Objekte)
Statement Execution Statistics Maintenance (Statistiken über ausgeführte Statement-Zeilen)
Variable Maintenance
List Object Source

2 Aktivieren Sie den Debugger:

- Geben Sie an einer Eingabeaufforderung das Kommando `TEST ON` ein.
- Oder: Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `T` ein.

3 Führen Sie das Natural-Objekt oder die Natural-Anwendung aus.

Der Debugger hält die Programmausführung an den angegebenen Debug-Einträgen an und ruft das **Debug Break**-Fenster auf.

Um den Debugger zur Fehlerbehandlung aufzurufen:

- Setzen Sie zu Beginn der Sitzung den Natural-Profilparameter `DBGERR` auf `ON`.

  Siehe auch *DBGERR - Automatischer Debugger-Start bei Laufzeitfehler* in der Natural-*Parameter-Referenz*-Dokumentation.

  Oder:

  Geben Sie während der Sitzung an einer Eingabeaufforderung das Kommando `TEST ON` ein oder geben Sie in einem **Debug Maintenance**-Hauptmenü den Funktionscode `T` ein.

  Der Debugger ruft das **Debug Break**-Fenster auf, wenn ein Natural-Fehler auftritt.

Siehe auch Abschnitt ***Fehlerbehandlung***.

Standard-Objekt

Die Maintenance-Funktionen des Debuggers, wie sie in den entsprechenden Abschnitten beschrieben sind, beziehen sich auf Objekte, die Sie entweder in den entsprechenden Namensfeldern von Menüs oder mit Direktkommandos angeben. Wenn Sie keinen Objektnamen angeben, nimmt der Debugger standardmäßig den Namen des aktuellen Objekts an, wie er im Feld **Object** oben rechts im Hauptmenü **Debug Main Menu** angezeigt wird. Bei Angabe eines Standardobjekts, ist kein Objektname in Direktkommandos und Menüoptionen erforderlich, die zur Angabe von

Breakpoints oder Watchpoints verwendet werden. Um das Standardobjekt zu ändern, siehe Syntax des Kommandos SET im Abschnitt ***Debug-Kommandoübersicht und -syntax***.

5 Test-Modus ein- und ausschalten

Um eine zuvor erstellte Debug-Umgebung zu aktivieren, müssen Sie den Test-Modus auf `ON` setzen.

Um den Test-Modus auf ON oder OFF zu setzen:

- Geben Sie in einem **Debug Maintenance**-Hauptmenü den Funktionscode `T` ein, um den Test-Modus auf `ON` (Ein) oder `OFF` (Aus) zu setzen.

  Oder:

  Geben Sie eines der folgenden Direktkommandos ein:

  ```
  TEST ON
  ```

  oder

  ```
  TEST OFF
  ```

Wenn Sie bei eingeschaltetem Test-Modus (`ON`) ein Natural-Objekt ausführen, prüft der Debugger kontinuierlich alle Debug-Einträge ab, ob eventuell eine Maßnahme erforderlich ist.

Wenn Sie bei ausgeschaltetem Test-Modus (`OFF`) ein Natural-Objekt ausführen, werden alle Debug-Einträge ignoriert.

Das Kommando `TEST` und somit die gesamte Anwendung können durch Natural Security geschützt werden, siehe *Command Restrictions* im Abschnitt *Library Maintenance* in der *Natural Security*-Dokumentation.

6 Debug Environment Maintenance (Verwaltung der Debug-Umgebung)

- Test-Modus für Debug-Umgebung ON/OFF setzen 57
- Debug-Umgebung laden 57
- Debug-Umgebung speichern 57
- Debug-Umgebung zurücksetzen 58
- Debug-Umgebung löschen 59
- Debug-Umgebung in verschiedenen Libraries verwalten 59

Da eine Debug-Umgebung hauptsächlich aus Debug-Einträgen besteht, wird sie erstellt, indem Breakpoints und Watchpoints gesetzt werden, wie in den entsprechenden Maintenance-Abschnitten beschrieben.

Nachdem eine Debug-Umgebung erstellt worden ist, kann sie zur anschließenden Benutzung gespeichert werden. Die Datei, in der Debug-Umgebungen gespeichert werden, kann mit dem Debugger-Kommando `PROFILE` angegeben werden (siehe *Kommandos zum Navigieren und Anzeigen von Informationen*). Sie können eine Debug-Umgebung auch löschen oder ihre Zähler auf die Anfangswerte zurücksetzen.



Anmerkung: Siehe auch die Einschränkungen, die für die Benutzung gelten, im Abschnitt *Voraussetzungen für den Betrieb*.

Die folgenden Elemente sind ebenfalls Teil einer Debug-Umgebung und werden daher jedes Mal gespeichert oder geladen, wenn Sie eine Debug-Umgebung speichern oder laden:

- Die Test-Modus-Einstellung (`ON` oder `OFF`),
- alle Optionen, die mit dem Debugger-Kommando `PROFILE` gesetzt werden können (mit Ausnahme der Datei zum Laden und Speichern von Debug-Umgebungen),
- die Einstellungen der Funktion **Statement execution statistics maintenance** (`ON`, `OFF` oder `COUNT`).

Um die Funktion Debug Environment Maintenance aufzurufen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `E` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  EM
  ```

  Das Menü **Debug Environment Maintenance** erscheint.

Dieser Abschnitt beschreibt die Funktionen, die im Menü **Debug Environment Maintenance** zur Verfügung stehen und enthält Anleitungen, wie Sie Maintenance-Funktionen in verschiedenen Libraries ausführen können.

Bei jeder gewählten Funktion müssen Sie den Namen der Debug-Umgebung angeben, die Sie verwalten wollen.

Test-Modus für Debug-Umgebung ON/OFF setzen

Siehe Abschnitt *Test-Modus ein- und ausschalten*.

Debug-Umgebung laden

Um eine Debug-Umgebung aus Ihrer Benutzer-Systemdatei (FUSER) zu laden:

- Geben Sie im Menü **Debug Environment Maintenance** den Funktionscode L und den Namen der Umgebung ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  LOAD ENVIRONMENT name
  ```

  Die angegebene Debug-Umgebung wird geladen.

Falls Sie keinen Namen angeben, wird die Standardumgebung mit dem Namen Noname geladen.

Um eine Liste aller verfügbaren Debug-Umgebungen zu erhalten, können Sie einen Stern (*) eingeben. In der Liste können Sie die gewünschte Umgebung mit dem Zeilenkommando LO markieren, um sie in den Debug-Pufferspeicher zu laden, oder mit dem Zeilenkommando DE, um sie zu löschen.

Debug-Umgebung speichern

Um eine Debug-Umgebung zu speichern:

- Geben Sie im Menü **Debug Environment Maintenance** den Funktionscode S und den Namen der Umgebung ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  SAVE ENVIRONMENT name
  ```

Die angegebene Umgebung wird **zurückgesetzt** (siehe unten) und in der Datei gespeichert, die mit dem Debugger-Kommando `PROFILE` als Speicherort angegeben wird (siehe Abschnitt *Kommandos zum Navigieren und Anzeigen von Informationen*).

Wenn Sie keinen Namen angeben, wird die Standardumgebung mit dem Namen `Noname` gespeichert.

Falls bereits eine Debug-Umgebung mit dem angegebenen Namen existiert, werden Sie aufgefordert, zu bestätigen, dass die alte Umgebung überschrieben wird.

Debug-Umgebung zurücksetzen

Die Debug-Umgebung sollte vor jedem Testlauf zurückgesetzt werden. Das Zurücksetzen der Umgebung bewirkt Folgendes:

- Die aktuellen Zustände aller Debug-Einträge werden auf ihren Anfangszustand gesetzt.
- Alle Ereigniszählungen werden auf Null gesetzt.
- Die Aufrufstatistiken im Debug-Pufferspeicher werden gelöscht. Weitere Informationen siehe Abschnitt *Call Statistics Maintenance (Statistiken über gerufene Objekte)*.

Um eine Debug-Umgebung zurückzusetzen:

- Geben Sie im Menü **Debug Environment Maintenance** den Funktionscode `R` und den Namen der Umgebung ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  RESET ENVIRONMENT name
  ```

  Die angegebene Debug-Umgebung wird zurückgesetzt.

Wenn Sie keinen Umgebungsnamen angeben, wird die aktuelle Debug-Umgebung zurückgesetzt.

Debug-Umgebung löschen

Um eine Debug-Umgebung zu löschen:

1 Geben Sie im Menü **Debug Environment Maintenance** den Funktionscode `D` und den Namen der Umgebung ein.

Oder:

Geben Sie folgendes Direktkommando ein:

```
DELETE ENVIRONMENT name
```

Ein Bestätigungsfenster erscheint.

2 Geben Sie im Bestätigungsfenster `Y` (Yes) ein, um die Löschung zu bestätigen.

Die angegebene Debug-Umgebung wird gelöscht.

Wenn Sie keinen Umgebungsnamen angeben, wird die aktuelle Debug-Umgebung gelöscht.

Debug-Umgebung in verschiedenen Libraries verwalten

Mit den Funktionen der Natural-Utility `SYSMAIN` können Sie Debug-Umgebungen zwischen verschiedenen Libraries und/oder Systemdateien kopieren oder verschieben und eine Debug-Umgebung löschen, auflisten oder umbenennen.

Wenn eine Debug-Umgebung von einer Library in eine andere verschoben oder kopiert worden ist, beziehen sich die Breakpoints und Watchpoints immer noch auf die alte (Quell-)Library. Um die Debug-Umgebung an die neue (Ziel-)Library anzupassen, müssen Sie die entsprechenden Breakpoints oder Watchpoints ändern (siehe auch ***Breakpoint ändern*** im Abschnitt *Breakpoint Maintenance (Verwaltung der Breakpoints)* bzw. ***Watchpoint ändern*** im Abschnitt *Watchpoint Maintenance (Verwaltung der Watchpoints)*). Beim Ausführen der `Modify`-Funktion brauchen Sie keine der existierenden Definitionen zu verändern. Wenn Sie das Kommando `Save` (`PF5`) zum Speichern ausführen, ändert sich die Library-Referenz automatisch auf die neue Library. Sie erkennen die Änderung am Eintrag im Feld **Library** im Bildschirm **Modify Breakpoint** bzw. **Modify Watchpoint**.

Verwandtes Thema:

Debug-Umgebungen bearbeiten - *SYSMAIN Utility, Debugger und Dienstprogramme (Utilities)*-Dokumentation

7 Spy Maintenance (Verwaltung der Debug-Einträge)

- Test-Modus für Debug-Einträge ON/OFF setzen 62
- Spy-Funktion aktivieren 62
- Spy deaktivieren 63
- Debug-Einträge (Spies) löschen 63
- Debug-Einträge (Spies) anzeigen 63
- Debug-Einträge (Spies) ändern 64

Mit der Funktion **Spy Maintenance** (Verwaltung der Debug-Einträge) können Sie alle Debug-Einträge („Spies"), d.h. Breakpoints *und* Watchpoints, aktivieren, deaktivieren, auflisten oder löschen. Zudem bietet Ihnen die Funktion **Spy Maintenance** einen alternativen Zugang zu den Bildschirmen zur Verwaltung der Breakpoints und Watchpoints. Die Beschreibungen dieser Bildschirme finden Sie in den Abschnitten *Watchpoint Maintenance (Verwaltung der Watchpoints)* und *Breakpoint Maintenance (Verwaltung der Breakpoints)*.

Um die Funktion Spy Maintenance aufzurufen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `S` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  SM
  ```

  Das Menü **Spy Maintenance** erscheint.

Die Funktionen, die im Menü **Spy Maintenance** zur Verfügung stehen, werden in den folgenden Abschnitten beschrieben:

Test-Modus für Debug-Einträge ON/OFF setzen

Siehe Abschnitt *Switch Test Mode On and Off*.

Spy-Funktion aktivieren

Um den aktuellen Status angegebener Debug-Einträge (Spies) auf aktiv zu setzen:

- Geben Sie im Menü **Spy Maintenance** den Funktionscode `A` und eine Spy-Nummer *oder* einen Spy-Namen ein.

  Oder:

  Benutzen Sie das Direktkommando `ACTIVATE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn Sie die Spy-Nummer *oder* den Spy-Namen nicht eingeben, werden *alle* Spies (Breakpoints und Watchpoints) aktiviert.

Spy deaktivieren

Um den aktuellen Status angegebener Debug-Einträge (Spies) auf inaktiv zu setzen:

- Geben Sie im Menü **Spy Maintenance** den Funktionscode `B` und eine Spy-Nummer *oder* einen Spy-Namen ein.

  Oder:

  Benutzen Sie das Direktkommando `DEACTIVATE`. Syntax-Beschreibung siehe Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Wenn Sie die Spy-Nummer *oder* den Spy-Namen nicht eingeben, werden *alle* Spies (Breakpoints und Watchpoints) deaktiviert.

Debug-Einträge (Spies) löschen

Um angegebene Debug-Einträge (Spies) zu löschen:

- Geben Sie im Menü **Spy Maintenance** den Funktionscode `C` und eine Spy-Nummer *oder* einen Spy-Namen ein.

  Oder:

  Benutzen Sie das Direktkommando `DELETE`. Syntax-Beschreibung siehe Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Wenn Sie die Spy-Nummer *oder* den Spy-Namen nicht eingeben, werden *alle* Spies (Breakpoints und Watchpoints) gelöscht.

Debug-Einträge (Spies) anzeigen

Um angegebene Debug-Einträge (Spies) anzuzeigen:

- Geben Sie im Menü **Spy Maintenance** den Funktionscode `D` und eine Spy-Nummer *oder* einen Spy-Namen ein.

  Oder:

Benutzen Sie das Direktkommando `DISPLAY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Ist der angegebene Debug-Eintrag (Spy) eindeutig, erscheint der Bildschirm **Display Breakpoint** bzw. **Display Watchpoint**, und es werden alle Angaben zu diesem Breakpoint bzw. Watchpoint angezeigt.

Wenn der angegebene Debug-Eintrag (Spy) nicht eindeutig ist, wird eine Liste der betreffenden Debug-Einträge (Spies) angezeigt. In der Liste können Sie einen Debug-Eintrag (Spy) mit einem Zeilenkommando markieren, um folgende Aktionen auszuführen:

Kommando	Aktion
`AC`	Debug-Eintrag aktivieren
`DA`	Debug-Eintrag deaktivieren
`DI`	Debug-Eintrag anzeigen
`MO`	Debug-Eintrag ändern
`DE`	Debug-Eintrag löschen

Wenn Sie die Spy-Nummer *oder* den Spy-Namen nicht eingeben, werden *alle* Spies (Breakpoints und Watchpoints) angezeigt.

Debug-Einträge (Spies) ändern

Um angegebene Debug-Einträge (Spies) zu ändern:

- Geben Sie im Menü **Spy Maintenance** den Funktionscode `M` und eine Spy-Nummer *oder* einen Spy-Namen ein.

  Oder:

  Benutzen Sie das Direktkommando `MODIFY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Ist der angegebene Debug-Eintrag (Spy) eindeutig, erscheint der Bildschirm **Modify Breakpoint** bzw. **Modify Watchpoint**, und die Angaben zu diesem Breakpoint bzw. Watchpoint können geändert werden.

  Wenn der angegebene Debug-Eintrag (Spy) nicht eindeutig ist, wird eine Liste der betreffenden Debug-Einträge (Spies) angezeigt. In der Liste können Sie einen Debug-Eintrag (Spy) mit einem Zeilenkommando markieren, um folgende Aktionen auszuführen:

Kommando	Aktion

AC	Debug-Eintrag aktivieren
DA	Debug-Eintrag deaktivieren
DI	Debug-Eintrag anzeigen
MO	Debug-Eintrag ändern
DE	Debug-Eintrag löschen

Wenn Sie die Spy-Nummer *oder* den Spy-Namen nicht eingeben, werden *alle* Spies (Breakpoints und Watchpoints) zum Auswählen und Ändern angezeigt.

8 Breakpoint Maintenance (Verwaltung der Breakpoints)

- Verwendungsbedingungen 68
- Test-Modus für Breakpoints ON/OFF setzen 69
- Breakpoint aktivieren 70
- Breakpoint deaktivieren 70
- Breakpoint löschen 70
- Breakpoint anzeigen 71
- Breakpoint ändern 73
- Breakpoint setzen 74
- Felder und Spalten in Breakpoint-Bildschirmen 75

Ein Breakpoint (Haltepunkt) bewirkt die Unterbrechung der Ausführung eines Natural-Objekts an einer bestimmten Statement-Zeile.

In diesem Abschnitt erfahren Sie, wie und wann Sie Breakpoints setzen.

Beachten Sie, dass Sie die hier beschriebenen Verwaltungsfunktionen auch aus einem Objekt-Quellcode heraus aufrufen können, indem Sie die Funktion **List object source** benutzen (siehe *List Object Source (Objekt-Quellcode anzeigen)*)

Um die Breakpoint Maintenance (Verwaltung der Breakpoints) aufzurufen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `B` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  BM
  ```

  Das Menü **Breakpoint Maintenance** erscheint.

Dieses Kapitel beschreibt Bedingungen für die Benutzung der Breakpoint-Verwaltung, die im Menü **Breakpoint Maintenance** zur Verfügung stehenden Funktionen und die in einem Breakpoint-Bildschirm vorhandenen Felder und Spalten.

Verwendungsbedingungen

Einen Breakpoint können Sie setzen, indem Sie den Namen des zu verarbeitenden Natural-Objekts und im Quellcode des Objekts die Zeilennummer angeben, wo der Breakpoint ausgeführt werden soll.

Sobald ein Breakpoint angegeben wurde, bleibt er für die gesamte Natural-Sitzung gesetzt, es sei denn, Sie löschen ihn.

Ein Breakpoint bezieht sich auf eine bestimmte Zeilennummer im Quellcode. Deshalb kann eine nachträgliche Änderung des Quellcode zur Folge haben, dass der Breakpoint nicht mehr für das gewünschte Statement gilt und somit das Natural-Objekt nicht mehr an der gewünschten Stelle unterbrochen wird. Um dieses Problem bei Programmschleifen zu umgehen, können in diesen Schleifen Statement-Labels gesetzt werden (siehe auch Beispiel mit Statement-Labels im *Leitfaden zur Programmierung*). Für diese Statement-Labels gesetzte Breakpoints werden auf die korrekte Zeilennummer angepasst, wenn Statement-Zeilen eingefügt oder gelöscht werden.

Der eindeutige Bezeichner für einen Breakpoint ist die vom Debugger zugewiesene Spy-Nummer.

An folgenden Stellen können keine Breakpoints gesetzt werden:

- In Kommentarzeilen,
- in einer beliebigen Statement-Zeile außer der ersten Zeile (wenn ein einzelnes Statement mehr als eine Programmzeile belegt,
- in Zeilen, die lediglich eines der folgenden Statements enthalten:
  - `AT BREAK OF`
  - `AT END OF DATA`
  - `AT END OF PAGE`
  - `AT START OF DATA`
  - `AT TOP OF PAGE`
  - `BEFORE BREAK`
  - `DECIDE`

    Siehe auch die Einschränkungen, die für die Benutzung gelten, im Abschnitt ***Voraussetzungen für den Betrieb***.
  - `DEFINE SUBROUTINE`
  - `DEFINE WINDOW`
  - `FORMAT`
  - `IF NO RECORDS FOUND`
  - `ON ERROR`
  - `OPTIONS`

Ob es möglich ist oder nicht, Breakpoints für Zeilen zu setzen, die mit dem Natural Optimizer Compiler kompiliert wurden, ist abhängig von der Einstellung der Option `NODBG` im `OPTIONS`-Statement. Weitere Informationen siehe *Natural-Statement OPTIONS* im Abschnitt *Optimizer Compiler aktivieren* in der *Natural Optimizer Compiler*-Dokumentation.

Test-Modus für Breakpoints ON/OFF setzen

Siehe Abschnitt ***Test-Modus ein- und ausschalten***.

Breakpoint aktivieren

Um den aktuellen Status angegebener Breakpoints auf aktiv zu setzen:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode `A` und einen Objektnamen und/oder eine Zeilennummer ein.

  Oder:

  Benutzen Sie das Direktkommando `ACTIVATE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Wenn Sie den Objektnamen oder eine Zeilennummer nicht eingeben, werden *alle* Breakpoints aktiviert.

Breakpoint deaktivieren

Um den aktuellen Status angegebener Breakpoints auf inaktiv zu setzen:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode `B` und einen Objektnamen und/oder eine Zeilennummer ein.

  Oder:

  Benutzen Sie das Direktkommando `DEACTIVATE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Wenn Sie den Objektnamen oder eine Zeilennummer nicht eingeben, werden *alle* Breakpoints deaktiviert.

Breakpoint löschen

Um angegebene Breakpoints zu löschen:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode `C` und einen Objektnamen und/oder eine Zeilennummer ein.

  Oder:

Benutzen Sie das Direktkommando DELETE. Syntax-Beschreibung siehe Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Wenn Sie den Objektnamen oder eine Zeilennummer nicht eingeben, werden *alle* Breakpoints gelöscht.

Breakpoint anzeigen

Um einen Breakpoint anzuzeigen:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode D, einen Objektnamen und/oder eine Zeilennummer ein.

  Wenn Sie den Objektnamen nicht angeben, wird das **Standard-Objekt** (falls angegeben) benutzt.

  Oder:

  Benutzen Sie das Direktkommando DISPLAY. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Wenn für das angegebene Objekt und Zeilennummer ein Breakpoint gesetzt worden ist, erscheint der Bildschirm **Display Breakpoint** mit allen Breakpoint-Definitionen, zum Beispiel:

```
11:16:12              ***** NATURAL TEST UTILITIES *****            2006-02-07
Test Mode ON                  - Display Breakpoint -              Object

Spy number ..............   1
Initial state ........... active         Current state .. active
Breakpoint name ......... BRK0130        DBID/FNR ....... 10/32
Object name ............. DEBPGM1        Library ........ SAG
Line number ............. 0130
Label ...................
Skips before execution ..     0
Max number executions ...     0
Number of activations ...     0
Error in definition ..... - none -

Commands ... BREAK









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

Falls kein eindeutiger Breakpoint gefunden wird, erscheint der Bildschirm **List Breakpoints** (Beschreibung siehe unten).

Beschreibung der Felder im Bildschirm **Display Breakpoint** und der Spalten im Bildschirm **List Breakpoints** siehe *Felder und Spalten in Breakpoint-Bildschirmen*.

Um Breakpoints aufzulisten:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode `D`, einen Objektnamen oder eine Zeilennummer ein. Sie können Stern-Notation (*) benutzen, um einen Bereich von Objektnamen anzugeben, zum Beispiel: `ABC*`. Wenn Sie nur einen Stern (*) eingeben, werden alle Objektnamen ausgewählt. Wenn Sie keinen Objektnamen angeben, wird das **Standard-Objekt** (falls angegeben) benutzt.

  Oder:

  Benutzen Sie das Direktkommando `DISPLAY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Der Bildschirm **List Breakpoints** erscheint. Darin werden alle Breakpoints aufgelistet, die für das bzw. die Objekte oder eine Zeilennummer gesetzt sind, zum Beispiel:

```
11:41:56              ***** NATURAL TEST UTILITIES *****             2006-01-30
Test Mode ON                    - List Breakpoints -                Object
                                                                          All
Co No. BP Name      Library  Object   Line  DBID   FNR Stat Skips Execs Count E
       *____________ *_______ *_______ 0000              I  C
__   1 BRK0130      SAG      DEBPGM1  0130    10    32 A  A      0     0     0
__   2 BRKPGM3-END  SAG      DEBPGM3  END     10    32 A  A      0     0     0
__   3 BRKPGM3-300  SAG      DEBPGM3  0300    10    32 A  A      0     0     0
__   4 BRKPGM2-400  SAG      DEBPGM2  0400    10    32 A  A      0     0     0
__   5 BRKPGM2-430  SAG      DEBPGM2  0430    10    32 A  A      0     0     0
__   6 BRKPGM1-END  SAG      DEBPGM1  END     10    32 A  A      0     0     0
__   7 BRKPGM1-ALL  SAG      DEBPGM1  ALL     10    32 A  A      0     0     0









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

Die Liste ist in aufsteigender Reihenfolge nach den Debug-Eintrag-Nummern (**Spy-Nummern**) sortiert, die in der Spalte **No.** enthalten sind.

Informationen zu den Spalten im Bildschirm **List Breakpoints** und zu den Zeilenkommandos, die bei einem Listeneintrag ausgeführt werden können, siehe *Felder und Spalten in Breakpoint-Bildschirmen*.

Breakpoint ändern

Um einen Breakpoint zu ändern:

1 Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode `M`, einen Objektnamen und eine Zeilennummer ein. Wenn Sie keinen Objektnamen angeben, wird das **Standard-Objekt** (falls angegeben) benutzt.

Oder:

Benutzen Sie das Direktkommando `MODIFY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn ein eindeutiger Breakpoint angegeben wurde, erscheint der Bildschirm **Modify Breakpoint**, in dem Sie die Feldeinträge ändern können. Beschreibung der Felder im Bildschirm **Modify Breakpoint** siehe *Felder und Spalten in Breakpoint-Bildschirmen*.

Wird kein eindeutiger Breakpoint gefunden, dann erscheint der Bildschirm **List Breakpoints** (siehe Abschnitt *Breakpoint anzeigen*).

2 Wenn Sie die Bearbeitung der Breakpoint-Definitionen beendet haben, drücken Sie PF3 (Exit) oder PF5 (Save), um die Änderungen zu speichern. Informationen zu Validierungen an Debug-Einträgen siehe Abschnitt *Verwaltung und Validierung der Debug-Einträge*.

Wenn Sie PF12 (Canc) drücken, wird die Bearbeitung abgebrochen und der Breakpoint bleibt unverändert.

Breakpoint setzen

Um einen Breakpoint für eine Sitzung zu setzen:

- Geben Sie im Menü **Breakpoint Maintenance** den Funktionscode S, einen Objektnamen und/oder eine Zeilennummer ein.

  Oder:

  Benutzen Sie das Direktkommando SET. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Wenn Sie keinen Objektnamen, sondern eine gültige Zeilennummer angeben, wird der Name des **Standard-Objekts** (falls angegeben) benutzt. Falls kein Standard-Objekt angegeben ist, erscheint ein Auswahlfenster, in dem alle in der aktuellen Library enthaltenen Objekte angezeigt werden.

  Ein für einen Copycode gesetzter Breakpoint kann jedoch nur dann auf Gültigkeit geprüft werden, wenn ein Programm ausgeführt wird, das den Copycode enthält. Informationen zu Validierungen an Debug-Einträgen siehe Abschnitt *Verwaltung und Validierung der Debug-Einträge*.

  Der Breakpoint erhält das Standard-Kommando (BREAK), sein Anfangsstatus und sein aktueller Status werden auf aktiv gesetzt und es werden keine Ausführungseinschränkungen angegeben. Achtung: Wenn Sie beim Setzen eines Breakpoint das Kommando BREAK löschen und kein Kommando eingeben, das einen Dialog öffnet, hat der Debugger keine Möglichkeit, während der Programmunterbrechung die Steuerung zu übernehmen.

Felder und Spalten in Breakpoint-Bildschirmen

Die folgende Tabelle enthält die Beschreibung der Felder in den Bildschirmen **Display Breakpoint** und **Modify Breakpoint** und die Spalten im Bildschirm **List Breakpoints**:

Feld	Spalte	Erläuterung
Test Mode		Gibt an, ob der Test-Modus auf `ON` oder `OFF` gesetzt ist.
Object		Zeigt den Namen des **Standard-Objekts** (siehe *Debugger starten*), falls angegeben.
	Co	Feld zur Eingabe eines der folgenden Zeilenkommandos: **Kommando Aktion** `AC` Breakpoint aktivieren `DA` Breakpoint deaktivieren `DI` Breakpoint anzeigen `MO` Breakpoint ändern `DE` Breakpoint löschen `?` Gültige Zeilenkommandos auflisten `.` Breakpoint-Bildschirm verlassen
Spy number	**No.**	Eine eindeutige Nummer, die vom Debugger beim Setzen des Breakpoint zugewiesen wird.
Initial state	**Stat I**	Gibt den Anfangszustand (Initial) und den aktuellen Zustand (Current) des Breakpoint an: Aktiv (`A`) oder Inaktiv (`I`) an.
Current state	**Stat C**	
Breakpoint name	**BP Name**	Der Name des Breakpoint. Gültige Werte: 1 bis 12 Zeichen. Der Standard-Name für einen Breakpoint besteht aus dem Namen des Objekts und der Zeilennummer.
DBID/FNR	**DBID**	Die Datenbankkennung (DBID) und die Dateinummer (FNR) der Systemdatei, in der das Natural-Objekt gespeichert ist.
	FNR	
Library	**Library**	Der Name der Library, die das Objekt enthält.
Object name	**Object**	Der Name des Objekts, das in der aktuellen Library oder in einer ihrer Steplibs verfügbar ist.
Line number	**Line**	Die Zeilennummer eines Statement im Quellcode des Objekts. Siehe auch *Verwendungsbedingungen*. Als Zeilennummern können Sie auch `BEG`, `END` oder `ALL` angeben: `BEG` Gibt den Breakpoint an, der die Programmausführung bei dem ersten Statement, das in einem Objekt ausgeführt wird, unterbrechen soll.

Feld	Spalte	Erläuterung
		BEG-Breakpoints können nicht für Copycode angegeben werden. END Gibt den Breakpoint an, der die Programmausführung bei dem letzten Statement, das in einem Objekt ausgeführt wird, unterbrechen soll, zum Beispiel, ein END- oder ein FETCH-Statement. END-Breakpoints können nicht für Copycode angegeben werden. ALL Gibt an, dass ein Breakpoint die Programmausführung bei jeder Zeile, die ein ausführbares Statement enthält, unterbrechen soll.
Label		Bezieht sich auf ein zuvor im Quellcode eines Objekts gesetztes Label für Statements, die Verarbeitungsschleifen definieren, siehe auch *Systemvariablen anzeigen* weiter oben. Gültige Werte: 1 bis 32 Zeichen.
Skips before execution	**Skips**	Legt fest, dass der Breakpoint erst dann ausgeführt werden soll, wenn die entsprechende Statement-Zeile eine bestimmte Anzahl von Malen ausgeführt wurde. Gültige Werte: 0 (Standardwert) bis 32767.
Max number executions	**Execs**	Ein beliebiger Wert größer als Null (0) legt die maximale Anzahl der Breakpoint-Ausführungen fest. Gültige Werte: 0 (Standardwert) bis 32767.
Number of activations	**Count**	Gibt an, wie viele Male ein Breakpoint für die betreffende Statement-Zeile aktiviert wurde. Der Zähler wird zurückgesetzt, wenn das Programm auf Level 1 gestartet wird.
Error in definition	**E**	Zeigt an, dass die Statement-Zeile in der Breakpoint-Definition während der Programmausführung nicht in dem katalogisierten Objekt gefunden werden kann. Dieser Fehler kann verursacht werden, wenn ein Objekt während des Debugging-Vorgangs geändert und neu katalogisiert wird.
Commands		Bis zu sechs Debug-Kommandos. Geben Sie jeweils ein Kommando pro Zeile ein. Eine Zusammenfassung der zur Verfügung stehenden Kommandos finden Sie im Abschnitt *Debug-Kommandoübersicht und -syntax*. **Vorsicht:** Wenn Sie beim Setzen eines Breakpoint das Kommando BREAK löschen und kein Kommando eingeben, das einen Dialog öffnet, hat der Debugger keine Möglichkeit, während der Programmunterbrechung die Steuerung zu übernehmen.

9 Watchpoint Maintenance (Verwaltung der Watchpoints)

- Test-Modus für Watchpoints ON/OFF setzen 79
- Watchpoint aktivieren 79
- Watchpoint deaktivieren 79
- Watchpoint löschen 80
- Watchpoint anzeigen 80
- Watchpoint ändern 83
- Watchpoint setzen 84
- Felder und Spalten in Watchpoint-Bildschirmen 86

Ein Watchpoint ist ein Haltepunkt, bei dem die Ausführung eines Natural-Objekts immer dann unterbrochen wird, wenn sich der Wert einer Variablen ändert.

Darüber hinaus können Sie die Unterbrechung, so wie im Abschnitt ***Watchpoint-Operatoren*** beschrieben, von einer Bedingung mit Bezug zu einem spezifischen Variablenwert abhängig machen (siehe auch *Watchpoint setzen* weiter unten).

Durch die Verwendung von Watchpoints können Sie unbeabsichtigte Änderungen von Variablen entdecken, die durch Fehler enthaltende Objekte verursacht werden.

Eine Variable gilt als geändert, wenn sich ihr aktueller Wert entweder von dem Wert unterscheidet, der beim letzten Auslösen des Watchpoints aufgezeichnet wurde, oder wenn er sich vom Anfangswert unterscheidet. Die vergleichende Validierung von Watchpoint-Werten ist auf eine Feldlänge von 253 Byte beschränkt. Bei großen Variablen, die die maximale Länge überschreiten, werden nur die ersten 253 Bytes für den Vergleich herangezogen.

Die Definition eines Watchpoint erfolgt durch Angabe des Namens des Natural-Objekts und des Namens der betreffenden Variablen.

Der eindeutige Bezeichner für einen Watchpoint ist die vom Debugger vergebene Spy-Nummer.

Sobald ein Watchpoint festgelegt wurde, bleibt er für die gesamte Sitzung festgelegt, es sei denn, Sie löschen ihn.

Um die Funktion zur Verwaltung der Watchpoints aufzurufen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `W` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  WM
  ```

  Das Menü **Watchpoint Maintenance** erscheint.

Dieses Kapitel beschreibt die im Menü **Watchpoint Maintenance** zur Verfügung stehenden Funktionen und die in einem Watchpoint-Bildschirm vorhandenen Felder und Spalten.

Test-Modus für Watchpoints ON/OFF setzen

Siehe Abschnitt *Test-Modus ein- und ausschalten*.

Watchpoint aktivieren

Um den aktuellen Zustand angegebener Watchpoints auf aktiv zu setzen:

- Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `A`, einen Objektnamen und/oder einen Variablennamen ein.

  Oder:

  Benutzen Sie das Direktkommando `ACTIVATE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn Sie kein Objekt oder keine Variable angeben (oder den standardmäßig vorhandenen Stern (*) im Feld **Variable** lassen), werden *alle* Watchpoints aktiviert.

Watchpoint deaktivieren

Um den aktuellen Zustand angegebener Watchpoints auf inaktiv zu setzen:

- Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `B`, einen Objektnamen und/oder einen Variablennamen ein.

  Oder:

  Benutzen Sie das Direktkommando `DEACTIVATE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn Sie kein Objekt oder keine Variable angeben (oder den standardmäßig vorhandenen Stern (*) im Feld **Variable** lassen), werden *alle* Watchpoints deaktiviert.

Watchpoint löschen

Um einen Watchpoint zu löschen:

- Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `C`, einen Objektnamen und/oder einen Variablennamen ein.

  Oder:

  Benutzen Sie das Direktkommando `DELETE`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn Sie kein Objekt oder keine Variable angeben (oder den standardmäßig vorhandenen Stern (*) im Feld **Variable** lassen), werden *alle* Watchpoints gelöscht.

Watchpoint anzeigen

Um einen Watchpoint anzuzeigen:

1 Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `D`, einen Objektnamen und/oder einen Variablennamen ein. Wenn Sie keinen Objektnamen eingeben, wird das **Standard-Objekt** (falls angegeben) verwendet.

  Oder:

  Benutzen Sie das Direktkommando `DISPLAY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Wenn für das angegebene Objekt und den Variablennamen ein Watchpoint gesetzt worden ist, erscheint der Bildschirm **Display Watchpoint** mit allen Watchpoint-Definitionen, zum Beispiel:

```
10:25:32              ***** NATURAL TEST UTILITIES *****             2006-02-14
Test Mode ON                  - Display Watchpoint -               Object

Spy number ..............  12
Initial state ........... active          Current state .. active
Watchpoint name ......... WATCHTEST1      DBID/FNR ....... 10/32
Object name ............. WATCHPGM        Library ........ SAG
Variable name ........... WATCHVARIABLE
Skips before execution ..     0           Format/length .. A 10
Max number executions ...     0           Persistent ..... N   Act.level ...  0
Number of activations ...     0
Error in definition ..... - none -

Commands ... BREAK









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

Die Felder im Bildschirm **Display Watchpoint** sind im Abschnitt *Felder und Spalten in Watchpoint-Bildschirmen* beschrieben.

Wird kein eindeutiger Watchpoint gefunden, dann erscheint der Bildschirm **List Watchpoints** (siehe unten).

2 Im Bildschirm **Display Watchpoint** können Sie sich die mit dem Watchpoint-Operator angegebene Bedingung für die Watchpoint-Aktivierung ansehen (siehe auch *Watchpoint-Operatoren*):

Drücken Sie PF10 (Alpha), um den Operator und/oder den Operandenwert in alphanumerischem Format anzuzeigen.

Oder:

Drücken Sie PF11 (Hex), um den Operator und/oder den Operandenwert in hexadezimalem Format anzuzeigen.

Sie können PF22 (Cmds) drücken, um zur Standardansicht des Bildschirms **Display Watchpoint** zurückzukehren. Dort befindet sich das Eingabefeld **Commands**.

Um Watchpoints aufzulisten:

- Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode D, einen Objektnamen und/oder einen Variablennamen ein. Sie können Stern-Notation (*) benutzen, um einen Bereich

von Objektnamen und/oder Variablennamen anzugeben, zum Beispiel: `ABC*`. Wenn Sie nur einen Stern (*) eingeben, werden alle Namen ausgewählt. Wenn Sie keinen Objektnamen angeben, wird das **Standard-Objekt** (falls angegeben) benutzt.

Oder:

Benutzen Sie das Direktkommando `DISPLAY`. Syntax-Beschreibung siehe Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Der Bildschirm **List Watchpoints** erscheint (s. Beispiel). Er enthält eine Liste aller Watchpoints, die für das oder die angegebenen Objekte oder den Variablennamen gesetzt wurden:

```
10:14:05              ***** NATURAL TEST UTILITIES *****             2006-02-14
Test Mode ON                    - List Watchpoints -                Object
                                                                      Top of data
Co No. WP Name      Library  Object    DBID   FNR Stat Skips Execs Count P E
       *___________ *_______ *_______             I  C
      *_____________________________________________________________________
__    1 NAME        SAG      DEBPGM      10    32 A  A     0     0     0 N
        EMPLOYEES-VIEW.NAME
__    5 #MAKE       SAG      DEBPGM      10    32 A  A     0     0     0 N
        #MAKE
__   10 LEAVE-DUE   SAG      DEBPGM      10    32 A  A     0     0     0 N
        EMPLOYEES-VIEW.LEAVE-DUE
__   11 WATCHTEST2  SAG      DEBPGM      10    32 A  A     0     0     0 N
        TESTWP
__   12 WATCHTEST1  SAG      WATCHPGM    10    32 A  A     0     0     0 N
        WATCHVARIABLE
__   13 WATCHTEST3  SAG      DEBPGM      10    32 A  A     0     0     0 N
        WPTEST



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

Die Liste ist in aufsteigender Reihenfolge nach den **Spy-Nummern** in der Spalte **No.** sortiert.

Informationen zu den Spalten im Bildschirm **List Watchpoints** und zu den Zeilenkommandos, die bei einem Listeneintrag ausgeführt werden können, siehe ***Felder und Spalten in Watchpoint-Bildschirmen***.

Watchpoint ändern

Um einen Watchpoint zu ändern:

1 Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `M`, einen Objektnamen und einen Variablennamen ein. Wenn Sie keinen Objektnamen eingeben, wird das **Standard-Objekt** (falls angegeben) verwendet.

Oder:

Benutzen Sie das Direktkommando `MODIFY`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

Wenn ein eindeutiger Watchpoint angegeben wurde, erscheint der Bildschirm **Modify Watchpoint**, in dem Sie die Feldeinträge ändern können. Beschreibung der Felder im Bildschirm **Modify Watchpoint** siehe *Felder und Spalten in Watchpoint-Bildschirmen*.

Wird kein eindeutiger Watchpoint gefunden, dann erscheint der Bildschirm **List Watchpoints** (siehe Abschnitt *Watchpoint anzeigen*).

2 Im Bildschirm **Modify Watchpoint** können Sie die mit dem Watchpoint-Operator angegebene Bedingung für die Aktivierung eines Watchpoint ändern (siehe auch *Watchpoint-Operatoren*):

Drücken Sie PF10 (Alpha), um den Operator und/oder den Operandenwert in alphanumerischem Format zu ändern.

Oder:

Drücken Sie PF11 (Hex), um den Operator und/oder den Operandenwert in hexadezimalem Format zu ändern.

Drücken Sie PF22 (Cmds), um zur Standardansicht des Bildschirms **Modify Watchpoint** zurückzukehren. Dort befindet sich das Eingabefeld **Commands**.

3 Wenn Sie die Bearbeitung der Watchpoint-Definitionen beendet haben, drücken Sie PF3 (Exit) oder PF5 (Save), um die Änderungen zu speichern.

Wenn Sie PF12 (Canc) drücken, wird die Bearbeitung abgebrochen und der Watchpoint bleibt unverändert.

Watchpoint setzen

Um einen Watchpoint für eine Sitzung hinzuzufügen:

- Geben Sie im Menü **Watchpoint Maintenance** den Funktionscode `S`, einen Objektnamen und einen Variablennamen ein.

  Oder:

  Benutzen Sie das Direktkommando `SET`. Syntax-Beschreibung siehe Abschnitt *Debug-Kommandoübersicht und -syntax*.

  Oder:

  Bevor Sie ein Natural-Objekt ausführen:

  - Rufen Sie den Bildschirm **List Object Source** auf (siehe *List Object Source (Objekt-Quellcode anzeigen)*).
  - Positionieren Sie in der Spalte **Source** den Cursor auf einem Variablennamen und drücken Sie PF18 (Se Wp).

  Wenn Sie keinen Objektnamen, aber einen gültigen Variablennamen angeben, wird der Name des **Standard-Objekts** (siehe Abschnitt *Debugger starten*) benutzt.

  Falls kein Standard-Objekt angegeben ist, erscheint ein Auswahlfenster, in dem alle in der aktuellen Library enthaltenen Objekte angezeigt werden.

  Wenn die Namen des Objekts und der Variablen korrekt angegeben sind, wird der Watchpoint sofort gesetzt und eine entsprechende Informationsmeldung wird im Bildschirm angezeigt. Ein für eine dynamische Variable oder ein X-Array gesetzter Watchpoint wird nur während der Programmausführung validiert. Weitere Informationen siehe *Verwaltung und Validierung der Debug-Einträge*.

  Der Watchpoint erhält das Standard-Kommando (`BREAK`), sein Anfangszustand und sein aktueller Zustand werden auf aktiv gesetzt und es werden keine Ausführungseinschränkungen angegeben. Achtung: Wenn Sie beim Setzen eines Watchpoint das Standardkommando `BREAK` löschen und kein Kommando eingeben, das einen Dialog öffnet, hat der Debugger keine Möglichkeit, während der Programmunterbrechung die Steuerung zu übernehmen.

Watchpoint-Operatoren

Sie können für die Aktivierung eines Watchpoint eine Bedingung angeben, indem Sie in einem Watchpoint-Verwaltungsbildschirm einen (Vergleichs-)Operator und (falls relevant) einen dazugehörigen Operanden eingeben.

Um Watchpoint-Operatoren anzugeben:

1 Drücken Sie im Bildschirm **Set Watchpoint** oder **Modify Watchpoint** des gewählten Watchpoint die Funktionstaste PF10 (Alpha), wenn Sie einen Operanden eines Operators in alphanumerischem Format angeben möchten.

Oder:

Drücken Sie im Bildschirm **Watchpoint setzen** oder **Modify Watchpoint** des gewählten Watchpoint die Funktionstaste PF11 (Hex), wenn Sie einen Operanden eines Operators in hexadezimalem Format angeben möchten.

In der unteren Bildschirmhälfte erscheinen zwei Eingabefelder.

2 Im linken Eingabefeld geben Sie einen der Watchpoint-Operatoren ein, die in der folgenden Tabelle aufgelistet sind.

Im rechten Eingabefeld geben Sie den Operandenwert ein, der mit der Variablen verglichen werden soll. Bei Watchpoints mit Operatoren, die für dynamische Variablen (alphanumerisch oder binär) angegeben sind, werden die Operandenwerte von links nach rechts verglichen. Weil die Feldlänge einer dynamischen Variablen variiert, können Sie bis zu 253 Bytes als Vergleichswert eingeben. Ist die aktuelle Länge der dynamischen Variablen kürzer als die maximale Vergleichswertlänge von 253 Bytes, dann erfolgt der Vergleich nur in der aktuellen Länge der dynamischen Variablen.

Operator	Erklärung
MOD	Änderung (*Modification*). Aktiviert den Watchpoint jedes Mal, wenn eine Änderung der Variablen erfolgt. Dies ist die Standardeinstellung.
EQ	Gleich (*Equal to*). Aktiviert den Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen gleich dem angegebenen Operatorwert ist.
NE	Ungleich (*Not equal to*). Aktiviert den Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen nicht gleich dem angegebenen Operatorwert ist.
GT	Größer (*Greater than*). Aktiviert den Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen größer als der angegebene Operatorwert ist.
GE	Größer gleich (*Greater than or equal to*).

Operator	Erklärung
	Aktiviert den Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen größer als oder gleich dem angegebenen Operatorwert ist.
LT	Kleiner (Less than). Aktiviert einen Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen kleiner als der angegebene Operatorwert ist.
LE	Kleiner gleich (*Less than or equal to*). Aktiviert einen Watchpoint, wenn die Variable geändert worden ist und wenn der aktuelle Wert der Variablen kleiner als oder gleich dem angebenen Operatorwert ist.
INV	Ungültiger Inhalt (*Invalid contents*). Aktiviert den Watchpoint jedes Mal, wenn der einer Variablen des Typs N, P, D oder T zugewiesene Wert nicht die folgenden Bedingungen erfüllt: N Numerisch, nicht gepackt (*Numeric unpacked*). P Numerisch, gepackt (*Packed numeric*). D Datumsbereich (*Date range*) von 1582-01-01 bis 2700-12-31. T Zeitbereich (*Time range*) von 1582-01-01 00:00:00.0 bis 2700-12-31 23:59:59.9.

Sie können PF22 (Cmds) drücken, um zur Standardansicht des Bildschirms **Watchpoint setzen** bzw. **Modify Watchpoint** zurückzukehren. Dort befindet sich das Eingabefeld **Commands**.

3 Drücken Sie PF5 (Save), um die Operator-Definitionen zu speichern.

Oder:

Drücken Sie PF12 (Canc), um die Operator-Definitionen unverändert zu lassen und den Bildschirm **Modify Watchpoint** zu verlassen.

Felder und Spalten in Watchpoint-Bildschirmen

Die folgende Tabelle enthält die Beschreibungen der Felder in den Bildschirmen **Display Watchpoint** und **Modify Watchpoint** und der Spalten im Bildschirm **List Watchpoint**:

Feld	Spalte	Erläuterung
Test Mode		Gibt an, ob der Test-Modus auf ON oder OFF gesetzt ist.
Object		Zeigt den Namen des **Standard-Objekts** (siehe *Debugger starten*), falls angegeben.
	Co	Feld zur Eingabe eines der folgenden Zeilenkommandos: **Kommando Aktion** AC Watchpoint aktivieren DA Watchpoint deaktivieren

Feld	Spalte	Erläuterung
		DI Watchpoint anzeigen MO Watchpoint ändern DE Watchpoint löschen ? Gültige Zeilenkommandos auflisten . Watchpoint-Bildschirm verlassen
Spy number	**No.**	Eine eindeutige Nummer, die vom Debugger beim Setzen des Watchpoint zugewiesen wird.
Initial state	**Stat I**	Gibt den Anfangszustand (Initial) und den aktuellen Zustand (Current) des Watchpoint an: Aktiv (A) oder Inaktiv (I) an.
Current state	**Stat C**	
Watchpoint name	**WP Name**	Der Name des Watchpoint. Der Standard-Name für einen Watchpoint ist der Name der betroffenen Variablen. Gültige Werte: 1 bis 12 Zeichen. Namen, deren Länge die Feldgröße übersteigt, werden nach 12 Zeichen abgeschnitten. Im Bildschirm **List Watchpoints** wird der Name des Watchpoint in der ersten Zeile über dem Namen der Variablen aufgelistet.
DBID/FNR	**DBID**	Die Datenbankkennung (DBID) und die Dateinummer (FNR) der Systemdatei, in der das Natural-Objekt gespeichert ist.
	FNR	
Library	**Library**	Der Name der Library, die das Objekt enthält.
Object name	**Object**	Der Name des Objekts, das in der aktuellen Library oder in einer ihrer Steplibs verfügbar ist. Wenn Sie eine Systemvariable als Watchpoint angeben möchten, geben Sie im Feld **Object name** einen Stern (*) ein.
Variable name		Der Name einer Benutzervariablen, einer globalen Variablen oder einer Systemvariablen. Wenn die Variable Teil einer Gruppe ist, kann sie den Gruppennamen als Präfix haben. Wenn Sie eine Systemvariable angeben möchten, geben Sie im Feld **Object name** einen Stern (*) ein. Für ein Array muss eine Indexbeschreibung angegeben werden (Watchpoints können nur für einzelne Elemente definiert werden.) Im Bildschirm **List Watchpoints** wird der Variablenname in der zweiten Zeile, unterhalb dem Watchpoint-Namen aufgelistet. Weitere Informationen siehe *Variable Maintenance*.

Feld	Spalte	Erläuterung
Skips before execution	**Skips**	Legt fest, dass der Watchpoint erst ausgeführt wird, wenn die für den Watchpoint gesetzte Bedingung erfüllt ist (siehe auch **Watchpoint-Operatoren**). Gültige Werte: `0` (Standardwert) bis `32767`.
Max number executions	**Execs**	Ein beliebiger Wert größer als Null (`0`) legt die maximale Anzahl der Watchpoint-Ausführungen fest. Gültige Werte: `0` (Standardwert) bis `32767`.
Number of activations	**Count**	Gibt an, wie viele Male die Watchpoint-Bedingung entsprechend der Angabe beim **Watchpoint-Operator** erfüllt wurde. Der Zähler wird zurückgesetzt, wenn das Programm auf Level 1 gestartet wird.
Format/length		Das Natural-Datenformat und die Datenlänge der Variablen, z.B. A10.
Persistent	**P**	Kennzeichnet einen Watchpoint als persistent. Persistente Watchpoints sind nicht auf das Natural-Objekt beschränkt, für das sie definiert sind, sondern gelten zusätzlich für alle untergeordneten Programmebenen. Persistente Watchpoints sind nur sinnvoll bei Variablen, die `BY REFERENCE` und nicht `BY VALUE RESULT` an ein Subprogramm übergeben werden, siehe relevante Parameterbeschreibung des `CALLNAT`-Statement im Abschnitt *Parameter - operand2*, in der *Statements*-Dokumentation. **Einschränkung:** Persistente Watchpoints sind nicht erlaubt bei Variablen, die in einer Parameter-Kontextklausel definiert sind. Gültiger Wert: `Y` (Yes) oder `N` (No). `N` ist der Standardwert.
Act. level		Bezieht sich auf **Persistent**. Zeigt die Programmebene an, auf der ein persistenter Watchpoint automatisch aktiviert wurde.
Error in definition	**E**	Zeigt eine ungültige Watchpoint-Definition an. Dieser Fehler kann auftreten, wenn das ausführende Programm während des Debugging-Vorgangs neu katalogisiert wird, nachdem die betreffende Variablendefinition geändert wurde. Ein Watchpoint, der für eine dynamische Variable oder ein X-Array (*eXtensible array* = erweiterbares Array) gesetzt ist, wird nur während der Programmausführung auf Gültigkeit geprüft.
Commands		Bis zu sechs Debug-Kommandos. Geben Sie jeweils ein Kommando pro Zeile ein. Eine Zusammenfassung der zur Verfügung stehenden Kommandos finden Sie im Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Feld	Spalte	Erläuterung
		Vorsicht: Wenn Sie das Kommando BREAK löschen und kein Kommando eingeben, das einen Dialog öffnet, hat der Debugger keine Möglichkeit, während der Programmunterbrechung die Kontrolle zu erhalten.

10 Call Statistics Maintenance (Statistiken über gerufene Objekte)

- Test-Modus für Call-Statistik ON/OFF setzen 92
- Call-Statistik ON/OFF setzen 92
- Alle Objekte anzeigen 93
- Aufgerufene Objekte anzeigen 94
- Nicht aufgerufene Objekte anzeigen 94
- Objekte drucken 95

Mit dieser Funktion erhalten Sie Statistikinformationen darüber, welche Natural-Objekte während der Ausführung einer Anwendung gerufen wurden und wie oft ein Objekt gerufen wurde. Die Call-Statistikinformationen werden nach dem Zurücksetzen der Debug-Umgebung gelöscht.

Um die Funktion Call Statistics Maintenance aufzurufen:

- Geben Sie Im Menü **Debug Main Menu** den Funktionscode `C` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  CS
  ```

  Das Menü **Call Statistics Maintenance** wird angezeigt.

Die im Menü **Call Statistics Maintenance** zur Verfügung stehenden Funktionen werden in den folgenden Abschnitten erklärt. Alle zugehörigen Druckfunktionen sind jedoch im Abschnitt *Objekte drucken* beschrieben.

Test-Modus für Call-Statistik ON/OFF setzen

Siehe Abschnitt *Test-Modus ein- und ausschalten*.

Call-Statistik ON/OFF setzen

Wenn während der Ausführung eines Natural-Objekts die Funktion **Call Statistics** auf `ON` gesetzt ist, werden alle Calls, die an ein bestimmtes Objekt gehen, gezählt und die resultierenden Statistikinformationen können anschließend angezeigt oder gedruckt werden.

Um die Funktion Call Statistics auf ON oder OFF zu setzen:

- Geben Sie im Menü **Call Statistics Maintenance** den Funktionscode `C` ein, um alle Call-Statistiken zu aktivieren oder zu deaktivieren.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  SET CALL ON
  ```

  oder

  ```
  SET CALL OFF
  ```

Anmerkung: Ist die Funktion **Call Statistics** ausgeschaltet und wurde keine Call-Statistik erstellt oder wurden alle Call-Statistiken durch Zurücksetzen der Debug-Umgebung gelöscht, dann werden die für Statements-Ausführungsstatistik gespeicherten Informationen (siehe *Statement Execution Statistics Maintenance (Statistiken über ausgeführte Statement-Zeilen)*) für die Anzeige verwendet. Damit können Sie die während der Ausführung einer Natural-Anwendung nicht aufgerufenen Natural-Objekte auffinden.

Alle Objekte anzeigen

Diese Funktion liefert eine Übersicht über die Aufrufhäufigkeit (*Call Frequency*) aller in einer Library enthaltenen Objekte.

Um die Aufrufhäufigkeit (*Call Frequency*) aller in einer Library enthaltenen Objekte anzuzeigen:

- Geben Sie im Menü **Call Statistics Maintenance** den Funktionscode `1` und einen Library-Namen ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY OBJECT library
  ```

  Siehe auch die Syntax des Direktkommandos `DISPLAY` in ***Debug-Kommandoübersicht und -syntax***.

  Wenn Sie keinen Library-Namen angeben, wird standardmäßig die Library angenommen, bei der Sie gerade angemeldet sind.

  Der Bildschirm **Display Called Objects** erscheint. Siehe Beispiel für **Display Called Objects** weiter unten.

  Der Bildschirm **Display Called Objects** enthält eine Auflistung aller Objekte in der angegebenen Library und zeigt in der Spalte **Calls** (ganz rechts) ihre Aufrufhäufigkeit. Für jedes Call-Statement, z.B. `FETCH` oder `CALLNAT` wird ein Eintrag mit dem Namen und einer Zählervariablen in den Debug-Pufferspeicher geschrieben. Der Zähler wird dann bei jedem Aufruf (Call) des entsprechenden Objekts erhöht.

Aufgerufene Objekte anzeigen

Der von dieser Funktion aufgerufene Bildschirm entspricht dem **Display Call Statistics**-Bildschirm, es werden jedoch nur diejenigen Objekte angezeigt, die aufgerufen worden sind.

Um aufgerufene Objekte einer Library anzuzeigen:

- Geben Sie im Menü **Call Statistics Maintenance** den Funktionscode `2` und einen Library-Namen ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY CALL library
  ```

  Siehe auch die Syntax des Direktkommandos `DISPLAY` in ***Debug-Kommandoübersicht und -syntax***.

  Der Bildschirm **Display Called Objects** erscheint (Beispiel):

```
16:06:53              ***** NATURAL TEST UTILITIES *****             2002-02-15
Test mode ON               - Display Called Objects -
                                                                          All
Object   Library  Type         DBID   FNR S/C Ver Cat Date   Time       Calls
*_______ SAG_____
MAINPGM  SAG      Program        10    32 S/C 3.1 2002-02-15 11:51          1
SUBPGM   SAG      Subprogram     10    32 S/C 3.1 2002-02-15 11:50          3
EMP-PGM  SAG      Program        10    32 S/C 3.1 2002-01-22 11:49          2
EMPLIND  SAG      Program        10    32 S/C 3.1 2001-08-13 11:18          1
```

Wenn Sie keinen Library-Namen angeben, wird standardmäßig die Library angenommen, bei der Sie gerade angemeldet sind.

Nicht aufgerufene Objekte anzeigen

Der von dieser Funktion aufgerufene Bildschirm entspricht dem **Display Call Statistics**-Bildschirm, es werden jedoch nur diejenigen Objekte angezeigt, die *nicht* aufgerufen worden sind.

Um nicht aufgerufene Objekte anzuzeigen:

- Geben Sie im Menü **Call Statistics Maintenance** den Funktionscode `3` und einen Library-Namen ein.

Oder:

Geben Sie folgendes Direktkommando ein:

```
DISPLAY NOCALL library
```

Siehe auch die Syntax des Direktkommandos DISPLAY in *Debug-Kommandoübersicht und -syntax*.

Wenn Sie keinen Library-Namen angeben, wird standardmäßig die Library angenommen, bei der Sie gerade angemeldet sind.

Siehe Beispiel-Bildschirm für **Display Called Objects**.

Objekte drucken

Mit den Druckfunktionen können Sie eine generierte Liste der Aufrufstatistiken direkt an einen Drucker weiterleiten oder die Liste auf einen PC herunterladen. Sie geben einen Drucker als Ausgabegerät im Bildschirm **User Profile** (Benutzerprofil) des Debuggers an. Benutzen Sie das Debugger-Kommando PROFILE (siehe Abschnitt *Kommandos zum Navigieren und Anzeigen von Informationen*), um diesen Bildschirm aufzurufen.

Wenn Sie keinen Library-Namen angeben, wird standardmäßig die Library angenommen, bei der Sie gerade angemeldet sind.

Wie unter *Druckoptionen* weiter unten angegeben, können Sie zum Aufrufen einer Druckfunktion entweder einen Funktionscode im Menü **Statement Execution Statistics Maintenance**, ein Zeilenkommando im Bildschirm **Display Statement Lines** oder ein Direktkommando an der Eingabeaufforderung eingeben.

Druckoptionen

Druckfunktion	Funktionscode	Direktkommando
All Objects (alle Objekte)	**4**	`PRINT OBJECT library`
Called Objects (gerufene Objekte)	**5**	`PRINT CALL library`
Non-Called Objects (nicht gerufene Objekte)	**6**	`PRINT NOCALL library`

Siehe auch die Syntax des Direktkommandos PRINT in *Debug-Kommandoübersicht und -syntax*.

Verwandtes Thema

Beispiel für das Erstellen und Drucken von Statistiken im Batch-Modus im Abschnitt *Batch-Verarbeitung*

Beispiel für einen PC Download

Wenn in Ihrer Umgebung die Produkte Entire Connection und Natural Connection installiert sind, können Sie, wie nachfolgend beschrieben, eine Statistikliste auf einen PC herunterladen.

Um eine Liste auf einen PC herunterzuladen:

1 Beim Start der Sitzung: Geben Sie den Profilparameter `PRINT` folgendermaßen ein:

```
PRINT=((1),AM=PC)
```

2 Nach dem Start der Sitzung: Benutzen Sie das folgende Terminalkommando, um die Verbindung zum PC zu aktivieren:

```
%+
```

3 Rufen Sie den Debugger auf und aktivieren Sie ihn.

4 Rufen Sie den Bildschirm **User Profile** auf, indem Sie das Debugger-Kommando `PROFILE` eingeben (siehe *Kommandos zum Navigieren und Anzeigen von Informationen*).

5 Ersetzen Sie im Bildschirm **User Profile** im Feld **Output device** den aktuellen Eintrag durch `PCPRNT01` und drücken Sie PF3 (Exit), um die Einstellungen zu speichern.

6 Aktivieren Sie die Funktion **Call Statistics** und führen Sie die Anwendung aus, für die der Debugger Statistikdaten sammeln soll.

7 Wählen Sie im Statistik-Bildschirm eine Druckfunktion aus.

In dem Entire Connection-Fenster, das dann erscheint, können Sie die Ausgabedatei und das PC-Verzeichnis angeben.

11 Statement Execution Statistics Maintenance (Statistiken über ausgeführte Statement-Zeilen)

- Test-Modus für Statement-Ausführungsstatistiken ON/OFF setzen 98
- Funktion Statement Execution Statistics auf ON/OFF/COUNT setzen 98
- Statement-Ausführungsstatistiken löschen 101
- Statement-Ausführungsstatistiken anzeigen 101
- Statements drucken 105

Mit dieser Funktion erhalten Sie Statistikinformationen darüber, welche Statement-Zeilen aufgerufener Natural-Objekte ausgeführt wurden. Außerdem zeigt die Funktion, wie oft ein Objekt aufgerufen und wie oft eine Statement-Zeile ausgeführt wurde.

Statistikinformationen über die Ausführung von Statements können für folgende Zwecke verwendet werden:

- Aufspüren von „totem“ Programmcode (der nie ausgeführt wird) in einer Anwendung,
- Abschätzen des Abdeckungsgrads eines Anwendungstests (wie viele Statement-Zeilen sind nicht wenigstens einmal beim Testen ausgeführt worden),
- Orten von häufig ausgeführten Programmabschnitten, die Auswirkungen auf die Performance der Anwendung haben könnten.

Um die Funktion Statement Execution Statistics Maintenance aufzurufen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `X` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  XS
  ```

  Das Menü **Statement Execution Statistics Maintenance** wird angezeigt.

Die im Menü **Statement Execution Statistics Maintenance** zur Verfügung stehenden Funktionen werden im folgenden Abschnitt erklärt. Alle zugehörigen Druckfunktionen sind jedoch im Abschnitt *Objekte drucken* beschrieben.

Test-Modus für Statement-Ausführungsstatistiken ON/OFF setzen

Siehe Abschnitt *Test-Modus ein- und ausschalten*.

Funktion Statement Execution Statistics auf ON/OFF/COUNT setzen

Mit dieser Funktion können Sie Statistiken über ausgeführte Statement-Zeilen von Natural-Objekten aktivieren.

Folgende Themen werden behandelt:

- Einstelloptionen

- Statement-Ausführungsstatistiken aktivieren und deaktivieren

Einstelloptionen

Wenn ein Natural-Objekt ausgeführt wird und die Funktion **Statement Execution Statistics** auf `ON` oder `COUNT` gesetzt ist, werden alle in einem bestimmten Objekt ausgeführten Statement-Zeilen in einem Statistikbericht aufgelistet.

- Wenn die Option `ON` gesetzt ist, hält der Debugger nur fest, ob eine spezifische Statement-Zeile ausgeführt wurde oder nicht.
- Wenn die Option `COUNT` gesetzt ist, zählt der Debugger, wie oft eine Statement-Zeile ausgeführt wurde

Sie können eine Library und einen Objektnamen angeben, um die Statement-Ausführungsstatistik auf die gewünschten Natural-Objekte einzuschränken. Standardmäßig werden Statistikdaten für alle Objekte in der aktuellen Library gesammelt. Sie können Stern-Notation (*) benutzen, um einen Bereich von Namen anzugeben.

Wenn Sie die Statement-Ausführungsstatistik von `ON` nach `COUNT` oder umgekehrt schalten, sind bereits vorhandene Statistiken davon nicht betroffen, d.h., ihr Status bleibt auf `ON` oder `COUNT`.

Die gesammelten Statistikdaten werden im Debug-Pufferspeicher gespeichert. Die Menge an Speicherplatz, die benötigt wird, um statistische Informationen für ein Objekt zu speichern, ist ungefähr:

(Anzahl der Quellcode-Zeilen) / 8 + 100 Bytes, wenn **Statement Execution Statistics** auf `ON` gesetzt ist,
und
(Anzahl der Quellcode-Zeilen) * 4 + 100 Bytes, wenn **Statement Execution Statistics** auf `COUNT` gesetzt ist.

Wenn Sie ein Natural-Objekt ändern, indem Sie Zeilen einfügen oder löschen und die Zeilen nicht neu nummerieren, bevor Sie es mit dem Kommando `STOW` speichern und katalogisieren, kann sich die für die Statistikdaten des Objekts benötigte Speichermenge erhöhen. Um dies zu vermeiden, können Sie in Ihrem Editor-Profil die Option **Auto Renumber** auf `Y` (Yes) setzen (siehe *Editor-Profil* in der *Editoren*-Dokumentation) oder das Systemkommando `CATALL` (siehe *Systemkommandos*-Dokumentation) benutzen, und zwar bei (standardmäßig) eingeschalteter Option **Renumber source-codes lines** (Quellcode-Zeilen neu nummerieren).

Sie können das Debugger-Kommando `PROFILE` benutzen (siehe *Kommandos zum Navigieren und Anzeigen von Informationen*), um die Größe des Debug-Pufferspeichers einzuschränken. Wenn die Option **Statement Execution Statistics** auf `COUNT` gesetzt ist, werden bei Objekten mit mehr als 8000 Statement-Zeilen keine Statement-Ausführungsstatistiken erfasst.

Statement-Ausführungsstatistiken unterliegen als Teil der Debug-Umgebung den Auswirkungen der Direktkommandos `SAVE ENVIRONMENT` und `LOAD ENVIRONMENT` (siehe auch Abschnitt *Debug Environment Maintenance - Debug-Umgebung verwalten*).

Statement-Ausführungsstatistiken aktivieren und deaktivieren

Dieser Abschnitt beschreibt, wie Sie Statement-Ausführungsstatistiken aktivieren und deaktivieren.

Sie können eine Library und/oder einen Objektnamen angeben, um die Statement-Ausführungsstatistik auf die gewünschten Natural-Objekte einzuschränken. Standardmäßig werden Statistikdaten für alle Objekte in der aktuellen Library gesammelt. Sie können Stern-Notation (*) benutzen, um einen Bereich von Namen anzugeben.

Um Statement-Ausführungsstatistiken zu aktivieren:

- Geben Sie im Menü **Statement Execution Statistics Maintenance** den Funktionscode `S`, den Namen einer Library und/oder den Namen eines Objekts an. Ändern Sie den Wert im Feld **State** auf `ON`.

  Oder:

  Geben Sie eines der folgenden Direktkommandos ein:

  ```
  SET XSTATISTICS ON library (object)
  ```

  oder

  ```
  SET XSTATISTICS COUNT library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `SET` in *Debug-Kommandoübersicht und -syntax*.

Wenn Sie keine Library und/oder ein Objekt angeben, werden die Statistikdaten zu allen Objekten in Ihrer aktuellen Library aktiviert.

Um Statement-Ausführungsstatistiken zu deaktivieren:

- Geben Sie im Menü **Statement Execution Statistics Maintenance** den Funktionscode `S`, den Namen einer Library und/oder den Namen eines Objekts an. Ändern Sie den Wert im Feld **State** auf `OFF`.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  SET XSTATISTICS OFF library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `SET` in *Debug-Kommandoübersicht und -syntax*.

Wenn Sie keine Library und/oder ein Objekt angeben, werden die Statistikdaten zu allen Objekten in Ihrer aktuellen Library deaktiviert.

Statement-Ausführungsstatistiken löschen

Um Statement-Ausführungsstatistiken zu löschen:

- Geben Sie im Menü **Statement Execution Statistics Maintenance** den Funktionscode `C`, den Namen einer Library und/oder den Namen eines Objekts an.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DELETE XSTATISTICS library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `DELETE` in ***Debug-Kommandoübersicht und -syntax***.

Wenn Sie keine Library und/oder ein Objekt angeben, werden die Statistikdaten zu allen Objekten in Ihrer aktuellen Library gelöscht.

Statement-Ausführungsstatistiken anzeigen

Mit dieser Funktion können Sie einen Bildschirm aufrufen, der eine Liste von Statement-Ausführungsstatistiken enthält.

Um Statement-Ausführungsstatistiken anzuzeigen:

1. Geben Sie im Menü **Statement Execution Statistics Maintenance** den Funktionscode `D`, den Namen einer Library und/oder den Namen eines Objekts an.

   Oder:

   Geben Sie folgendes Direktkommando ein:

   ```
   DISPLAY XSTATISTICS
   ```

   Der Bildschirm **List Statement Execution Statistics** wird angezeigt:

```
16:02:01              ***** NATURAL TEST UTILITIES *****             2002-02-15
Test Mode ON        - List Statement Execution Statistics -     Object
                                                                          All
Co Object   Library  Type        DBID   FNR Obj.Called Exec Exec   %  Total No.
   *________ *_______                          n Times able uted     Executions
__ TEST     SAG      Program       10    32          4   20   17  85          95
__ MAP01    SAG      Map           10    32          6    2    2 100          12
__ SPGM02   SAG      Subprogram    10    32          2    6    2  33           4
__ SAGTEST1 SAG      Program       10    32          2   20   10  50          17
__ DEBPGM   SAG      Program       10    32          1    6    6 100          34
```

Zu jedem Objekt werden folgende Informationen angezeigt:

- Aufrufhäufigkeit (*Called n Times*),
- Anzahl der ausführbaren Statements (*Exec able*),
- Anzahl der ausgeführten Statements (*Executed*),
- Prozentsatz (%) der ausgeführten Statements bezogen auf die Gesamtzahl der ausführbaren Statements,
- Gesamtzahl der ausgeführten Statements (*Total No. Executions*).

Falls Daten fehlen oder möglicherweise inkonsistent sind, wird der betroffene Listeneintrag hervorgehoben dargestellt.

2 Sie können in der Liste einen Eintrag mit einem Zeilenkommando zur Weiterverarbeitung markieren:

Zeilenkommando	Erklärung
DE	Statement-Ausführungsstatistik löschen, s. **oben**.
DS	**Alle Statement-Zeilen anzeigen**.
DX	**Nur die ausgeführten Statement-Zeilen anzeigen**.
DN	**Nur die nicht ausgeführten Statement-Zeilen anzeigen**.
I	Informationen zum katalogisierten Objekt und zu Fehlern anzeigen.
PS	Alle Statement-Zeilen drucken.
PX	Nur die ausgeführten Statement-Zeilen drucken.
PN	Nur die nicht ausgeführten Statement-Zeilen drucken.

Weitere Informationen zu Druckfunktionen siehe ***Statements drucken***.

Im folgenden Abschnitt werden die Bildschirme beschrieben, die mit den Anzeigekommandos aufgerufen werden können:

- Alle Statement-Zeilen anzeigen
- Nur die ausgeführten Statement-Zeilen anzeigen

- Nur die nicht ausgeführten Statement-Zeilen anzeigen

Alle Statement-Zeilen anzeigen

Der Bildschirm **Display Statement Lines** zeigt den Objekt-Quellcode und gibt an, ob eine Statement-Zeile ausgeführt worden ist oder nicht.

Um den Bildschirm Display Statement Lines anzuzeigen:

- Markieren Sie im Bildschirm **List Statement Execution Statistics** den gewünschten Eintrag mit dem Zeilenkommando `DS`.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY STATEMENT library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `DISPLAY` in ***Debug-Kommandoübersicht und -syntax***.

  Der Bildschirm **Display Statement Lines** erscheint. Wenn die Funktion **Statement Execution Statistics** auf `COUNT` gesetzt worden ist, wird die Ausführungshäufigkeit der Statements-Zeile angezeigt. Beispiel:

```
16:04:01              ***** NATURAL TEST UTILITIES *****              2002-02-15
Test Mode ON              - Display Statement Lines -             Object SAGTEST

Line Source                                                                Count
0200   RD1. READ EMPLOYEES-VIEW BY NAME                                        2
0210       STARTING FROM #NAME-START THRU #NAME-END
0220 *
0230     IF LEAVE-DUE >= 20                                                    1
0240       PERFORM MARK-SPECIAL-EMPLOYEES                          not executed
0250     ELSE                                                      not executed
0260       RESET #MARK                                                       1
0270     END-IF
0280 *
0290     RESET #MAKE #MODEL                                                    1
0300     CALLNAT 'SPGM02' PERSONNEL-ID #MAKE #MODEL                            1
0310 *
0320     WRITE TITLE / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DU              1
0330       / '***    ARE MARKED WITH AN ASTERISK        ***' //
0340     DISPLAY   '//N A M E' NAME                                            2
```

Falls kein eindeutiges Objekt angegeben wurde, erscheint der Bildschirm **List Statement Execution Statistics**.

Nur die ausgeführten Statement-Zeilen anzeigen

Der Bildschirm **Display Executed Statement Lines** entspricht dem Bildschirm **Display Statement Lines**, aber es werden nur die Statement-Zeilen angezeigt, die ausgeführt worden sind.

Um den Bildschirm Display Executed Statement Lines aufzurufen:

- Markieren Sie im Bildschirm **List Statement Execution Statistics** den gewünschten Eintrag mit dem Zeilenkommando `DX`.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY EXEC library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `DISPLAY` in *Debug-Kommandoübersicht und -syntax*.

  Falls kein eindeutiges Objekt angegeben wurde, erscheint der Bildschirm **List Statement Execution Statistics**.

Nur die nicht ausgeführten Statement-Zeilen anzeigen

Der Bildschirm **Non-Executed Statement Lines** entspricht dem Bildschirm **Display Statement Lines**, aber es werden nur die Statement-Zeilen angezeigt, die nicht ausgeführt worden sind.

Um den Bildschirm Display Non-Executed Statement Lines aufzurufen:

- Markieren Sie im Bildschirm **List Statement Execution Statistics** den gewünschten Eintrag mit dem Zeilenkommando `DN`.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY NOEXEC library (object)
  ```

  Siehe auch die Syntax des Direktkommandos `DISPLAY` in *Debug-Kommandoübersicht und -syntax*.

  Falls kein eindeutiges Objekt angegeben wurde, erscheint der Bildschirm **List Statement Execution Statistics**.

Statements drucken

Mit den Druckfunktionen können Sie eine generierte Liste der Statement-Ausführungsstatistiken direkt an einen Drucker weiterleiten oder die Liste auf einen PC herunterladen. Sie geben einen Drucker als Ausgabegerät im Bildschirm **User Profile** (Benutzerprofil) des Debuggers an. Benutzen Sie das Debugger-Kommando `PROFILE` (siehe Abschnitt *Kommandos zum Navigieren und Anzeigen von Informationen*), um diesen Bildschirm aufzurufen.

Wenn Sie keinen Library-Namen angeben, wird standardmäßig die Library angenommen, bei der Sie gerade angemeldet sind.

Wie unter *Druckoptionen* weiter unten angegeben, können Sie zum Aufrufen einer der Druckfunktionen entweder einen Funktionscode im Menü **Statement Execution Statistics Maintenance**, ein Zeilenkommando im Bildschirm **Display Statement Lines** oder ein Direktkommando eingeben.

Druckoptionen

Druckfunktion	Funktionscode	Zeilenkommando	Direktkommando
Statement-Ausführungsstatistik drucken	**1**		`PRINT XSTATISTICS library (object)`
Alle Statements drucken	**2**	`PS`	`PRINT STATEMENT library (object)`
Ausgeführte Statements drucken	**3**	`PX`	`PRINT EXEC library (object)`
Nicht ausgeführte Statements drucken	**4**	`PN`	`PRINT NOEXEC library (object)`

Siehe auch die Syntax des Direktkommandos `PRINT` im Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Verwandte Themen:

- ***Beispiel für einen PC Download*** in *Objekte drucken* im Abschnitt *Call Statistics Maintenance (Statistiken über gerufene Objekte)*
- ***Beispiel für das Erstellen und Drucken von Statistiken im Batch-Modus*** im Abschnitt *Batch-Verarbeitung*

12 Variable Maintenance (Anzeigen und Ändern von Variablen)

- Benutzervariablen, globale Variablen und datenbankbezogene Systemvariablen anzeigen 108
- Systemvariablen anzeigen 111
- Variablen ändern 112

Mit dieser Funktion können Sie innerhalb des Debuggers Variablen anzeigen und ändern, wenn ein Natural-Objekt unterbrochen wurde.

Die Funktion **Variable maintenance** zeigt für das unterbrochene Natural-Objekt Benutzervariablen, globale Variablen und die datenbankbezogenen Systemvariablen `*COUNTER`, `*ISN` und `*NUMBER` mit Natural-Datenformaten, Längen und Inhalten an.

Benutzervariablen, globale Variablen und datenbankbezogene Systemvariablen anzeigen

Dieser Abschnitt beschreibt, wie Sie entweder den Bildschirm **Display Variables** (Übersicht) mit einer Liste aller Variablen oder den Bildschirm **Display Variable** (einzeln) mit allen Einzelheiten zu einer bestimmten Variablen aufrufen.

- Variablen anzeigen - Übersicht
- Variablen anzeigen - einzeln

Variablen anzeigen - Übersicht

Um eine Übersicht über Benutzervariablen, globale Variablen und datenbankbezogene Systemvariablen anzuzeigen:

- Geben Sie im Hauptmenü **Debug Main Menu** oder im Fenster **Debug Break** den Funktionscode `V` ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY VARIABLE variable,variable,...
  ```

Der Bildschirm **Display Variables** (Übersicht) erscheint mit einer Liste der Variablen, die für das unterbrochene Natural-Objekt angegeben sind. Lange Werte können im Bildschirm abgeschnitten angezeigt werden. Bei Arrays wird nur der Inhalt der ersten Ausprägung angezeigt.

Sie können mit den Funktionstasten PF10 (Alpha) und PF11 (Hex) zwischen der alphanumerischen und der hexadezimalen Darstellung des Variableninhalts wechseln.

Mit PF5 (Zoom) können Sie zwischen der verkürzten Anzeige einer Variablen und der vollständigen Namensanzeige mit Gruppenname, Variablenname und Indizes (falls relevant) umschalten.

Für *variable* können Sie auch eine Systemvariable angeben. Weitere Informationen siehe ***Systemvariablen anzeigen***.

Variablen anzeigen - einzeln

Um eine einzelne Variable vollständig anzuzeigen:

- Wählen Sie im Bildschirm **Display Variables** (Übersicht) eine Variable aus und markieren Sie sie mit dem Zeilenkommando `DI`.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  DISPLAY VARIABLE variable
  ```

  Oder:

  Positionieren Sie im Bildschirm **List Object Source** in der Spalte **Source** den Cursor auf einem Variablennamen und drücken Sie PF18 (Di Va).

  Bei der Benutzung von PF18 (Di Va) gelten die folgenden Einschränkungen:

  Wenn sich ein Variablenname (einschließlich der Ausprägungen eines Arrays) über mehr als eine Zeile erstreckt, wird nur der Inhalt der ersten Zeile ausgewertet.
  Wenn auf einen Array-Namen kein Index folgt, wird das gesamte Array angezeigt.
  Wenn der Inhalt eines Arrays konstant ist, z.B. Array (3,2,6), wird nur diese Ausprägung angezeigt.
  Wenn der Inhalt eines Arrays variabel ist, z.B. Array (i,j) oder Array (3:i), werden die Variablen ausgewertet, bevor die jeweiligen Ausprägungen des Arrays angezeigt werden.

  Oder:

  Positionieren Sie im Bildschirm **List Object Source** in der **Source** den Cursor auf einem Variablennamen und drücken Sie ENTER.

  Bei der Benutzung von ENTER gelten die gleichen Einschränkungen wie bei PF18, siehe oben. Allerdings wird die Variable oder die Variablen-Ausprägung in einem Fenster **Display Variable** statt über den Bildschirm **Display Variable** (einzeln) angezeigt, wenn der Index für ein Array nicht mehr als eine Ausprägung angibt. Wenn der Index für ein Array mehr als eine Ausprägung angibt, werden die Daten über den Bildschirm **Display Variable** (einzeln) angezeigt.

  Oder:

  Anstatt den Cursor manuell zu positionieren und ENTER zu drücken, können Sie zur einfacheren Bedienung auch Entire Connection benutzen. Hier positioniert ein Doppelklick mit der linken Maustaste den Cursor und simuliert die ENTER-Taste.

Es erscheint der Bildschirm **Display Variable** (einzeln) bzw. ein Fenster mit allen relevanten Angaben zur jeweiligen Variablen.

Wenn Daten über ein Fenster angezeigt werden und die Länge des Variableninhalts 256 Bytes überschreitet, werden nur die ersten 256 Bytes angezeigt. Beim Bildschirm **Display Variable** (einzeln) gibt es keine solchen Einschränkungen und Sie können, wie nachfolgend beschrieben, durch den gesamten Inhalt der Variablen navigieren.

Um den gesamten Inhalt der Variablen anzuzeigen oder im Inhalt zu navigieren:

- Mit PF22 können Sie vorwärts, mit PF23 rückwärts navigieren.

  Oder:

  Geben Sie im Feld **Position** einen numerischen Wert ein, damit die Anzeige an einer bestimmten Stelle beginnt.

Mit PF10 (Alpha) und PF11 (Hex) können Sie zwischen alphanumerischer und hexadezimaler Darstellung des Variableninhalts wechseln.

Um alle Ausprägungen eines Arrays mittels Bildschirmfunktionen anzuzeigen:

- Wählen Sie im Bildschirm **Display Variables** eine Variable aus und markieren Sie sie mit dem Zeilenkommando `DI`.

  Oder:

  Mit PF7 (-) und PF8 (+) können Sie zwischen den einzelnen Ausprägungen hin und her wechseln.

Um eine oder mehrere Ausprägungen eines Arrays mittels Direktkommando anzuzeigen:

- Benutzen Sie folgendes Direktkommando:

  ```
  DISPLAY VARIABLE variable-name(index-specification)
  ```

  Dabei ist *variable-name* der Name der Variablen und *index-specification* eines der Folgenden: eine Index-Notation, ein Index-Bereich oder ein Stern (*) für alle Ausprägungen einer Dimension. Variablen, die Teil einer Indexangabe (*index-specification*) sind, werden ausgewertet, bevor die jeweiligen Ausprägungen angezeigt werden.

  Beispiele:

`DISPLAY VARIABLE ARRAY1(*)`	Eindimensionales Array: Zeigt alle Ausprägungen des eindimensionalen Arrays `ARRAY1` an.
`DISPLAY VARIABLE ARRAY1(1)` or	Eindimensionales Array: Zeigt die erste Ausprägung des eindimensionalen Arrays `ARRAY1` an.

`DISPLAY VARIABLE ARRAY1`	
`DISPLAY VARIABLE ARRAY2(2,3:4)`	Zweidimensionales Array: Zeigt die zweite Ausprägung der ersten Dimension und die Index-Notation der zweiten Dimension des zweidimensionalen Arrays `ARRAY2` an.
`DISPLAY VARIABLE ARRAY3(1,3:4,*)`	Dreidimensionales Array: Zeigt die erste Ausprägung der ersten Dimension, die Index-Notation der zweiten Dimension und alle Ausprägungen der dritten Dimension des dreidimensionalen Arrays `ARRAY3` an.
`DISPLAY VARIABLE ARRAY4(I,J + 1)`	Zweidimensionales Array: Zeigt die Ausprägung des zweidimensionalen Arrays `ARRAY4` an, angegeben durch den Wert des Ausdrucks `J + 1`.

Systemvariablen anzeigen

Um Systemvariablen anzuzeigen (außer datenbankbezogenen Systemvariablen):

- Geben Sie folgendes Direktkommando ein:

  ```
  SYSVARS
  ```

  Der Bildschirm **System Variables** erscheint mit einem eingeschränkten Satz Systemvariablen.

Um eine einzelne Systemvariable anzuzeigen:

- Benutzen Sie folgendes Direktkommando:

  ```
  DISPLAY VARIABLE system-variable-name
  ```

  Dabei ist *system-variable-name* der Name der Systemvariablen, der auch mit dem Direktkommando `SYSVARS` angezeigt werden kann.

Bei Variablen des Typs Handle wird der Name der Class der Instanz, auf die sich das Handle bezieht, in alphanumerischer Darstellung angezeigt. Wenn der Name der Class nicht verfügbar ist, wird stattdessen der Globally Unique Identifier (GUID) angezeigt. Wenn die Class innerhalb von Natural definiert wurde, wird der Name der Class oder die GUID mit dem Suffix `(NAT)` versehen.

Der Inhalt von Properties (Eigenschaften) einer Instanz kann im Debugger nicht angezeigt werden.

Variablen ändern

Diese Funktion gilt nicht bei Systemvariablen.

Mit dieser Funktion können Sie den Wert von Benutzervariablen, globalen Variablen und datenbankbezogenen Variablen ändern.

Um den Inhalt einer Variablen vom Bildschirm Modify Variable aus zu ändern:

1 Rufen Sie den Bildschirm **Modify Variable** auf, indem Sie die betreffende Variable mit dem Zeilenkommando `MO` markieren.

Oder:

Drücken Sie im Bildschirm **Display Variable** die Taste PF5 (Mod).

2 Ändern Sie im Bildschirm **Modify Variable** im Feld **Contents** den Wert der Variablen.

Der neue Inhalt muss in Bezug auf das Natural-Datenformat der geänderten Variablen gültig sein, weil innerhalb des Debuggers keine Formatänderung bei einer Variablen vorgenommen werden kann.

Sie können im Bildschirm **Modify Variable** mit PF10 (Alpha) und PF11 (Hex) zwischen alphanumerischer und hexadezimaler Darstellung des Variablenwerts umschalten.

Um den Inhalt einer Variablen mittels Direktkommando zu ändern:

- Geben Sie folgendes Direktkommando ein:

```
MODIFY VARIABLE variable = new value
```

Eine Meldung erscheint, die die Änderung des Variablenwerts bestätigt.

Anmerkung: Die Benutzung der Funktion **Modify Variables** oder des Kommandos `MODIFY VARIABLE` kann durch Natural Security unterbunden werden. Siehe Abschnitt *Components of an Environment Profile* in der *Natural Security*-Dokumentation.

13 List Object Source (Objekt-Quellcode anzeigen)

- Breakpoints verwalten - Maintain Breakpoints 115

Mit der Funktion **List object source** können Sie den Quellcode eines Objekts anzeigen und Breakpoints verwalten. Dazu muss sich die entsprechende Source in Ihrer aktuellen Library oder in einer ihrer Steplibs befinden.

Um den Objekt-Quellcode eines Objekts anzuzeigen:

- Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `L` und einen Objektnamen ein.

  Oder:

  Geben Sie folgendes Direktkommando ein:

  ```
  LIST object
  ```

  Siehe auch die Syntax des Direktkommandos `LIST` in *Debug-Kommandoübersicht und -syntax*.

  Der Bildschirm **List Object Source** erscheint und der Quellcode des Objekts wird angezeigt, wobei rechts im Bildschirm in der Spalte **Message** alle Breakpoints aufgelistet werden.

  Mit PF7 (-) oder PF8 (+) können Sie seitenweise nach oben oder unten blättern.

Wenn Sie ein Natural-Objekt ausführen, unterbricht der Debugger die Ausführung bei jedem von Ihnen gesetzten Breakpoint oder Watchpoint und das Fenster **Debug Break** erscheint (siehe *Debug Break-Fenster* in *Debugger-Konzept*).

Um den Quellcode eines unterbrochenen Natural-Objekts aufzulisten:

- Wählen Sie im Fenster **Debug Break** den Funktionscode `L` für **List break**.

  Oder:

  Falls relevant: Drücken Sie auf einem Debugger-Bildschirm PF9 (Li Br) oder geben Sie folgendes Direktkommando ein:

  ```
  LIST BREAK
  ```

  Der Bildschirm **List Object Source** erscheint. Darin wird der Quellcode an der Position angezeigt, an der eine Unterbrechung (Breakpoint oder Watchpoint) auftrat. Rechts im Bildschirm in der Spalte **Message** wird der Name des Breakpoint oder Watchpoint angezeigt. Der entsprechende Quellcode wird hervorgehoben dargestellt.

Breakpoints verwalten - Maintain Breakpoints

Sie können die Funktion **List object source** benutzen, um innerhalb des Objekt-Quellcode Funktionen zum Verwalten oder direkten Ausführen von Breakpoint-Verwaltungsfunktionen aufzurufen oder direkt auszuführen. Eine Anleitung zum Setzen von Breakpoints und allgemeine Informationen zu Breakpoints finden Sie im Abschnitt ***Systemvariablen anzeigen*** in *Breakpoint Maintenance (Verwaltung der Breakpoints)*.

Um eine Breakpoint-Verwaltungsfunktion aus einer Objekt-Source heraus aufzurufen:

1 Geben Sie im Hauptmenü **Debug Main Menu** den Funktionscode `L` und einen Objektnamen ein.

Oder:

Geben Sie folgendes Direktkommando ein:

```
LIST object
```

Siehe auch die Syntax des Direktkommandos `LIST` in ***Debug-Kommandoübersicht und -syntax***.

Der Quellcode des angegebenen Objekts wird angezeigt.

Rechts im Bildschirm in der Spalte **Message** werden Namen von bereits gesetzten Breakpoints angezeigt.

Zum Navigieren in der Source-Liste können Sie eines der folgenden Kommandos in der Kommandozeile eingeben:

- `+` (Pluszeichen) oder `-` (Minuszeichen), um seitenweise nach unten bzw. oben zu blättern.
- `TOP`, um an den Anfang zu blättern.
- `BOTTOM`, um an das Ende zu blättern.
- `LEFT`, um nach Links zu blättern.
- `RIGHT`, um nach Rechts zu blättern.

2 Markieren Sie in der Objekt-Source eine oder mehrere Zeilen mit einem der folgenden Kommandos:

Zeilenkommando	Erklärung
AC	Aktiviert Breakpoints.
DA	Deaktiviert Breakpoints.
DE	Löscht Breakpoints.
DI	Zeigt Breakpoints an.
MO	Verzweigt in den Verwaltungsbildschirm **Modify Breakpoint**.
SE	Setzt Breakpoints.
SM	Verzweigt in den Verwaltungsbildschirm **Set Breakpoint**.

Nach erfolgreicher Kommandoausführung wird in der Spalte **Message** rechts im Bildschirm eine entsprechende Meldung angezeigt.

14 Fehlerbehandlung

- Fehler während der Anwendungsausführung 118
- Fehler während der Debugger-Ausführung 119

Dieser Abschnitt enthält Informationen zur Behandlung von Fehlern, die bei der Benutzung des Debuggers auftreten können.

Fehler während der Anwendungsausführung

Sie können den Debugger benutzen, um einen Natural-Systemfehler zu analysieren, der die Programmausführung unterbricht. Bei Test-Modus auf `ON` (siehe ***Test-Modus ein- und ausschalten***) oder Natural-Profilparameter `DBGERR` auf `ON` (siehe *DBGERR - Automatischer Debugger-Start bei Laufzeitfehler* in der *Parameter-Referenz*-Dokumentation) übernimmt der Debugger die Steuerung, wenn ein Fehler auftritt. In diesem Fall erscheint ein **Debug Break**-Fenster, zum Beispiel:

```
+------------------- Debug Break -------------------+
! Break by NATURAL error 1316                       !
! at line   60 in program SAGTEST (level 1)         !
!                                                   !
!         G   Go                                    !
!         L   List break                            !
!         M   Debug Main Menu                       !
!         N   Next break command                    !
!         R   Run (set test mode OFF)               !
!         S   Step mode                             !
!         V   Variable maintenance                  !
!                                                   !
! Code .. G                                         !
!                                                   !
! Index not within array structure.                 !
! PF2=Step,PF13=Next,PF14=Go,PF15=Menu,PF17=SkipS   !
+---------------------------------------------------+
```

Sie können die Funktion **List break** benutzen, um den Quellcode des Programms an der Stelle anzuzeigen, an der das letzte Statement ausgeführt wurde. Die Natural-Fehlernummer wird rechts im Bildschirm in der Spalte **Message** angezeigt und die entsprechende Quellcode-Zeile ist hervorgehoben.

Sie können dann beispielsweise den Inhalt der Variablen in dem Programm untersuchen, um den Grund für den Fehler zu bestimmen.

Fehler während der Debugger-Ausführung

Falls während der Debugger-Ausführung in einer Anwendung (nachfolgend: „Debugging“) ein Fehler festgestellt wird, beendet der Debugger seine Arbeit und ruft ein Fenster mit einer Fehlermeldung ähnlich der im folgenden Beispiel gezeigten auf:

```
+------------------ NATURAL Debug Error ---------------------------+
! NATURAL error 3009 has occurred in the NATURAL Debugger.         !
! Last transaction backed out of database 10. Subcode 3            !
!                                                                  !
! Error occurred on level 5 in line 4150 in                       !
! subprogram DBGTEST in library TEST.                              !
! DBGTEST has been loaded from FNAT=(10,932).                      !
! DBGTEST has been cataloged on 2005-04-12 14:43:07.               !
!                                                                  !
! Debugging terminates.                                            !
! Pass this error to application for error processing ? (Y/N): N   !
+------------------------------------------------------------------+
```

Wenn Sie diese Fehlermeldung durch Eingabe von `N` (No - dies ist die Standardeinstellung) bestätigen, geschieht Folgendes:

- Der Debugger beendet das Debugging und schaltet den Test-Modus aus (`OFF`).
- Das Natural-Laufzeitsystem ignoriert den Fehler und setzt die Ausführung der Anwendung fort.

Wenn Sie die Fehlermeldung durch Eingabe von `Y` (Yes) bestätigen, geschieht Folgendes:

- Der Debugger beendet das Debugging und schaltet den Test-Modus aus (`OFF`).
- Das Natural-Laufzeitsystem reagiert auf den Fehler und übergibt ihn an die Anwendung:

  Wenn ein `ON ERROR`-Statement (siehe *Statements*-Dokumentation) verwendet wird, bestimmt die Anwendung das weitere Vorgehen, nachdem ein Fehler zur Ausführungszeit aufgetreten ist. Zum Beispiel, im Fall eines Fehlers NAT3009, bei dem eine Transaktion aus der Datenbank zurückgesichert wird (*Back out)*, kann die Anwendung entsprechende Maßnahmen ergreifen.

  Falls kein `ON ERROR`-Statement verwendet wird, beendet das Natural-Laufzeitsystem die Ausführung der Anwendung und kehrt zur Anzeige einer Natural-Eingabeaufforderung zurück.

15 Kommandos zur Ausführungssteuerung

- ESCAPE BOTTOM 122
- ESCAPE ROUTINE 122
- EXIT 122
- GO 123
- NEXT 123
- RUN 123
- STEP 124
- STEP SKIPSUBLEVEL 124
- STEP SKIPSUBLEVEL n 124
- STOP 124

Dieser Abschnitt beschreibt die Direktkommandos, die im Debugger für die Steuerung des Programmablaufs während einer Debug-Sitzung zur Verfügung stehen. Eine Zusammenfassung aller im Debugger verfügbaren Kommandos finden Sie im Abschnitt ***Debug-Kommandoübersicht und -syntax***.

Die im Folgenden aufgeführten Kommandos gelten nur, wenn der Debugger die Programmausführung unterbricht.

ESCAPE BOTTOM

Dieses Kommando kann nur dann benutzt werden, wenn ein Natural-Objekt innerhalb einer Verarbeitungsschleife unterbrochen worden ist.

Wenn Sie dieses Kommando eingeben, wird das unterbrochene Natural-Objekt mit dem ersten Statement, das auf die Verarbeitungsschleife folgt, fortgesetzt.

Anmerkung: Natural Security kann die Ausführung dieses Kommandos unterbinden, siehe *Components of an Environment Profile* in der *Natural Security*-Dokumentation.

ESCAPE ROUTINE

Wenn Sie dieses Kommando eingeben, wird die Verarbeitung des unterbrochenen Natural-Objekts gestoppt und die Verarbeitung wird ab dem Objekt fortgesetzt, von dem aus das unterbrochene Natural-Objekt aufgerufen wurde. Sie wird mit dem Statement fortgesetzt, das auf das entsprechende `CALLNAT`-, `PERFORM`- oder `FETCH RETURN`-Statement folgt.

Wenn Sie das Kommando `ESCAPE ROUTINE` bei einem Hauptprogramm anwenden, beendet Natural das Programm und kehrt zum Kommandoeingabemodus zurück.

Anmerkung: Natural Security kann die Ausführung dieses Kommandos unterbinden, siehe *Components of an Environment Profile* in der *Natural Security*-Dokumentation.

EXIT

Wenn Sie aus dem Hauptmenü **Debug Main Menu** heraus die Exit-Funktion aufrufen möchten, können Sie PF3 (Exit) drücken oder das Ausführungssteuerungskommando `EXIT` eingeben. Der Debugger kehrt dann an eine der folgenden Stellen zurück:

- entweder zum aufrufenden Programm (d.h. zum unterbrochenen Natural-Objekt, das dann fortgesetzt wird),

- oder zu einer Kommandoeingabeaufforderung, wenn der Debugger mit dem Direktkommando TEST aufgerufen worden ist,
- oder zu dem entsprechenden Eingabefeld, wenn der Debugger mit dem Terminalkommando %<TEST aufgerufen worden ist.

Wenn aber zurzeit ein Breakpoint oder Watchpoint aktiv ist, wird das nächste Kommando dieses Breakpoint oder Watchpoint ausgeführt.

Wenn Sie sich jedoch nicht im Hauptmenü **Debug Main Menu** befinden und das Direktkommando EXIT eingeben oder PF3 (Exit) drücken, dann verlassen Sie die aktuelle Funktion und kehren zum vorigen Schritt Ihrer Debugging-Sitzung zurück.

GO

Wenn Sie das Direktkommando GO eingeben (oder PF14 drücken), gibt der Debugger die Steuerung an die Ausführung des unterbrochenen Natural-Objekts zurück. Wenn zum Zeitpunkt, als das Natural-Objekt unterbrochen wurde, ein Breakpoint oder Watchpoint aktiv war, werden die verbleibenden Kommandos dieses Breakpoint oder Watchpoint *nicht* ausgeführt.

NEXT

Wenn Sie das Direktkommando NEXT eingeben (oder PF13 drücken), wird das nächste für einen Breakpoint oder Watchpoint angegebene Kommando ausgeführt. Falls kein weiteres Kommando angegeben ist, wird die Programmausführung fortgesetzt.

RUN

Wenn Sie das Direktkommando RUN eingeben, wird der Test-Modus ausgeschaltet und die Programmausführung fortgesetzt, ohne dass weitere Breakpoints oder Watchpoints untersucht werden.

STEP

Wenn Sie das Direktkommando `STEP` eingeben, wird ein unterbrochenes Natural-Objekt um *n* ausführbare Statements fortgesetzt. Der Standardwert für *n* ist 1.

STEP SKIPSUBLEVEL

Wenn Sie das Direktkommando `STEP SKIPSUBLEVEL` bei einem Statement eingeben, das ein anderes Objekt aufruft (zum Beispiel, `CALLNAT`), erfolgt die Fortsetzung der Verarbeitung mit dem nächsten ausführbaren Statement im aktuellen Objekt, anstatt mit dem ersten ausgeführten Statement im aufgerufenen Objekt.

Wenn Sie dieses Direktkommando bei einem Statement anwenden, das kein anderes Objekt aufruft, reagiert der Debugger so, als ob Sie das Direktkommando `STEP` eingegeben hätten.

STEP SKIPSUBLEVEL n

Mit dem Direktkommando `STEP SKIPSUBLEVEL` können Sie eine übergeordnete Level-Nummer *n* angeben. Der Step-Modus wird dann beim nächsten Objekt auf der angegebenen Ebene fortgesetzt. Beispiel: Wenn Sie `STEP SKIPSUBLEVEL 2` in einem Objekt auf Ebene 4 eingeben, dann setzen Sie den Step-Modus in dem Objekt auf Ebene 2 fort.

Informationen zur Ebene eines Objekts können Sie mit dem Kommando `OBJCHAIN` aufrufen. Siehe Abschnitt *Kommandos zum Navigieren und Anzeigen von Informationen*.

STOP

Wenn Sie das Direktkommando `STOP` eingeben, werden sowohl der Debugger als auch ein unterbrochenes Natural-Objekt beendet.



Anmerkung: Natural Security kann die Ausführung dieses Kommandos unterbinden, siehe *Components of an Environment Profile* in der *Natural Security*-Dokumentation.

16 Kommandos zum Navigieren und Anzeigen von Informationen

- BREAK 126
- FLIP 126
- LAST 126
- OBJCHAIN 126
- ON/OFF 127
- PROFILE 127
- SCAN 128
- SCREEN 128
- SET OBJECT 128
- STACK 129
- SYSVARS 129
- TEST ON/OFF 129

Dieser Abschnitt beschreibt die Direktkommandos, die der Debugger zum Navigieren durch die Debugging-Bereiche, zum Blättern in den Bildschirmanzeigen, zum Abrufen verschiedener Informationen über Objekte und Variablen und zum Angeben von Profilen bereitstellt.

Eine Übersicht über alle mit dem Debugger verfügbaren Kommandos finden Sie unter *Debug-Kommandoübersicht und -syntax*.

BREAK

Das Kommando `BREAK` ist das Standard-Kommando, das automatisch beim Anlegen eines neuen Debug-Eintrags gesetzt wird. Es bewirkt die Anzeige des **Debug Break**-Fensters. Beschreibung siehe *Debug Break-Fenster* im Abschnitt *Debugger-Konzept*.

Wenn beim Ändern eines Debug-Eintrags das Kommando `BREAK` gelöscht wird, erscheint kein **Debug Break**-Fenster. Es werden jedoch sonstige angegebene Kommandos ausgeführt und der Ereigniszähler wird erhöht.

FLIP

Das Kommando `FLIP` schaltet in Bildschirmen die Anzeige der beiden PF-Tastenzeilen um: (PF1 bis PF12 und PF13 bis PF24).

LAST

Das Kommando `LAST` zeigt das zuletzt eingegebene Kommando an. Die letzten drei Kommandos werden gespeichert und können angezeigt und erneut ausgeführt werden.

OBJCHAIN

Das Kommando `OBJCHAIN` kann nur benutzt werden, wenn ein Natural-Objekt unterbrochen worden ist.

Dieses Kommando zeigt die Objekte auf der aktuellen Ebene und auf allen übergeordneten Ebenen sowie, falls zutreffend, die aktuelle Global Data Area (GDA) an. Außerdem werden Informationen über die Unterbrechung angezeigt.

ON/OFF

Wenn Sie im Debugger das Kommando `ON` oder `OFF` eingeben, wird der Test-Modus ein- bzw. ausgeschaltet. Siehe auch *TEST ON/OFF*.

PROFILE

Das Kommando `PROFILE` ruft den Bildschirm **User Profile** auf. Dort können Sie die Profileinstellungen für den Debugger ändern.

User Profile-Bildschirm

Der Bildschirm **User Profile** bietet folgende Optionen:

Option	Erklärung
Reset debug environment automatically on exit	Gibt an, dass ein automatischer Reset Ihrer Debug-Umgebung erfolgt, wenn Sie den Debugger verlassen.
File for loading/saving debug environments	Gibt an, in welche bzw. aus welcher Systemdatei Debug-Umgebungen gespeichert bzw. geladen werden sollen: `FUSER` (Standardeinstellung), `FNAT` oder `SPAD` (Scratch-Pad File).
Confirm EXIT/CANCEL before execution	Gibt an, dass vor der Ausführung eines `EXIT`- oder `CANCEL`-Kommandos eine Bestätigungsabfrage erfolgen soll. Die Standardeinstellung ist `N` (No).
Stack unknown commands	Gibt an, dass ein unbekanntes Debug-Kommando, das eingegeben wird (z.B. der Name eines gerufenen Programms) im Natural-Stack zwischengespeichert werden soll. Wenn dies der Fall ist, wird nach der Eingabe eines unbekannten Debug-Kommandos der Debugger sofort verlassen und das Kommando wird ausgeführt. Wenn diese Option nicht angegeben worden ist, führt ein unbekanntes Debug-Kommando zu einer entsprechenden Fehlermeldung. Die Standardeinstellung ist `Y` (Yes).
Output device	Angabe eines Druckers für die Funktionen **Call statistics maintenance** (siehe *Objekte drucken*) und **Statement execution statistics maintenance** (siehe *Statements drucken*). Der Standardwert ist `HARDCOPY`. Wenn Sie die Ausgabe an einen anderen Drucker leiten möchten, müssen Sie `HARDCOPY` durch den Namen eines anderen, von Ihrem Systemadministrator verfügbar gemachten gültigen Druckers ersetzen.

Option	Erklärung
Maximum Debug buffer size in KB	Gibt die maximale Größe (in Kilobytes) des Debug-Pufferspeichers an. Der Debug-Pufferspeicher wird bedarfsabhängig automatisch vergrößert, jedoch nur bis zur hier angegebenen maximalen Größe. Geben Sie `0` ein, um anzugeben, das keine Einschränkung besteht, oder einen Wert von `4` - `16384` (muss ein Vielfaches von 4 sein). Bei Überschreiten des Limits können keine weiteren Debug-Einträge definiert werden, und es werden keine zusätzlichen Call- oder Statement-Ausführungsstatistiken generiert.

SCAN

Kann nur bei der Funktion **List object source** angewendet werden, siehe *List Object Source (Objekt-Quellcode anzeigen)*.

Dieses Kommando ermöglicht die Suche nach einer Zeichenkette im Quellcode eines Objekts.

- `SCAN` sucht nach dem Wert, der begrenzt wird durch Leerzeichen oder beliebige Zeichen, bei denen es sich weder um Buchstaben noch um numerische Zeichen handelt.
- `SCAN ABS` führt zu einer absoluten Durchsuchung des Quellcode nach dem angegeben Wert, unabhängig davon welche anderen Zeichen den Wert umgeben.

Siehe auch die Syntax-Diagramme in *Debug-Kommandoübersicht und -syntax*.

SCREEN

Wenn Sie bei einer Unterbrechung eines Natural-Objekts das Kommando `SCREEN` eingeben, wird die aktuelle Bildschirmausgabe des unterbrochenen Natural-Objekts angezeigt. Wenn Sie ENTER drücken, erfolgt die Rückkehr in den Debug-Modus.

SET OBJECT

Das Kommando `SET OBJECT` dient zum Ändern des **Standard-Objekts**, wie im entsprechenden Abschnitt in *Debugger starten* beschrieben. Siehe auch die Syntax des Direktkommandos `SET` im Abschnitt *Debug-Kommandoübersicht und -syntax*.

STACK

Wenn Sie das Kommando `STACK` eingeben, wird der Inhalt des Eintrags, der sich oben auf dem Natural-Stack befindet, angezeigt. Es können bis zu 15 oben auf dem Stack abgelegte Eintragselemente angezeigt werden. Elemente, die mehr als 55 Zeichen haben, werden abgeschnitten und mit einem Stern (*) markiert.



Anmerkung: Wenn irgendein Einzelelement länger als 249 Zeichen ist, wird eine Fehlermeldung angezeigt.

SYSVARS

Wenn Sie das Kommando `SYSVARS` eingeben, werden die aktuellen Werte eines begrenzten Satzes an Systemvariablen angezeigt.

TEST ON/OFF

Das Kommando `TEST ON` bzw. `TEST OFF` dient zum **Ein- bzw. Ausschalten des Test-Modus**. Im Debugger brauchen Sie, wie weiter **oben** beschrieben, nur `ON` oder `OFF` einzugeben.

Die Benutzung des Kommandos `TEST` kann durch Natural Security unterbunden werden, siehe *Command Restrictions* im Abschnitt *Library Maintenance* in der *Natural Security*-Dokumentation.

17 Debug-Kommandoübersicht und -syntax

- Alle Debug-Kommandos 132
- Syntax-Diagramme 138

Dieser Abschnitt beschreibt alle Debugger-Kommandos, mit denen Sie Debugging-Funktionen per Direktaufruf ausführen oder in Debugger-Bildschirmen navigieren können.

Eine Erklärung komplexerer Kommandostrukturen mit benutzerdefinierten Operanden finden Sie im Abschnitt *Syntax-Diagramme* weiter unten.

Alle Debug-Kommandos

Die in der folgenden Tabelle aufgelisteten Kommandos können in der Kommandozeile eines beliebigen Debugger-Bildschirms eingegeben werden. Ein unterstrichener Teil eines Debug-Kommandos oder -Unterkommandos stellt dessen minimale Abkürzung dar.

Kommando	Unterkommando(s)	Erklärung
-		In einer Liste eine Seite nach oben blättern.
--		An den Anfang einer Liste blättern.
TOP		
+		In einer Liste eine Seite nach unten blättern
++		An das Ende einer Liste blättern.
BOTTOM		
ACTIVATE (**Syntax** siehe unten)	BREAKPOINT oder BP	**Breakpoints aktivieren**, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	SPY	Breakpoints *und* Watchpoints aktivieren, siehe auch **Spy aktivieren** in *Spy Maintenance (Verwaltung der Debug-Einträge)*.
	WATCHPOINT oder WP	**Watchpoints aktivieren**, siehe *Watchpoint Maintenance (Verwaltung der Watchpoints)*.
BM		Menü **Breakpoint Maintenance** aufrufen, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
BREAK		Fenster **Debug Break** anzeigen, siehe auch BREAK in *Kommandos zum Navigieren und Anzeigen von Informationen*.
CANCEL		Aktuelle Operation abbrechen und/oder Bildschirme ohne Speicherung der Änderungen verlassen.

Kommando	Unterkommando(s)	Erklärung
DBLOG	A oder Q oder D	Utility DBLOG aus dem Debugger heraus aufrufen (siehe *Utilities*-Dokumentation). Um eine Datenbankumgebung anzugeben, können Sie eines der folgenden Unterkommandos benutzen: ■ A = Adabas (Standardeinstellung) ■ Q = SQL **Anmerkung:** Während einer Debug-Unterbrechung können Sie nur eines der oben genannten Unterkommandos angeben.
DEACTIVATE oder DA (**Syntax** siehe unten)	BREAKPOINT oder BP	**Breakpoint deaktivieren**, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	SPY	Breakpoints *und* Watchpoints deaktivieren, siehe auch ***Spy deaktivieren***.
	WATCHPOINT oder WP	**Watchpoint deaktivieren**, siehe auch *Watchpoint Maintenance (Verwaltung der Watchpoints)*.
DELETE (**Syntax** siehe unten)	BREAKPOINT oder BP	**Breakpoint löschen**, siehe auch *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	SPY	Breakpoints *und* Watchpoints löschen, siehe auch ***Spy löschen***.
	WATCHPOINT oder WP	**Watchpoint löschen**, siehe *Watchpoint Maintenance (Verwaltung der Watchpoints)*.
	ENVIRONMENT	Angegebene Debug-Umgebung löschen, siehe auch ***Debug-Umgebung löschen***.
DISPLAY (**Syntax** siehe unten)	BREAKPOINT oder BP	**Breakpoint anzeigen**, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	SPY	Breakpoints *und* Watchpoints anzeigen, siehe auch ***Spy anzeigen***.
	WATCHPOINT oder WP	**Watchpoint anzeigen**, siehe *Watchpoint Maintenance (Verwaltung der Watchpoints)*.

Kommando	Unterkommando(s)	Erklärung
	CALL	Statistiken anzeigen zu Natural-Objekten, die während der Ausführung einer Operation aufgerufen worden sind, siehe auch ***Aufgerufene Objekte anzeigen***.
	EXEC	Statistiken anzeigen zu ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe auch ***Nur die ausgeführten Statement-Zeilen anzeigen***.
	HEXADECIMAL	Inhalt von Variablen in hexadezimalem Format anzeigen.
	NOCALL	Statistiken anzeigen zu Natural-Objekten, die während der Ausführung einer Operation nicht aufgerufen worden sind, siehe auch ***Nicht aufgerufene Objekte anzeigen***.
	NOEXEC	Statistiken anzeigen zu nicht ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe auch ***Nur die nicht ausgeführten Statement-Zeilen anzeigen***.
	OBJECT	Statistiken anzeigen zur Aufrufhäufigkeit von Objekten, siehe auch ***Alle Objekte anzeigen***.
	STATEMENT	Statistiken anzeigen zu ausgeführten und nicht ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe ***Alle Statement-Zeilen anzeigen*** .
	VARIABLE	**Variablen anzeigen** für unterbrochene Natural-Objekte, siehe ***Variable Maintenance (Anzeigen und Ändern von Variablen)***.
	XSTATISTICS	Statistik-Zusammenfassung von Ausführungsstatistiken anzeigen, siehe auch ***Statement-Ausführungsstatistiken anzeigen***.
EM		Menü **Debug Environment Maintenance** aufrufen, siehe *Debug Environment Maintenance (Verwaltung der Debug-Umgebung)*.
ESCAPE	BOTTOM	Verarbeitung einer Schleife stoppen und Verarbeitung mit dem ersten Statement nach der Schleife fortsetzen, siehe `ESCAPE BOTTOM` in *Kommandos zur Ausführungssteuerung*.
	ROUTINE	Verarbeitung eines unterbrochenen Natural-Objekts stoppen und, falls vorhanden, mit einem anderen Objekt fortsetzen, siehe `ESCAPE ROUTINE` in *Kommandos zur Ausführungssteuerung*.
EXIT		Aktuellen Bildschirm verlassen, siehe `EXIT` in *Kommandos zur Ausführungssteuerung*.
FLIP		Anzeige der beiden PF-Tastenreihen umschalten (PF1 bis PF12 und PF13 bis PF24).
GO		Rückgabe der Steuerung an die Ausführung des unterbrochenen Natural-Objekts, siehe `GO` in *Kommandos zur Ausführungssteuerung*.
LAST		Zuletzt eingegebenes Kommando anzeige. Die drei letzten Kommandos werden gespeichert und können abgerufen werden.
LEFT		Zur linken Seite einer Quellcode-Auflistung verschieben.
LIST		Quellcode eines Objekts anzeigen.

Kommando	Unterkommando(s)	Erklärung
(**Syntax** siehe unten)	BREAK	Quellcode eines Objekts mit der aktuellen Unterbrechung anzeigen. Die relevante Statement-Zeile wird hervorgehoben.
	LASTLINE	Quellcode eines Objekts mit der letzten Zeile anzeigen, die vor der aktuellen Unterbrechung ausgeführt wurde.
LOAD (**Syntax** siehe unten)	ENVIRONMENT	Angegebene Debug-Umgebung laden, siehe ***Debug-Umgebung laden***.
MENU		Hauptmenü **Debug Main Menu** aufrufen.
MODIFY (**Syntax** siehe unten)	BREAKPOINT oder BP	**Breakpoint ändern**, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	SPY	Bildschirm **Modify Breakpoint** oder **Modify Watchpoint** aufrufen, siehe auch ***Debug-Einträge (Spies) ändern*** in *Spy Maintenance (Verwaltung der Debug-Einträge)*.
	WATCHPOINT oder WP	**Watchpoint ändern**, siehe *Watchpoint Maintenance (Verwaltung der Watchpoints)*.
	HEXADECIMAL	Inhalt von Variablen in hexadezimalem Format ändern.
	VARIABLE	Bildschirm **Display Variable** zwecks Änderung aufrufen, siehe ***Variable ändern***.
NEXT		Nächstes, für einen Breakpoint oder Watchpoint angegebenes Kommando ausführen.
OBJCHAIN		Ausgeführte Objekte auf verschiedenen Programmebenen anzeigen, siehe OBJCHAIN in *Kommandos zum Navigieren und Anzeigen von Informationen*.
ON oder OFF		Test-Modus ein- und ausschalten, siehe auch ***Test-Modus ein- und ausschalten***.
PRINT (**Syntax** siehe unten)	CALL	Statistiken drucken zu Natural-Objekten, die während der Ausführung einer Anwendung aufgerufen wurden, siehe auch ***Aufgerufene Objekte anzeigen***.
	EXEC	Statistiken drucken zu ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe auch ***Nur die ausgeführten Statement-Zeilen anzeigen***.
	NOCALL	Statistiken drucken zu Natural-Objekten, die während der Ausführung einer Anwendung nicht aufgerufen wurden, siehe auch ***Nicht aufgerufene Objekte anzeigen***.

Kommando	Unterkommando(s)	Erklärung
	NOEXEC	Statistiken drucken zu nicht ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe auch ***Nur die nicht ausgeführten Statement-Zeilen anzeigen***.
	OBJECT	Statistiken drucken zur Aufrufhäufigkeit (*Call Frequency*) von Objekten, siehe auch ***Aufgerufene Objekte anzeigen***.
	STATEMENT	Statistiken drucken zu ausgeführten und nicht ausgeführten Statement-Zeilen von aufgerufenen Natural-Objekten, siehe auch ***Alle Statement-Zeilen anzeigen***.
	XSTATISTICS	Statistiken drucken zu ausgeführten Statement-Zeilen, siehe auch ***Statement-Ausführungsstatistiken anzeigen***.
PROFILE		Bildschirm **User Profile** anzeigen. Dort können Sie die Profileinstellungen des Debuggers ändern, siehe ***Kommandos zum Navigieren und Anzeigen von Informationen***.
RESET (**Syntax** siehe unten)	ENVIRONMENT	Aktuelle Debug-Umgebung zurücksetzen, siehe ***Debug-Umgebung zurücksetzen***.
RIGHT		Zur rechten Seite einer Quellcode-Auflistung verschieben.
RUN		Test-Modus ausschalten und Programmausführung fortsetzen.
SAVE (**Syntax** siehe unten)	ENVIRONMENT	Aktuelle Umgebung zurücksetzen und die Debug-Angaben speichern, siehe auch ***Debug-Umgebung speichern***.
SCAN	ABS	Nur anwendbar bei der Funktion **List object source** (siehe *List Object Source (Objekt-Quellcode anzeigen)*). Suche nach einem Wert im Quellcode eines Objekts, siehe SCAN in *Kommandos zum Navigieren und Anzeigen von Informationen* und ***Syntax-Diagramme*** weiter unten.
SCREEN		Wenn bei Unterbrechung eines Objekts eingegeben, wird die aktuelle Bildschirmausgabe des unterbrochenen Natural-Objekts angezeigt. Rückkehr in den Debug-Modus nach Drücken von ENTER.
SET (**Syntax** siehe unten)	BREAKPOINT oder BP	Bildschirm **Set Breakpoint** aufrufen, siehe *Breakpoint Maintenance (Verwaltung der Breakpoints)*.
	CALL ON oder CALL OFF	Call-Statistik aktivieren oder deaktivieren, siehe *Call Statistics Maintenance (Statistiken über gerufene Objekte)*.
	OBJECT	Das für den Debugger angegebene Standard-Objekt ändern, siehe auch SET OBJECT in *Kommandos zum Navigieren und Anzeigen von Informationen*.

Kommando	Unterkommando(s)	Erklärung
	WATCHPOINT oder WP	Bildschirm **Watchpoint setzen** aufrufen, siehe ***Watchpoint Maintenance (Verwaltung der Watchpoints)***.
	XSTATISTICS ON oder XSTATISTICS COUNT oder XSTATISTICS OFF	Statement-Ausführungsstatistik aktivieren (ON oder COUNT) deaktivieren (OFF), siehe ***Funktion Statement Execution Statistics auf ON/OFF/COUNT setzen***.
SM		Menü **Spy Maintenance** aufrufen, siehe *Spy Maintenance (Verwaltung der Debug-Einträge)*.
STACK		Inhalt des obersten Eintrags im Natural-Stack anzeigen, siehe STACK in *Kommandos zum Navigieren und Anzeigen von Informationen*.
STEP	[*n*]	Unterbrochenes Natural-Objekt für eine mit dem Kommando angegebene Anzahl (*n*) ausführbarer Statements fortsetzen. Wenn Sie *n* nicht angeben, wird standardmäßig ein ausführbares Statement übersprungen. Siehe auch STEP in *Kommandos zur Ausführungssteuerung*.
	SKIPSUBLEVEL [*n*]	Schrittweise Bearbeitung von Natural-Objekten fortsetzen, ohne Programme auf Unterebenen zu erfassen. Sie können eine Ebenen-Nummer (*n*) angeben. Siehe auch SKIPSUBLEVEL in *Kommandos zur Ausführungssteuerung*.
STOP		Debugger und unterbrochenes Natural-Objekt beenden. Die Eingabeaufforderung NEXT erscheint.
SYSVARS		Aktuelle Werte eines begrenzten Satzes an Systemvariablen anzeigen (außer datenbankbezogene Systemvariablen). Siehe auch ***Systemvariablen anzeigen***.
TEST ON oder TEST OFF		**Test-Modus ein- und ausschalten**.
WM		Ruft das Menü **Watchpoint Maintenance** auf. Beschreibung siehe *Watchpoint Maintenance (Verwaltung der Watchpoints)*.

Syntax-Diagramme

Die im Folgenden aufgeführten Syntax-Diagramme beziehen sich auf komplexere Kommandosequenzen.

Ausführliche Erläuterungen zu den Symbolen, die innerhalb der Syntax-Beschreibungen verwendet werden, siehe Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation.

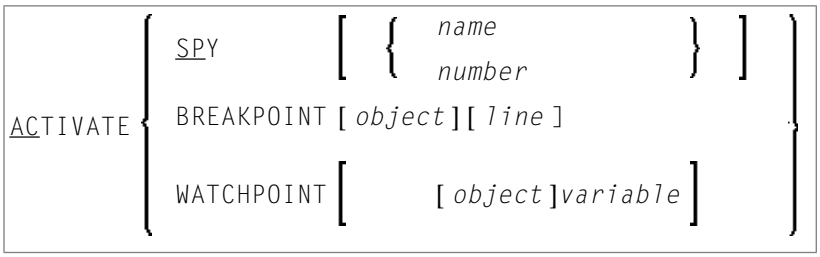
Zur besseren Lesbarkeit sind synonyme Schlüsselwörter in den unten aufgeführten Syntax-Diagrammen weggelassen worden. Eine Unterstreichung eines Schlüsselwortteils bedeutet, dass Sie das Schlüsselwort auch in entsprechend abgekürzter Form als eingeben können.

Gültige synonyme Schlüsselwörter sind:

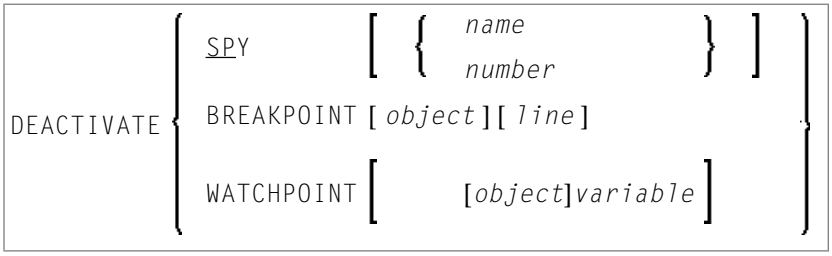
Schlüsselwort	Synonym
BREAKPOINT	BP
DEACTIVATE	DA
WATCHPOINT	WP

- ACTIVATE
- DEACTIVATE
- DELETE
- DISPLAY
- LIST
- LOAD
- MODIFY
- PRINT
- RESET
- SAVE
- SET

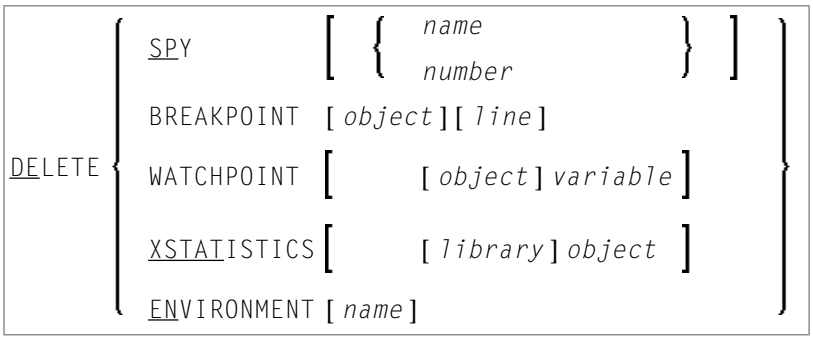
ACTIVATE



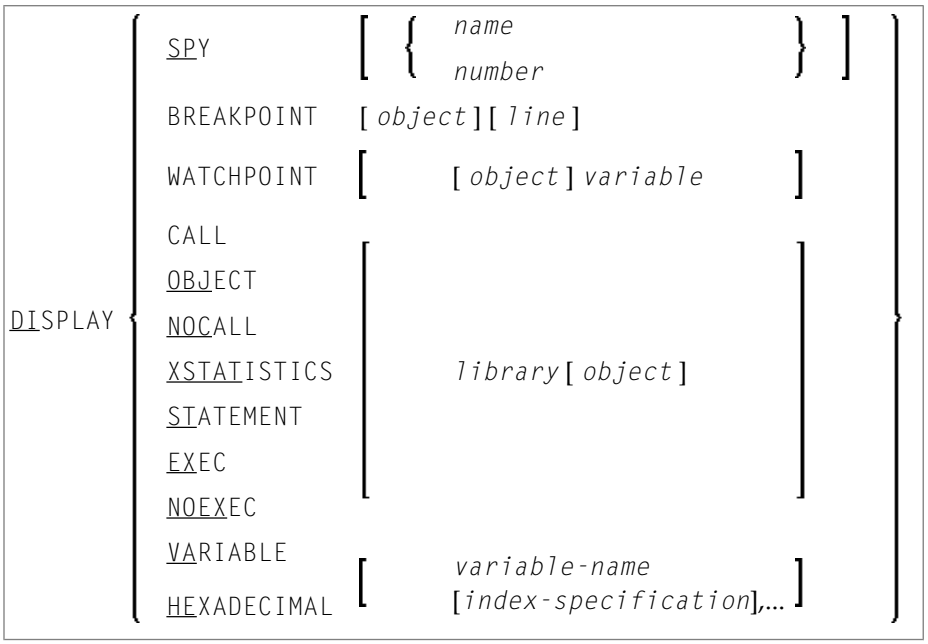
DEACTIVATE



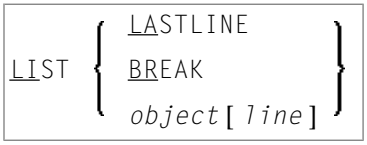
DELETE



DISPLAY



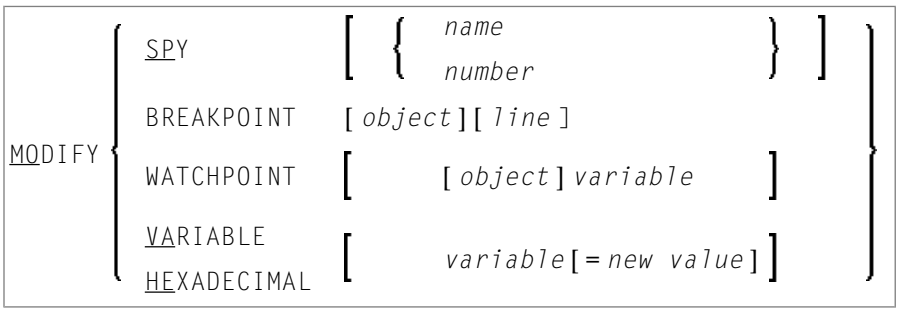
LIST



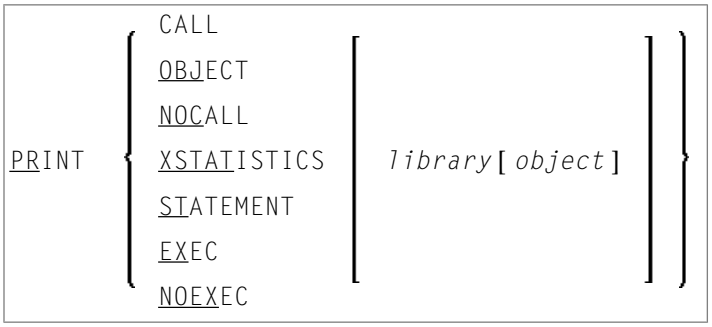
LOAD



MODIFY



PRINT



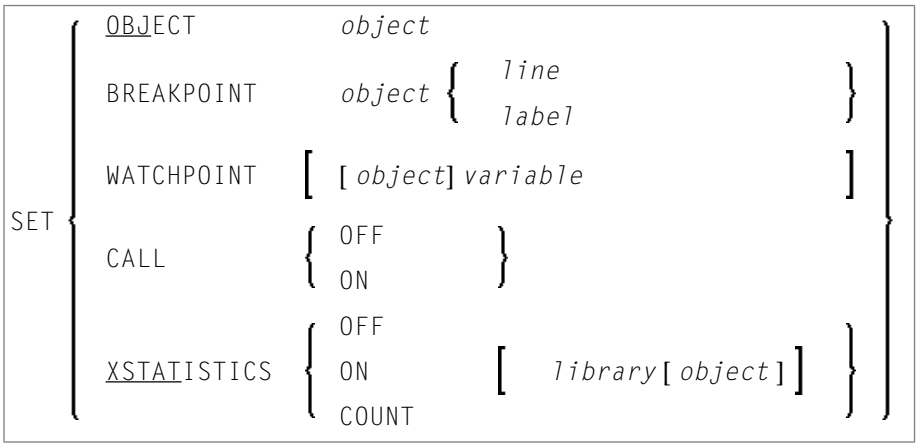
RESET



SAVE

```
SAVE ENVIRONMENT[ name ]
```

SET



18 Natural für Attached-Debugging vorbereiten

- Einleitung 144
- Voraussetzungen für Attached Debugging 144
- Beispiel für z/OS Batch 145

Einleitung

Dieses Dokument liefert Informationen, wie Sie den Debug Attach Server (DAS) aktivieren können, um das Debugging einer externe Natural-Anwendung mittels NaturalONE durchzuführen.

Eine externe Natural-Anwendung läuft in einer Natural-Umgebung, speichert aber ihren Quellcode in einem NaturalONE-Project. Der Debug Attach Server (DAS) wird benutzt, um auf ein NaturalONE-Project zuzugreifen.

Weitere Informationen zur Benutzung des Debug Attach Server (DAS) siehe *NaturalONE*-Dokumentation.

Voraussetzungen für Attached Debugging

Damit Sie aus einer Natural-Sitzung heraus auf den NaturalONE-Debugger zugreifen können, müssen folgende Voraussetzungen erfüllt sein:

- Die Natural-Sitzung läuft in einer z/OS-Umgebung.
- NaturalONE ist installiert.
- Der Natural Development Server ist installiert und die installierte Version muss verbundenes Debugging unterstützen.
- Das Modul `NATAD`*`vrs`* (oder `NCIAD`*`vrs`* für eine CICS-Session auf z/OS) wird aus der Natural Development Server Library generiert und die Natural-Sitzung kann darauf zugreifen.
- Der Profilparameter `DBGAT` ist angegeben.
- Der Profilparameter `RCA` ist auf `NATATDBG` gesetzt.
- Der Profilparameter `RCALIAS` ist auf `(NATATDBG,NATAD`*`vrs`*`)` gesetzt, für CICS auf z/OS auf `(NATATDBG,NCIAD`*`vrs`*`)`.
- Der Debug Attach Server (DAS) läuft und kann über TCP/IP adressiert werden. Der DAS wird mit NaturalONE als `NATDAS.EXE`-Datei ausgeliefert.

Ausführliche Informationen zu den oben erwähnten Natural-Profilparametern, siehe *Parameter-Referenz*-Dokumentation.

Beispiel für z/OS Batch

Eine Natural-Batch-Anwendung soll mit NaturalONE auf Programmfehler untersucht werden. Der DAS-Server ist unter dem TCP/IP-Namen `DASSERV` verfügbar und empfängt am Port `50882`. Der NaturalONE-Debugger hat sich beim Debug Attach Server mit der Client ID `FRED` identifiziert. Die angebundene Debug-Schnittstelle befindet sich in der Library `DSN NDV`*`vrs`*`.LOAD`:

```
//NATBAT  EXEC PGM=NATBATvr
//STEPLIB DD   DISP=SHR,DSN=NATvrs.LOAD
//        DD   DISP=SHR,DSN=NDVvrs.LOAD
//CMPRMIN DD   *
RCA=NATATDBG,RCALIAS=(NATATDBG,NATADvrs)
DBGAT=(ACTIVE=ON,HOST=DASSERV,PORT=50882,CLID=FRED)
/*
```

II Dienstprogramme gruppiert nach Verwendungszweck

19 Dienstprogramme gruppiert nach Verwendungszweck

Die folgende Liste gibt eine Übersicht über die Natural-Dienstprogramme (Utilities), sortiert nach Verwendungszweck.

Systemverwaltung	Debugging	Monitoring	Objekt-Transfer
SYSAPI	ADACALL	Natural Profiler	INPL
SYSCP	DBLOG	SYSBPM	Object Handler
NATRJE	Debugger	SYSEDT	SYSMAIN
SYSBPM	DUMP (s. *Systemkommandos*)	SYSRDC	
SYSEDT	NATPAGE	SYSTP	
SYSERR	Recording		
SYSEXT	SYSRDC		
SYSEXV			
SYSNCP			
SYSPARM			
SYSRPC			
SYSTP			

III Utility-Aktivierung

20 Utility-Aktivierung

Beim Aufrufen eines Natural-Dienstprogramms (Utility) führt Natural keine Anmeldung (Logon) bei der entsprechenden Utility Library in der Systemdatei FNAT aus, was zur Folge hat, dass der globale Datenbereich (Global Data Area, GDA) und/oder die anwendungsunabhängigen Variablen (Application-Independent Variables, AIVs) erhalten bleiben. Die aktuelle Benutzer-Library und die Einstellungen werden beibehalten. (Zum Zurücksetzen von GDA und/oder AIV siehe Profilparameter `FREEGDA` in der *Parameter-Referenz*-Dokumentation.)

Damit die Einstellungen Ihrer Anwendungsumgebung erhalten bleiben, dürfen Sie sich nicht bei einer Utility Library anmelden (d.h. kein Logon durchführen). Benutzen Sie stattdessen das Natural-Systemkommando, das der Utility entspricht.

Nach dem Beenden einer Utility kehren Sie wieder zu der Library zurück, aus der Sie die Utility aufgerufen haben.

Wenn Sie sich jedoch explizit bei einer Utility Library anmelden, bevor Sie die Utility aufrufen, bleiben Sie nach Beendigung der Utility in dieser (Utility) Library.

Ausnahme: Bei den Utilities **SYSEXT** und **SYSEXV** wird ein implizites Logon bei der entsprechenden Utility Library durchgeführt, weil das Bearbeiten von Source-Objekten nur in einer aktiven Library möglich ist.

Benutzung von Utilities mit/ohne Natural Security

Wenn Natural Security in Ihrer Umgebung installiert ist, können Sie die Informationen, wie die Benutzung der Natural Utilities mit Natural Security geregelt werden kann, dem Abschnitt *Protecting Utilities* in der *Natural Security*-Dokumentation entnehmen.

Wenn Natural Security nicht installiert ist, können Sie die Benutzung der Natural Utilities mit der User Exit Routine UTI-EX01 regeln. Das Source-Programm für diese User Exit Routine wird als Source-Objekt UTI-SX01 in der Library SYSEXT ausgeliefert.

Um UTI-EX01 zu aktivieren:

1 Benutzen Sie das Systemkommando `CATALOG` oder `STOW`, um das Source-Objekt UTI-SX01 unter dem Namen UTI-EX01 zu katalogisieren.

   Es werden absichtlich verschiedene Namen verwendet, um sicherzustellen, dass das (möglicherweise gemäß Ihren Erfordernissen geänderte) Source-Objekt und das katalogisierte Objekt der User Exit Routine nicht bei einer Update-Installation überschrieben werden.

2 Kopieren Sie UTI-EX01 in die Library SYSTEM in der Systemdatei FNAT oder FUSER.

Eine ausführliche Beschreibung der User Exit Routine finden Sie beim Source-Objekt von UTI-SX01 in der Library SYSEXT.

IV ADACALL Utility

21 ADACALL Utility

- ADACALL aufrufen 158
- ADACALL-Parameter 159
- ADACALL-Kommandos und PF-Tasten 161
- User Exit ADAEXIT 164

Mit dem Dienstprogramm („Utility“) ADACALL können Sie Adabas-Direktaufrufe (native Kommandos) an eine Adabas-Datenbank zum Lernen und Testen sowie zur Problemanalyse ausgeben.

Die ADACALL Utility ist in der Natural Library SYSADA enthalten.

ADACALL aufrufen

Um ADACALL aufzurufen:

- Geben Sie folgendes Systemkommando ein:

```
SYSADA
```

Der Hauptbildschirm der ADACALL Utility wird angezeigt (Beispiel):

```
15:53:32            ***** NATURAL ADACALL UTILITY *****            2006-12-14
 User SAG                  - ADABAS Direct Calls -
 Mode Char                                                 Call No. 45
 *** Control Block ***         First Byte 30
  Cmd L3        Cmd ID SAG           File 316             Database 10
 Resp 0            ISN 382            ISQ 0                   ISL 0
  FBL 210          RBL 980            SBL 140     VBL 140     IBL 0
 COP1             COP2          User Area                 Cmd Time 4
 Addition1        Addition2  Addition3        Addition4        Addition5
 AA]?             227  48
 *** Buffer Areas ***
 Format  AA,AC,AE.

 Record  11111003ARTHUR              DENT

 Search

  Value

    ISN
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Main  Exit             Char  Hex   View  Prnt  Run   Init  Canc
```

Geben Sie im **Hauptbildschirm der ADACALL Utility** die notwendigen Parameterwerte an und führen Sie das Adabas-Kommando aus, indem Sie entweder PF10 (Run) drücken oder das ADACALL-Kommando `EXEC` in der Kommandozeile eingeben.

Im obigen Beispiel-Bildschirm wurde das Adabas-Kommando L3 abgesetzt, um einen logischen Lesevorgang (Logical Read) der Employees-Datei auszuführen.

Mit Ausnahme des Steuerblocks, der vollständig angezeigt wird, wird nur ein Teil des des Puffers angezeigt. Sie können die Puffer in ihrer Gesamtheit anzeigen, indem Sie eines der unten aufgeführten ADACALL-Direktkommandos oder die entsprechende PF-Taste benutzen.

ADACALL-Parameter

Die folgende Tabelle enthält die Parameter, die Sie im **Hauptbildschirm der ADACALL Utility** angeben können. Sie können die Online-Hilfefunktion der ADACALL Utility benutzen, um sich eine Zusammenfassung der Parameterbeschreibungen anzeigen zu lassen.

Um die Online-Hilfefunktion aufzurufen:

- Positionieren Sie den Cursor in dem Feld, zu dem Sie Hilfe benötigen, und geben Sie ein Fragezeichen (?) ein oder drücken Sie PF1. (Bei schreibgeschützten Feldern können Sie nur PF1 benutzen.)

Weitere Informationen siehe Adabas-Dokumentation *Command Reference* und *Messages and Codes*.

Parameter	Erklärung
Mode	Zeigt den Anzeigemodus für die Pufferinhalte an: Char: Zeichenwerte. Hex: Hexadezimalwerte. Informationen zum Ändern des Anzeigemodus siehe ADACALL-Kommandos CHAR und HEX.
Call No.	Anzahl der Kommandos, die seit dem Session-Start ausgeführt worden sind.
First Byte	Das erste Byte des Adabas-Steuerblocks. Zeigt an, ob 1-Byte- oder 2-Byte-Datenbankkenungen (DBID) und Dateinummern (FNR) verwendet werden: H'00' = 1-Byte-DBID, FNR (Dateinummern 1 - 255) H'30' = 2-Byte-DBID, FNR (Dateinummern > 255)
Cmd	Adabas-Kommando. Geben Sie das **Adabas-Kommando** OP ein und führen Sie es aus, um die Parameter anzugeben, die im Abschnitt weiter unten beschrieben sind.
Cmd ID	Kommandokennung (ID).

Parameter	Erklärung
File	Dateinummer. Wenn **First Byte** auf `H'00'` gesetzt ist: 3-stellige Dateinummer, **Database** ungleich 0. Wenn **First Byte** auf `H'30'` gesetzt ist: 5-stellige Dateinummer.
Database	Datenbankkennung (DBID). Standardmäßig ist dies die DBID der Systemdatei FUSER der aktuellen Natural-Session (siehe **File** weiter oben). Wenn **First Byte** auf `H'30'` gesetzt ist, wird zur Ausführungszeit die Datenbanknummer in das Rückmeldecode-Feld des Adabas-Steuerblocks verschoben.
Resp	Response Code. Rückmeldecode, der zurückgegeben wird, nachdem das Kommando ausgeführt worden ist.
ISN	Internal Sequence Number (interne Folgenummer).
ISQ	ISN Quantity (ISN-Menge).
ISL	Niedrigster ISN-Wert für ISN-Listen.
FBL	Format Buffer Length (Formatpufferlänge) in Bytes (maximal 210).
RBL	Record Buffer Length (Datensatzpufferlänge) in Bytes (maximal 980).
SBL	Search Buffer Length (Suchpufferlänge) in Bytes (maximal 140).
VBL	Value Buffer Length (Wertpufferlänge) in Bytes (maximal 140).
IBL	ISN Buffer Length (Pufferlänge für die interne Folgenummer) in Bytes (maximal 200).
COP1	Command Option 1.
COP2	Command Option 2.
User Area	Benutzerbereich für den Steuerblock.
Cmd Time	Die Zeit für die Ausführung des Kommandos, der Einfachheit halber umgerechnet in 1/100 Sekunden.
Addition1	Additions 1.
Addition2	Additions 2. Wenn der Aufruf erfolgreich war, zeigt dieses Feld die komprimierte Länge des gelesenen Datensatzes und die dekomprimierte Länge der über den Formatpuffer angeforderten Daten an. Wenn eine Antwort ungleich Null zurückgegeben wird und der Fehler auf einen ungültigen Formatpuffer zurückzuführen ist, werden das fehlerhafte Feld und sein Offset im Formatpuffer angezeigt.
Addition3	Additions 3.

Parameter	Erklärung
Addition4	Additions 4. Wenn eine VSAM-Datei gelesen wird, wird dieses Feld auf VSAM gesetzt, wenn es initialisiert ist.
Addition5	Additions 5.
Format	Formatpuffer. (Der abschließende Punkt ist erforderlich.)
Record	Record Buffer (Datensatzpuffer).
Search	Search Buffer (Suchpuffer). (Der abschließende Punkt ist erforderlich.)
Value	Value Buffer (Wertpuffer).
ISN	ISN Buffer (Puffer für die interne Folgenummer).

Adabas-Kommando OP

Wenn Sie das Adabas-Kommando `OP` (Open) ausführen, dann zeigt ADACALL ein Fenster an, in dem Sie die folgenden Parameter angeben können:

- maximale Anzahl an ISNs, die im internen ISN-Puffer gespeichert werden sollen,
- maximale Anzahl an Datensätzen, die im Hold-Status sein dürfen,
- maximale Anzahl an Kommandokennungen (CIDs /Command IDs), die aktiv sein dürfen,
- maximale Zeit, die für die Ausführung eines `Sx`-Kommandos erlaubt ist.

Geben Sie in dem Fenster die relevanten Informationen ein und drücken Sie ENTER.

Informationen zu den Parametern und den gültigen Werten siehe *Adabas Command Reference*-Dokumentation.

ADACALL-Kommandos und PF-Tasten

Die in der folgenden Tabelle aufgeführten Parameter dienen dazu, ADACALL-Parametereinstellungen zu ändern oder zwischen Bildschirmanzeigen umzuschalten. Dazu können Sie entweder ein Kommando in der Kommandozeile eingeben oder die entsprechende PF-Taste drücken.

In der Kommandozeile können Sie nicht nur ADACALL-Kommandos, sondern auch Natural-Systemkommandos absetzen.

In der folgenden Tabelle stellt ein unterstrichener Teil eines Kommandos eine akzeptable Abkürzung dar.

Kommando	PF-Taste	Funktion
	PF1	Aufruf der Hilfefunktion für ADACALL. Wenn Sie den Cursor auf einem der ADACALL-Parameter positionieren und PF1 drücken, wird ein Hilfetext zu diesem Parameter angezeigt.
	PF2	Rückkehr zum **Hauptbildschirm der ADACALL Utility**. **Mode** wird dabei auf CHAR gesetzt.
BACK	PF5	Rückwärts blättern zum vorherigen Puffer, wenn die Puffer in ihrer Gesamtheit angesehen werden. Nur gültig, nachdem das Kommando VIEW angewendet worden ist, d.h., das Kommando kann nicht im **Hauptbildschirm der ADACALL Utility** angewendet werden.
CB		Anzeige des Steuerblockpuffers in seiner Gesamtheit. Gilt nur im hexadezimalen Modus.
CHAR	PF6	Ändern des aktuellen Modus nach Zeichenmodus (EBCDIC).
D		Erweiterte Anzeige des Fehlermeldungstextes beim Rückmeldecode. Wenn eine andere Adabas-Rückmeldung als 0 (Null) zurückgegeben wird, wird der entsprechende Kurztext der Fehlermeldung in der Meldungszeile angezeigt. Der Langtext kann mit diesem Kommando angezeigt werden.
EXEC oder RUN	PF10	Ausführen des Direktkommandos mit den angegebenen Parametern.
EXIT oder STOP oder Q oder .	PF3 oder PF12	Beenden. Nach Drücken im **Hauptbildschirm der ADACALL Utility** wird die ADACALL Utility beendet. Nach Drücken während der Anzeige eines der anderen Puffer-Bildschirme wird Hauptbildschirm der ADACALL Utility angezeigt. **Mode** bleibt unverändert.
FB		Anzeige des Formatpuffers in seiner Gesamtheit.
FWD	PF4	Vorwärts blättern zum nächsten Puffer, wenn die Puffer in ihrer Gesamtheit angesehen werden.. Nur gültig, nachdem das Kommando VIEW angewendet worden ist, d.h., das Kommando kann nicht im **Hauptbildschirm der ADACALL Utility** angewendet werden.
HEX	PF7	Aktuellen Modus in hexadezimal ändern.
IB		Anzeige des ISN Buffer (Puffer für die interne Folgenummer) in seiner Gesamtheit.

<table>
<tr><th>Kommando</th><th>PF-Taste</th><th>Funktion</th></tr>
<tr><td>INIT</td><td>PF11</td><td>Initialisieren/Zurücksetzen der Puffer.

Es erscheint ein Fenster, in dem Sie zu den angezeigten Puffern einen der folgenden Werte eingeben können:

<table>
<tr><td></td><td></td></tr>
<tr><td>H</td><td>Initialisieren des oder der entsprechenden Puffer mit binären Nullen (H'00').</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Beliebiges Zeichen außer H oder leer</td><td>Initialisieren des oder der entsprechenden Puffer mit Leerzeichen (H'40').</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Leerzeichen</td><td>Die entsprechenden Puffer nicht initialisieren.</td></tr>
<tr><td></td><td></td></tr>
</table>
Bei Eingabe von INIT ALL werden alle Puffer mit Ausnahme des Steuerblocks mit Leerzeichen initialisiert.

Alternativ kann das Kommando INIT FB RB SB VB IB (es müssen nicht alle Puffer angegeben werden) eingegeben werden. Dann werden alle in der Auflistung enthaltenen Puffer mit Leerzeichen initialisiert.

Anmerkung: Der ISN-Puffer wird immer mit binären Nullen initialisiert.</td></tr>
<tr><td>PRINT</td><td>PF9</td><td>Generieren und Anzeigen eines Reports über den Status aller Puffer.

Das Natural-Terminalkommando %H kann benutzt werden, um eine Hardcopy zu erzeugen.</td></tr>
<tr><td>RB</td><td></td><td>Vollständige Anzeige des Record Buffer (Datensatzpuffer).</td></tr>
<tr><td>RUN</td><td></td><td>Wie EXEC.</td></tr>
<tr><td>SB</td><td></td><td>Vollständige Anzeige des Search Buffer (Suchpuffer).</td></tr>
<tr><td>VB</td><td></td><td>Vollständige Anzeige des Value Buffer (Wertpuffer)</td></tr>
<tr><td>VIEW</td><td>PF8</td><td>Vollständige Anzeige aller Puffer.

Der erste anzuzeigende Puffer ist der Record Buffer (Datensatzpuffer). Mit dem FWD-Kommando können Sie durch die Anzeigen der übrigen Puffer blättern.

Wenn Sie den Datensatzpuffer per VIEW-Kommando im hexadezimalen Modus betrachten, werden die Daten auf vier Seiten angezeigt:

Um eine Seite weiter zu blättern, können Sie das Kommando FWD benutzen oder PF4 drücken.
Um eine Seite zurück zu blättern, können Sie das Kommando BACK benutzen oder PF5 drücken.
Um eine bestimmte Seite anzuzeigen, können Sie eine Seitenzahl von 1 bis 4 im Feld Specify next page number eingeben.</td></tr>
</table>

Kommando	PF-Taste	Funktion	
		Zur Einzelanzeige der Puffer können Sie eines der folgenden Kommandos eingeben:	
		FB	Format Buffer (Formatpuffer)
		RB	Record Buffer (Datensatzpuffer)
		SB	Search Buffer (Suchpuffer)
		VB	Value Buffer (Wertpuffer)
		IB	ISN Buffer (Puffer für die interne Folgenummer)
		CB	Steuerblock (Standardeinstellung). Gilt nur im hexadezimalen Modus: Ändern in HEX vor Ausführung von VIEW.
VSAM		Wenn für die aktuelle Natural-Session VSAM definiert wurde, kann das Direktkommando VSAM abgesetzt werden, um auf VSAM-Dateien zuzugreifen oder sie zu ändern. Wenn Sie das Kommando eingeben, erscheint ein Fenster, in dem Sie aufgefordert werden, den Namen der VSAM-Datei einzugeben. Wenn das Kommando ausgeführt wird, wird es an die entsprechende VSAM-Datei geleitet.	

User Exit ADAEXIT

Mit der ADACALL Utility können Sie Direktkommandos an eine beliebige Datenbank absetzen. Für Security-Zwecke wird ein User Exit mitgeliefert. Er hat den Namen ADAEXIT und befindet sich in der Library SYSADA. Sie können den User Exit ADAEXIT ändern, um ihn an Ihre Erfordernisse anzupassen.

Der Adabas-Steuerblock wird als Parameter ADAEXIT übergeben. Um den Inhalt des Steuerblocks zu ändern, können Sie den Quellcode des User Exit ändern.

Sie können verhindern, dass Datenbankaufrufe ausgeführt werden, indem Sie einfach die Datenbankkennung oder die Dateinummer ändern oder den Kommandocode nach XX ändern.

V SYSBPM Utility

Die SYSBPM-Dokumentation umfasst die folgenden Kapitel:

- **Allgemeines zum Dienstprogramm SYSBPM**
- **SYSBPM aufrufen und benutzen**
- **Objekte auflisten**
- **Objekte löschen**
- **Verzeichnisinformationen anzeigen**
- **Hexadezimale Anzeige**
- **In Arbeitsdatei schreiben**
- **Sortierten Auszug anzeigen**
- **Buffer Pool-Statistiken anzeigen**
- **BP Cache-Statistiken anzeigen**
- **Message Pool-Statistiken anzeigen**
- **Buffer Pool auswählen**
- **Message Buffer Pool auswählen**
- **Sperrliste verwalten**
- **Preload-Liste verwalten**
- **Performance-Aspekte**
- **SYSBPM-Direktkommandos**
- **Batch-Verarbeitung**
- **Anwendungsprogrammierschnittstellen**

Anmerkungen:

1. In dieser Dokumentation wird manchmal anstelle des Begriffs „Buffer Pool“ die Abkürzung „BP“ verwendet.

2. Anstelle des Begriffs „Message Buffer Pool“ wird in der Benutzungsoberfläche und in der Dokumentation häufig die Abkürzung „Message Pool“ verwendet.

22 Allgemeines zum Dienstprogramm SYSBPM

Das Dienstprogramm SYSBPM dient zur Verwaltung lokaler und globaler Buffer Pools des Typs `Natural`, `DL/I` oder `SORT` sowie von Message Pools.

Neben den Verwaltungsfunktionen bietet SYSBPM auch statistische Informationen über den aktuellen Status eines Buffer Pools beliebigen Typs, einschließlich des Buffer Pool Cache (BP Cache) und eines Message Pool (falls aktiviert), sowie Informationen über die in einen Buffer Pool und den BP Cache geladenen Natural-Objekte und die in einen Message Pool geladenen Meldungen.

Eine Beschreibung des Natural Buffer Pools (d.h. des Buffer Pool vom Typ `Natural`) finden Sie unter *Natural Buffer Pool - General* in der *Operations*-Dokumentation.

Der Buffer Pool wird mit dem Makro `NTBPI` im Natural-Parametermodul oder mit dem entsprechenden dynamischen Profilparameter `BPI` definiert. Beide Optionen werden in der *Parameter-Referenz*-Dokumentation beschrieben. Der Typ eines Buffer Pool wird durch den Subparameter `TYPE` von `NTBPI` oder `BPI` bestimmt.

Anmerkungen:

1. Natural-Objekte, die mit dem Systemkommando `RUN` ausgeführt werden, werden nicht in einen Buffer Pool geladen, sondern nur Natural-Objekte, die mit dem Systemkommando `EXECUTE` ausgeführt werden, d.h. die Objekte müssen gestowed oder katalogisiert worden sein.
2. Der BP Cache und der Message Pool sind optionale Speicherbereiche, die separat aktiviert werden müssen.

23 SYSBPM aufrufen und benutzen

- SYSBPM aufrufen und beenden 170
- Online-Hilfe 172
- SYSBPM Hauptmenü - Felder, Funktionen und Kommandos 173
- SYSBPM in einer z/OS-Parallel-Sysplex-Umgebung 176

Dieses Kapitel beschreibt das Aufrufen des Dienstprogramms SYSBPM und die Benutzung des SYSBPM-Hauptmenüs (**Main Menu**) einschließlich der relevanten Felder, Funktionen und Kommandos. Darüber enthält es Informationen zum Aufrufen der Online-Hilfe und zur Verwendung von SYSBPM in einer z/OS Parallel Sysplex-Umgebung.

SYSBPM aufrufen und beenden

Um das Dienstprogramm SYSBPM aufzurufen:

- Geben Sie das folgende Natural-Systemkommando ein:

```
SYSBPM
```

Es erscheint das SYSBPM-Hauptmenü (**Main Menu**, Beispiel):

```
 10:02:50            ***** NATURAL SYSBPM UTILITY *****            2023-05-23
 BPNAME QA92GBP                - Main Menu -                   Type Global Nat
 BPPROP OFF                                                      Loc DAEF QA92
                                                               Preload QA92GBPL

           Object Functions               Object Pool Statistics

           L  List Objects                A  Buffer Pool
           D  Delete Objects              C  BP Cache
           I  Directory Information       M  Message Buffer Pool
           H  Hexadecimal Display
           W  Write to Work File          Other Functions
           X  Display Sorted Extract
           ?  Help                        S  Select Buffer Pool
           .  Exit                        B  Blacklist Maintenance
                                          P  Preload List Maintenance
   Code .. _   Library ... *_______
               Object .... *________________________________________
               DBID ...... 0____  FNR .. 0____  Object Pool ... * (B,C,*) (M)


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

Um Auswahlfelder für Meldungsobjekte anzuzeigen:

- Geben Sie den Funktionscode `L` in das Eingabefeld **Code ..** und `M` in das Auswahlfeld **Object Pool** ein.

Ein Fenster zur Angabe von Message Pool-Parametern wird angezeigt (Beispiel):

```
10:20:54            ***** NATURAL SYSBPM UTILITY *****            2023-05-23
BPNAME QA92GBP                 - Main Menu -                   Type Global Nat
BPPROP OFF                                                       Loc DAEF QA92
                                                               Preload QA92GBPL

        +--------------Specify Message Pool Parameter--------------+
        ! Message number .. 1___ - 9999   Library .. *________      !
        ! Language code ... 0_            DBID ..... 0____         !
        ! Codepage ........ *________      FNR ...... 0___          !
        !                                                          !
        +----------------------------------------------------------+
           W  Write to Work File           Other Functions
           X  Display Sorted Extract
           ?  Help                         S  Select Buffer Pool
           .  Exit                         B  Blacklist Maintenance
                                           P  Preload List Maintenance
   Code .. L   Library ... *________
               Object .... *________________________________________
               DBID ...... 0____  FNR .. 0____  Object Pool ... M (B,C,*) (M)


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

Um die Anzeige des Fensters abzubrechen:

- Drücken Sie PF3 oder PF12.

Um das Dienstprogramm SYSBPM zu beenden:

- Drücken Sie PF3 oder PF12.

  Oder:

  Geben Sie in der Kommandozeile einen Punkt (.) oder `EXIT` ein.

Nachdem Sie das SYSBPM-Hauptmenü (**Main Menu**) aufgerufen haben, können Sie einen Funktionscode oder eine PF-Taste aus einem SYSBPM-Menü auswählen, um eine SYSBPM-Funktion auszuführen. Alternativ können Sie auch ein **SYSBPM-Direktkommando** verwenden, wie im entsprechenden Abschnitt beschrieben.

Die im SYSBPM-Hauptmenü angebotenen Funktionen sind in drei Abschnitte unterteilt:

- Der Abschnitt **Object Functions** enthält Funktionen zur Anzeige oder Bearbeitung von Natural-Objekten im Buffer Pool oder BP Cache und Natural-System- oder benutzerspezifische Meldungen im Message Pool.
- Der Abschnitt **Object Pool Statistics** enthält Funktionen zum Abrufen objektunabhängiger statistischer Daten über den Buffer Pool oder BP Cache, einschließlich Hash-Tabellen, und den Message Pool. Objektunabhängige Daten enthalten keine individuellen Informationen über das Objekt, wie z.B. Objektname, Größe oder Adressen.
- Der Abschnitt **Other Functions** enthält Funktionen zur Auswahl eines Buffer Pool im aktiven Subsystem (Subsid) und zur Angabe von Objekten, die in den Buffer Pool oder den Message Pool geladen werden sollen bzw. nicht geladen werden sollen.

Eine Beschreibung der verfügbaren Funktionen finden Sie unter *Funktionen*.

Zusätzlich zur Auswahl einer Funktion können Sie auch die Natural-Objekte oder Meldungen auswählen, auf die die Funktion angewendet werden soll, indem Sie eine der folgenden Optionen wählen:

- Füllen Sie die Eingabefelder (entweder für Natural-Objekte oder Meldungen) aus, wie in *SYSBPM Hauptmenü - Felder, Funktionen und Kommandos* beschrieben.
- Oder: Geben Sie in der Kommandozeile ein SYSBPM-Direktkommando ein, wie in *SYSBPM-Direktkommandos* beschrieben.

Online-Hilfe

Die Online-Hilfe des Dienstprogramms SYSBPM bietet Informationen zu den **SYSBPM-Direktkommandos** oder zu gültigen Eingabewerten für Felder, die auf SYSBPM-Bildschirmen vorhanden sind.

Um die Online-Hilfe für SYSBPM-Direktkommandos aufzurufen:

- Positionieren Sie auf einem SYSBPM-Bildschirm den Cursor in der Kommandozeile und drücken Sie PF1 oder geben Sie ein Fragezeichen (?) ein.

  Es erscheint das **Help**-Fenster mit einer Liste aller verfügbaren SYSBPM Direktkommandos.

Um die Online-Hilfe für ein SYSBPM-Eingabefeld aufzurufen:

- Positionieren Sie auf einem SYSBPM-Bildschirm den Cursor in dem betreffenden Eingabefeld und drücken Sie PF1 oder geben Sie ein Fragezeichen (?) ein.

  Es erscheint das **Help**-Fenster für das entsprechende Feld mit einer Liste aller gültigen Eingabewerte.

SYSBPM Hauptmenü - Felder, Funktionen und Kommandos

In diesem Abschnitt werden die folgenden Themen behandelt:

- Felder
- Funktionen
- PF-Tasten und Direktkommandos

Felder

Die folgende Tabelle beschreibt die Felder des SYSBPM-Hauptmenüs (**Main Menu**). Außerdem wird angegeben, ob ein Feld nur für einen Object Pool-Typ spezifisch ist (Buffer Pool und BP Cache bzw. Message Pool).

Felder	Erläuterung
BPNAME	Der Name des globalen Buffer Pools, wie mit dem Profilparameter `BPNAME` angegeben. Bei einem lokalen Buffer Pool wird kein Name, sondern ein leeres **BPNAME**-Feld angezeigt. Siehe auch *BPNAME - Name des globalen Natural Buffer Pool* in der *Parameter-Referenz*-Dokumentation.
BPPROP	Die Einstellung des Profilparameters `BPPROP` zur Steuerung der Weitergabe von Änderungen an einem Objekt in einem Buffer Pool. Siehe auch *BPPROP - Global Buffer Pool-Weiterleitung* in der *Parameter-Referenz*-Dokumentation.
Type	Der Typ des Buffer Pool, z.B. `Global Nat`, `Local Nat`, `Global Sort` oder `Global DL/I`.
Loc	Der Standort. Zeigt die Host-ID (im der obigen Beispielabbildung: `DAEF`) und die Subsystem-ID (im obigen Beispiel: `QA41`) an.
Preload	Der Name einer Preload-Liste, falls geladen. Siehe auch ***Preload-Liste verwalten***.
Library	Der Name der Library, in der das ausgeführte Objekt (entweder aus dem Buffer Pool oder dem Message Pool) gespeichert ist. Sie können einen Namen angeben oder Stern-Notation (*) verwenden. Der standardmäßig vorhandene Stern (*) dient zur Auswahl aller Libraries. Dieses Feld gilt auch für Message Pools.
Object	Der Name des ausgeführten Objekts, das in den Buffer Pool geladen wurde. Sie können einen Namen angeben oder Stern-Notation (*) verwenden. Der standardmäßig vorhandene Stern (*) dient zur Auswahl aller Objekte.

<table>
<tr><th>Felder</th><th>Erläuterung</th></tr>
<tr><td>DBID</td><td>Die Datenbankkennung (DBID) der Systemdatei FNAT oder FUSER, in der das ausgeführte Objekt aus dem Buffer Pool oder Message Pool gespeichert ist und aus der es geladen wird.

Wenn Sie 0 (Null; dies ist der Standardwert) als DBID angeben, wird/werden das/die angegebene(n) Objekt(e) unabhängig von ihrer DBID ausgewählt. Jeder andere Wert als 0 steht für eine bestimmte DBID-Angabe.

Dieses Feld gilt auch für Message Pools.</td></tr>
<tr><td>FNR</td><td>Die Dateinummer (FNR) der Systemdatei FNAT oder FUSER, in der das ausgeführte Objekt aus dem Buffer Pool oder Message Pool gespeichert ist und aus der es geladen wird.

Wenn Sie 0 (Null; dies ist der Standardwert) als FNR angeben, wird/werden das/die angegebene(n) Objekt(e) unabhängig von ihrer FNR ausgewählt. Jeder andere Wert als 0 steht für eine bestimmte FNR-Angabe.

Dieses Feld gilt auch für Message Pools.</td></tr>
<tr><td>Object Pool</td><td>Wählt den Typ des/der zu verwendenden Object Pool(s) aus.
<table>
<tr><td></td><td></td></tr>
<tr><td>B</td><td>Buffer Pool</td></tr>
<tr><td>C</td><td>BP Cache.</td></tr>
<tr><td>*</td><td>Sowohl Buffer Pool als auch BP Cache. Dies ist die Standardeinstellung.</td></tr>
<tr><td>M</td><td>Message Pool</td></tr>
</table>
Anmerkung:

1. Wenn Sie den Message Pool (M) gewählt haben, können Sie nur die Funktionen List Objects, Delete Objects, Select Buffer Pool und Preload List Maintenance ausführen, siehe Funktionen.
2. Der angegebene Object Pool-Typ bestimmt den Teil der Objektliste, der zuerst angezeigt wird. Wenn Sie z.B. das Auswahlkriterium C eingeben, enthält der Teil die im BP Cache geladenen Objekte. Wenn Sie nach oben blättern, werden die im Buffer Pool geladenen Objekte angezeigt. Wenn kein Objekt im BP Cache gefunden wird, werden nur die Objekte des Buffer Pool angezeigt.
3. Wenn die Funktion List Objects auf den Buffer Pool und den BP Cache angewendet wird (durch Auswahl von *), werden alle Objekte angezeigt, die entweder in den Buffer Pool oder in den BP Cache geladen sind. Im Buffer Pool geladene Objekte werden zuoberst angezeigt.</td></tr>
</table>

Spezifische Felder für Meldungen

Die folgenden Felder sind nur für die Auswahl von Meldungen verfügbar:

Felder	Erläuterung
Message Number	Die Nummer der Meldung.
Language Code	Der Sprachcode, unter dem die Meldung gespeichert wurde.
Codepage	Die Codepage, unter der die Meldung gespeichert wurde.

Funktionen

Im Folgenden sind die einzelnen Funktionen aufgeführt.

Sie rufen eine Funktion auf, indem Sie den einbuchstabigen Code, der der gewünschten Funktion entspricht, in das Feld **Code..** eingeben, z.B. `L` für **List Objects**. Beachten Sie, dass Sie für den Message Pool nur die Funktionen **List Objects**, **Delete Objects**, **Select Buffer Pool** und **Preload List Maintenance** angeben können.

Code	Funktion	Erläuterung
`L`	**List Objects**	Zeigt Informationen zu den Objekten an, die entweder in den Buffer Pool und/oder den BP Cache (falls verwendet) oder den Message Pool (falls aktiviert) geladen sind. Auf jedes Listenelement kann einzeln zugegriffen werden, und auf jedes Objekt können verschiedene Zeilenkommandos ausgeführt werden.
`D`	**Delete Objects**	Löscht ein oder mehrere Objekte aus dem Buffer Pool, BP Cache oder Message Pool.
`I`	**Directory Information**	Zeigt die vollständigen Verzeichnisinformationen eines bestimmten Objekts an, das in den Buffer Pool oder den BP Cache geladen wurde.
`H`	**Hexadecimal Display**	Zeigt ein angegebenes, in den Buffer Pool geladenes Objekt im hexadezimalen Format an.
`W`	**Write to Work File**	Schreibt die Objektverzeichnisinformationen, die sich im Buffer Pool und/oder im BP Cache befinden, in eine lokale Datei oder eine PC-Textdatei.
`X`	**Display Sorted Extract**	Zeigt eine sortierte Liste von 50 Objektverzeichnissen an, die sich im Buffer Pool oder BP Cache befinden. Die Listenelemente können nach einem der angegebenen Sortierkriterien geordnet werden.
`A`	**Buffer Pool**	Ruft das Menü **Buffer Pool Statistics** auf. Von diesem Menü aus können Sie objektunabhängige Statistikfunktionen für den Buffer Pool aufrufen, einschließlich Hash-Tabellen-Statistiken.
`C`	**BP Cache**	BP Cache erforderlich. Ruft das Menü **BP Cache** auf. Von diesem Menü aus können Sie objektunabhängige Statistikfunktionen für den BP Cache aufrufen, einschließlich Hash-Tabellen-Statistiken.

Code	Funktion	Erläuterung
M	**Message Pool Statistics**	Message Pool erforderlich. Ruft das objektunabhängige Menü **Message Pool Statistics** auf.
S	**Select Buffer Pool**	Zeigt eine Auswahlliste aller verfügbaren Buffer Pools an.
S und **Object Pool** M	**Select Message Pool**	Zeigt eine Auswahlliste aller verfügbaren Message Pools an.
B	**Blacklist Maintenance**	Ruft das Menü **Blacklist Maintenance** auf, das zur Pflege einer Sperrliste von Objekten dient, die *nicht* ausgeführt werden sollen.
P	**Preload List Maintenance**	Ruft das Menü **Preload List Maintenance** für den Buffer Pool oder den Message Pool (falls aktiviert) auf. In einer Preload-Liste können Sie die Namen von Objekten angeben, die vorab in den Buffer Pool oder den Message Pool geladen werden sollen.

PF-Tasten und Direktkommandos

Im SYSBPM-Hauptmenü (**Main Menu**) können Sie die PF-Tasten oder SYSBPM-Direktkommandos verwenden, die in der folgenden Tabelle aufgeführt sind. Ein unterstrichener Teil eines Kommandos stellt dessen Mindestabkürzung dar. Weitere Kommandos finden Sie unter ***SYSBPM-Direktkommandos***.

PF Key	Kommando	Funktion
PF1		Liefert SYSBPM-Hilfeinformationen: siehe auch ***Online-Hilfe***.
PF3	EXIT	Die aktuelle Funktion/der aktuelle Bildschirm wird verlassen und der vorherige Bildschirm angezeigt.
PF4	LAST	Zeigt das zuletzt eingegebene SYSBPM-Direktkommando an.
PF6	FLIP	Schaltet die PF-Tastenzeile um: Wechsel zwischen der Anzeige von PF1 bis PF12 und PF13 bis PF24.
PF12	CANCEL	Wie EXIT.
PF15	MENU	Rückkehr zum SYSBPM-Hauptmenü (**Main Menu**).

SYSBPM in einer z/OS-Parallel-Sysplex-Umgebung

Wenn Natural zu einem anderen Betriebssystem-Image (Host) wechselt, dann wechselt Natural auch die Buffer Pools. Ein Wechsel der Buffer Pools wird durch eine andere Host-ID angezeigt, die im Feld **Loc** eines SYSBPM-Bildschirms angezeigt wird.

Der Wechsel kann nach jeder Terminal-Ein-/Ausgabe erfolgen, d.h. nach der Wahl einer beliebigen Funktionstaste oder durch Drücken von ENTER. Nach dem Wechsel der Buffer Pools werden die

Kommandos zum Blättern und Positionieren (`TOP`, `BOTTOM`, `+`, `-`, `LEFT`, `RIGHT`). nicht ausgeführt. Stattdessen beginnt die Liste mit dem Anfang des neuen Buffer Pools.

Wenn der Profilparameter `BPPROP` (siehe *BPPROP - Global Buffer Pool-Weiterleitung* in der *Parameter-Referenz*-Dokumentation) auf `PLEX` oder `GPLEX` gesetzt ist, werden SYSBPM-Kommandos, die Sperrlisten verändern, Objekte löschen oder den Buffer Pool initialisieren, zunächst wie üblich ausgeführt und dann an andere Buffer Pools weitergegeben, die im selben Subsystem verfügbar sind. Wenn ein BP-Wechsel dazu führt, dass eine Funktion abgebrochen oder weitergegeben wird, wird eine entsprechende Meldung angezeigt. Eine entsprechende Meldung wird auch angezeigt, wenn Natural erfolgreich zu einem anderen Host gewechselt und die Buffer Pools gewechselt hat.

24 Objekte auflisten

- Natural-Objekte auflisten 180
- Meldungen auflisten 192

Mit der Funktion **List Objects** rufen Sie den gleichnamigen Bildschirm auf, auf dem Sie statistische Daten zu den Verzeichnissen der Natural-Objekte abrufen können, die derzeit in den Buffer Pool oder den BP Cache (falls verwendet) geladen sind. Bei Anwendung dieser Funktion auf den Message Pool wird eine Liste aller Natural-Systemmeldungen oder Benutzernachrichten angezeigt, die derzeit in den Message Pool geladen sind.

Dieses Kapitel enthält Informationen zu den statistischen Daten, die auf dem Bildschirm **List Objects** angezeigt werden, sowie zu den Kommandos und Funktionen, die für die Auswahl eines Objekts oder einer Reihe von Objekten, die Änderung ihres aktuellen Zustands und die Navigation auf dem Bildschirm **List Objects** zur Verfügung stehen.

Da sich der Bildschirm **List Objects** für Natural-Objekte von dem für Meldungen unterscheidet, wird jeder Bildschirm in einem eigenen Kapitel beschrieben:

Natural-Objekte auflisten

In diesem Abschnitt werden die folgenden Themen behandelt:

- Funktion List Objects für Natural-Objekte aufrufen
- Bildschirm List Object für Natural-Objekte: Spalten und Auswahlmöglichkeiten
- PF-Tasten und Direktkommandos für die Liste der Natural-Objekte
- Zeilenkommandos für die Liste der Natural-Objekte

Funktion List Objects für Natural-Objekte aufrufen

Um den Bildschirm List Objects für Natural-Objekte aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `L` in das Eingabefeld **Code ..** und `B`, `C` oder `*` in das Auswahlfeld **Object Pool** ein und geben Sie die Objekte an, siehe gültige **Feldeingabewerte** im Abschnitt ***SYSBPM aufrufen und benutzen***.

  Oder:

  Gehen Sie direkt zur Liste der Objekte im Buffer Pool, indem Sie das folgende SYSBPM-Direktkommando eingeben:

  ```
  DISPLAY LIST library-name object-name dbid fnr
  ```

  Oder:

  Gehen Sie direkt zur Liste der Objekte im BP Cache, indem Sie das folgende SYSBPM-Direktkommando eingeben:

  ```
  DISPLAY CLIST library-name object-name dbid fnr
  ```

Es erscheint ein **List Objects**-Bildschirm ähnlich dem folgenden Beispiel:

```
17:13:17              ***** NATURAL SYSBPM UTILITY *****              2002-09-16
BPNAME QA41GBP                  - List Objects -                  Type Global Nat
BPPROP OFF                                                          Loc DAEF QA41
C  Library  Object    DBID  FNR Loc RLD Use Max  Reuse     TotalUC ObjSize Sto
   *_______ *_______ _____ ____ ___ ___ ___ ___ ______ ___________ _______ ___
__ SYSBPM   BPMCALL     10 1640 B           1   1                  5   8,516  12
__ SYSBPM   BPMNSC      10  410 B               1                  4   3,380   4
__ SYSDLINP PCNDL02    255  253 B   R           1                 19     292   4
__ SYSLIBS  NAT00017    10  410 B               1                  1   5,000   8
__ SYSLIB   ATEST       10 1640 B   R           1                340  16,148  16
__ SYSLIB   CATALL10    10  410 B               1                  1   4,256   8
__ SYSBPM   BPM141-M    10 1640 B               1                  1   5,944   8
__ SYSDLINS U246005    255  253 B   R           1                 14      52   4
__ SYSBPM   MENU        10 1640 B               1                  5  10,392  12
__ SYSLIBS  NAT00040    10  410 B               1                  1   2,816   4
__ SYSLIBS  NAT00034    10  410 B               1                  1   2,672   4
__ SYSDLIND DNDL01     255  253 B   R           3                 42     552   4
__ SYSLIB   ACATALL     10  410 B               1                  3  55,728  56
__ SYSDLINS U246004    255  253 B   R           2                 28     172   4

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

Siehe auch die Funktion **Display Sorted Extract** für eine sortierte Anzeige von Objekten.

Der Bildschirm **List Objects** listet alle Einzelobjekte auf

1. die derzeit im Buffer Pool geladen sind (erster Teil der Anzeige) und
2. die derzeit in den BP Cache geladen sind (zweiter Teil der Anzeige).

Die angezeigten Statistiken sind Momentaufnahmen des Inhalts des Buffer Pool, die jedes Mal, wenn Sie ENTER drücken, aufgefrischt werden.

Hinweis zu im Buffer Pool geladenen GDA- und LDA-Objekten:

Auf dem Bildschirm **List Objects** können zwei Einträge für einen GDA (Global Data Area) oder LDA (Local Data Area) angezeigt werden: ein Eintrag enthält die GDA/LDA-Daten selbst und der andere Eintrag enthält die interne Natural-Symboltabelle für diesen GDA/LDA. Dies kann vorkommen, wenn ein Programm katalogisiert wurde, das auf diesen GDA/LDA verweist.

Bildschirm List Object für Natural-Objekte: Spalten und Auswahlmöglichkeiten

Dieser Abschnitt beschreibt die Spalten des Bildschirms **List Objects** für Natural-Objekte, einschließlich der zugehörigen Eingabefelder und Auswahlkriterien, um die Liste der angezeigten Objekte zu verkürzen.

Die folgenden Themen werden behandelt:

- Auswahlkriterien für die anzuzeigenden Objekte angeben
- Auswahlfelder für Natural-Objekte
- Namen, Datum und Zeit angeben

Auswahlkriterien für die anzuzeigenden Objekte angeben

Um Auswahlkriterien für die anzuzeigenden Objekte anzugeben:

- Geben Sie in die Eingabefelder unter den Spaltenüberschriften einen gültigen Wert oder Bereich ein, wie für die entsprechenden Felder beschrieben.

  Der Standardwert ist ein Leerzeichen oder ein Stern (*), der alle Objekte auswählt.

Auswahlfelder für Natural-Objekte

Spalte	Erläuterung	
C	In dieser Spalte können Sie ein Kommando eingeben, um eine Funktion auf das Objekt auszuführen. Siehe auch ***Zeilenkommandos.***	
Library	Die Library, aus der das Objekt geladen wurde. Zur Angabe von Auswahlkriterien siehe ***Namens- und Bereichsangabe***.	
Object	Der Name des Objekts. Zur Angabe von Auswahlkriterien siehe ***Namens- und Bereichsangabe***.	
DBID	Die Datenbankkennung der Natural-Systemdatei, aus der das Objekt geladen wurde. Um Objekte einer bestimmten Datenbank auszuwählen, geben Sie einen gültigen numerischen Wert ein.	
FNR	Die Dateinummer der Natural-Systemdatei, aus der das Objekt geladen wurde. Um Objekte aus einer bestimmten Datei auszuwählen, geben Sie einen gültigen numerischen Wert ein.	
Loc	Speicherort des Objekts:	
	B	Buffer Pool.
	B/C	Buffer Pool und BP Cache.
	C	BP Cache.
	C/B	BP Cache und Buffer Pool.

Spalte	Erläuterung	
	Wenn B an erster Stelle aufgeführt ist, stammen die statistischen Daten aus dem Buffer Pool. Steht C an erster Stelle, stammen die Daten aus dem BP Cache. Außerdem gelten je nach dieser Positionierung unterschiedliche Zeilenkommandos für die Felder auf dem Statistik-Bildschirm (siehe auch *Zeilenkommandos*). Um die Speicherorte der Objekte anzugeben, geben Sie einen der folgenden Werte ein:	
	B	Wählt alle Objekte aus, die nur im Buffer Pool geladen sind.
	B/C	Wählt alle Objekte aus, die sowohl in den Buffer Pool als auch in den BP Cache geladen sind.
	B*	Wählt alle Objekte aus, die in den Buffer Pool oder sowohl in den Buffer Pool als auch in den BP Cache geladen sind (B* = B + B/C).
	C	Wählt alle Objekte aus, die nur in den BP Cache geladen sind.
	C/B	Wählt alle Objekte aus, die sowohl in den BP Cache als auch in den Buffer Pool geladen sind.
	C*	Wählt alle Objekte aus, die in den BP Cache oder sowohl in den BP Cache als auch in den Buffer Pool geladen sind (C* = C + C/B).
RLD	Aktueller Status des Objekts im Buffer Pool oder im BP Cache. Ein BP Cache-Status bezieht sich nur auf die Objektsperrung und wird daher nur unterhalb des **L** (Locked) der Spalte **RLD** angezeigt. Buffer Pool:	
	R	Markiert als resident. Resident bedeutet, dass das Objekt nicht aus dem Buffer Pool gelöscht wird, auch nicht, wenn sich der entsprechende Wert in der Spalte **Use** auf 0 (Null) ändert, was bedeutet, dass das Objekt nicht mehr verwendet wird.
	L	Gesperrt, während die Ladefunktion ausgeführt wird.
	D	Ein Löschaufruf für das Objekt steht an. Das Objekt wird aus dem Buffer Pool gelöscht, sobald sich der Wert in der Spalte **Use** auf 0 (Null) ändert.
	BP Cache:	
	L	Gesperrt, während die Ladefunktion ausgeführt wird.
	D	Gesperrt zum Löschen.
	Um alle Objekte mit einem bestimmten Status, wie oben beschrieben, auszuwählen, geben Sie den Code R, L oder D ein.	

Spalte	Erläuterung	
Use	Nur für Buffer Pool. Die Anzahl der Natural-Anwendungen, die das Objekt zurzeit ausführen. Um Objekte auszuwählen, können Sie eine der folgenden Angaben machen:	
	value	Ein numerischer Wert. Wählt alle Objekte mit dieser Nummer aus. Beispiel: `10` Ausgewählt: 10
	>*value*	Ein numerischer Startwert (>). Wählt alle Objekte aus, deren Zahl größer als *value* ist. Beispiel: `>10` Ausgewählt: 11, 21 Nicht ausgewählt: 10
	<*value*	Ein numerischer Endwert (<). Wählt alle Objekte mit einer Zahl kleiner als *value* aus. Beispiel: `<10` Ausgewählt: 9, 8 Nicht ausgewählt: 10
Max	Nur für Buffer Pool. Die maximale Anzahl der Anwendungen, die das Objekt ausgeführt haben, seit es in den Buffer Pool geladen wurde. Zur Auswahl von Objekten siehe die gültigen Eingabewerte unter **Use** (Verwendung) oben.	
Reuse	Nur für Buffer Pool Cache. Gibt an, wie oft das Objekt aus dem BP Cache in den Buffer Pool geladen (wiederverwendet) wurde. Zur Auswahl von Objekten siehe die gültigen Eingabewerte unter **Use** (Verwendung) oben.	
TotalUC	Total Use Count: Die Gesamtzahl der Locate-Aufrufe des Objekts, seit es in den Buffer Pool geladen wurde. **Anmerkung:** Dieser Wert erhöht sich nicht nur, wenn ein Natural-Objekt (z. B. ein Programm, eine Map (Maske) oder ein Subprogramm über `CALLNAT`) zum ersten Mal aufgerufen wird, sondern auch, wenn Natural das Objekt nach Rückkehr von E/A-Operationen oder von außerhalb erfolgten Aufrufen im Buffer Pool neu lokalisieren muss. Der Grund dafür ist, dass das Objekt während dieser Zeit möglicherweise aus dem Buffer Pool entfernt wurde, sodass eine neue Suche erforderlich ist. Daher spiegelt **TotalUC** die Häufigkeit wider, mit der auf	

<table>
<tr><th>Spalte</th><th colspan="2">Erläuterung</th></tr>
<tr><td></td><td colspan="2">den Buffer Pool zugegriffen wird, und nicht nur die Anzahl der eindeutigen oder ersten Aufrufe.

Wenn ein BP Cache verwendet wird, geht dieser Wert nicht verloren, wenn das Objekt aus dem Buffer Pool entfernt und im BP Cache gespeichert wird. Daher gibt dieser Wert an, wie oft das Objekt verwendet wurde, seit es aus der Systemdatei geladen wurde.

Für Buffer Pool-Objekte wird dieser Wert regelmäßig aktualisiert. Bei BP Cache-Objekten wird dieser Wert nur aktualisiert, nachdem das Objekt aus dem Buffer Pool entfernt und im BP Cache gespeichert wurde.

Um Objekte auszuwählen, können Sie eine der folgenden Angaben machen:</td></tr>
<tr><td></td><td>value oder
>value</td><td>Ein numerischer Wert oder ein numerischer Startwert (>). Wählt alle Objekte aus, deren Zahl größer oder gleich value. ist.

Beispiel: >10

Ausgewählt: 10, 11, 21

Nicht ausgewählt: 9</td></tr>
<tr><td></td><td><<i>value</i></td><td>Ein numerischer Endwert (<). Wählt alle Objekte aus, deren Zahl kleiner als <i>value</i> oder gleich <i>value</i> ist.

Beispiel: <10

Ausgewählt: 10, 9, 8

Nicht ausgewählt: 11</td></tr>
<tr><td>ObjSize</td><td colspan="2">Die Größe des Objekts.

Zur Auswahl von Objekten siehe die gültigen Eingabewerte in TotalUC weiter oben.</td></tr>
<tr><td>Sto</td><td colspan="2">Speicherplatz, der für das Objekt im Buffer Pool oder BP Cache zugeordnet werden muss. Die Textdatensatzgröße des Buffer Pool wird bei der Initialisierung des Buffer Pool festgelegt.

Zur Auswahl von Objekten siehe die gültigen Eingabewerte in Use weiter oben.</td></tr>
<tr><td>BP Load Time*</td><td colspan="2">Das Datum und die Uhrzeit, wann das Objekt zum ersten Mal in den Buffer Pool geladen wurde.

Dieses Datum und diese Uhrzeit werden beibehalten, bis das Objekt sowohl aus dem Buffer Pool als auch aus dem BP Cache entfernt wird (wenn es nur aus dem BP Cache gelöscht wird, werden Datum und Uhrzeit nicht angezeigt).

Zur Auswahl von Objekten siehe Datumsangabe und Zeitangabe.</td></tr>
<tr><td>BP Last Action*</td><td colspan="2">Nur für Buffer Pool.

Das Datum und die Uhrzeit, zu der das Objekt zuletzt von einer Anwendung verwendet wurde.</td></tr>
</table>

Spalte	Erläuterung
	Zur Auswahl von Objekten siehe *Datumsangabe* und *Zeitangabe*.
BPC Load Time*	Nur für BP Cache (BPC). Das Datum und die Uhrzeit, wann das Objekt zum ersten Mal in den BP Cache geladen wurde. Dieses Datum und diese Uhrzeit werden beibehalten, bis das Objekt aus dem BP Cache entfernt wird. Zur Auswahl von Objekten siehe *Datumsangabe* und *Zeitangabe*.
BPC Last Get*	Nur für BP Cache (BPC). Datum und Uhrzeit, wann das Objekt zuletzt aus dem Buffer Pool in den BP Cache ausgelagert wurde. Dieser Zeitstempel wird auch aktualisiert, wenn das Objekt bereits im BP Cache vorhanden war und daher nicht erneut in den BP Cache geschrieben wurde. Zur Auswahl von Objekten siehe *Datumsangabe* und *Zeitangabe*.
BPC Last Put*	Nur für BP Cache (BPC). Datum und Uhrzeit, wann das Objekt zuletzt aus dem BP Cache in den Buffer Pool geladen wurde. Zur Auswahl von Objekten siehe *Datumsangabe* und *Zeitangabe*.
1.BPperiod*	Nur für BP Cache (BPC). Der Zeitraum, in dem das Objekt im Buffer Pool verfügbar war, beginnend mit dem Zeitpunkt, zu dem das Objekt zum ersten Mal geladen wurde, und endend mit dem Zeitpunkt, zu dem das Objekt zum ersten Mal aus dem Buffer Pool in den BP Cache ausgelagert wurde. Nach 24 Stunden wird die Anzeige des Zeitrahmens abgebrochen und durch dieses Zeichen ersetzt: **:**:**.***:**:**.*
NatVers*	Die Nummer der Natural-Version, mit der ein Objekt katalogisiert ist.

* Damit diese Spalten angezeigt werden, müssen Sie PF11 drücken. Siehe auch *PF-Tasten und Direktkommandos*.

Namen, Datum und Zeit angeben

Dieser Abschnitt enthält Regeln für die Angabe von Auswahlkriterien in Form von Bereichen für die Spalten **Library** und **Object**:

- Namens- und Bereichsangabe
- Datumsangabe

- Zeitangabe

Namens- und Bereichsangabe

Sie können die Liste der auf dem Bildschirm **List Objects** angezeigten Objekte verkürzen, indem Sie einen Namen oder einen Bereich von Namen in die Eingabefelder für **Library** und/oder **Object** eingeben.

In der nachstehenden Liste der Optionen ist *value* (Wert) eine beliebige Kombination aus einem oder mehreren Zeichen:

Eingabewert	Ausgewählte Libraries/Objekte
*	Alle Libraries/Objekte. Dies ist die Standardeinstellung.
value	Alle Libraries/Objekte mit einem Namen, der gleich *value* ist.
*value**	Alle Libraries/Objekte mit einem Namen, der mit *value* beginnt. Beispiel: AB* Ausgewählt: AB, AB1, ABC, ABEZ Nicht ausgewählt: AA1, ACB
value?	Alle Libraries/Objekte mit einem Namen, der mit *value* beginnt und mit einem beliebigen Zeichen für jedes eingegebene Fragezeichen (?) endet. Beispiel: ABC? Ausgewählt: ABCA, ABCZ Nicht ausgewählt: AXC, ABCAA
value?*value*? *value**value*? **value*?*value**	Alle Elemente, die mit *value* übereinstimmen, kombiniert mit Stern (*) und Fragezeichen (?) in beliebiger Reihenfolge. Beispiel: A?C*Z Ausgewählt: ABCZ, AXCBBBZ, ANCZ Nicht ausgewählt: ACBZ, ABDEZ, AXCBBBZA
value>	Alle Libraries/Objekte mit einem Namen, der größer oder gleich *value* ist. Beispiel: AB> Ausgewählt: AB, AB1, BBB, ZZZZZZZ Nicht ausgewählt: AA1, AAB
value<	Alle Libraries/Objekte mit einem Namen, der kleiner oder gleich *value* ist. Beispiel: AX< Ausgewählt: AB, AWW, AX Nicht ausgewählt: AXA, AY

Datumsangabe

Sie können die Liste der auf dem Bildschirm **List Objects** angezeigten Objekte verkürzen, indem Sie ein Datum, einen Datumsbereich, ein spezielles Datum oder einen Bereich spezieller Datumswerte in die Eingabefelder für Datumsangaben eingeben.

Ein Datum muss im Format *YYYYMMDD* (*YYYY* = Jahr, *MM* = Monat, *DD* = Tag) angegeben werden.

In der nachstehenden Liste der Optionen ist *value* (Wert) eine beliebige Kombination aus einer oder mehreren Ziffern:

Eingabewert	Ausgewählte Objekte
YYYYMMDD	Alle Objekte mit einem Datum gleich *YYYYMMDD*. Beispiel: 20070630
*value**	Alle Objekte mit einem Datum, das mit *value* beginnt. Beispiel: 2007* Ausgewählt: 20070101 to 20071231 Nicht ausgewählt: 20061231, 20080101
value>	Alle Objekte mit einem Datum, das größer oder gleich *value* ist. Beispiel: 2007> Ausgewählt: 20070101 to 20070101 Nicht ausgewählt: 20061231
value<	Alle Objekte mit einem Datum kleiner als *value*. Beispiel: 2007< Ausgewählt: 20060101 to 20061231 Nicht ausgewählt: 20070101, 20071231
Besondere Datumswerte	
TODAY oder TO+/-*n*	Alle Objekte mit dem Datum des aktuellen Tages oder eines Tages vor oder nach dem aktuellen Tag:
TODAY	Alle Objekte mit dem Datum des aktuellen Tages.
TO+/-*n*	Alle Objekte mit dem Datum des aktuellen Tages plus oder minus *n* Tage. Beispiel: Das aktuelle Datum ist der 30. Juni 2007. TO-5 wählt 20070625 aus.
YESTERDAY	Alle Objekte mit dem Datum des Vortages des aktuellen Tages. Beispiel: Das aktuelle Datum ist der 30. Juni 2007. YESTERDAY wählt 20070625 aus.

Eingabewert	Ausgewählte Objekte
MONTH	Alle Objekte mit dem Datumsbereich des aktuellen Monats. Beispiel: Der aktuelle Monat ist der Juni des Jahres 2007. MONTH wählt 20070601 bis 20070630 aus.
YEAR	Alle Objekte mit dem Datumsbereich des aktuellen Jahres. Beispiel: Das aktuelle Jahr ist 2007. YEAR wählt 20070101 bis 20071231 aus

Zeitangabe

Sie können die Liste der auf dem Bildschirm **List Objects** angezeigten Objekte verkürzen, indem Sie eine Uhrzeit oder einen Bereich von Uhrzeiten in die Eingabefelder für Uhrzeiten eingeben.

Die Uhrzeit muss im folgenden Format angegeben werden: *HH*:*II*:*SS* (*HH* = Stunden, *II* = Minuten, *SS* = Sekunden).

In der nachstehenden Liste der Optionen kann *value* (Wert) eine beliebige Kombination aus einer oder mehreren Ziffern sein:

Eingabewert	Ausgewählte Objekte
HH:*II*:*SS*	Alle Objekte mit einer Zeit gleich *HH*:*II*:*SS*. Beispiel: 14:15:16
*value**	Alle Objekte mit einer Zeit, die mit *value*. beginnt. Beispiel: 13* Ausgewählt: 13:00:00, 13:10:53, 13:59:59 Nicht ausgewählt: 12:59:59, 14:00:00
value>	Alle Objekte mit einer Zeit, die größer oder gleich *value* ist. Beispiel: 12:30> Ausgewählt: 12:30:00, 12:30:01, 16:34:01 Nicht ausgewählt: 12:29:59
value<	Alle Objekte mit einer Zeit, die kleiner als *value* ist. Beispiel: 12:30< Ausgewählt: 12:29:59 Nicht ausgewählt: 12:30:00

PF-Tasten und Direktkommandos für die Liste der Natural-Objekte

Auf dem Bildschirm **List Objects** können Sie die in der nachstehenden Tabelle aufgeführten PF-Tasten oder SYSBPM-Direktkommandos verwenden. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos finden Sie unter *SYSBPM-Direktkommandos*.

PF-Taste	Kommando	Funktion
PF1		Liefert SYSBPM-Hilfeinformationen. Wird diese Funktion gewählt, wenn der Cursor auf der Spalte **C** steht, werden alle verfügbaren Kommandos und Funktionen aufgelistet. Wenn der Cursor auf den Eingabefeldern unter den Spaltenüberschriften steht, werden alle zulässigen Eingabewerte für die Objektauswahl aufgelistet.
PF3	EXIT	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF4	LAST	Anzeige des zuletzt eingegebenen SYSBPM-Direktkommandos.
PF5	CACHE	Nur anwendbar, wenn BP Cache-Daten vorhanden sind. Blättern an den Anfang der Liste mit statistischen Daten zu BP Cache-Objekten.
PF6	-	Blättern an den Anfang der Liste mit statistischen Daten zu Buffer Pool-Objekten.
PF7	-	In der Liste um eine Seite nach oben blättern.
PF8	+	In der Liste um eine Seite nach unten blättern.
PF9	++	An das Ende der Liste blättern.
PF10	<	In der Liste nach links blättern. Drücken Sie PF11, um nach rechts zu blättern.
	LEFT	
PF11	>	Blättern in der Liste nach rechts und Anzeige der zusätzlichen Bildschirmspalten: **BP Load Time**, **BP Last Action**, **BPC Load Time**, **BPC Last Get**, **BPC Last Put** und **1.BPperiod**. Drücken Sie PF10, um nach links zu blättern.
	RIGHT	
PF12	CANCEL	Wie EXIT.
PF15	MENU	Rückkehr zum SYSBPM-Hauptmenü (**Main Menu**).

Zeilenkommandos für die Liste der Natural-Objekte

Auf dem Bildschirm **List Objects** können Sie in Spalte **C** für jedes angezeigte Objekt eines der unten aufgeführten Zeilenkommandos eingeben:

Kommando	Funktion
CL	Nur Buffer Pool. Gibt ein als resident gekennzeichnetes Objekt frei.
DE	Markiert ein Objekt, das aus dem Buffer Pool oder BP Cache gelöscht werden soll. Das Objekt wird gelöscht, sobald der entsprechende **Use Count** auf 0 wechselt (Objekt wird nicht mehr verwendet). Bei einem Buffer Pool-Objekt wird das Objekt sowohl aus dem Buffer Pool als auch aus dem BP Cache gelöscht. Wenn es für ein BP Cache-Objekt abgesetzt wird, wird das Objekt nur aus dem BP Cache gelöscht.
HD	Nur Buffer Pool. Zeigt die Verzeichnisinformationen eines Objekts im hexadezimalen Format an.
HE	Nur Buffer Pool. Entspricht der Funktion **Hexadecimal Display**, siehe entsprechenden Abschnitt.
FO	Nur Buffer Pool. Löscht ein Objekt sofort aus dem Buffer Pool, unabhängig vom relevanten **Use Count**.
LD	Entspricht der Funktion **Directory Information**, siehe entsprechenden Abschnitt.
RE	Nur Buffer Pool. Markiert ein Objekt als resident.
ZO	Vergrößert die Anzeige der Felder **Object**, **Use**, **Max**, **Reuse**, **TotalUC**, **ObjSize** und **Sto** und zeigt sie in voller Länge an. Zum Verkleinern drücken Sie ENTER.

Für jedes eingegebene Kommando erscheint für die entsprechende Zeile eine Bestätigungsmeldung, die den Text der auf dem Bildschirm angezeigten Zeilen überschreibt. Mögliche Meldungen sind:

- Failed - Fehlgeschlagen (als Antwort auf eine Funktion, die nicht erfolgreich ausgeführt wurde),
- Deleted - Gelöscht (als Antwort auf das Kommando DE oder FO),
- Released - Freigegeben (als Antwort auf das Kommando CL) und
- Resident (als Antwort auf das Kommando RE).

Meldungen auflisten

Dieser Abschnitt behandelt die folgenden Themen:

- Bildschirm List Objects für System- und Benutzermeldungen aufrufen
- List Objects-Bildschirm für Meldungen: Spalten und Auswahlmöglichkeiten
- PF-Tasten und Direktkommandos für die Liste der Meldungen
- Zeilenkommandos für die Liste der Meldungen

Bildschirm List Objects für System- und Benutzermeldungen aufrufen

Um den Bildschirm List Objects für System- und Benutzermeldungen aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `L` in das Feld **Code..** und das Auswahlkriterium `M` in das Auswahlfeld **Object Pool** ein und geben Sie die Meldungen an. **Gültige Feldeingabewerte** siehe Abschnitt ***SYSBPM aufrufen und benutzen***.

  Oder:

  Gehen Sie direkt zur Liste der Objekte im Buffer Pool, indem Sie das folgende SYSBPM-Direktkommando eingeben:

  ```
  DISPLAY MLIST start-number end-number library-name language dbid fnr codepage
  ```

Es erscheint ein **List Objects**-Bildschirm ähnlich dem folgenden Beispiel:

```
 17:13:17             ***** NATURAL SYSBPM UTILITY *****             2002-09-16
 BPNAME QA41GBP                - List Objects -                 Type Global Nat
 BPPROP OFF                                                       Loc DAEF QA41
  C  Library  Nmbr  DBID  FNR Code Page LC Message Text
     *_______ ____ _____ ____ *_______  __
  __ XXXLIB     25   177    8 CPAGE02    2 U0025 TEST TEST TEST
  __ <NATSYS>   80    10 2430 IBM01141   1 Command / program name must start wi
  __ <NATSYS>   82    10 2430 IBM01141   1 Invalid command, or :1: :2: does not
  __ XXXLIB    102   177    8 CPAGE02    2 U0102 TEST TEST TEST
  __ GGSLIB    296   177    7 CPAGE22   34 U0296 TEST TEST TEST
  __ <NATSYS>  631    10 2430 IBM01141   1 Invalid index specified in arithmeti
  __ <NATSYS>  660    10 2430 IBM01141   1 Time-stamp inconsistency for segment
  __ XXXLIB    759   177    8 CPAGE02    2 U0759 TEST TEST TEST
  __ XXXLIB    789   177    8 CPAGE02    2 U0789 TEST TEST TEST
  __ XXXLIB    806   177    8 CPAGE02    2 U0806 TEST TEST TEST
  __ XXXLIB    853   177    8 CPAGE02    2 U0853 TEST TEST TEST
  __ <NATSYS>  932    10 2430 IBM01141   1 Program version error.
  __ <NATSYS>  933    10 2430 IBM01141   1 GDA time-stamp conflict.
  __ <NATSYS>  933    10 2430 IBM01140   1 GDA time-stamp conflict.
Top of List
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit              --    -     +     ++               Canc   ↩
```

Auf dem Bildschirm **List Objects** werden alle Einzelmeldungen aufgelistet, die derzeit in den Message Pool geladen sind. Der Bildschirm wird jedes Mal aktualisiert, wenn Sie ENTER drücken.

List Objects-Bildschirm für Meldungen: Spalten und Auswahlmöglichkeiten

Dieser Abschnitt beschreibt die Spalten des Bildschirms **List Objects** für Meldungen, einschließlich der zugehörigen Eingabefelder und Auswahlkriterien zur Verkürzung der Liste der angezeigten Objekte.

- Angabe von Auswahlkriterien für die anzuzeigenden Objekte
- Auswahlfelder für Meldungen
- Namens- und Bereichsangabe

Angabe von Auswahlkriterien für die anzuzeigenden Objekte

Um Auswahlkriterien für die anzuzeigenden Objekte anzugeben:

- Geben Sie in den Eingabefeldern unter den Spaltenüberschriften einen gültigen Wert oder Bereich ein, wie bei den entsprechenden Feldern beschrieben.

  Der Standardwert ist ein Leerzeichen oder ein Stern (*), wodurch alle Objekte ausgewählt werden.

Auswahlfelder für Meldungen

Dieser Abschnitt enthält eine tabellarische Übersicht über die Felder des Bildschirms **List Object**.

Spalte	Erläuterung
C	In dieser Spalte können Sie ein Zeilenkommando eingeben, um eine Funktion für die entsprechende Meldung auszuführen. Siehe auch *Zeilenkommandos*.
Library	Die Library, aus der die Meldung geladen wurde. Zur Angabe von Auswahlkriterien siehe *Namens- und Bereichsangabe*.
Nmbr	Die Nummer der Meldung. Um Meldungen mit einer bestimmten Nummer auszuwählen, geben Sie einen gültigen numerischen Wert ein.
DBID	Die Datenbankkennung der Natural-Systemdatei, aus der die Meldung geladen wurde. Um eine Meldung einer bestimmten Datenbank auszuwählen, geben Sie einen gültigen numerischen Wert ein.
FNR	Die Dateinummer der Natural-Systemdatei, aus der die Meldung geladen wurde. Um Meldungen einer bestimmten Datei auszuwählen, geben Sie einen gültigen numerischen Wert ein.
Code Page	Die Codepage, unter der die Meldung gespeichert wurde.
LC	Der Sprachcode, unter dem die Nachricht gespeichert wurde.
Message Text	Der Anfang des Kurztextes einer Meldung.

* Damit diese Spalten angezeigt werden, müssen Sie PF11 drücken. Siehe auch *PF-Tasten und Direktkommandos*.

Namens- und Bereichsangabe

Siehe den Abschnitt *Namens- und Bereichsangabe* im Kapitel *Natural-Objekte auflisten*.

PF-Tasten und Direktkommandos für die Liste der Meldungen

Auf dem Bildschirm **List Objects** können Sie die in der nachstehenden Tabelle aufgeführten PF-Tasten oder SYSBPM-Direktkommandos verwenden. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos siehe *SYSBPM-Direktkommandos*.

PF-Taste	Kommando	Funktion
PF1		Liefert SYSBPM-Hilfeinformationen. Wird diese Funktion gewählt, wenn der Cursor auf der Spalte **C** steht, werden alle verfügbaren Kommandos und Funktionen aufgelistet. Wenn der Cursor auf den Eingabefeldern unter den Spaltenüberschriften steht, werden alle zulässigen Eingabewerte für die Objektauswahl aufgelistet.
PF3	EXIT	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF6	-	An den Anfang der Liste der Meldungen im Message Pool blättern.
PF7	-	In der Liste um eine Seite nach oben blättern.
PF8	+	In der Liste um eine Seite nach unten blättern.
PF9	++	An das Ende der Liste blättern.
PF12	CANCEL	Wie EXIT.
PF15	MENU	Rückkehr zum SYSBPM-Hauptmenü (**Main Menu**).

Zeilenkommandos für die Liste der Meldungen

Auf dem Bildschirm **List Objects** können Sie in Spalte **C** für jedes angezeigte Objekt eines der unten aufgeführten Zeilenkommandos eingeben:

Kommando	Funktion
DE	Löscht eine Meldung aus dem Message Pool. Der Eintrag wird durch die Bestätigung „deleted“ (gelöscht) ersetzt.

25 Objekte löschen

- Objekte aus dem Buffer Pool und/oder dem BP Cache löschen 198
- Objekte aus dem Message Pool löschen 198

Mit dieser Funktion können Sie ein oder mehrere Objekte aus dem Buffer Pool (BP) und/oder dem BP Cache löschen. Sie können die Funktion auch auf den Message Pool anwenden, um die zu löschenden Meldungen auszuwählen.

Objekte aus dem Buffer Pool und/oder dem BP Cache löschen

Ein Objekt, das einen **Current Use Count** (siehe *Verzeichnisinformationen anzeigen*) von `0` (Null) hat, wird sofort gelöscht. `0` bedeutet, dass ein solches Objekt nicht mehr verwendet wird. Ein Objekt mit einem **Current Use Count** größer als `0` wird zum Löschen markiert und gelöscht, sobald sein **Current Use Count** auf `0` wechselt.

Um Natural-Objekte zu löschen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `D` ein, den Code für den Typ des Objekt-Pools (`B`, `C` oder `*`), und geben Sie das oder die zu löschenden Objekte an. **Gültige Feldeingabewerte** siehe unter *SYSBPM aufrufen und benutzen*.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DELETE library-name object-name dbid fnr
  ```

Objekte aus dem Message Pool löschen

Um Meldungen zu löschen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `D`, den Code für den Message Pool (`M`) und das oder die zu löschenden Meldungsobjekte ein. **Gültige Feldeingabewerte** siehe unter *SYSBPM aufrufen und benutzen*.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DELETE MP start-number end-number library-name language dbid fnr codepage
  ```

26 Verzeichnisinformationen anzeigen

- Felder für Buffer-Pool-Objekte 200
- Felder für BP Cache-Objekte 202
- PF-Tasten und Direktkommandos 203

Mit der Funktion **Directory Information** können Sie das vollständige Verzeichnis eines Natural-Objekts anzeigen, das zurzeit in den Buffer Pool oder den BP Cache geladen ist.

Um Verzeichnisinformationen aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) Folgendes ein:

  Geben Sie den Funktionscode `I` ein.

  Geben Sie in den Feldern **Object**, **Library**, **DBID** und **FNR** das Objekt an, für das das Verzeichnis angezeigt werden soll. **Gültige Eingabewerte** sind unter *SYSBPM aufrufen und benutzen* aufgeführt.

  Geben Sie im Feld **Object Pool** den Code ein, der dem Objekt-Pool entspricht, aus dem die Objektverzeichnisinformationen gelesen werden sollen:

`B` oder * (Stern)	Buffer Pool
`C`	BP Cache

  Oder:

  Verwenden Sie eines der folgenden SYSBPM-Direktkommandos:

  - `DISPLAY DIRECTORY library-name object-name dbid fnr`

    (Für das Verzeichnis des angegebenen Objekts, das in den Buffer Pool geladen wurde.)
  - `DISPLAY CDIRECTORY library-name object-name dbid fnr`

    (Für das Verzeichnis des angegebenen Objekts, das in den BP Cache geladen wurde.)

  Der Bildschirm **Directory Information** erscheint. Je nach ausgewähltem Objekt-Pool (Buffer Pool oder BP Cache) enthält der Bildschirm unterschiedliche Felder, wie im folgenden Abschnitt beschrieben.

Felder für Buffer-Pool-Objekte

Zu Verzeichnissen von Objekten, die in den Buffer Pool geladen wurden, bietet der Bildschirm **Directory Information** die folgenden Felder und Informationen zu einem bestimmten Objekt:

Feld		Erläuterung
Directory of		Typ, z.B. Map (Maske) und Name des Objekts.
Loaded	**from Library**	Der Name der Library, aus der das Objekt in den Buffer Pool geladen wurde.
	on DBID/FNR	Die Datenbankkennung (DBID) und Dateinummer (FNR) der Systemdatei FNAT oder FUSER, aus der das Objekt in den Buffer Pool geladen wurde.
	on	Das Datum und die Uhrzeit, wann das Objekt in den Buffer Pool geladen wurde.
	by User	Die Kennung des Benutzers, der das Objekt ausgeführt hat.
Last Action on		Das Datum und die Uhrzeit, wann das Objekt zuletzt von einer Anwendung verwendet wurde.
BP Directory at Address		Die Adresse des Verzeichnisses des Objekts im Buffer Pool.
Object at Address		Die Adresse des Verzeichnisses des Objekts im Buffer Pool.
Allocated Size (KB)		Die Größe, die im Buffer Pool für das Objekt zugeordnet wurde.
Object Size		Die Größe des Objekts.
Status (RLD)		Der Status des Objekts: `R` – Das Objekt ist im Buffer Pool resident. Resident bedeutet, dass das Objekt nicht aus dem Buffer Pool gelöscht wird, auch dann nicht, wenn sich sein **Current Use Count** (siehe unten) auf `0` (Null) ändert. `L` – Das Objekt ist zurzeit gesperrt. `D` – Ein Löschaufruf für das Objekt steht an. Das Objekt wird aus dem Buffer Pool gelöscht, sobald sein **Current Use Count** (siehe unten) auf `0` (Null) sinkt.
Current Use Count		Die Anzahl der Anwendungen, die das Objekt derzeit ausführen. Ein Wert von `0` (Null) bedeutet, dass das Objekt nicht mehr verwendet wird.
Maximum Use Count		Die maximale Anzahl der Anwendungen, die das Objekt ausgeführt haben, seit es in den Buffer Pool geladen wurde.
BP Total Use		Die Gesamtzahl der Ausführungen eines Objekts, seit es aus der Systemdatei in den Buffer Pool geladen wurde. Wenn ein BP Cache verwendet wird, geht dieser Wert nicht verloren, wenn das Objekt aus dem Buffer Pool entfernt und im BP Cache gespeichert wird. Daher gibt dieser Wert an, wie oft das Objekt verwendet wurde, seit es aus der Systemdatei geladen wurde.
Cataloged		Die im Abschnitt **Cataloged** des Bildschirms **Directory Information** angezeigten Informationen sind identisch mit den Informationen, die mit dem in der *Systemkommandos*-Dokumentation beschriebenen Natural-Systemkommando `LIST DIRECTORY` bereitgestellt werden.

Felder für BP Cache-Objekte

Für Verzeichnisse von Objekten, die in den BP Cache geladen wurden, bietet der Bildschirm **Directory Information** die folgenden Felder und Informationen zu einem bestimmten Objekt:

Feld	Erläuterung
object-type	Der Typ, z.B. Map (Maske), und der Name des Objekts.
Library	Der Name der Library, aus der das Objekt in den Buffer Pool geladen wurde.
DBID	Die Datenbankkennung der Systemdatei FNAT oder FUSER, aus der das Objekt ursprünglich in den Buffer Pool geladen wurde.
FNR	Die Dateinummer der Systemdatei FNAT oder FUSER, aus der das Objekt ursprünglich in den Buffer Pool geladen wurde.
Last Put	Das Datum und die Uhrzeit, zu der das Objekt zuletzt aus dem BP Cache in den Buffer Pool geladen wurde.
BP Load Time	Datum und Uhrzeit, wann das Objekt zum ersten Mal in den Buffer Pool geladen wurde.
Cache Load Time	Datum und Uhrzeit, wann das Objekt zum ersten Mal in den BP Cache geladen wurde.
Last Get	Datum und Uhrzeit, wann das Objekt zuletzt aus dem Buffer Pool in den BP Cache ausgelagert wurde.
Position Index	Fortlaufend nummerierter interner Natural-Positionsindex der Objekte im BP Cache.
First Data Block Offset	Die Adresse des Verzeichnisses des Objekts im BP Cache.
Allocated Size (KB)	Die Größe, die im BP Cache für das Objekt zugeordnet wurde.
Object Size	Die Größe des Objekts.
Status	Der Status des Objekts: (leer): (leer) L: Gesperrt, während die Ladefunktion ausgeführt wird. (leer): (leer) D: Gesperrt zum Löschen.
Reuse	Gibt an, wie oft das Objekt aus dem BP Cache an den Buffer Pool zurückgegeben wurde.
BP Total Use	Die Gesamtzahl der Ausführungen eines Objekts, seit es zuerst aus der Systemdatei in den Buffer Pool und dann in den BP Cache geladen wurde.
Cataloged	Die im Abschnitt **Cataloged** des Bildschirms **Directory Information** angezeigten Informationen sind identisch mit den Informationen, die mit dem in der *Systemkommandos*-Dokumentation beschriebenen Natural-Systemkommando `LIST DIRECTORY` bereitgestellt werden.

PF-Tasten und Direktkommandos

Auf dem Bildschirm **Directory Information** können Sie die PF-Tasten oder SYSBPM-Direktkommandos verwenden, die in der folgenden Tabelle aufgeführt sind. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos finden Sie unter *SYSBPM-Direktkommandos*.

PF-Taste	Kommando	Funktion
PF1		Liefert Hilfeinformationen zu SYSBPM-Direktkommandos.
PF2	NEXT	Gilt nur, wenn ein Bereich von Objekten ausgewählt wurde. Zeigt ein Objekt nach dem anderen an und zeigt dann wieder den Bildschirm an, auf dem NEXT eingegeben wurde.
PF3	EXIT	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF4	LAST	Anzeige des zuletzt eingegebenen SYSBPM-Direktkommandos.
PF6	FLIP	Schaltet die PF-Tastenzeile um: Wechsel zwischen der Anzeige von PF1 bis PF12 und PF13 bis PF24.
PF12	CANCEL	Wie EXIT.
PF15	MENU	Rückkehr zum SYSBPM-Hauptmenü (**Main Menu**).
	FDELETE	Nur Buffer Pool. Löscht ein Objekt sofort aus dem Buffer Pool, unabhängig vom relevanten **Use Count**.
	RESIDENT	Nur Buffer Pool. Markiert ein Objekt als resident. Resident bedeutet, dass das Objekt nicht aus dem Buffer Pool gelöscht wird, auch dann nicht, wenn sein **Use Count** 0 ist (Objekt wird nicht mehr verwendet).
	CLEAR	Nur Buffer Pool. Gibt ein als resident gekennzeichnetes Objekt frei.
	DELETE	Markiert ein Objekt zur Löschung. Siehe Status D des **Buffer Pools** oder **BP Cache**, der im entsprechenden früheren Abschnitt erwähnt wurde.

27 Hexadezimale Anzeige

- PF-Tasten und Direktkommandos 206

Mit der Funktion **Hexadecimal Display** wird der Code eines Natural-Objekts, das gerade in den Buffer Pool geladen ist, in hexadezimaler Form angezeigt.

Um die hexadezimale Anzeige aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `H` ein und geben Sie ein Objekt an. **Gültige Feld-Eingabewerte** siehe Abschnitt *SYSBPM aufrufen und benutzen*.

  Oder:

  Geben Sie das folgende SYSPBM-Direktkommando ein:

  ```
  DISPLAY HEX library-name object-name dbid fnr
  ```

  Es erscheint der Bildschirm **Hexadecimal Display** mit der Anzeige des Objektcodes im hexadezimalen Format.

PF-Tasten und Direktkommandos

Innerhalb des auf dem Bildschirm angezeigten Objekts können Sie an eine bestimmte Stelle springen, indem Sie entweder eine absolute hexadezimale Adresse oder einen hexadezimalen Offset relativ zu Ihrer aktuellen Position eingeben.

Im Bildschirm **Hexadecimal Display** können Sie die in der folgenden Tabelle aufgeführten PF-Tasten oder SYSBPM-Direktkommandos verwenden. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos siehe *SYSBPM-Direktkommandos*.

PF-Taste	Kommando	Funktion
PF1		Liefert Hilfeinformationen zu den SYSBPM-Direktkommandos.
PF2	`NEXT`	Gilt nur, wenn ein Bereich von Objekten ausgewählt wurde. Zeigt ein Objekt nach dem anderen an und zeigt dann wieder den Bildschirm an, auf dem `NEXT` eingegeben wurde.
PF3	`EXIT`	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF4	`LAST`	Anzeige des zuletzt eingegebenen SYSBPM-Direktkommandos.
PF6	`-`	An den Anfang der Anzeige blättern
PF7	`-`	Eine Seite nach oben blättern.
PF8	`+`	Eine Seite nach unten blättern.
PF9	`++`	An das Ende der Anzeige blättern
PF12	`CANCEL`	Wie `EXIT`.
PF15	`MENU`	Rückkehr zum SYSBPM-Hauptmenü (**Main Menu**).

PF-Taste	Kommando	Funktion
	NEXT	Gilt nur, wenn ein Bereich von Objekten ausgewählt wurde. Zeigt ein Objekt nach dem anderen an und zeigt dann wieder den Bildschirm an, auf dem NEXT eingegeben wurde.

28 In Arbeitsdatei schreiben

Die Funktion **Write to Work File** schreibt die Verzeichnisinformationen der aktuell in den Buffer Pool und/oder den BP Cache geladenen Natural-Objekte in eine Arbeitsdatei.

Um die Funktion aufzurufen:

1 Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) Folgendes ein:

- Geben Sie im Feld **Code** den Funktionscode `W` ein.
- Geben Sie in das Feld **Object Pool** den Code ein, der dem Objekt-Pool entspricht, aus dem die Objektverzeichnisinformationen gelesen werden sollen:

`B`	Buffer Pool.
`C`	BP Cache.
* (Stern)	Sowohl Buffer Pool als auch BP Cache.

Oder:

Geben Sie eines der folgenden SYSBPM-Direktkommandos ein:

- `WRITE ALL`

  (speichert Buffer Pool- und BP Cache-Daten)
- `WRITE BP`

  (speichert nur Buffer Pool-Daten)
- `WRITE BPC`

  (speichert nur Buffer-Pool-Cache-Daten)

Das Fenster **Work File Selection** erscheint.

2 Geben Sie Folgendes an:

- Wählen Sie die Ziel-Arbeitsdatei:

  Geben Sie `N` (Nein, dies ist die Standardeinstellung) ein, um die Daten in die Natural-Arbeitsdatei 1 auszugeben.

  Oder: Wenn Entire Connection installiert ist, geben Sie Y (Ja) ein, um die Daten mit Hilfe von Natural-Arbeitsdatei 7 in eine PC-Textdatei auszugeben.
- Geben Sie das Trennzeichen ein, z. B. ein Semikolon (;), das zur Trennung der Spalten in der Arbeitsdatei verwendet werden soll. Die Standardeinstellung ist ein Leerzeichen.

Die statistischen Daten, die in die Arbeitsdatei geschrieben werden, sind Momentaufnahmen der Liste, die mit der Funktion **List Objects** erzeugt wurde. Erläuterungen zu den Spalten finden Sie unter *Objekte auflisten*.

Die PC-Textdatei kann als Grundlage für eine Tabellenkalkulation verwendet werden.

29 Sortierten Auszug anzeigen

Die Funktion **Display Sorted Extract** erzeugt einen sortierten Auszug von 50 Verzeichniseinträgen der zurzeit in den Buffer Pool geladenen Natural-Objekte. Diese Liste kann für Auswertungszwecke verwendet werden, z.B. um die Objekte zu bestimmen, die als resident markiert oder in eine Preload-Liste aufgenommen werden sollen, wie im entsprechenden Abschnitt beschrieben.

Um die Funktion aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `X` ein und geben Sie im Feld **Object Pool** den Typ des Pools an, indem Sie `B` (Buffer Pool) oder `C` (BP Cache) wählen.

  Das Fenster **Specify Sort Criteria** (Auswahlkriterien anzeigen) erscheint. Geben Sie in den Eingabefeldern einen der folgenden Codes ein, um eine Spalte und die Reihenfolge auszuwählen, nach der die Statistikspalten sortiert werden sollen:

Sort Value	`O`	**ObjSize** = Object Size/Objektgröße. Dies ist die Standardeinstellung.
	`T`	**TotalUC** = Total Use Count. Siehe Beschreibung von **BP Total Use** in *Verzeichnisinformationen anzeigen*.
	`L`	**BP Last Action** (gilt nur für den Buffer Pool). Das Datum und die Uhrzeit, wann die Objekte zuletzt von einer Anwendung verwendet wurden.
Sort Order	`D`	Absteigende Reihenfolge. Dies ist die Standardeinstellung.
	`A`	Aufsteigende Reihenfolge.

  Oder:

  Benutzen Sie das SYSBPM-Direktkommando `SORT` oder `SORT BPC`, siehe *SYSBPM-Direktkommandos*.

  Der Bildschirm **BP Extract** erscheint und zeigt angegebenen Sortierkriterien. Erläuterungen zu den Spalten und den entsprechenden Kommandos siehe ***Objekte auflisten***.

30 Buffer Pool-Statistiken anzeigen

- Allgemeine Buffer Pool-Statistiken anzeigen 214
- Buffer Pool Load/Locate-Statistiken 217
- Buffer Pool-Fragmentierung 222
- Nutzung der internen Funktionen 223
- Buffer Pool Hash Table-Statistiken 224
- Performance-Informationen anzeigen 226
- PF-Tasten und Direktkommandos für die Buffer Pool-Statistiken 231

Die Funktion **Buffer Pool Statistics** ruft das gleichnamige Menü auf, in dem Sie Statistiken zum Buffer Pool abrufen können (einschließlich Hash-Tabellen-Statistiken).

Um die Funktion aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `A` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY STATISTICS
  ```

  Das Menü **Buffer Pool Statistics** wird angezeigt.

Die im Menü **Buffer Pool Statistics** verfügbaren Funktionen und die auf den Bildschirmen verfügbaren Kommandos zum Aufrufen dieser Funktionen werden in diesem Kapitel beschrieben.

Allgemeine Buffer Pool-Statistiken anzeigen

Die Funktion **General Buffer Pool Statistics** dient zur Überwachung der Performance des Buffer Pool und zeigt Statistiken über die Aktivität des Buffer Pool an.

Um die Funktion aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode `G` ein.

  ```
  G
  ```

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY GENERAL
  ```

  Der Bildschirm **General Buffer Pool Statistics** erscheint.

Die im Bildschirm **General Buffer Pool Statistics** angezeigte Statistik ist eine Momentaufnahme des Buffer Pool, die jedes Mal, wenn Sie ENTER drücken, aktualisiert wird. Die folgenden Informationen werden angezeigt:

Feld	Erläuterung
Buffer Pool Address	Die Adresse des Buffer Pool.
Directory Section	Die Adresse des Verzeichnisabschnitts des Buffer Pool relativ zum Anfang des Buffer Pool. Für jedes in den Buffer Pool geladene Objekt ist ein Verzeichniseintrag erforderlich, der Informationen über dieses Objekt enthält. Der Platz für diese Verzeichniseinträge wird im Buffer Pool zugeordnet.
Text Record Section	Die Adresse des Textsatzabschnitts relativ zum Anfang des Buffer Pool. Nachdem der von den Verzeichniseinträgen belegte Platz zugeordnet wurde, wird der verbleibende Platz in Blöcke aufgeteilt, die Textsätze genannt werden (deren Größe standardmäßig 4 KB beträgt). Ein Objekt kann je nach seiner Größe einen oder mehrere Textsätze belegen.
Cache attached	Der Name des Cache, der dem Buffer Pool zugeordnet ist. Dies ist entweder der Name des Datenraums oder, im Fall eines Speicherobjekts (64-Bit-Speicherung), `MEMOBJ`.
Buffer Pool Size (MB)	Die Größe des gesamten Buffer Pools in MB. Die Buffer Pool-Größe kann mit dem Makro `NTBPI` im Parametermodul oder mit dem Profilparameter `BPI` angegeben werden, siehe *Parameter Referenz*-Dokumentation.
Directory Entry Size	Die Größe eines Verzeichniseintrags in Bytes.
Text Record Size (KB)	Die Größe eines Textsatzes in KB. Die Größe eines Textsatzes kann mit dem Makro `NTBPI` im Parametermodul oder mit dem Profilparameter `BPI` angegeben werden, siehe *Parameter Referenz*-Dokumentation. Sie können die Textsatzgröße eines vorhandenen Buffer Pools ändern, wenn Sie den Buffer Pool mit dem Kommando `INITIALIZE` reinitialisieren. Die Standardgröße der Textsätze ist auf 4 KB eingestellt. Wenn Sie jedoch Anwendungen verwenden, die aus vielen kleinen Objekten bestehen, empfehlen wir Ihnen, die Größe auf 2 KB zu reduzieren. Dadurch wird der Prozentsatz des ungenutzten Platzes im Buffer Pool verringert, obwohl dies dazu führen kann, dass Algorithmus 2 (siehe *METHOD=S* in der *Operations*-Dokumentation) häufiger aufgerufen wird.
Buffer Pool Start	Das Datum und die Uhrzeit, wann der Buffer Pool ursprünglich gestartet wurde.

Feld	Erläuterung
Last Initialization	Das Datum und die Uhrzeit der letzten Initialisierung des Buffer Pool sowie die Kennung des Benutzers, der die Initialisierung durchgeführt hat. Der Buffer Pool wird initialisiert: ■ beim ursprünglichen Start des Buffer Pool, ■ bei der Ausführung des SYSBPM-Direktkommandos `INITIALIZE` oder ■ bei der Ausführung des Kommandos `REFRESH` des GBP-Betriebsprogramms, siehe auch *Global Buffer Pool Operating Functions* in der *Operations*-Dokumentation.
Text Records - Total	Die Gesamtzahl der Textsätze.
Text Records - Used	Die Anzahl der aktuell verwendeten Textsätze.
Text Records - Used in %	Der Prozentsatz der aktuell verwendeten Textsätze.
Text Records - Max Used	Die maximale Anzahl der verwendeten Textsätze.
Text Records - Total Size	Der von allen verwendeten Textsätzen belegte Gesamtspeicherplatz, d. h. **Text Records - Used** multipliziert mit der Größe eines einzelnen Textsatzes. Die Differenz zwischen der Gesamtgröße der Textsätze und der Gesamtgröße des Objekts zeigt die Menge des ungenutzten Platzes im Bereich der Textsätze an und kann dem Systemadministrator als Anhaltspunkt dienen, ob die Größe der Textsätze geändert werden sollte.
Text Records - Avg Usage %	Die durchschnittliche Nutzung in Prozent aller genutzten Textsätze, d. h. **Objects - Total Size** geteilt durch **Text Records - Total Size**. Dieser Wert sollte nicht wesentlich kleiner als 75 % sein. Wenn der Buffer Pool fast voll ist, deutet jeder Wert über 75% auf eine gute Auslastung des Buffer Pool hin. Liegt die Auslastung deutlich unter 75 %, sollte die Größe der Textsätze verringert werden.
Space Used %	Die tatsächliche Nutzung des Textsatzbereichs in Prozent, d. h. **Objects - Total Size** geteilt durch die Gesamtgröße der Textsatzbereichs (**Text Records**). **Tipp:** Wenn der Buffer Pool fast voll ist (d.h. der Wert im Feld **Text Records - Used** beträgt fast 100%), deutet jeder Wert über 75% auf eine gute Auslastung des Buffer Pool hin. Liegt die Auslastung deutlich unter 75%, sollte die Textsatzgröße reduziert werden.
Objects - Loaded	Die Anzahl der derzeit in den Buffer Pool geladenen Objekte.
Objects - Max Loaded	Die maximale Anzahl von Objekten, die seit dem Start des Buffer Pool jemals gleichzeitig in den Buffer Pool geladen wurden.
Objects - Total Size	Die Gesamtgröße der zurzeit geladenen Objekte in Bytes.

Feld	Erläuterung
Objects - Avg TR Used	Die durchschnittliche Anzahl der von einem Objekt verwendeten Textsätze.
Objects - SumOfUseCounts	Die Summe der Nutzungszahlen aller derzeit in den Buffer Pool geladenen Objekte. Die Nutzungszahl zählt alle Anwendungen, die ein Objekt derzeit ausführen. Wenn ein Objekt derzeit nicht verwendet wird, ändert sich sein Nutzungszähler auf 0 (Null).
Objects - AvgLifetimeUsed(min)	Die durchschnittliche Lebensdauer (in Minuten) der zurzeit in den Buffer Pool geladenen Objekte.
Objects - AvgLifetimeReplace(min)	Die durchschnittliche Lebensdauer (in Minuten) der Objekte, die bereits im Buffer Pool ersetzt wurden.

Buffer Pool Load/Locate-Statistiken

Die Funktion **Buffer Pool Load/Locate Statistics** liefert statistische Informationen über das Laden von Objekten in den Buffer Pool und das Auffinden von Objekten im Buffer Pool. Diese Informationen dienen auch als Anhaltspunkt für die Performance des Buffer Pool.

Um die Funktion Buffer Pool Load/Locate Statistics aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode L ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY LOAD
  ```

  Der Bildschirm **Buffer Pool Load/Locate Statistics** wird angezeigt.

Die auf dem Bildschirm **Buffer Pool Load/Locate Statistics** angezeigten Statistiken sind Momentaufnahmen des Buffer Pool, die jedes Mal, wenn Sie ENTER drücken, aktualisiert werden.

Die folgenden Informationen werden auf dem Bildschirm angezeigt:

Feld	Erläuterung
Total Locate Calls	Die Gesamtzahl der Aufrufe zum Auffinden eines Objekts, d.h. wie oft der Natural Buffer Pool Manager aufgefordert wurde, den Buffer Pool nach einem Objekt zu durchsuchen. Wenn die Suche erfolgreich war, wurde das Objekt im Buffer Pool oder im BP Cache gefunden und muss nicht aus einer Natural-Systemdatei geladen werden, wodurch Aufrufe und E/A eingespart werden.
Total Locate Calls - successful	Die Gesamtzahl der erfolgreichen Locate-Aufrufe als absolute Zahl.
Total Locate Calls - failed	Die Gesamtzahl der fehlgeschlagenen Locate-Aufrufe.
Quick Locate Calls	Die Gesamtzahl der Quick Locate-Aufrufe. Ein Quick Locate-Aufruf ist der Versuch, ein Objekt im Buffer Pool unter Verwendung der internen Fast Locate-Tabelle zu finden. Weitere Informationen finden Sie unter *Interne Fast-Locate-Tabelle*. In diesem Fall werden die Library (Name, Datenbankkennung und Dateinummer) und die Position des Objekts im Buffer Pool des letzten erfolgreichen Locate-Aufrufs für dieses Objekt erneut verwendet, um das Objekt zu finden. Dies ist die effizienteste Methode zum Auffinden eines Objekts.
Quick Locate Calls - successful	Die Anzahl der erfolgreich durchgeführten Quick-Locate-Aufrufe.
Quick Locate Calls - failed	Die Anzahl der fehlgeschlagenen Quick Locate-Aufrufe. Ein fehlgeschlagener Quick Locate-Aufruf zeigt an, dass ein Objekt nicht an seiner vorherigen Position im Buffer Pool unter Verwendung der **internen Fast Locate-Tabelle** gefunden werden konnte (siehe den entsprechenden Abschnitt unter *Performance-Aspekte*). Dies geschieht, wenn das Objekt entweder aus dem Buffer Pool gelöscht oder an eine andere Position verschoben wurde. Ein fehlgeschlagener Quick-Locate-Aufruf führt zu einem Normal-Locate-Aufruf, jedoch ohne Steplib-Suche, da die Natural-Laufzeit sich die Library merkt, die das Objekt enthält.
Normal after Quick	Die Anzahl der Normal-Locate-Aufrufe, die aus **Quick Locate Calls - failed** resultieren. Der Wert von **Normal after Quick** ist immer identisch mit dem Wert von **Quick Locate Calls - failed**.
Normal after Quick - successful	Die Anzahl der erfolgreichen Normal-Locate-Aufrufe, die aus **Quick Locate Calls - failed** resultieren. Ein Normal after Quick-Aufruf ist erfolgreich, wenn sich das angeforderte Objekt noch im Buffer Pool befindet, aber an einer anderen Position.

Feld	Erläuterung
Normal after Quick - failed	Die Anzahl der fehlgeschlagenen Normal-Locate-Aufrufe, die aus **Quick Locate Calls - failed** resultieren. Ein fehlgeschlagener Normal after Quick-Aufruf zeigt an, dass das angeforderte Objekt im Buffer Pool nicht mehr verfügbar ist. Infolgedessen wird das Objekt aus der Systemdatei unter Verwendung des Library-Eintrags in der i **internen Fast-Locate-Tabelle** neu geladen (siehe den entsprechenden Abschnitt in *Buffer Pool Load/Locate-Statistiken*).
Normal Locate Calls	Die Gesamtzahl der Normal Locate-Aufrufe. Ein Normal Locate-Aufruf ist der Versuch, ein Objekt im Buffer Pool zu finden, ohne die **interne Fast Locate-Tabelle** zu verwenden (siehe den entsprechenden Abschnitt in *Buffer Pool Load/Locate-Statistiken* Ein Normal Locate-Aufruf erfolgt immer dann, wenn ein Objekt zum ersten Mal innerhalb einer Natural-Sitzung referenziert wird, nachdem ein Systemkommando `LOGON` ausgeführt wurde. Im Gegensatz zu einem Quick Locate-Aufruf ist der Natural-Laufzeitumgebung noch nicht bekannt, in welcher Library sich das Objekt befindet.
Normal Locate Calls - successful	Die Anzahl der normalen Locate-Aufrufe, bei denen das benötigte Objekt im Buffer Pool gefunden werden konnte.
Normal Locate Calls - failed	Die Anzahl der fehlgeschlagenen normalen Locate-Aufrufe. Ein fehlgeschlagener Normal Locate-Aufruf zeigt an, dass ein Objekt (identifiziert durch seinen Namen und die Library, in der es sich befindet) nicht im Buffer Pool gefunden werden konnte. Ein fehlgeschlagener Normal Locate-Aufruf führt entweder zu einem Laden aus dem BP Cache oder einer Systemdatei oder zu einem Normal Locate-Aufruf für die nächste Library in der Steplib-Kette.
STEPLIB Searches	Die Anzahl der **Normal Locate**-Aufrufe, die durch fehlgeschlagene Versuche, ein Objekt in einer Steplib-Library zu finden, aufgetreten sind. Normale Locate-Aufrufe, die erfolglose Steplib-Suchen durchführen, führen nicht zum Laden eines Objekts aus dem BP Cache oder der Systemdatei. **STEPLIB Searches** zählt *keine* Locate-Aufrufe für Objekte, die weder in der aktuellen Library noch in einer Steplib oder Systemdatei enthalten sind (Fehlermeldung: Ungültiges Kommando, oder Programm existiert nicht in Library). Locate-Aufrufe, die aufgrund fehlerhafter Programmierung fehlschlagen, werden zum Zähler bei **Normal Locate Calls - failed** addiert. Die Anzahl der STEPLIB-Suchvorgänge (**STEPLIB Searches**) wird nach der folgenden Formel berechnet:

Feld	Erläuterung
	Normal Locate Calls - failed - (**Number Loads into BP** - **Normal after Quick - failed**) Je geringer die Anzahl der STEPLIB-Suchvorgänge ist, desto besser ist die Performance des Buffer Pool. Siehe auch ***Suche in Steplibs***.
Number Loads into BP	Die Anzahl der erfolgreich in den Buffer Pool durchgeführten Ladevorgänge. Das Laden in den Buffer Pool (Speicherzuordnungsanforderung) kann entweder durch ein Laden aus der Datenbank oder durch ein Laden aus dem BP Cache ausgelöst werden.
Loads from Cache	Die Gesamtzahl der erfolgreichen Locate-Aufrufe von Objekten, die sich im BP Cache befanden. Diese Information wird nur gezählt, wenn der vorherige Locate-Aufruf (**Normal after Quick - failed** oder **Normal Locate Calls - failed**) fehlgeschlagen ist. Sie gibt die Anzahl der gespeicherten Datenbankladungen an. Dies bedeutet, dass das Objekt ohne den BP Cache aus der Datenbank geladen werden müsste.
Loads from DB	Die Anzahl, wie oft ein Objekt aus einer Natural-Systemdatei in den Buffer Pool geladen wurde. Da mehrere Ladeaufrufe erforderlich sein können, um ein einzelnes Objekt zu laden, gibt dieser Wert die tatsächliche Anzahl der Objektladevorgänge seit der letzten Aktualisierung des Buffer Pool an. Beim Laden eines Objekts verwendet der Buffer Pool Manager verschiedene Suchalgorithmen: siehe *METHOD=S* und *METHOD=N* in der *Operations*-Dokumentation.
Loads from DB - finished	Die Anzahl der erfolgreich abgeschlossenen Objektladevorgänge. Ein Objektladevorgang kann nicht beendet werden, wenn der Ladevorgang aus einem der folgenden Gründe abgebrochen wird: ■ Ein gleichzeitiger Objektladevorgang ist aufgetreten: siehe **Loads from DB - concurrent**. ■ Während des Objektladevorgangs tritt ein Adabas-Antwortcode auf. ■ Während des Objektladevorgangs wird eine SYSBPM-Löschoperation für dieses Objekt ausgeführt.
Loads from DB - concurrent	Die Anzahl der Objektladevorgänge, die gleichzeitig für dasselbe Objekt durchgeführt wurden: Gleichzeitige Objektladevorgänge treten auf, wenn zwei oder mehr Natural-Sitzungen, die gleichzeitig laufen, das gleiche Objekt anfordern. Während ein Objekt von einer Sitzung geladen wird, fordern andere

Feld	Erläuterung
	Sitzungen dasselbe Objekt an und beginnen mit dem Laden, bevor eine Sitzung den Ladevorgang beendet hat. In diesem Fall wird dasselbe Objekt mehr als einmal geladen. Die erste Sitzung, die mit dem Laden des Objekts fertig ist, markiert das Objekt der anderen Sitzungen zum Löschen aus dem Buffer Pool. Die anderen Sitzungen hören dann auf, das Objekt zu laden, entfernen das zur Löschung markierte Objekt aus dem Buffer Pool und verwenden das von der ersten Sitzung erfolgreich geladene Objekt. Die Anzahl der Objekte, die von den Zählern **Loads from DB - finished** und **Loads from DB - concurrent** berechnet werden, sind normalerweise identisch. Die Zahlen unterscheiden sich nur, wenn das gleichzeitige Laden erst erkannt wird, nachdem beide Sitzungen das Laden beendet haben.
Load Calls	Die Gesamtzahl der Ladeaufrufe, die seit der Aktualisierung des Buffer Pool erfolgt sind. Die Ladeaufrufe sind mit dem Zugriff auf die Systemdatei korreliert, aus der die Objekte gelesen werden. Die Anzahl der Systemdateizugriffe wird wie folgt berechnet: ■ Adabas-Systemdatei: Anzahl der **Load Calls** plus Anzahl der **Loads from DB** (siehe oben). In der Gesamtzahl sind die Adabas-RC-Aufrufe nicht enthalten. ■ VSAM-Systemdatei: Die Anzahl der **Load Calls**.
Number Loads BP 2nd	Dieses Feld wird angezeigt, wenn METHOD=S (Selektionsverfahren) als Suchmethode für die Speicherzuordnung verwendet wird. In diesem Feld wird angezeigt, wie oft eine Speicherzuordnungsanfrage die Suchkriterien von Algorithmus 2 erfüllte, wie unter *METHOD=S* in der *Operations*-Dokumentation beschrieben.
Number Load Cycles	Dieses Feld wird angezeigt, wenn als Suchmethode für die Speicherzuordnung `METHOD=N` (nächstverfügbar) verwendet wird, wie in der *Operations*-Dokumentation beschrieben. Dieses Feld gibt an, wie oft eine Suche vom Anfang des Buffer Pool aus durchgeführt wurde. Diese Zahl gibt einen Anhaltspunkt für die Häufigkeit des Durchlaufens des Buffer Pool in einem Wrap-around-Verfahren.
Last Cycle Start	Dieses Feld wird angezeigt, wenn als Suchmethode für die Speicherzuordnung `METHOD=N` (nächstverfügbar) verwendet wird, wie in der *Operations*-Dokumentation beschrieben. Die Zeit und das Datum, wann die Anzahl der Ladezyklen (**Number Load Cycles**) zuletzt erhöht wurde.

Feld	Erläuterung
Number Lock Retries	Dieses Feld wird angezeigt, wenn als Suchmethode für die Speicherzuordnung `METHOD=N` (nächstverfügbar) verwendet wird, wie in der *Operations*-Dokumentation beschrieben. Dieses Feld gibt an, wie oft eine Kette von gesperrten Buffer Pool-Einträgen entsperrt werden musste, weil sie die Zuordnungsanforderung nicht erfüllen konnten.
Largest Alloc (TR)	Die größte bisher angeforderte einzelne Zuordnungsgröße, angegeben in Anzahl der Textsätze.
Number Load Failure	Die Gesamtzahl der Fehlschläge bei einem Objektladevorgang. Der Grund für einen Fehlschlag ist, dass entweder alle Verzeichniseinträge zum Zeitpunkt der Ladeanforderung belegt sind oder nicht genügend Speicherplatz im Textsatzbereich zur Verfügung steht, um den Ladevorgang durchzuführen.
Number Load Failure - Sizes failing last	Die Anzahl der Textsätze, die für die letzten drei fehlgeschlagenen Speicherplatzanforderungen erforderlich gewesen wären.

Ausführliche Informationen zu den Suchmethoden, die für die Zuweisung von Speicherplatz im Buffer Pool verwendet werden, finden Sie unter *Buffer Pool Search Methods* in der *Operations*-Dokumentation.

Buffer Pool-Fragmentierung

Die Funktion **Buffer Pool Fragmentation** bietet einen Überblick über die Fragmentierung des Buffer Pool, d.h. einen Überblick darüber, wie viele verschiedene Natural-Objekte wie viele Textsätze belegen und wie die Objektpositionen über den Buffer Pool verteilt sind.

Um die Funktion Buffer Pool Fragmentation aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode `F` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY FRAGMENTATION
  ```

  Der Bildschirm **Buffer Pool Fragmentation** wird angezeigt.

Einige der Felder auf dem Bildschirm **Buffer Pool Fragmentation** sind identisch mit den Feldern, die unter ***General Buffer Pool Statistics*** erläutert werden:

Buffer Pool Size

Buffer Pool Address

Text Record Section

Text Record Size

Number of Text Records (entspricht **Text Records - Total**)

Außerdem wird ein Diagramm angezeigt, aus dem hervorgeht, wie viele verschiedene Einzelobjekte wie viel Textsatzgröße beanspruchen.

Zum Beispiel:

```
1---+----10---+----20---+----30---+----40---+----50

005F0480
..._.+***_____++
....**_.+**_.++_ *..+**+__++++XX
```

Jedes Symbol im Diagramm steht für einen Textsatz, und jede Folge von gleichen Symbolen steht für ein anderes Einzelobjekt, das einen oder mehrere Textsätze belegt. Die Symbole haben die folgende Bedeutung:

_ und .	Objekte mit einem **Current Use Count** (siehe *Verzeichnisinformationen anzeigen*) von 0 (Null).
+ und *	Objekte mit einem **Current Use Count** größer als 0 (Null).
Leerzeichen	Ein unbenutzter Textsatz.
XX	Das Ende des Buffer Pool, d.h. es sind keine weiteren Textsätze vorhanden.

Im obigen Beispiel enthält der Buffer Pool 48 Textsätze. Drei davon sind nicht in Gebrauch. Der Rest wird von 24 anderen Objekten belegt, 12 davon mit einem **Current Use Count** von 0 (Null) und 12 mit einem **Current Use Count** größer als 0.

Nutzung der internen Funktionen

Die Funktion **Internal Function Usage** liefert statistische Informationen über die Aufrufe, die an den Natural Buffer Pool Manager abgesetzt wurden.

Um die Funktion Internal Function Usage aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode I ein.

  Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY FUNCTION
```

Der Bildschirm **Internal Function Usage** erscheint.

Die angezeigte Statistik ist eine Momentaufnahme des Buffer Pool, die jedes Mal, wenn Sie ENTER drücken, aktualisiert wird.

Das Feld **Total Calls** zeigt die Gesamtzahl aller internen Aufrufe an, die an den Buffer Pool Manager abgesetzt wurden.

Intern kann der Buffer Pool Manager für verschiedene Funktionen aufgerufen werden. Für jede Funktion wird angezeigt, wie oft sie aufgerufen wurde, sowohl absolut als auch prozentual. Darüber hinaus werden diese Zahlen in einem horizontalen Balkendiagramm dargestellt.

Buffer Pool Hash Table-Statistiken

Die Funktion **Buffer Pool Hash Table Statistics** gilt nur für Buffer Pools des Typs Natural.

Sie zeigt Statistiken zu Hash-Tabellen-Slots und Kollisionen pro Slot an. Die Statistiken geben Aufschluss über die Effizienz des verwendeten Hash-Algorithmus.

Weitere Informationen zu Hash-Tabellen finden Sie unter *Buffer Pool Hash Table* in der *Operations*-Dokumentation.

Die Statistiken sind in erster Linie nur für den internen Gebrauch durch unsere Mitarbeiter bestimmt.

Um die Funktion Buffer Pool Hash Table Statistics aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode `H` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY HASH
  ```

  Der Bildschirm **Hash Table Collisions** erscheint.

Die dort angezeigte Statistik ist eine Momentaufnahme der Hash-Tabelle, die jedes Mal erstellt wird, wenn Sie ENTER drücken. Es werden die folgenden Informationen angezeigt:

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Total Number of Slots</td><td>Die Gesamtzahl der Hash-Tabellen-Slots, d. h. die Gesamtzahl der möglichen Einträge in der Hash-Tabelle, die die Objektnamen mit dem Speicherort der Objekte verbinden.

Die Anzahl der Slots, d. h. die Größe der Hash-Tabelle, wird intern in Abhängigkeit von der Anzahl der Textsätze berechnet.</td></tr>
<tr><td>Number of Slots used</td><td>Die Anzahl der Slots in der Hash-Tabelle, auf die mindestens ein Objektname abgebildet ist.</td></tr>
<tr><td>Number of Slots free</td><td>Die Anzahl der Slots in der Hash-Tabelle, auf die kein Objektname abgebildet ist.</td></tr>
<tr><td>Max. Collisions per Slot</td><td>Die maximale Anzahl der Kollisionen eines beliebigen Slots. Die maximale Anzahl der Kollisionen ist der längste mögliche Suchpfad für ein Objekt.

Eine Kollision wird dann verursacht, wenn der Name von zwei verschiedenen Objekten durch den Hash-Algorithmus auf denselben Slot abgebildet wird. In diesem Fall wird eine Kollisionsauflösung verwendet, um einen anderen Slot zu finden.</td></tr>
<tr><td>Collisions</td><td>Die Anzahl der aktuellen Kollisionen. Abhängig von den aufgetretenen Kollisionen enthält die Tabelle bis zu 10 Zeilen:
<table>
<tr><td></td><td></td></tr>
<tr><td>0</td><td>Keine Kollision.</td></tr>
<tr><td>1</td><td>1 Kollision.</td></tr>
<tr><td>2</td><td>2 Kollisionen.</td></tr>
<tr><td>3</td><td>3 Kollisionen.</td></tr>
<tr><td>4</td><td>4 Kollisionen.</td></tr>
<tr><td>5</td><td>5 Kollisionen.</td></tr>
<tr><td>6 - 10</td><td>Zwischen 6 und 10 Kollisionen.</td></tr>
<tr><td>11 - 15</td><td>Zwischen 11 und 15 Kollisionen.</td></tr>
<tr><td>16 - 20</td><td>Zwischen 16 und 20 Kollisionen.</td></tr>
<tr><td>21</td><td>Mehr als 21 Kollisionen.</td></tr>
<tr><td></td><td></td></tr>
</table>
Keine Kollision bedeutet, dass nur ein Objektname pro Slot abgebildet wird. Um dieses Objekt zu finden, müssen Sie nur einmal auf die Hash-Tabelle zugreifen.

Wenn die Anzahl der Kollisionen größer als 0 (Null) ist, werden z. B. x, x+1 Objektnamen auf denselben Slot abgebildet. Um eines dieser Objekte zu finden, müssen Sie auf die Hash-Tabelle bis zu x+1 zugreifen.</td></tr>
<tr><td>Number of Slots</td><td>Die Anzahl der Slots bezogen auf die Anzahl der Kollisionen.

Darüber hinaus wird der prozentuale Anteil dieser Slots an allen verwendeten Slots angezeigt.</td></tr>
</table>

Feld	Erläuterung
Number of Slots Totaled	Die gleichen Werte wie **Number of Slots**, aber die Werte werden summiert.

Beispiel für Hash Table Statistics:

```
 14:21:57              ***** NATURAL SYSBPM UTILITY *****              2020-04-07
 BPNAME QA82GBP       - Buffer Pool Hash Table Statistics -       Type Global Nat
 BPPROP OFF                                                         Loc DAEF QA82
                                                                  Preload QA82GBPL
 Total Number of Slots ..          5,351
  Number of Slots used ..          1,280 (  23.9 %)     Max. Collisions
  Number of Slots free ..          4,071 (  76.0 %)       per Slot ..... 8

      Collisions            Number of Slots   Number of Slots Totaled
                0             283 (  22.1 %)             283 (  22.1 %)
                1             604 (  47.1 %)             887 (  69.2 %)
                2             207 (  16.1 %)           1,094 (  85.4 %)
                3             121 (   9.4 %)           1,215 (  94.9 %)
                4              46 (   3.5 %)           1,261 (  98.5 %)
                5              14 (   1.0 %)           1,275 (  99.6 %)
          6 - 10                5 (   0.3 %)           1,280 ( 100.0 %)




Command ===>


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

Performance-Informationen anzeigen

Die Funktion **Performance Hints** gilt nur für einen Buffer Pool und einen BP Cache des Typs `Nat` (Natural).

In diesem Abschnitt werden die folgenden Themen behandelt:

- Funktion Performance Hints aufrufen
- Bewertung von Performance-Hinweisen

Verwandtes Thema:

Performance-Aspekte

Funktion Performance Hints aufrufen

In diesem Abschnitt wird beschrieben, wie Sie die Funktion **Performance Hints** aufrufen und welches Statistikfeld auf dem gleichnamigen Bildschirm angezeigt wird.

Um die Funktion Performance Hints aufzurufen:

- Geben Sie im Menü **Buffer Pool Statistics** den Funktionscode `P` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY PERFORMANCE
```

  Es erscheint ein Bildschirm mit Hinweisen ähnlich dem folgenden Beispiel:

```
 13:22:51             ***** NATURAL SYSBPM UTILITY *****             2020-04-07
 BPNAME QA92GBP             - Performance Hints -               Type Global Nat
 BPPROP OFF                                                       Loc DAEF QA92
                                                                Preload QA92GBPL

                                                          Rating
                                                    (1=best - 6=worst)
 Buffer Pool
              Locates / Loads Ratio ......       476.16       2
              Wrap Time Last (hh:mm:ss) ..     00:02:34       5
              Wrap Time Avg (hh:mm:ss) ...     00:00:17       6

 BP Cache
              Object Reuse Factor ........         0.27       6
              Wrap Time Last (hh:mm:ss) .. no value returned
              Wrap Time Avg (hh:mm:ss) ... no value returned
              Get / Search % .............        16.38 %



Command ===>


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

Die Felder auf dem Bildschirm **Performance Hints** enthalten die folgenden Informationen:

Feld	Erläuterung
Buffer Pool - Locates / Loads Ratio	Das Verhältnis von **Total Locate Calls - successful** zu **Loads from DB**. Ein Wert größer als 1 bedeutet, dass Natural mehr Objekte im Buffer Pool gefunden als aus der Systemdatei geladen wurden. Dieses Verhältnis dient als Kennzahl für die Effizienz des Buffer Pool. Je größer die Zahl ist, desto besser ist die Performance des Buffer Pool. Dies ist die wichtigste Kennzahl für die Leistung von einer Buffer Pool-Sitzung zur nächsten.
Buffer Pool - Wrap Time Last	Dieses Feld wird angezeigt, wenn `METHOD=N` (nächstverfügbar) als Suchmethode für die Speicherzuordnung verwendet wird, wie in der *Operations*-Dokumentation beschrieben. Die Zeit in Stunden, Minuten und Sekunden (*hh*:*mm*:*ss*) seit der letzten vollständigen Wiederverwendung des Buffer Pool (Wrap Around). Die Objekte werden nacheinander in den Buffer Pool geladen (in sequenzieller Reihenfolge), wobei der obere Teil des Buffer Pool zuerst und der untere zuletzt gefüllt wird. Wenn das untere Ende des Buffer Pool erreicht ist, wird das Ende des Buffers umgebrochen und das nächste Objekt wird wieder an den oberen Rand des Buffer Pool geladen. Wenn der Buffer Pool zum ersten Mal vollständig gefüllt ist, überschreibt ein neues Objekt, das in den Buffer geladen wird, ein Objekt, das im vorangegangenen Wrap-around-Zyklus geladen wurde und das derzeit nicht gesperrt (als resident oder in Verwendung gekennzeichnet) ist. Jedem in den Buffer Pool geladenen Objekt wird ein Verzeichniseintrag zugeordnet, der Informationen wie den Namen des Objekts, die Library, in der es gespeichert ist, und den Zeitstempel **BP Load Time** des Ladens in den Buffer Pool enthält. **Wrap Time Last** wird jedes Mal ausgewertet, wenn ein Objekt in den Buffer Pool geladen wird. **Wrap Time Last** gibt die Zeitspanne zwischen dem letzten Laden eines Objekts (**BP Load Time**) und dem letzten Überschreiben eines Objekts an. Je länger der Zeitraum eines Wrap-around-Zyklus ist, desto besser ist die Leistung des Buffer Pool. Dieser Zeitraum kann nachts oder am Wochenende, wenn der Benutzer-Datenverkehr im Vergleich zu den normalen Arbeitszeiten gering ist, erheblich variieren.
Buffer Pool - Wrap Time Avg	Dieses Feld wird angezeigt, wenn `METHOD=N` (nächstverfügbar) als Suchmethode für die Speicherzuordnung verwendet wird, wie in der *Operations*-Dokumentation beschrieben. Die durchschnittliche Zeit in Stunden, Minuten und Sekunden (*hh*:*mm*:*ss*) eines Wrap-around-Zyklus seit der Initialisierung oder Auffrischung des Buffer Pool. **Wrap Time Avg** wird berechnet, indem die Lebensdauer des Buffer Pool durch die Anzahl der Wrap-Around-Zyklen geteilt wird. **Wrap Time Last** im Vergleich zu **Wrap Time Avg** zeigt an, ob der Buffer Pool derzeit häufiger als im Durchschnitt verwendet wird.

Feld	Erläuterung
BP Cache - Object Reuse Factor	Das Verhältnis von aus dem BP Cache ausgelagerten Objekten (**Get-Aufrufe**) zu in den BP Cache ausgelagerten Objekten (**Put-Aufrufe**). Der Wert wird berechnet, indem erfolgreiche Get-Aufrufe durch erfolgreiche Put-Aufrufe geteilt werden. Er zeigt den Gesamtwiederverwendungsfaktor an, d.h. wie oft ein einmal in den BP Cache geladenes Objekt erfolgreich in den Buffer Pool zurückgeladen werden konnte. Je höher der Wert, desto besser ist die Effizienz des BP Cache. Beispiel für einen **Object Reuse Factor**: Ein Verhältnis von `5.70` bedeutet, dass ein in den BP Cache geladenes Objekt im Durchschnitt 5,7 Mal in den Buffer Pool ausgelagert wurde.
BP Cache - Wrap Time Last	Die Zeit in Stunden, Minuten und Sekunden (*hh*:*mm*:*ss*) seit der letzten vollständigen Wiederverwendung (Wrap Around) des BP Cache. Die Objekte werden nacheinander in den BP Cache geladen (in sequenzieller Reihenfolge), wobei der obere Teil des BP Cache zuerst und der untere zuletzt gefüllt wird. Wenn das untere Ende des BP Cache erreicht ist, wird das Ende des BP Cache umgebrochen und das nächste Objekt wird wieder am oberen Ende des BP Cache geladen. Wenn der BP Cache zum ersten Mal vollständig gefüllt ist, überschreibt ein neues Objekt, das in den Buffer geladen wird, das Objekt, das im vorherigen Wrap-around-Zyklus geladen wurde. Jedem in den BP Cache geladenen Objekt wird ein Verzeichniseintrag zugewiesen, der Informationen wie den Namen des Objekts, die Library, in der es gespeichert ist, und den **BPC Load Time**-Zeitstempel, wann es in den BP Cache geladen wurde, enthält. **Wrap Time Last** wird jedes Mal ausgewertet, wenn ein Objekt in den GP-Cache geladen wird. **Wrap Time Last** gibt die Zeitspanne zwischen dem letzten Laden eines Objekts (**BPC Load Time**) und dem letzten Überschreiben eines Objekts an. Je länger der Zeitraum eines Wrap-around-Zyklus ist, desto besser ist die Performance des BP Cache. Dieser Zeitraum kann nachts oder am Wochenende, wenn der Benutzer-Datenverkehr im Vergleich zu den normalen Arbeitszeiten gering ist, erheblich variieren.
BP Cache - Wrap Time Avg	Die durchschnittliche Zeit in Stunden, Minuten und Sekunden (*hh*:*mm*:*ss*) eines Wrap-around-Zyklus seit dem Start des BP Cache. **Wrap Time Avg** wird berechnet, indem die Nutzungsdauer des BP Cache durch die Anzahl der Wrap-Around-Zyklen geteilt wird. **Wrap Time Last** im Vergleich zu **Wrap Time Avg** zeigt, ob der BP Cache derzeit häufiger als im Durchschnitt verwendet wird.
BP Cache - Get / Search %	Der Prozentsatz der vom BP Cache erfolgreich zurückgegebenen Objekte (**Get-Aufrufe**) im Vergleich zur Gesamtzahl der Suchanfragen (**Search-Aufrufe**), die der Buffer Pool an den BP Cache gesendet hat. Dieser Wert gibt den Prozentsatz der Objekte an, die der Buffer Pool aus dem BP Cache laden konnte, anstatt aus der Natural-Systemdatei FNAT oder FUSER. Je höher der Wert ist, desto besser ist die Cache-Effizienz.

Feld	Erläuterung
	Beachten Sie, dass eine Suche nach einem Objekt in einer Kette von Steplibs die Anzahl der Suchaufrufe erhöhen kann, die der Buffer Pool an den BP Cache sendet. Diese Aufrufe führen jedoch nicht zu einem erfolgreichen Get-Aufruf oder zu einem Laden aus der Natural-Systemdatei, da ein Objekt möglicherweise nicht in einer Steplib gefunden wird. **Get / Search %** berücksichtigt nicht die Suche durch eine lange Kette von Steplibs, die häufig verwendet wird. Wir empfehlen Ihnen, so wenig Steplibs wie möglich zu verwenden, um die Gesamt-Performance zu erhöhen. Beispiel für einen **Get / Search %**-Wert: Ein Wert von 70 % bedeutet, dass 70 % aller in den Buffer Pool geladenen Objekte aus dem BP Cache abgerufen und 30 % aus der Systemdatei geladen wurden.

Bewertung von Performance-Hinweisen

Die statistischen Werte, die auf dem Bildschirm **Performance Hints** angezeigt werden, bilden die Grundlage für ein Leistungsbewertungssystem, bei dem 1 die beste (höchste) und 6 die schlechteste (niedrigste) Performance bezeichnet.

Die Bewertungen sollen Ihnen einen Eindruck davon vermitteln, wie ein Buffer Pool oder ein BP Cache mit den für sie festgelegten Werten abschneidet. Die Einstufungswerte sollten den Anforderungen der meisten Systemumgebungen entsprechen.

Es gibt Umgebungen, in denen es sich bewährt hat, einen BP Cache zu haben, der fünfmal so groß wie der Buffer Pool ist. Wenn die Bewertungen tendenziell schlecht sind, sollte die Größe des Buffer Pool oder des BP Cache erhöht werden. Die Größe eines Buffer Pool oder eines BP Cache kann jedoch auch dann eine gute Performance garantieren, wenn die Bewertungen schlecht sind. Wenn die Bewertungen für einen BP Cache gut sind, fallen schlechte Bewertungen für den Buffer Pool nicht so stark ins Gewicht. In Umgebungen mit extremen Arbeitslasten können die Bewertungen jedoch nützliche Anhaltspunkte dafür sein, wann die Größe des Buffer Pool oder des BP Cache geändert werden sollte.

Die auf dem Bildschirm **Performance Hints** angezeigten statistischen Werte sind Momentaufnahmen des Natural Buffer Pool und des BP Cache, die jedes Mal aktualisiert werden, wenn ein Wraparound auftritt. Der Buffer Pool und der BP Cache sollten einige Zeit laufen und die Statistikwerte einen gewissen Umfang erreichen, um aussagekräftige Ergebnisse aus diesen Werten zu erhalten. Wenn z.B. die Größe eines Buffer Pools so groß ist, dass nur sehr wenige BP Cache-Aufrufe erforderlich sind, sind die Statistiken über den BP Cache nicht aussagekräftig.

Bei der Auswertung der Statistiken sollten Sie auch die Art der Systemumgebung (z.B. Produktion oder Test), die Art der Anwendungen, die den Buffer Pool nutzen (Batch, Online, benutzerdefiniert oder System), den Benutzer-Datenverkehr (Spitzenzeiten oder normale Zeiten) und außergewöhnliche Betriebsfaktoren berücksichtigen.

PF-Tasten und Direktkommandos für die Buffer Pool-Statistiken

Auf den Buffer Pool-Statistik-Bildschirmen können Sie die in der folgenden Tabelle aufgeführten PF-Tasten oder SYSBPM-Direktkommandos verwenden. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos finden Sie unter *SYSBPM-Direktkommandos*.

PF-Taste	Kommando	Funktion
PF1		Liefert SYSBPM-Hilfeinformationen: siehe auch *Online-Hilfe*.
PF3	EXIT	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF4	LAST	Anzeige des zuletzt eingegebenen SYSBPM-Direktkommandos.
PF6	FLIP	Schaltet die PF-Tastenzeile um: Wechsel zwischen der Anzeige von PF1 bis PF12 und PF13 bis PF24.
PF8 (Load)	DISPLAY LOAD	Gilt nur für den Bildschirm **General Buffer Pool Statistics**. Anzeige des Bildschirms **Buffer Pool Load/Locate Statistics**.
PF8 (Gen)	DISPLAY GENERAL	Gilt nur für den Bildschirm **Buffer Pool Load/Locate Statistics**. Anzeige des Bildschirms **General Buffer Pool Statistics**.
PF12	CANCEL	Wie EXIT.
PF15	MENU	Aufruf des SYSBPM-Hauptmenüs (**Main Menu**).

31 BP Cache-Statistiken anzeigen

- Allgemeine BP Cache-Statistiken anzeigen 234
- BP Cache Call-Statistiken 236
- BP Cache Hash Table-Statistiken anzeigen 238
- Performance-Hinweise 239
- PF-Tasten und Direktkommandos für die BP Cache-Statistiken 239

Die Funktion **BP Cache** gilt nur für einen Buffer Pool vom Typ `Natural`.

Diese Funktion ruft das Menü **BP Cache Statistics** auf, über das Sie statistische Informationen über den Buffer Pool Cache einholen können.

Beachten Sie, dass die Funktion **BP Cache** nur ausgeführt werden kann, wenn bei der Initialisierung eines globalen Buffer Pool ein BP Cache installiert wurde. Bei lokalen Buffer Pools gibt es keine BP Cache-Unterstützung.

Um das Menü BP Cache Statistics aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode C ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY CSTATISTICS
  ```

  Das Menü **BP Cache Statistics** wird angezeigt.

Die in diesem Menü verfügbaren Funktionen und die auf den Bildschirmen angebotenen Kommandos, durch die diese Funktionen aufgerufen werden, werden in diesem Kapitel erläutert.

Allgemeine BP Cache-Statistiken anzeigen

Die Funktion **General BP Cache Statistics** zeigt Adressen und Statistiken über die Aktivität des BP Cache an.

Um die Funktion General BP Cache Statistics aufzurufen:

- Geben Sie im Menü **BP Cache Statistics** den Funktionscode `G` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY CGENERAL
  ```

  Der Bildschirm **General BP Cache Statistics** wird angezeigt.

Die auf dem Bildschirm **General BP Cache Statistics** angezeigten Statistiken sind Momentaufnahmen des Buffer Pool, die jedes Mal, wenn Sie ENTER drücken, aktualisiert werden. Die folgenden Informationen werden angezeigt:

Feld	Erläuterung
Dataspace - Name	Der Name des Datenraums, in dem sich der BP Cache resident befindet.
Dataspace - SToken	Der Begriff SToken (für Space Token) kennzeichnet einen Datenbereich.
Dataspace - ALET	Der Begriff ALET (für Address List Entry Token) kennzeichnet einen Index für den Zugriff auf den Datenbereich.
Dataspace - Size (MB)	Die Größe des BP Cache in MB.
Dataspace - Current state	Der Status des BP Cache: `not initialized` nicht initialisiert `locked for init` gesperrt für Initialisierung `closed` geschlossen `free for operation` frei für den Betrieb `undefined` nicht definiert
Dataspace - Initialization	Das Datum und die Uhrzeit, wann der BP Cache initialisiert wurde.
Internal buffer offsets - Header buffer	Der Header des BP Cache, der allgemeine BP Cache-Informationen enthält.
Internal buffer offsets - Hash buffer	Enthält die Hash-Tabelle (siehe auch *BP Cache Hash Table-Statistik*).
Internal buffer offsets - Directory buffer	Die Adresse des Verzeichnisabschnitts des BP Cache relativ zum Anfang des BP Cache. Jedes in den BP Cache geladene Natural-Objekt erfordert einen Verzeichniseintrag, der Informationen über dieses Objekt enthält. Der Platz für diese Verzeichniseinträge wird aus dem BP Cache selbst bezogen.
Internal buffer offsets - Text buffer	Die Adresse des Text Buffers relativ zum Anfang des BP Cache. Nach Zuordnung des Platzes für alle anderen Puffer wird der verbleibende Platz in Textsätze mit einer Größe von 4 KB aufgeteilt. Ein Objekt kann je nach seiner Größe einen oder mehrere Textsätze belegen.
Tot. Text Records	Die Gesamtzahl der Textsätze im BP Cache. Die Anzahl der Textsätze hängt von der Größe des BP Cache ab. Die Textsatzgröße für den BP Cache beträgt 4 KB.
Insert position	Die Indexnummer des Textsatzes, in den das nächste Objekt eingefügt werden soll. Objekte werden in den BP Cache eingefügt, wenn sie aus dem Buffer Pool entfernt werden müssen.
Reuse cycles	Die Anzahl der vollständigen Wiederverwendungen des BP Cache. Jedes Mal, wenn der BP Cache voll ist, verwendet der BP Cache Manager den BP Cache von Anfang an wieder und überschreibt das/die Objekt(e) ab dort. Die Objekte verbleiben im BP Cache, bis der BP Cache erneut verwendet wird.

Feld	Erläuterung
Objects - Max Loaded	Die maximale Anzahl der zurzeit in den BP Cache geladenen Objekte.
Objects - Loaded	Die Anzahl der zurzeit in den GP-Cache geladenen Objekte.

BP Cache Call-Statistiken

Die Funktion **BP Cache Call Statistics** liefert statistische Informationen über das Laden (put), Abrufen (get) und Löschen von Objekten in/aus dem BP Cache. Diese Informationen dienen auch als Anhaltspunkte für die Performance des BP Cache.

Um die Funktion aufzurufen:

- Geben Sie im Menü **BP Cache Statistics** den Funktionscode `L` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY CLOAD
  ```

  Der Bildschirm **BP Cache Call Statistics** wird angezeigt.

Die auf dem Bildschirm **BP Cache Call Statistics** angezeigten Statistiken sind Momentaufnahmen des Buffer Pool, die jedes Mal aktualisiert werden, wenn Sie ENTER drücken. Die folgenden Informationen werden angezeigt:

Feld	Erläuterung
Search calls	Die Anzahl der Suchaufrufe, die der Buffer Pool an den BP Cache gesendet hat, während er versucht hat, ein Objekt im BP Cache zu finden. Wenn ein Objekt gefunden wird, führt ein Search-Aufruf zu einem Get-Aufruf.
Get calls (from BP cache)	Die Anzahl der Get-Aufrufe, die der Buffer Pool an den BP Cache gesendet hat, während er versucht hat, ein Objekt aus dem BP Cache in den Buffer Pool zu laden.
Get calls - successful	Die Anzahl der erfolgreichen Get-Aufrufe, die der BP Cache durchgeführt hat, d.h. die Anzahl der Objekte, die der BP Cache in den Buffer Pool ausgelagert hat. Ein Get-Aufruf ist erfolgreich, wenn ein Objekt, das der Buffer Pool zu laden versucht hat, tatsächlich aus dem BP Cache in den Buffer Pool geladen wurde. Ein Get-Aufruf ist nicht erfolgreich, wenn ein Objekt beispielsweise gelöscht wurde, nachdem es durch den Search-Aufruf gefunden wurde.

Feld	Erläuterung
Put calls (to BP cache)	Die Anzahl der Put-Aufrufe, die der Buffer Pool an den BP Cache gesendet hat, während er versucht hat, ein Objekt aus dem Buffer Pool in den BP Cache auszulagern.
Put calls - successful	Die Anzahl der Put-Aufrufe, die dazu führten, dass ein Objekt aus dem Buffer Pool in den BP Cache ausgelagert wurde.
Put calls - obj. already cached	Die Anzahl der Put-Aufrufe, die der Buffer Pool an den BP Cache für Objekte gesendet hat, die bereits in den BP Cache geladen waren.
Delete calls	Die Anzahl der Delete-Aufrufe, die der Buffer Pool an den BP Cache gesendet hat, während versucht wurde, ein Objekt aus dem BP-Cache zu löschen. Ein Delete-Aufruf fordert entweder ein einzelnes Objekt oder einen Bereich von Objekten an (siehe auch den Abschnitt ***Objekte löschen***).
Delete calls - successful	Die Anzahl der erfolgreichen Löschaufrufe, die der Buffer Pool an den BP Cache gesendet hat. Ein Löschaufruf ist erfolgreich, wenn mindestens ein Objekt tatsächlich aus dem BP Cache gelöscht wird. Ein Löschaufruf ist erfolglos, wenn das angeforderte Objekt nicht in den BP Cache geladen wurde und daher nicht gelöscht werden konnte. Verglichen mit der Gesamtzahl der Löschaufrufe kann die Anzahl der erfolgreichen Löschaufrufe sehr gering sein. Dies ist z. B. der Fall, wenn mehrere Natural-Objekte mit dem `CATALL`-Kommando katalogisiert werden. In diesem Fall sendet Natural für jedes erfolgreich katalogisierte Objekt einen Löschaufruf an den BP Cache. Zum Zeitpunkt des Löschaufrufs sind jedoch die meisten katalogisierten Objekte in der Regel noch nicht in den BP Cache geladen und der Löschversuch schlägt fehl.
Nbr objects deleted - by roll-over	Die Anzahl der Natural-Objekte, die aufgrund eines vollen Cache gelöscht wurden, bevor neue Objekte geladen werden konnten.
Nbr objects deleted - by command	Die Anzahl der Natural-Objekte, die alle durch die SYSBPM-Funktion oder durch Natural-Kommandos bzw. Dienstprogramme wie `CATALOG`, `STOW`, `CATALL` bzw. SYSMAIN gelöscht wurden. **Anmerkung:** Sie können mehrere Objekte per Delete-Aufruf löschen.
Initialization	Das Datum und die Uhrzeit, wann der BP Cache initialisiert wurde.
Last reuse cycle	Das Ladedatum und die Uhrzeit des Objekts, das zuletzt überschrieben wurde. Ein Objekt wird im BP Cache überschrieben, wenn sein Platz wiederverwendet werden muss, um ein anderes Objekt zu laden. Das Objekt, das zuerst in den BP Cache geladen wurde, wird zuerst ausgelagert. Das bedeutet, dass das Datum und die Uhrzeit des Ladens des Objekts, das sich am längsten im BP Cache befindet, dem Datum und der Uhrzeit des letzten **Last reuse cycle** (Wiederverwendungszyklus) entspricht.
Last access	Datum und Uhrzeit des letzten Zugriffs des Buffer Pool auf den BP Cache.

Feld	Erläuterung
Last Put (to BP cache)	Datum und Uhrzeit, wann der Buffer Pool zuletzt einen Put-Aufruf an den BP Cache gesendet hat.
Last Get (from BP cache)	Datum und Uhrzeit, wann der Buffer Pool zuletzt einen Get-Aufruf an den BP Cache gesendet hat.
Last Delete	Datum und Uhrzeit, wann der Buffer Pool zuletzt einen Delete-Aufruf an den BP Cache gesendet hat.

BP Cache Hash Table-Statistiken anzeigen

Die Funktion **BP Cache Hash Table Statistics** zeigt Statistiken über Hash-Tabellen-Slots und Kollisionen pro Slot an. Die Statistiken ermöglichen die Bestimmung der Effizienz des verwendeten Hash-Algorithmus.

Um die Funktion aufzurufen:

- Geben Sie im Menü **BP Cache Statistics** den Funktionscode `H` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY CHASH
  ```

  Der Bildschirm **Cache Hash Table Collision Statistics** wird angezeigt.

Die dort angezeigte Statistik ist eine Momentaufnahme der Hash-Tabelle, die jedes Mal, wenn Sie ENTER drücken, aktualisiert wird. Die folgenden Informationen werden angezeigt:

Feld	Erläuterung
Total Number of Slots	Die Gesamtzahl der Slots in der Hash-Tabelle, d. h. die Gesamtzahl der möglichen Einträge, die den Objektnamen mit dem Ort (Location) des Objekts verbinden. Die Anzahl der Slots, d. h. die Größe der Hash-Tabelle, wird intern in Abhängigkeit von der Anzahl der Textsätze berechnet.
Number of Slots used	Die Anzahl der Slots, die einen oder mehrere Einträge enthalten.
Number of Slots free	Die Anzahl der Slots, die keinen Eintrag haben.
Max. Collisions per Slot	Die maximale Anzahl der Kollisionen aller Slots. Die maximale Anzahl der Kollisionen ist der längste mögliche Suchpfad für ein Objekt.

Feld	Erläuterung
Collisions	Die Anzahl der möglichen Kollisionen. 0 (Null) bedeutet keine Kollision oder ein Eintrag. Bei mehr als 5 Kollisionen wird die Anzahl der Kollisionen in Bereichen angegeben (z. B. 6 - 10).
Number of Slots	Die Anzahl der Slots, gruppiert nach Anzahl der Kollisionen. Beträgt die Anzahl der Kollisionen z. B. 3, muss der Suchalgorithmus maximal 3 Seitwärtsschritte machen, um ein Objekt zu finden. Außerdem wird der prozentuale Anteil dieser Slots im Verhältnis zu allen verwendeten Slots angezeigt.
Number of Slots Totaled	Die gleichen Werte wie **Number of Slots** aber die Werte werden summiert.

Performance-Hinweise

Siehe ***Performance-Hinweise*** im Abschnitt *Buffer Pool-Statistiken anzeigen*.

PF-Tasten und Direktkommandos für die BP Cache-Statistiken

Auf den BP Cache Statistics-Bildschirmen können Sie die in der folgenden Tabelle aufgeführten PF-Tasten oder SYSBPM-Direktkommandos verwenden. Ein unterstrichener Teil eines Kommandos steht für seine Mindestabkürzung. Weitere Kommandos finden Sie unter ***SYSBPM-Direktkommandos***.

PF-Taste	Kommando	Funktion
PF1		Liefert SYSBPM-Hilfeinformationen: siehe auch ***Online-Hilfe***.
PF3	EXIT	Verlassen der aktuellen Funktion/des aktuellen Bildschirms und Anzeige des vorherigen Bildschirms.
PF4	LAST	Anzeige des zuletzt eingegebenen SYSBPM-Direktkommandos.
PF6	FLIP	Schaltet die PF-Tastenzeile um: Wechsel zwischen der Anzeige von PF1 bis PF12 und PF13 bis PF24.
PF8 (CLoad)	DISPLAY CLOAD	Gilt nur für den Bildschirm **General BP Cache Statistics**. Anzeige des Bildschirms **BP Cache Call Statistics**.
PF8 (CGen)	DISPLAY CGENERAL	Gilt nur für den Bildschirm **BP Cache Call Statistics**. Anzeige des Bildschirms **General BP Cache Statistics**.
PF12	CANCEL	Wie EXIT.
PF15	MENU	Aufruf des SYSBPM-Hauptmenüs (**Main Menu**).

32 Message Buffer Pool-Statistiken anzeigen

Die Funktion **Message Buffer Pool Statistics** ruft das gleichnamige Menü auf, in dem Sie Statistiken über den Message Buffer Pool abrufen können.

Informationen über den Message Buffer Pool siehe *Message Buffer Pool verwenden* in der *Operations*-Dokumentation.

Um die Funktion Message Buffer Pool Statistics aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `M` ein:

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY MSTATISTICS
  ```

  Der Bildschirm **Message Buffer Pool Statistics** wird angezeigt. Die dort angezeigten Statistiken sind Momentaufnahmen des Message Pool, die jedes Mal, wenn Sie ENTER drücken, aktualisiert werden.

Die folgenden Informationen werden angezeigt:

Feld	Erläuterung
Message Pool Size (MB)	Die Größe des gesamten Message Pool in MB. Die Größe des Message Pool kann mit dem Makro `NTBPI` im Parametermodul oder mit dem Profilparameter `BPI` festgelegt werden. Siehe *Parameter-Referenz*-Dokumentation.
Available text entries	Die maximale Anzahl von Texten, die der Message Pool enthalten kann.
Free text entries	Anzahl der zusätzlichen Texte, die in den Message Pool geladen werden können.
Start date	Das Datum, wann der der Message Pool ursprünglich gestartet wurde.
Start time	Die Uhrzeit, wann der der Message Pool ursprünglich gestartet wurde.

Feld	Erläuterung
Full conditions	Anzahl der Fälle, in denen eine Meldung nicht in den Message Pool aufgenommen werden konnte, weil keine freien Texteinträge mehr vorhanden waren.
Total number of locates	Die Gesamtzahl der Abrufe von Meldungen aus dem Message Pool.
Successful direct locates	Die Anzahl der Male, in denen eine Meldung ohne Verwendung der Hash-Tabelle abgerufen werden konnte.
Collisions high water	Die maximale Anzahl von Kollisionen. Mit anderen Worten, die maximale Anzahl von Texten, die denselben Eintrag in der Hash-Tabelle verwenden.

33 Buffer Pool auswählen

- Funktion Select Buffer Pool aufrufen 244
- Buffer Pools anzeigen 245
- Buffer Pool zurücksetzen 245

SYSBPM bietet Funktionen zum Anzeigen und Verwalten der für Ihre Natural-Systemumgebung definierten Buffer Pools (einschließlich des BP Cache). Ein Buffer Pool, der in der aktuellen Natural-Sitzung beim Start nicht definiert ist, muss ausgewählt werden, bevor er verwaltet werden kann.

Funktion Select Buffer Pool aufrufen

Um die Funktion Select Buffer Pool aufzurufen:

1 Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Code `S` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
SELECT BP
```

Das Fenster **Select Buffer Pool** erscheint mit den folgenden Informationen zu Ihrem aktuellen Buffer Pool (global oder lokal) und zu allen anderen globalen Buffer Pools, die zurzeit in Ihrer Natural-Systemumgebung verfügbar sind:

BPNAME	Der Name des Buffer Pool.
Type	Der Typ des Buffer Pool: `Global Nat`, `Local Nat`, `Global Sort`, `Global DL/I`, `Edit` (Editor) oder `Mon` (Monitor).
Status	Der aktuelle Status.
Preload	Der Name der Preload-Liste (falls geladen).
Address	Die Adresse des Buffer Pool.
Loc	Der Standort des Buffer Pool, angegeben durch **HostID** und **SubsID** (Subsystem ID).

Weitere Informationen zu den Feldern siehe *SYSBPM Hauptmenü - Felder, Funktionen und Kommandos*.

2 Geben Sie in der Spalte **C** ein beliebiges Zeichen vor dem Buffer Pool ein, den Sie auswählen möchten, und drücken Sie ENTER.

Beachten Sie, dass nur Puffer Pools des Typs `Natural`, `DL/I` oder `Sort` ausgewählt werden können. Buffer Pools eines anderen Typs können nur angezeigt werden.

Der angegebene Buffer Pool ist nun in Ihrer aktuellen Sitzung definiert und der Inhalt der Buffer Pool-relevanten Felder (wie oben erwähnt), die oben im SYSBPM-Hauptmenü (**Main Menu**) angezeigt werden, wird entsprechend geändert.

Sobald Sie einen Buffer Pool aus dem Fenster **Select Buffer Pool** ausgewählt haben, gelten alle SYSBPM-Funktionen für diesen Buffer Pool. Ihre Natural-Sitzung selbst wird jedoch weiterhin mit dem Buffer Pool ausgeführt, der beim Start der Sitzung verwendet wurde.

Buffer Pools anzeigen

Um die in Ihrem Natural-System verfügbaren Buffer Pools anzuzeigen:

- Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY BUFFERPOOL
  ```

  Oder:

  Führen Sie den **ersten Schritt** der Anleitung unter ***Funktion Select Buffer Pool aufrufen*** aus.

  Es erscheint das Fenster **Display Buffer Pools** , das dieselben Informationen enthält wie das Fenster **Select Buffer Pool**.

Buffer Pool zurücksetzen

Um einen Buffer Pool zurückzusetzen:

- Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  RESET BUFFERPOOL
  ```

  SYSBPM schaltet auf den Buffer Pool zurück, der ursprünglich für Ihre aktuelle Natural-Sitzung definiert wurde, und der Inhalt der Buffer Pool-relevanten Felder (wie unter ***Funktion Select Buffer Pool aufrufen*** erwähnt), die oben im SYSBPM-Hauptmenü (**Main Menu**) angezeigt werden, ändert sich entsprechend.

  Oder:

  Rufen Sie die Funktion **Select Buffer Pool** auf (wie zuvor beschrieben) und wählen Sie den Startup-Buffer Pool aus.

34 Message Buffer Pool auswählen

- Funktion Select Message Pool aufrufen 248
- Message Pools anzeigen 249
- Message Pool zurücksetzen 249

Informationen über den Message Buffer Pool siehe *Message Buffer Pool verwenden* in der *Operations*-Dokumentation.

Funktion Select Message Pool aufrufen

Um die Funktion Select Message Pool aufzurufen:

1 Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) im Feld **Object Pool** ein `M` und im Feld **Code** ein `S` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
SELECT MP
```

Es erscheint das Fenster **Select Message Pool** mit den folgenden Informationen zu Ihrem aktuellen Message Pool und zu allen anderen Message Pools, die derzeit in Ihrer Natural-Systemumgebung verfügbar sind:

BPNAME	Der Name des Message Pool.
Type	`MSG`, der Typ für Message Pool.
Preload	Der Name der Preload-Liste (falls geladen).
ALET	Der Begriff ALET (für Address List Entry Token) kennzeichnet einen Index für den Zugriff auf den Datenraum.
Loc	Der Standort des Buffer Pool, angegeben durch **HostID** und **SubsID** (Subsystem ID).

Weitere Informationen zu den Feldern siehe *SYSBPM Hauptmenü - Felder, Funktionen und Kommandos*.

2 Geben Sie in der Spalte **C** ein beliebiges Zeichen vor dem Message Pool ein, den Sie auswählen möchten, und drücken Sie ENTER. Der angegebene Message Pool ist nun in Ihrer aktuellen Sitzung definiert und der Inhalt der Message Pool-relevanten Felder (wie oben erwähnt), die oben auf den SYSBPM-Bildschirmen angezeigt werden, wird entsprechend geändert.

Beachten Sie, dass nur die Funktionen **L** (**List Objects**), **D** (**Delete Objects**) und **M** (**Message Pool Statistics**) oder das entsprechende Direktkommando verfügbar sind.

Sobald Sie einen Message Pool im Fenster **Select Message Pool** ausgewählt haben, gelten alle SYSBPM-Funktionen für diesen Message Pool. Ihre Natural-Sitzung selbst wird jedoch weiterhin mit dem Message Pool ausgeführt, der beim Start der Sitzung verwendet wurde.

Message Pools anzeigen

Um die in Ihrer Natural-Systemumgebung verfügbaren Message Pools anzuzeigen:

- Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  DISPLAY MP
  ```

  Oder:

  Führen Sie den **ersten Schritt** der Anleitung unter *Funktion Select Message Pool aufrufen* aus

  Es erscheint das Fenster **Display Message Pool**, das die gleichen Informationen enthält wie das Fenster **Select Message Pool**.

Message Pool zurücksetzen

Um einen Message Pool zurückzusetzen:

- Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  RESET MP
  ```

  SYSBPM schaltet auf den ursprünglich für Ihre aktuelle Natural-Sitzung definierten Message Pool zurück und der Inhalt der Message Pool-relevanten Felder (wie unter *Funktion Select Message Pool aufrufen*), die oben auf den SYSBPM-Bildschirmen angezeigt werden, wird entsprechend geändert.

  Oder:

  Rufen Sie die Funktion **Select Message Pool** (wie oben beschrieben) auf und wählen Sie den Startup-Message Pool aus.

35 Sperrliste verwalten

- Funktion Maintain Blacklist aufrufen 252
- Objektsets auflisten 256
- Objektset bearbeiten 257
- Objektset zur Sperrliste hinzufügen 259
- Objektset aus Sperrliste löschen 260
- Quellobjekt des Objektsets löschen 261
- Zusätzliche Objektset-Verwaltung mittels Dienstprogrammen (Utilities) 261
- Sperrliste im Batch-Modus verwalten 262

Mit der Funktion **Blacklist Maintenance** können Sie eine Sperrliste für Natural-Objekte verwalten.

In einer Sperrliste können Sie die Natural-Objekte angeben, die nicht in den Buffer Pool geladen werden sollen. Objekte, die bereits in den Buffer Pool geladen sind, werden dann gelöscht. Wenn der BP Cache aktiviert ist, werden die Natural-Objekte auch aus dem BP Cache gelöscht. Die Sperrliste gilt immer für den gerade aktiven Buffer Pool.

In einer Sperrliste können Sie einzelne Natural-Objekte und/oder Libraries und Objektsets pflegen, die mehrere Natural-Objekte enthalten. In einem Objektset geben Sie die Objekte an, die nicht ausgeführt werden sollen, und fügen ein einzelnes Set (anstelle mehrerer einzelner Objekte) zur Sperrliste hinzu. Sie können auch beides kombinieren: Objekte einzeln oder in Gruppen (Sets) pflegen.

Der Quellcode eines Objektsets wird als Natural-Quellcodeobjekt vom Typ `Text` in der aktuellen Natural-Library und -Systemdatei gespeichert. Die erste Zeile des Quellcodes eines Objektsets enthält den Kommentar `**BBL**` (Buffer Pool Blacklist).

Weitere Informationen über die Sperrliste (Blacklist) finden Sie in dem entsprechenden Abschnitt in *Natural Buffer Pool* in der *Operations*-Dokumentation.

Um die Funktion Blacklist Maintenance aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Funktionscode `B` ein.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  BLACKLIST
  ```

  Das Menü **Blacklist Maintenance** erscheint.

Die Funktionen im Menü **Blacklist Maintenance** werden im folgenden Abschnitt erläutert. Dort finden Sie auch Informationen über weitere Funktionen zur Pflege der Sperrliste.

Funktion Maintain Blacklist aufrufen

Mit der Funktion **Maintain Blacklist** rufen Sie den gleichnamigen Bildschirm auf, auf dem Sie alle zurzeit in der Sperrliste vorhandenen Natural-Objekte anzeigen und pflegen können.

Um die Funktion Maintain Blacklist aufzurufen:

- Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `M` ein.

  Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY BLACKLIST
```

Es erscheint der Bildschirm **Maintain Blacklist** mit der aktuellen Sperrliste.

Drücken Sie PF7, um eine Seite rückwärts zu blättern, und PF8, um eine Seite vorwärts zu blättern.

Je nach dem Modus, der bei einem früheren Aufruf der Funktion **Maintain Blacklist** während einer SYSBPM-Sitzung eingestellt wurde, erscheint der Bildschirm **Maintain Blacklist** im **Display**-Modus (Anzeigen gemäß Voreinstellung bei der Initialisierung von SYSBPM) oder im **Add**-Modus (Hinzufügen).

Benutzen Sie PF9, um von einem Modus in den anderen zu wechseln.

Dieser Abschnitt behandelt die folgenden Themen:

- Objekte zur Sperrliste hinzufügen
- Objekte in der Sperrliste ändern
- Objekte in der Sperrliste löschen

Objekte zur Sperrliste hinzufügen

Um Objekte zur Sperrliste hinzuzufügen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode M ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY BLACKLIST
```

Der Bildschirm **Maintain Blacklist Display Mode** erscheint.

2 Falls erforderlich, können Sie Durch Drücken von PF9 in den **Add Mode** (Hinzufügen-Modus) wechseln.

Der Bildschirm **Maintain Blacklist Add Mode** erscheint mit leeren Eingabefeldern.

3 Geben Sie in die entsprechenden Felder den Namen der Library, in der die Objekte gespeichert sind, die Namen der Objekte und die entsprechenden Datenbankkennungen (DBID) und Dateinummern (FNR) ein. Wenn Sie DBID und FNR leer lassen, werden sie aus Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), aus der aktuellen Systemdatei FUSER oder FNAT übernommen.

Wenn Sie den Inhalt des Bildschirms **Maintain Blacklist Add Mode** löschen müssen, geben Sie in der Kommandozeile eines der folgenden Kommandos ein:

```
CLE
```

oder

```
CLEAR
```

4 Drücken Sie PF5, um das Hinzufügen zu bestätigen.

Oder:

Geben Sie in der Kommandozeile eines der folgenden Kommandos ein:

```
UP
```

oder

```
UPDATE
```

Es erscheint eine entsprechende Meldung.

Objekte in der Sperrliste ändern

Um Objekte in der Sperrliste zu ändern:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `M` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY BLACKLIST
```

Der Bildschirm **Maintain Blacklist** erscheint.

2 Falls erforderlich, können Sie durch Drücken von PF9 in den **Display Mode** (Anzeigen-Modus) wechseln, um eine Liste die Liste aller zurzeit in der Sperrliste enthaltenen Objekte zu erhalten.

3 Ersetzen Sie in dem/den entsprechenden Eingabefeld(ern) die bestehenden Einträge durch neue Werte.

4 Drücken Sie PF5, um die Änderung zu bestätigen.

Oder:

Geben Sie in der Kommandozeile eines der folgenden Kommandos ein:

```
UP
```

oder

```
UPDATE
```

Es erscheint eine entsprechende Meldung.

Objekte in der Sperrliste löschen

Um Objekte in der Sperrliste zu löschen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `M` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY BLACKLIST
```

Der Bildschirm **Maintain Blacklist** erscheint.

2 Falls erforderlich, können Sie durch Drücken von PF9 in den **Display Mode** (Anzeigen-Modus) wechseln, um eine Liste die Liste aller zurzeit in der Sperrliste enthaltenen Objekte zu erhalten.

3 Geben Sie in der Spalte **C**, neben dem/den gewünschten Objekt(en), das folgende Kommando ein:

```
DE
```

4 Drücken Sie ENTER, um das Löschen zu bestätigen.

Es erscheint eine entsprechende Meldung.

Um alle Objekte in der Sperrliste zu löschen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `M` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DISPLAY BLACKLIST
```

Der Bildschirm **Maintain Blacklist** erscheint.

2 Drücken Sie PF2 (DeAll).

Das Fenster **Confirm Delete** (Löschen bestätigen) erscheint.

3 Bestätigen Sie die Löschung, indem Sie `Y` (Ja) eingeben.

Oder:

Brechen Sie die Löschung ab, indem Sie `N` (Nein, dies ist der Standardwert) eingeben oder PF3 drücken, ohne etwas in das Fenster einzugeben.

4 Drücken Sie ENTER, um die Aktion zu bestätigen.

Es erscheint eine entsprechende Meldung.

Objektsets auflisten

Die Funktion **List Object Sets** ruft den gleichnamigen Bildschirm auf, der eine Liste aller vorhandenen Objektsets anzeigt.

Um den Bildschirm List Object Sets aufzurufen:

- Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `L`, einen Library-Namen und einen Objektset-Namen ein.

  Im Feld **Object Set** ist auch Stern-Notation (*) ist zulässig.

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

  ```
  LIST SET library-name set-name
  ```

  Bei *set-name* ist auch Stern-Notation (*) zulässig.

  Der Bildschirm **List Object Sets** erscheint mit dem/den angegebenen Set(s).

Vom Bildschirm **List Object Sets** aus können Sie ein Objektset mit einem der angebotenen Zeilenkommandos bearbeiten, um ein Set zu ändern und es zur Sperrliste hinzuzufügen oder daraus zu löschen. Eine Liste der möglichen Kommandos erhalten Sie, wenn Sie ein Fragezeichen (?) in die linke Spalte des Bildschirms eingeben (Präfixinformationen).

Eine Liste der Kommandos, die in die Kommandozeile des Bildschirms List Objects Sets eingegeben werden können, erhalten Sie, wenn Sie ein Fragezeichen (?) in die Kommandozeile eingeben und somit das Hilfefenster aufrufen.

Drücken Sie bei Bedarf PF7, um rückwärts zu blättern, oder PF8, um im Fenster vorwärts zu blättern.

Objektset bearbeiten

Mit der Funktion **Edit Object Set** rufen Sie das gleichnamige Fenster Edit auf, in dem Sie ein neues Objektset erstellen, Objekte zu einem bestehenden Set hinzufügen oder ändern oder Objekte aus einem Set löschen können.

Die Bearbeitungsfunktionen auf dem Bildschirm **Edit Object Set** sind eine Teilmenge der Funktionen des Software AG Editors (siehe *Editoren*-Dokumentation).

Eine Liste der verfügbaren Zeilenkommandos erhalten Sie, wenn Sie ein Fragezeichen (`?`) in die linke Spalte des Bildschirms eingeben (Präfixinformationen).

Eine Liste der Kommandos, die in der Kommandozeile des Bildschirms **Edit Object Set** eingegeben werden können, erhalten Sie, indem Sie ein Fragezeichen (`?`) in die Kommandozeile eingeben.

Drücken Sie bei Bedarf PF7, um rückwärts zu blättern, und PF8, um im Fenster vorwärts zu blättern.

Dieser Abschnitt behandelt die folgenden Themen:

- Objektsets anlegen
- Objektsets ändern

Objektsets anlegen

Um Objektsets anzulegen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `E` und einen Library-Namen ein. Geben Sie *nicht* den Namen eines Objektsets ein, sondern löschen Sie den Inhalt (falls vorhanden) des entsprechenden Feldes.

Der Bildschirm **Edit Object Set** erscheint.

2 Geben Sie in die entsprechenden Eingabefelder den Namen der Library, in der die Objekte gespeichert sind, die Namen der Objekte sowie die entsprechenden Datenbankkennungen (DBID) und Dateinummern (FNR) ein.

Wenn DBID und FNR leer gelassen werden, werden sie in Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), aus der aktuellen Systemdatei FUSER bzw. FNAT übernommen.

3 Um das Objektset als Quellcodeobjekt vom Typ `Text` zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Object Set** folgendes ein:

```
SA set-name
```

Objektsets ändern

Dieser Abschnitt beschreibt das Hinzufügen von Objekten zu einem Objektset, das Ändern bestehender Objekte und das Löschen von Objekten aus einem Objektset. Beachten Sie, dass die aktuelle Sperrliste durch diese Objektset-Änderungen *nicht* aktualisiert wird.

Um ein neues Objekt zu einem Objektset hinzuzufügen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `E`, einen Library-Namen und einen Objektset-Namen ein.

Oder:

Geben Sie im Bildschirm **List Object Sets** in der Spalte ganz links neben dem betreffenden Objektset das folgenden Zeilenkommando `E` ein:

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
EDIT SET library-name set-name
```

Es erscheint der Bildschirm **Edit Object Set** mit dem angegebenen Objektset.

2 Füllen Sie die Eingabefelder aus, indem Sie den Namen der Library, in der die Objekte gespeichert sind, die Namen der Objekte sowie die entsprechenden Datenbankkennungen (DBID) und Dateinummern (FNR) eingeben.

Wenn DBID und FNR leer gelassen werden, werden sie in Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), aus der aktuellen Systemdatei FUSER oder FNAT übernommen.

3 Um die Änderungen zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Object Set** das Kommando `SA` ein:

Um ein Objekt eines Objektsets zu ändern:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `E`, einen Library-Namen und einen Objektset-Namen ein.

Oder:

Geben Sie im Bildschirm **List Object Sets** in der Spalte ganz links neben dem betreffenden Objektset das Zeilenkommando `E` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
EDIT SET library-name set-name
```

Es erscheint der Bildschirm **Edit Object Set** mit dem angegebenen Objektset.

2 Ersetzen Sie in den entsprechenden Eingabefeldern die bestehenden Einträge durch neue Werte.

3 Um die Änderungen zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Object Set** das Kommando `SA` ein:

Um ein Objekt aus einem Objektset zu löschen:

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `E`, einen Library-Namen und einen Objektset-Namen ein.

Oder:

Geben Sie im Bildschirm **List Object Sets** in der Spalte ganz links neben dem/den betreffenden Objektset(s) das Zeilenkommando `E` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
EDIT SET library-name set-name
```

Es erscheint der Bildschirm **Edit Object Set** mit dem angegebenen Objektset.

2 Geben Sie im Bildschirm **List Object Sets** in der Spalte ganz links neben dem betreffenden Objektset das Zeilenkommando `D` ein.

Das angegebene Objekt wird aus dem Objektset entfernt.

3 Um die Änderungen zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Object Set** das Kommando `SA` ein:

Objektset zur Sperrliste hinzufügen

Mit der Funktion **Add Object Set to Blacklist** können Sie alle Objekte eines Objektsets in die Sperrliste aufnehmen.

Um einen Objektset zur Sperrliste hinzuzufügen

- Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `A`, einen Library-Namen und einen Objektset-Namen ein.

  Oder:

Geben Sie im Bildschirm **List Object Sets** in der Spalte ganz links bei dem/den erforderlichen Objektset(s) das Zeilenkommando `AC` ein.

Oder:

Geben Sie auf dem Bildschirm **Edit Object Set** in der Kommandozeile folgendes ein:

```
AC
```

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
ADD SET library-name set-name
```

Es erscheint eine Meldung, die bestätigt, dass das Objektset zur Sperrliste hinzugefügt wurde.

Anmerkung: Das Kommando `AC` steht für `ACTIVATE`, was **Add Object Set to Blacklist** entspricht.

Objektset aus Sperrliste löschen

Mit der Funktion **Delete Object Set from Blacklist** können Sie alle Objekte eines Objektsets aus der Sperrliste löschen. Beachten Sie, dass die Funktion nicht das Objektset als Quellcodeobjekt löscht. Die Objekte des Objektsets können, wie oben beschrieben, jederzeit wieder in die Sperrliste aufgenommen werden. Siehe auch *Quellobjekt des Objektsets löschen*.

Um einen Objektset aus Sperrliste zu löschen:

- Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `D`, einen Library-Namen und einen Objektset-Namen ein.

  Oder:

  Geben Sie auf dem Bildschirm **List Object Sets** in der Spalte ganz links neben dem/den gewünschten Objektset(s) das Zeilenkommando `DA` ein:

  Oder:

  Geben Sie auf dem Bildschirm **Edit Object Set** in der Kommandozeile folgendes Kommando ein:

  ```
  DA
  ```

  Oder:

  Geben Sie das folgende SYSBPM-Direktkommando ein:

```
DELETE SET library-name set-name
```

Es erscheint eine Meldung, die bestätigt, dass der Objektset aus der Sperrliste gelöscht wurde.

Anmerkung: Das Kommando `DA` steht für `DEACTIVATE`, was dem Kommando **Delete Object Set from Blacklist** entspricht.

Quellobjekt des Objektsets löschen

To delete the source object of an object set

1 Geben Sie im Menü **Blacklist Maintenance** den Funktionscode `L`, einen Library-Namen und einen Objektset-Namen ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

```
LIST SET library-name list-name
```

Der Bildschirm **List Object Sets** wird angezeigt.

2 Geben Sie in der Spalte ganz links neben dem zu löschenden Objektset das Zeilenkommando `D` ein.

Das Fenster **DELETE** erscheint.

3 Bestätigen Sie den Löschvorgang, indem Sie den Namen des Objektsets eingeben.

Es erscheint eine Bestätigungsmeldung.

Anmerkung: Durch das Löschen des Quellobjekts eines Objektsets wird die aktuelle Sperrliste *nicht* aktualisiert.

Zusätzliche Objektset-Verwaltung mittels Dienstprogrammen (Utilities)

Die Natural-Dienstprogramme SYSMAIN und Object Handler bieten zusätzliche Funktionen zur Pflege von Objektsätzen. Zu den Funktionen gehören die Übertragung von Objektsets zwischen verschiedenen Natural-Libraries und Systemdateien und/oder verschiedenen Großrechnerplattformen sowie das Löschen oder Auffinden von Objektsets in einer anderen Umgebung.

Bei der Verwendung eines Natural-Dienstprogramms wird ein Objektset wie jedes andere Quellobjekt vom Typ `Text` behandelt.

Einzelheiten hierzu finden Sie in den entsprechenden Abschnitten der Dokumentation über Dienstprogramme.

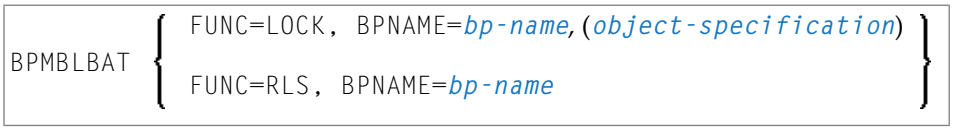
Sperrliste im Batch-Modus verwalten

Die SYSBPM-Funktionen zur Sperrlistenverwaltung können auch im Batch-Modus ausgeführt werden. Weitere Informationen siehe ***Batch-Verarbeitung***.

Darüber hinaus bietet die Natural-Systembibliothek SYSBPM das Kommando `BPMBLBAT`, das die Batch-Verarbeitung der Sperrlistenverwaltungsfunktionen **Add** (Hinzufügen eines Objekts) und **Delete** (Löschen aller Objekte) weiter erleichtert.

Melden Sie sich bei der Library SYSBPM an und führen Sie das Kommando `BPMBLBAT` aus, indem Sie die in dem folgenden Diagramm angegebene Syntax verwenden.

Die Bedeutung der im folgenden Diagramm verwendeten Symbole wird im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.



Dieser Abschnitt behandelt folgende Themen:

- Erläuterung der Syntax
- Eingabe-Beispiele

Erläuterung der Syntax

Die Schlüsselwörter und die *object-specification*-Klausel, die im obigen Syntax-Diagramm des Kommandos `BPMBLBAT` angegeben sind, werden im folgenden Abschnitt beschrieben. ***Beispiele für Eingaben*** veranschaulichen die Verwendung der Schlüsselwörter.

Syntax-Element	Erläuterung
`FUNC=LOCK`	Fügt Objektnamen zu einer Sperrliste hinzu, siehe ***Objekte hinzufügen*** im Abschnitt *Sperrliste verwalten*.
`FUNC=RLS`	Löscht eine komplette Sperrliste mit allen in der Liste enthaltenen Objektnamen. Diese Funktion entspricht der Löschfunktion, siehe unter ***Um alle Objekte aus der Sperrliste zu löschen*** im Abschnitt *Sperrliste verwalten*. Im Batch-Modus können Sie keine einzelnen Objektnamen aus einer Sperrliste löschen.
`BPNAME`	Der Name des Buffer Pool, in den die Sperrliste geladen wird.

Syntax-Element	Erläuterung	
object-specification	Die Schlüsselwörter, die für die *object-specification*-Klausel gelten (siehe die entsprechende Syntax unten), sind:	
	LIB	Der Name der Library, in der die Objekte gespeichert sind.
	DBID FNR	Die Datenbankkennung (DBID) und die Dateinummer (FNR), in der die Objekte gespeichert sind. Wenn DBID und FNR leer gelassen werden, werden sie in Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), der aktuellen Systemdatei FUSER oder FNAT entnommen.
	object-name	Name(n) des Objekts/der Objekte, das/die in die Sperrliste aufgenommen werden soll(en): Geben Sie jeden Namen in einer separaten Zeile an den Positionen 1 bis 8 ein. Um das Ende der Eingabe zu kennzeichnen, geben Sie in einer separaten Zeile einen Punkt (.) ein.

Syntax der *object-specification*-Klausel

Im Folgenden wird die Syntax dargestellt und erläutert, die für die *object-specification*-Klausel gilt:

```
LIB=lib-name, DBID dbid, FNR fnr [object-name]...
```

Eingabe-Beispiele

Beispiel 1 - Objekte zu einer Sperrliste hinzufügen

Das folgende Beispiel zeigt, wie in die Sperrliste im Buffer Pool NATGBP die Objekte A, B und C hinzugefügt wird:

```
/*Job
.
.
.
*/Job

LOGON SYSBPM
BPMBLBAT
FUNC=LOCK,BPNAME=NATGBP,LIB=SAGTEST,DBID=10,FNR=32
A
B
C
.
FIN
```

Beispiel 2 - Eine Sperrliste löschen

Das folgende Beispiel zeigt, wie eine Sperrliste im Buffer Pool `NATGBP` gelöscht wird:

```
/*Job
.
.
.
*/Job
LOGON SYSBPM
BPMBLBAT
FUNC=RLS,BPNAME=NATGBP
.
FIN
```

36 Preload-Liste verwalten

- Preload-Listen auflisten 266
- Preload-Liste bearbeiten 267
- Preload-Liste generieren 271
- Preload-Liste löschen 273
- Zusätzliche Verwaltungsfunktionen mit Dienstprogrammen 274

Die in diesem Kapitel beschriebene Funktion **Preload List Maintenance** gilt nur für Buffer Pools des Typs `Natural` und den Message Pool. Sie dient zum Verwalten von Preload-Listen. In einer Preload-Liste können Sie die Namen von Natural-Objekten oder von Natural-Meldungen angeben, die bei der Initialisierung in den Buffer Pool oder den Message Pool geladen werden sollen.

Der Quellcode einer Preload-Liste wird als Natural-Quellcode-Objekt vom Typ `Text` in der Natural System Library SYSBPM gespeichert. Die erste Zeile des Quelltextes einer Preload-Liste enthält den Kommentar `**BPL**` (Buffer Pool Preload-Liste) oder `**MPL**` (Message Pool Preload-Liste).

Weitere Informationen zur Preload-Liste siehe entsprechenden Abschnitt unter *Natural Buffer Pool* bzw. *Message Pool* in der *Operations*-Dokumentation.

Um die Funktion Preload List Maintenance aufzurufen:

- Geben Sie im SYSBPM-Hauptmenü (**Main Menu**) den Code `B`, `C` oder `*` ein, um einen Natural Buffer Pool auszuwählen, oder `M`, um einen Message Pool auszuwählen, sowie den Funktionscode `P`:

  Oder:

  Geben Sie eines der folgenden SYSBPM-Direktkommandos ein:

  `PRELOADLIST` - um das Menü für Natural-Objekte aufzurufen.

  bzw.

  `MPRELOADLIST` - um das Menü für Natural-Meldungen aufzurufen.

  Das Menü **Preload List Maintenance** erscheint.

Die Funktionen im Menü **Preload List Maintenance** werden im folgenden Abschnitt erläutert. In diesem Kapitel finden Sie außerdem Informationen zu weiteren **Preload List Maintenance**-Funktionen.

Preload-Listen auflisten

Die Funktion **List Preload Lists** ruft den gleichnamigen Bildschirm auf, der eine Liste aller vorhandenen Preload-Listen anzeigt.

Um die Funktion List Preload Lists aufzurufen:

- Geben Sie im Menü **Preload List Maintenance** den Funktionscode `L` und den Namen einer Preload-Liste ein.

Für den Namen einer Preload-Liste ist auch Stern-Notation (*) zulässig.

Oder:

Geben Sie eines der folgenden SYSBPM-Direktkommandos ein:

`LIST PRELOADLIST list-name` - um Preload-Listen mit Natural-Objekten aufzulisten

bzw.

`LIST MPRELOADLIST list-name` - um Preload-Listen aufzulisten, die Natural-Meldungen enthalten

Für *list-name* ist auch Stern-Notation (*) zulässig.

Der Bildschirm **List Preload Lists** wird angezeigt.

Um eine Liste der verfügbaren Zeilenkommandos zu erhalten, geben Sie ein Fragezeichen (?) in die linke Bildschirmspalte ein (Präfix-Information).

Das Hilfefenster **Help** mit einer Liste der Kommandos, die in der Kommandozeile des Bildschirms **List Preload Lists** eingegeben werden können, rufen Sie auf, indem Sie ein Fragezeichen (?) in die Kommandozeile eingeben.

Drücken Sie bei Bedarf PF7, um rückwärts zu blättern, und PF8, um im Fenster vorwärts zu blättern.

Preload-Liste bearbeiten

Die Funktion **Edit Preload List** ruft den gleichnamigen Bildschirm auf, in dem Sie eine neue Preload-Liste erstellen, Objekte zu einer bestehenden Liste hinzufügen oder Objekte aus ihr löschen können.

Wichtig: Die Bearbeitungsfunktionen auf dem Bildschirm **Edit Preload List** sind eine Teilmenge der Funktionen, die der Software AG Editor bietet (beschrieben in der Dokumentation dieses Editors). Setzen Sie daher vor dem Start einer Natural-Sitzung zur Bearbeitung einer Preload-Liste den Natural-Profilparameter `EDPSIZE` auf einen Wert größer als `0` (Null), siehe *Parameter-Referenz*-Dokumentation. Wir empfehlen, `EDPSIZE` auf einen Mindestwert von `100` zu setzen.

Eine Liste der verfügbaren Zeilenkommandos erhalten Sie, wenn Sie in der linken Bildschirmspalte ein Fragezeichen (`?`) eingeben (Präfix-Information).

Eine Liste der Kommandos, die in die Kommandozeile des Bildschirms **Edit Preload List** eingegeben werden können, erhalten Sie, indem Sie ein Fragezeichen (`?`) in die Kommandozeile eingeben.

Drücken Sie bei Bedarf PF7, um rückwärts zu blättern, und PF8, um im Fenster vorwärts zu blättern.

Dieser Abschnitt behandelt die folgenden Themen:

- Preload-Listen anlegen
- Preload-Listen ändern

Preload-Listen anlegen

Um eine Preload-Liste anzulegen:

1 Führen Sie im Menü **Preload List Maintenance** die folgenden Schritte aus:

- Geben Sie den Funktionscode `E` ein.
- Löschen Sie den Inhalt des Feldes **Preload List Name**, d.h. geben Sie *keinen* Namen für eine Preload-Liste ein.
- Belassen Sie in den Feldern **Library** und **Objects** den standardmäßig vorgeschlagenen Stern (`*`).

Der Bildschirm **Edit Preload List** erscheint.

2
- **Bei Natural-Objekten:**
  Geben Sie in den entsprechenden Eingabefeldern den Namen der Library, in der die Objekte gespeichert sind, die Namen der Objekte und die entsprechenden Datenbankkennungen (**DBID**) und Dateinummern (**FNR**) ein.

  Werden **DBID** und **FNR** nicht angegeben, wird in Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), die aktuelle Systemdatei FUSER bzw. FNAT als Voreinstellung verwendet.

  Das Kennzeichen für resident wird in der Spalte **R** auf dem Bearbeitungsbildschirm auf `Y` (Ja) gesetzt, wenn kein Wert eingegeben wird. Resident bedeutet, dass das Objekt nicht aus dem Buffer Pool gelöscht wird, auch nicht, wenn sich sein **Use Count** auf `0` (Null) ändert. (**Use Count** entspricht dem im Kapitel *Verzeichnisinformationen anzeigen* beschriebenen **Current Use Count**).
- **Bei Meldungen:**
  Geben Sie in die entsprechenden Eingabefelder den Namen der Library ein, in der die Benutzermeldungen gespeichert sind (ein Leerzeichen für Natural-Systemmeldungen), die Nummern der Meldungen (dieses Feld darf nicht leer gelassen werden), die entsprechenden Datenbankkennungen (**DBID**) und Dateinummern (**FNR**), die Codepage und den Sprachcode (**Language Code**).

  Wenn **DBID** und **FNR** nicht angegeben werden, werden beim Preload-Vorgang durch die aktuelle Systemdatei FUSER oder FNAT in Libraries, deren Namen mit SYS beginnen (außer der Library SYSTEM), Standardwerte bereitgestellt. Wenn in FUSER oder FNAT keine Meldungen gefunden werden, wird die Suche auf Steplibs ausgedehnt.

Wenn die Codepage nicht angegeben ist, wird während des Preload-Vorgangs eine Standard-Codepage anhand der Session-Codepage bereitgestellt.

Wenn der Sprachcode nicht angegeben ist, wird während des Preload-Vorgangs eine Standardsprache aus der Systemvariablen `*LANGUAGE` bereitgestellt.

3 Um die Preload-Liste als Quellcodeobjekt vom Typ `Text` in der Library SYSBPM zu speichern, geben Sie das folgende Kommando in die Kommandozeile ein:

```
SA set-name
```

Siehe auch *Preload-Liste aus Buffer Pool generieren*.

Preload-Listen ändern

Um ein neues Objekt zu einer Preload-Liste hinzuzufügen:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `E` und den Namen einer Preload-Liste ein.

Oder:

Geben Sie auf dem Bildschirm **List Preload Lists** in der linken Spalte neben der gewünschten Preload-Liste das Zeilenkommando `E` ein:

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

`EDIT PRELOADLIST list-name` - für eine Preload-Liste mit Natural-Objekten

oder

`EDIT MPRELOADLIST list-name` - für eine Preload-Liste mit Meldungen.

Es erscheint der Bildschirm **Edit Preload List** mit der angegebenen Preload-Liste.

2 Füllen Sie die Eingabefelder wie in **Schritt 2** unter *Preload-Listen anlegen* beschrieben aus.

3 Geben Sie Ihre Änderungen in die von Ihnen ausgewählte Preload-Liste ein.

4 Um die Änderungen zu sichern, geben Sie in der Kommandozeile des Bildschirms **Edit Preload List** das Kommando `SA` ein.

Um ein Objekt einer Preload-Liste zu ändern:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `E`, einen Library-Namen und den Namen einer Preload-Liste ein.

Oder:

Geben Sie auf dem Bildschirm **List Preload Lists** in der linken Spalte neben der gewünschten Preload-Liste das Zeilenkommando `E` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

`EDIT PRELOADLIST` *`list-name`* - für eine Preload-Liste mit Natural-Objekten

oder

`EDIT MPRELOADLIST` *`list-name`* - für eine Preload-Liste mit Meldungen.

Der Bildschirm **Edit Preload List** erscheint mit der angegebenen Preload-Liste.

2 Ersetzen Sie in den entsprechenden Eingabefeldern die vorhandenen Einträge durch neue Werte. Es gelten die gleichen Regeln wie in **Schritt 2** unter *Preload-Listen anlegen* beschrieben.

3 Um die Änderungen zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Preload List** das Kommando `SA` ein.

Um ein Objekt aus einer Preload-Liste zu löschen:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `E`, einen Library-Namen und den Namen einer Preload-Liste ein.

Oder:

Geben Sie im Bildschirm **List Preload Lists** in der linken Spalte neben der/den zu löschenden Vorladeliste(n) das Zeilenkommando `E` ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

`EDIT PRELOADLIST` *`list-name`* - für eine Preload-Liste mit Natural-Objekten

oder

`EDIT MPRELOADLIST` *`list-name`* - für eine Preload-Liste mit Meldungen.

Es erscheint der Bildschirm **Edit Preload List** mit der angegebenen Preload-Liste.

2 Geben Sie in der Spalte ganz links neben dem zu löschenden Objekt das Kommando `D` ein.

Das angegebene Objekt wird aus der Preload-Liste entfernt.

3 Um die Änderung zu speichern, geben Sie in der Kommandozeile des Bildschirms **Edit Preload List** das Kommando `SA` ein:

Preload-Liste generieren

- Preload-Liste aus Buffer Pool generieren
- Preload-Liste aus Message Pool generieren

Preload-Liste aus Buffer Pool generieren

Mit der Funktion **Generate Preload List** können Sie eine neue Preload-Liste generieren, indem Sie die Namen der zurzeit im Buffer Pool geladenen Objekte verwenden. Sie können aus den Objekten, die sich zurzeit im Buffer Pool befinden, diejenigen auswählen, die Sie in die Preload-Liste aufnehmen möchten.

Um eine Preload-Liste zu generieren, wählen Sie eine der folgenden Optionen:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `G` und den Namen einer Preload-Liste ein. Geben Sie in den Feldern **Library**, **Objects**, **Resident**, **Use Count** und **Total Use Count** die Objekte an, die in die Liste aufgenommen werden sollen:

- Um alle Objekte in die Liste aufzunehmen, die sich derzeit im Buffer Pool befinden, geben Sie in den Feldern **Library**, **Objects** und **Resident** einen Stern (*) ein und lassen die Felder **Use Count** und **Total Use Count** leer.
- Oder: Um bestimmte Objekte in den Buffer Pool aufzunehmen, können Sie in die unten beschriebenen Felder die folgenden Werte eingeben:

Library	Ein einzelner Name oder ein Stern (*).
Objects	Ein einzelner Name oder ein Stern (*).
Resident	Ein Stern (*) für alle Objekte oder `Y` (Yes) für alle Objekte, die derzeit als resident im Buffer Pool markiert sind.
Use Count	Ein numerischer Startwert (>), z. B. `>10`. Wählt alle Objekte aus, deren **Use Count** größer als oder gleich dem Wert ist. Dieses Feld entspricht dem unter *Verzeichnisinformationen anzeigen* beschriebenen **Current Use Count**.
Total Use Count	Ein numerischer Startwert (>), z. B. `>10`. Wählt alle Objekte aus, deren **Total Use Count** größer als oder gleich dem Wert ist. Dieses Feld entspricht dem unter *Verzeichnisinformationen anzeigen* beschriebenen **BP Total Use**.

2 Geben Sie eines der folgenden SYSBPM-Direktkommandos ein:

```
GENERATE PRELOADLIST list-name
```

oder

```
GENERATE PRELOADLIST list-name gen-library
```

Siehe auch die Erläuterungen zu den Feldwerten oben.

Es erscheint eine Meldung, die bestätigt, dass die Preload-Liste aus dem Buffer Pool generiert wurde.

Alle Preload-Listenobjekte werden standardmäßig als resident (Eintrag `Y` in Spalte **R**) generiert. Wählen Sie manuell, welche Objekte Sie aus der Liste entfernen möchten.

Objekte aus der Library SYSBPM werden nicht in die generierte Preload-Liste aufgenommen, da davon ausgegangen werden kann, dass es sich um Objekte handelt, die nur in den Buffer Pool geladen wurden, um diese Funktion auszuführen.

Preload-Liste aus Message Pool generieren

Mit der Funktion **Generate Preload List from Message Pool** können Sie eine neue Preload-Liste anhand der Namen der aktuell im Message Pool geladenen Objekte erstellen. Sie können aus den Objekten, die sich derzeit im Message Pool befinden, diejenigen auswählen, die Sie in die Preload-Liste aufnehmen möchten.

Um eine Preload-Liste zu erstellen, verwenden Sie eine der folgenden Optionen:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `G` und den Namen einer Preload-Liste ein. Geben Sie in den Feldern **Library**, **Message**, **DBID**, **FNR**, **Language Code** und **Code Page** die Meldungen an, die in die Liste aufgenommen werden sollen:

- Um alle Meldungen, die sich derzeit im Message Pool befinden, in die Liste aufzunehmen, geben Sie einen Stern (*) in das Feld **Library** ein, geben Sie einen Wert im Bereich `1 - 9999` in das Feld **Message Number** (Meldungsnummer) ein, `0` in die Felder **DBID** (Datenbankkennung), **FNR** (Dateinummer) und **Language Code** (Sprachcode) und lassen Sie das Feld **Code Page** leer.
- Oder: Um bestimmte Meldungen in den Message Pool aufzunehmen, können Sie in den unten beschriebenen Feldern die folgenden Werte eingeben:

Message Number	Zwei Zahlen zwischen 1 und 9999, in aufsteigender Reihenfolge.
Library	Ein einzelner Name oder ein Stern (*).
DBID	Ein numerischer Wert zwischen `0` und `65535`, wobei `0` für alle Meldungen steht.
FNR	Ein numerischer Wert zwischen `0` und `5000`, wobei `0` für alle Meldungen steht.
Language Code	Ein numerischer Wert zwischen `1` und `60`.
Code Page	Ein einzelner Name oder ein Leerzeichen für alle Werte.

2 Geben Sie eines der folgenden SYSBPM-Direktkommandos ein:

```
GENERATE MPRELOADLIST list-name
```

oder

```
GENERATE MPRELOADLIST  list-name start-number end-number gen-library language ↩
dbid fnr codepage
```

Siehe auch die Erläuterungen zu den Feldwerten oben.

Es erscheint eine Meldung, die bestätigt, dass die Preload-Liste aus dem Message Pool generiert wurde.

Preload-Liste löschen

Um eine Preload-Liste zu löschen:

1 Geben Sie im Menü **Preload List Maintenance** den Funktionscode `L` und den Namen einer Preload-Liste ein.

Oder:

Geben Sie das folgende SYSBPM-Direktkommando ein:

`LIST PRELOADLIST list-name` - für eine Preload-Liste, die das Natural-Objekt enthält

oder

`LIST MPRELOADLIST list-name` - für eine Preload-Liste, die Meldungen enthält.

Oder:

```
LIST MPRELOADLIST list-name
```

Der Bildschirm **List Preload Lists** erscheint.

2 Geben Sie in der linken Spalte neben dem gewünschten Objekt das Zeilenkommando `D` ein:

Das Fenster **DELETE** erscheint.

3 Bestätigen Sie die Löschung durch Eingabe des Namens der Preload-Liste.

Es erscheint eine Bestätigungsmeldung.

Zusätzliche Verwaltungsfunktionen mit Dienstprogrammen

Die Natural-Dienstprogramme SYSMAIN und Object Handler bieten zusätzliche Funktionen zur Verwaltung von Preload-Listen. Dazu gehören das Übertragen von Preload-Listen zwischen verschiedenen Natural-Libraries und Systemdateien und/oder verschiedenen Großrechnerplattformen sowie das Löschen oder Auffinden von Preload-Listen in einer anderen Umgebung.

Bei der Verwendung eines Natural-Dienstprogramms wird eine Preload-Liste wie jedes andere Quellcodeobjekt vom Typ `Text` behandelt.

Einzelheiten hierzu finden Sie in den entsprechenden Abschnitten der Dokumentation über Dienstprogramme.

37 Performance-Aspekte

- Interne Fast-Locate-Tabelle 276
- Suche in Steplibs 277
- Wiederverwendung und Beibehaltung von Objekten 278
- Lokaler oder globaler Buffer Pool? 279

Dieser Abschnitt enthält allgemeine Ratschläge bei leistungsrelevanten Problemen im Zusammenhang mit dem Buffer Pool und den BP Cache.

Erläuterungen zu den in diesem Abschnitt genannten Statistikelementen finden Sie unter *Buffer Pool Load/Locate-Statistiken*.

Verwandtes Thema:

Performance-Hinweise

Interne Fast-Locate-Tabelle

Wenn innerhalb einer Natural-Sitzung ein Natural-Objekt erstmalig referenziert wird, erstellt der Buffer Pool Manager einen Verzeichniseintrag für dieses Objekt. Ein Verzeichniseintrag dient zur Identifizierung eines Objekts und enthält Informationen wie den Namen des Objekts, die Library, in der es sich befindet (Name, Datenbankkennung und Dateinummer) und die Adresse des Objekts (Position) im Buffer Pool.

Das Natural-Laufzeitsystem speichert die Namen der Objekte, die kürzlich ausgeführt wurden, die Libraries (Name, Datenbankkennung und Dateinummer), in denen sie sich befinden, und (zur Schnellsuche) die Adressen der entsprechenden Buffer Pool-Verzeichniseinträge in der internen Fast-Locate-Tabelle für die Dauer einer Natural-Sitzung.

Wenn ein Benutzer ein Objekt aufruft, das zuvor in der Natural-Sitzung verwendet wurde, gibt das Natural-Laufzeitsystem die Informationen aus der internen Fast-Locate-Tabelle an den Buffer Pool Manager weiter, der dann die normale Lokalisierungsprozedur (Normal Locate) umgehen und einen zeitsparenden Schnellsuche-Aufruf (Fast Locate) durchführen kann (siehe **Quick Locate Calls**). Dies ist der effizienteste Weg, um ein Objekt zu lokalisieren.

Wenn sich die Position eines Objekts im Buffer Pool geändert hat, plant der Buffer Pool Manager automatisch einen Normal Locate-Aufruf. Die Position ändert sich in der Regel, wenn ein Objekt, das aus dem Buffer Pool gelöscht oder durch ein anderes Objekt überschrieben wurde, erneut in den Buffer Pool geladen wird.

Die interne Schnellsuche-Tabelle enthält maximal 128 Einträge. Die Schnellsuche-Tabelle wird mit dem Systemkommando `LOGON` (Anmelden) zurückgesetzt und muss mit Normal Locate-Aufrufen erneut gefüllt werden. Daher verliert eine Anwendung, die ein `LOGON` durchführt, an Leistung.

Ein hohes Verhältnis von **Normal Locate Calls** zu **Quick Locate Calls** deutet auf die Verwendung von `LOGON`-Kommandos in einer Natural-Anwendung hin, da jedes `LOGON` ein Zurücksetzen der internen Fast Locate-Tabelle bewirkt.

- Unerwartete Suchergebnisse bei gesetztem BPSFI-Parameter

Unerwartete Suchergebnisse bei gesetztem BPSFI-Parameter

Wenn der Profilparameter `BPSFI` (Objektsuche zuerst im Buffer Pool) auf `OFF` gesetzt ist, kann das folgende Szenario zu unerwarteten Ergebnissen führen:

- Die Liste der Steplibs enthält die Libraries S1 und S2, wobei S1 vor S2 durchsucht wird.
- Während der aktuellen Natural-Sitzung wird auf ein Objekt aus S2 zugegriffen.
- Eine andere Natural-Sitzung kopiert eine neue Version dieses Objekts in S1.

Wird dann während der aktuellen Sitzung (ohne neues `LOGON`) erneut auf dieses Objekt zugegriffen, wird die neue Version des Objekts nicht verwendet.

Beachten Sie Folgendes, um solche unerwarteten Suchergebnisse zu vermeiden:

- Setzen Sie ein `LOGON`-Systemkommando ab.
- Da ein LOGON-Kommando in einer Client/Server-Umgebung nicht abgesetzt werden kann, können Sie die Anwendungsprogrammierschnittstelle (API) **USR3004N** verwenden, um die Fast-Locate-Tabelle zu löschen.

  Weitere Informationen zur Verwendung von APIs finden Sie in der Beschreibung des Dienstprogramms **SYSEXT**.
- Wenn `BPSFI=ON` gesetzt ist, sollten Objektnamen in allen Libraries, die an Objektsuchoperationen beteiligt sind, immer eindeutig sein. Damit ist auch gewährleistet, dass solche Szenarien nicht auftreten.

Suche in Steplibs

Die Suche nach einem Natural-Objekt in einer langen Kette von Steplib-Libraries hat einen negativen Einfluss auf die Performance.

Für jede Steplib ruft die Natural-Laufzeit den Buffer Pool Manager auf, bis das gesuchte Objekt gefunden wird. Jeder unnötige Aufruf einer Steplib-Library, die das angeforderte Objekt nicht enthält, kann jedoch normalerweise vermieden werden.

Abhängig von der Einstellung des Profilparameters `BPSFI` (siehe *Parameter-Referenz*-Dokumentation) können zusätzliche Datenbankaufrufe erforderlich sein.

Eine lange Steplib-Kette zeigt sich durch ein hohes Verhältnis von **Normal Locate Calls** (einschließlich Steplib-Suchen) zu **Normal Locate Calls** (ohne Steplib-Suchen), das wie folgt berechnet wird:

Normal Locate Calls: (Normal Locate Calls - STEPLIB Searches)

Beispiel für eine Steplib-Suche

Beim Durchsuchen der Standard-Steplib-Kette (Library SYSTEM von FUSER, Library SYSTEM von FNAT) führt jeder Versuch, ein Objekt aus SYSTEM (FNAT) zu laden, zu Folgendem:

3 Normal Locate Calls und **2 STEPLIB Searches**

Erläuterung: 3 Normal Locate-Aufrufe resultieren aus Suchen in der aktuellen Library, der Library SYSTEM (FUSER) und der Library SYSTEM (FNAT). Es gibt (mindestens) 2 Normal Locate-Aufrufe, die fehlschlagen, da das Objekt weder in der aktuellen Library noch in der Library SYSTEM (FUSER) gespeichert ist. Bei Anwendung der obigen Formel ergibt sich ein Verhältnis von 3:1.

Befindet sich das Objekt in der aktuellen Library, ist das Ergebnis wie folgt:

1 Normal Locate Call und 0 (Null) STEPLIB Searches.

Bei Anwendung der obigen Formel ergibt sich ein Verhältnis von 1:1.

Wiederverwendung und Beibehaltung von Objekten

Eine Anwendung, die viele Natural-Objekte enthält, wobei jedes Objekt nur selten ausgeführt wird, hat starke Auswirkungen auf die Leistung des Buffer Pool. Keines der Objekte verbleibt lange im Buffer Pool und viele Objekte müssen aus einer Systemdatei geladen werden. Aus Performance-Gründen sollte eine Anwendung Objekte so oft wie möglich wiederverwenden, z. B. indem sie identischen Quellcode, der in mehreren Objekten enthalten ist, in ein einziges Objekt verschiebt.

Sie können die Verwendung von Objekten mit der Funktion **List Objects** überprüfen (siehe Abschnitt *Objekte auflisten*). Die Spalte **Max** enthält z.B. Informationen über die maximale Anzahl von Anwendungen, die ein Objekt ausgeführt haben, und die Spalte **TotalUC** enthält die Gesamtzahl der Locate-Aufrufe eines Objekts, das in den Buffer Pool geladen wurde.

Objekte können einzeln mit der Funktion **List Objects** oder durch Angabe in einer Preload-Liste resident gemacht werden (siehe *Preload-Liste verwalten*).

Lokaler oder globaler Buffer Pool?

Es gibt keine allgemeinen Empfehlungen, wann ein lokaler oder ein globaler Buffer Pool verwendet werden sollte, da verschiedene Anwendungen unterschiedliche Anforderungen haben. Aufgrund von Erfahrungswerten können wir jedoch allgemeine Ratschläge geben:

- Lokale Buffer Pools verwenden
- Globalen Buffer Pool verwenden

Lokale Buffer Pools verwenden

Die Performance kann durch die Verwendung mehrerer kleiner lokaler Buffer Pools anstelle eines einzigen globalen Buffer Pools verbessert werden, wenn die lokalen Buffer Pools verschiedenen Anwendungsumgebungen zugewiesen werden können.

In CICS-Umgebungen beispielsweise verbessert sich die Leistung in der Regel durch die Verwendung eines lokalen Buffer Pools für jede AOR (Application Operating Region).

Globalen Buffer Pool verwenden

Bei verschiedenen Batch-Anwendungen, die auf die gleichen Natural-Objekte verweisen, kann sich die Performance verbessern, wenn diese Anwendungen einen gemeinsamen globalen Buffer Pool anstelle eines lokalen Buffer Pools für jede einzelne Anwendung verwenden. In diesem Fall ist die Wahrscheinlichkeit höher, dass die von jeder Anwendung benötigten Objekte bereits von einer der anderen Anwendungen in den globalen Buffer Pool geladen wurden.

38 SYSBPM-Direktkommandos

Die in diesem Kapitel beschriebenen SYSBPM-Direktkommandos können verwendet werden, um SYSBPM-Funktionen direkt auszuführen oder in SYSBPM-Bildschirmen im Online- oder Batch-Modus zu navigieren. Weitere SYSBPM-Direktkommandos, die sich nur auf bestimmte Bildschirme beziehen, finden Sie in den Abschnitten, in denen die einzelnen SYSBPM-Funktionen dokumentiert sind.

SYSBPM-Direktkommandos, die sich auf den BP Cache oder die Buffer Pool Hash Table beziehen, gelten nur für Buffer Pools des Typs `Natural` (`TYPE=NAT`), siehe Schlüsselwort-Subparameter *TYPE - Buffer Pool-Typ* des Profilparameters `BPI` (Buffer Pool-Initialisierung) in der *Parameter-Referenz*-Dokumentation.

Die folgende Tabelle enthält eine Liste aller Direktkommandos von SYSBPM (einschließlich Unterkommandos), die Parameter, die mit den Kommandos verwendet werden können, und die entsprechenden SYSBPM-Menüfunktionen.

Die unten aufgeführten SYSBPM-Direktkommandos können in der Kommandozeile eines beliebigen SYSBPM-Bildschirms eingegeben werden. Um ein SYSBPM-Direktkommando von einer anderen Eingabeaufforderung aus oder im Batch-Modus auszuführen, muss dem Direktkommando das Schlüsselwort SYSBPM vorangestellt werden, zum Beispiel:

```
SYSBPM ADD BLACKLIST
```

Ein unterstrichener Teil eines Kommandos von SYSBPM stellt dessen Mindestabkürzung dar. Parameterwerte, die für ein Kommando erforderlich sind, werden durch kursiv gedruckte Buchstaben dargestellt.

Kommando	Parameter	Funktion
+	keine	In einer Liste eine Seite nach unten blättern.
-	keine	In einer Liste eine Seite nach oben blättern.
ADD BLACKLIST	keine	Ruft den Bildschirm **Maintain Blacklist** auf, der unter *Sperrliste verwalten* beschrieben ist.
ADD SET	*library-name set-name*	Fügt alle Objekte eines angegebenen Objektsatzes zur Sperrliste hinzu, wie unter ***Objektset zur Sperrliste hinzufügen*** beschrieben.
BLACKLIST	keine	Ruft das Menü **Blacklist Maintenance** auf.
BOTTOM	keine	An das Ende einer Liste blättern.
CANCEL	keine	Wie EXIT.
CHECK HASH oder CHECK HT	keine	Prüft die BP Hash-Tabelle auf Konsistenz und gibt die Anzahl der gefundenen Inkonsistenzen zurück. Siehe auch REBUILD HASH.
CLOSE BPC	keine	BP Cache erforderlich. Schließt den BP Cache. Der Buffer Pool läuft danach ohne BP Cache. Sie können den BP Cache mit dem Kommando INITIALIZE BPC neu starten.
DELETE	keine	Löscht alle Objekte aus dem Buffer Pool und dem BP Cache. Bei Eingabe auf dem Bildschirm **Directory Information**: siehe DELETE in ***PF-Tasten und Direktkommandos***.
DELETE	*library-name object-name dbid fnr*	Löscht das/die angegebene(n) Objekt(e) aus dem Buffer Pool und dem GP-Cache, siehe Kapitel ***Objekte löschen***.
DELETE ALL	keine	Löscht alle Objekte aus der Sperrliste, siehe ***Objektset aus Sperrliste löschen***.
DELETE BUFFERPOOL oder DELETE BP	keine	Löscht alle Objekte nur aus dem Buffer Pool.
	library-name object-name dbid fnr	Löscht nur das/die angegebene(n) Objekt(e) aus dem Buffer Pool.
DELETE BPC	keine	BP Cache erforderlich. Löscht alle Objekte nur aus dem BP Cache.
	library-name object-name dbid fnr	BP Cache erforderlich. Löscht nur das/die angegebene(n) Objekt(e) aus dem BP Cache.

Kommando	Parameter	Funktion
DELETE BLACKLIST	keine	Ruft den Bildschirm **Maintain Blacklist** auf, von dem aus Sie die Einträge aus der Sperrliste löschen können, siehe *Sperrliste verwalten*.
DELETE MP	keine	Message Pool erforderlich. Löscht alle Meldungen aus dem Message Pool.
	start-number end-number library-name language dbid fnr code-page	Message Pool erforderlich. Löscht die angegebenen Meldungen aus dem Message Pool.
DELETE SET	*library-name set name*	Löscht alle Objekte des angegebenen Objektset aus der Sperrliste, siehe ***Objektset aus Sperrliste löschen***.
DISPLAY ALL	keine	Wie DISPLAY LIST.
DISPLAY BUFFERPOOL oder DISPLAY BP	keine	Siehe ***Buffer Pools anzeigen*** unter *Buffer Pool auswählen*.
DISPLAY BLACKLIST	keine	Ruft den Bildschirm **Maintain Blacklist** auf, siehe *Sperrliste verwalten*.
DISPLAY CDIRECTORY	keine	BP Cache erforderlich. Ruft den Bildschirm **Directory Information** auf.
DISPLAY CGENERAL	keine	BP Cache erforderlich. Ruft den Bildschirm **General BP Cache Statistics** auf, siehe *Allgemeine BP Cache-Statistiken anzeigen*.
DISPLAY CHASH	keine	Ruft die Funktion **BP Cache Hash Table Statistics** auf, siehe *BP Cache Hash Table-Statistiken anzeigen*, und zeigt den Bildschirm **Cache Hash Table Collisions**.
DISPLAY CLIST	*library-name object-name dbid fnr*	BP Cache erforderlich. Ruft den Bildschirm **List Objects** auf. Im Gegensatz zum Kommando DISPLAY LIST erzeugt dieses Kommando einen Statistikbericht, der Daten über BP Cache-Objekte am Anfang der Liste anzeigt.
DISPLAY CLOAD	keine	BP Cache erforderlich. Ruft den Bildschirm **BP Cache Call Statistics** auf, siehe *BP Cache-Statistiken anzeigen*.
DISPLAY CSTATISTICS	keine	BP Cache erforderlich. Ruft den Bildschirm **BP Cache Statistics** auf, siehe ***BP Cache-Statistiken anzeigen***.

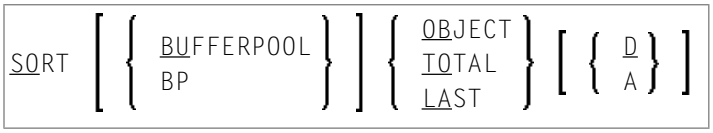
Kommando	Parameter	Funktion
DISPLAY DIRECTORY	*library-name object-name dbid fnr*	Ruft den Bildschirm **Directory Information** auf.
DISPLAY FRAGMENTATION	keine	Ruft den Bildschirm **Buffer Pool Fragmentation** auf, siehe *Buffer Pool-Statistiken anzeigen*.
DISPLAY FUNCTION	keine	Ruft den Bildschirm **Internal Function Usage** auf, siehe *Buffer Pool-Statistiken anzeigen*.
DISPLAY GENERAL	keine	Ruft den Bildschirm **General Buffer Pool Statistics** auf, siehe *Buffer Pool-Statistiken anzeigen*.
DISPLAY HASH oder DISPLAY HT	keine	Ruft die Funktion **Buffer Pool Hash Table Statistics** auf und zeigt den Bildschirm **Hash Table Collisions** an.
DISPLAY HDIRECTORY	*library-name object-name dbid fnr*	Ruft den Bildschirm **Directory Information Hex** auf, der die Verzeichnisinformationen eines Objekts im Hexadezimalformat anzeigt.
DISPLAY HEX	*library-name object-name dbid fnr*	Ruft den Bildschirm **Hexadecimal Display** auf, der den Quellcode eines Objekts im Hexadezimalformat anzeigt.
DISPLAY LIST	*library-name object-name dbid fnr*	Ruft den Bildschirm **List Objects** auf. Im Gegensatz zum Kommando `DISPLAY CLIST` erzeugt dieses Kommando einen Statistikbericht, der Daten über Buffer Pool-Objekte am Anfang der Liste anzeigt.
DISPLAY LOAD	keine	Ruft den Bildschirm **Buffer Pool Load/Locate Statistics** auf, siehe *Buffer Pool-Statistiken anzeigen*.
DISPLAY MLIST	*start-number end-number library-name language dbid fnr code-page*	Message Pool erforderlich. Ruft den Bildschirm **List Objects** auf. Listet die ausgewählten Meldungen des Message Pool auf.
DISPLAY MP	keine	Message Pool erforderlich. Listet alle verfügbaren Message Pools auf, die im aktuellen Subsystem laufen.
DISPLAY MSTATISTICS	keine	Message Pool erforderlich. Zeigt die **Message Pool Statistics** an.
DISPLAY PERFORMANCE	keine	Ruft den Bildschirm **Performance Hints** mit leistungsrelevanten Statistiken eines Natural Buffer Pool und eines BP Cache auf, siehe *Buffer Pool-Statistiken anzeigen* und *BP Cache-Statistiken anzeigen*.
DISPLAY STATISTICS	keine	Ruft das Menü **Buffer Pool Statistics** auf, siehe *Buffer Pool-Statistiken anzeigen*.

Kommando	Parameter	Funktion
EDIT PRELOADLIST	*list-name*	Ruft den Bildschirm **Edit Preload List** auf, siehe *Preload-Liste verwalten*.
EDIT MPRELOADLIST	*list-name*	Message Pool erforderlich. Ruft den Bildschirm **Edit Message Preload List** auf, siehe *Preload-Liste verwalten* .
EDIT SET	*library-name set-name*	Ruft den Bildschirm **Edit Object Set** auf, siehe *Sperrliste verwalten*.
EXIT	keine	Verlassen des aktuellen Bildschirms und Anzeigen des vorherigen Bildschirms.
FLIP	keine	Schaltet die PF-Tastenzeile um: Wechsel zwischen der Anzeige von PF1 bis PF12 und PF13 bis PF24.
GENERATE MPRELOADLIST	*list-name start-number end-number gen-library language dbid fnr code-page*	Message Pool erforderlich. Ruft die Funktion **Generate Preload List from Message Pool** auf.
GENERATE PRELOADLIST	*list-name gen-library*	Ruft die Funktion **Generate Preload List from Buffer Pool** auf.
INITIALIZE	keine oder: 1, 2, 4, 8, 12, 16	Reinitialisiert den Buffer Pool und den BP Cache. Wenn keine Textsatzgröße angegeben ist, wird die aktuelle Textsatzgröße verwendet. Verwenden Sie diese Funktion nur, wenn der **Current Use Count** (siehe ***Verzeichnisinformationen anzeigen***) gleich 0 ist (siehe die Warnung unten) oder wenn der Buffer Pool zerstört wurde. **Vorsicht:** Wenn Sie versuchen, den Buffer Pool neu zu initialisieren, während Objekte von aktiven Sitzungen in diesem Buffer Pool ausgeführt werden, erscheint das Fenster **Confirm Initialization** mit dem **Current Use Count** für diesen Buffer Pool (der SYSBPM-Benutzer selbst wird nicht mitgezählt). Wenn **Current Use Count** ungleich 0 (Null) ist und Sie ein Y eingeben, um die Reinitialisierung des Puffers zu bestätigen, sind die Ergebnisse der aktiven Sitzungen unvorhersehbar und Natural kann sogar abbrechen.
INITIALIZE BP	keine oder: 1, 2, 4, 8, 12, 16	Reinitialisiert nur den Buffer Pool. Wenn keine Textsatzgröße angegeben wird, wird die aktuelle Textsatzgröße verwendet. Siehe auch die Warnung oben.

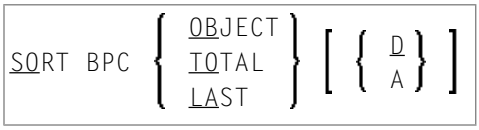
Kommando	Parameter	Funktion
		Siehe auch die *Warnung* oben.
INITIALIZE BPC	keine	BP Cache erforderlich. Reinitialisiert nur den BP Cache. Die Textsatzgröße des BP Cache ist fest (4 KB).
INITIALIZE MP	keine	Message Pool erforderlich. Reinitialisiert den für SYSBPM aktiven Message Pool, d.h. der Inhalt des Message Pools wird gelöscht.
LAST	keine	Zeigt das zuletzt eingegebene SYSBPM-Direktkommando an.
LIST MPRELOADLIST	*list-name*	Message Pool erforderlich. Ruft den Bildschirm **Generate Preload List from Message Pool** für das angegebene Objekt auf, siehe *Preload-Liste verwalten*.
LIST PRELOADLIST	*list-name*	Ruft den Bildschirm **List Preload Lists** für das angegebene Objekt auf, siehe *Preload-Liste verwalten*.
LIST SET	*library-name* *set-name*	Ruft den Bildschirm **List Object Sets** für die angegebene Library oder das angegebene Objekt auf, siehe *Sperrliste verwalten*. Für *set-name* ist auch ein Stern (*) zulässig.
MENU	keine	Ruft das SYSBPM-Hauptmenü (**Main Menu**), siehe *SYSBPM aufrufen und benutzen*.
MPRELOADLIST	keine	Message Pool erforderlich. Ruft das Menü **Preload List Maintenance** auf, siehe *Preload-Liste verwalten*.
PRELOADLIST	keine	Ruft das Menü **Preload List Maintenance** auf, siehe *Preload-Liste verwalten*.
QUIT	keine	Wie EXIT.
REBUILD HASH oder REBUILD HT	keine	Baut Hash-Tabellen neu auf, wenn mit CHECK HASH **Inkonsistenzen gefunden werden.** REBUILD HASH löscht die aktuelle Hash-Tabelle und baut eine neue Hash-Tabelle aus dem aktuellen Inhalt des Buffer Pool auf.
RESET BUFFERPOOL oder RESET BP	keine	Setzt den Buffer Pool zurück, siehe *Buffer Pool zurücksetzen*.

Kommando	Parameter	Funktion
RESET MP	keine	Setzt den für SYSBPM aktiven Message Pool auf den Message Pool zurück, der beim Start von Natural verwendet wurde.
SELECT BUFFERPOOL oder SELECT BP	keine oder: *buffer pool name*	Gilt nur für Buffer Pools des Typs Natural, DL/I oder Sort. Ruft eine Auswahlliste mit Buffer Pools auf, siehe ***Buffer Pool auswählen***. Wenn ein Buffer Pool Name eingegeben wird, wird er ausgewählt, ohne dass die Auswahlliste angezeigt wird.
SELECT MP	keine oder: *message pool name*	Message Pool erforderlich. Ruft eine Auswahlliste von Message Pools auf, siehe ***Buffer Pool auswählen***. Wenn ein Message Pool-Name eingegeben wird, wird er ausgewählt, ohne dass die Auswahlliste angezeigt wird.
SORT BPC	See ***Command Syntax for SORT BPC*** below.	Sortiert den BP Cache, siehe ***Sortierten Auszug anzeigen***.
SORT BUFFERPOOL	See ***Command Syntax for SORT*** below.	Sortiert den BP Cache, siehe ***Sortierten Auszug anzeigen***.
STOP	keine	Verlassen des SYSBPM-Dienstprogramms.
TOP	keine	An den Anfang einer Liste blättern.
WRITE BP oder WRITE BPC oder WRITE ALL		Schreibt die Objektverzeichnisdaten in eine lokale Datei oder eine PC-Textdatei. Siehe auch ***In Arbeitsdatei schreiben***.

Kommando-Syntax für SORT



Kommando-Syntax für SORT BPC



39 Batch-Verarbeitung

- Verwandte Themen 290

Die vom Dienstprogramm SYSBPM bereitgestellten Funktionen können auch im Batch-Modus ausgeführt werden.

Zu diesem Zweck empfehlen wir Ihnen die Verwendung der SYSBPM-Anwendungsprogrammierschnittstelle USR4340N oder USR0340N. Weitere Informationen zu diesen APIs siehe Kapitel *Anwendungsprogrammierschnittstellen*.

Sie können das Dienstprogramm SYSBPM auch im Batch-Modus verwenden, indem Sie die Abfolge der Online-Eingabekommandos simulieren.

Da das Dienstprogramm SYSBPM die Funktionalität des Software AG-Editors nutzt, sollten Sie beim Schreiben Ihres Batch-Jobs die folgende Anleitung beachten:

- Setzen Sie den Profilparameter `EDPSIZE`.

  Alternativ können Sie den Profilparameter `BPI` mit `TYPE=EDIT` für den Buffer Pool des Software AG Editors setzen und die Arbeitsdatei des Editors im Batch-Job definieren.
- Beachten Sie, dass das Feld **Code** nicht auf jedem SYSBPM-Bildschirm verfügbar ist.
- Beachten Sie, dass Sie eventuell Eingabefelder überspringen müssen, um für die Eingabe von Direktkommandos den Cursor im Eingabefeld **Command** (Kommandozeile) zu positionieren.
- Simulieren Sie PF-Tasten, indem Sie die `%K`-Terminalkommandos verwenden, um durch das SYSBPM-Dienstprogramm zu navigieren. Verwenden Sie zum Beispiel `%K3`, um den Bildschirm **List Objects** zu verlassen.
- Verwenden Sie das mit dem Session-Parameter `CF` definierte Fortsetzungszeichen (standardmäßig ist dies das Prozentzeichen `%`), um mehr als eine Eingabezeile für eine Bildschirmmaske einzugeben.

Im Folgenden sind Themen und Quellen in der Natural-Dokumentation zusammengefasst, die sich auf die obige Anleitung beziehen.

Verwandte Themen

Thema	Dokumentation
Sperrliste verwalten in Batch Mode	*SYSBPM Utility*
SYSBPM-Direktkommandos	*SYSBPM Utility*
CF - Steuerzeichen für Terminalkommandos	*Parameter-Referenz*
EDPSIZE - Größe des Software AG Editor-Hilfs-Buffer-Pool	*Parameter-Referenz*
BPI - Buffer Pool-Initialisierung	*Parameter-Referenz*
Installing the Software AG Editor	*Installation for z/OS*
%K und %KP - Simulieren von PF- und PA-Tasten	*Terminalkommandos*
EDBP - Software AG Editor Buffer Pool Definitions	*Parameter-Referenz*

Thema	Dokumentation
Editor Buffer Pool	*Operations*
Editor Work File	*Operations*
Natural in Batch Mode	*Operations*
INPUT-Statement unter Nicht-Screen-Modi	*Statements*
INPUT-Statement im Batch-Betrieb	*Statements*

40 Anwendungsprogrammierschnittstellen

Dieses Kapitel beschreibt die Anwendungsprogrammierschnittstellen (APIs) USR0340N, USR0341N, USR3004N, USR4340N, USR4341N und USR4342N. Diese APIs werden für die Handhabung von Natural-Objekten verwendet, die zurzeit in den Buffer Pool, den BP Cache und/oder die interne Fast Locate Table geladen sind. Die APIs werden in der Natural-System-Library SYSEXT bereitgestellt.

Weitere Informationen zu diesen APIs finden Sie im Folgenden:

- Das entsprechende Natural-Quellcodeobjekt vom Typ `Text` und die Beispielprogramme in der Natural-System-Library SYSEXT. Siehe auch Beschreibung des Dienstprogramms SYSEXT.
- Die SYSBPM-Funktionen, auf die in der folgenden Tabelle verwiesen wird.
- Informationen zur Benutzung dieser Anwendungsprogrammierschnittstellen siehe Kapitel *Batch-Verarbeitung*.

API	Funktionalität
USR0340N	■ Löschen von Objekten aus dem Buffer Pool und/oder BP Cache. ■ Markieren von Objekten als resident. ■ Entfernen des Resident-Flags von Objekten. ■ Lesen von Objektverzeichnisinformationen. ■ Abrufen der allgemeinen Buffer Pool-Statistiken und der Pufferpool-Lade-/Lokalisierungsstatistiken. Entsprechende SYSBPM-Funktionen: **Delete Objects** **List Objects** **Directory Information** **General Buffer Pool Statistics** **Buffer Pool Load/Locate Statistics**

API	Funktionalität
USR0341N	Sammeln von Datenmüll, um den Buffer Pool zu bereinigen, indem nicht mehr benötigte Objekte entfernt werden. Auswahlkriterien für bestimmte Objekte sind das relative Alter eines Objekts. Das relative Alter ist die Zeit, seit der das Objekt in den Buffer Pool geladen wurde. Es wird aus dem Datum der letzten Aktion des BP (**BP Last Action**) berechnet. Das minimale Alter beträgt 30 Minuten. Siehe auch die SYSBPM-Funktion **List Objects**.
USR3004N	Löschen der internen Fast-Locate-Tabelle. Siehe auch ***Interne Fast-Locate-Tabelle***.
USR4340N	Wir empfehlen, diese Schnittstelle für die Batch-Verarbeitung zu verwenden, anstatt das Dienstprogramm SYSBPM im Batch-Modus zu benutzen. Listet die in den Buffer Pool und/oder BP Cache geladenen Objekte sortiert nach Objektgröße (**ObjSize**), Total Use Count (**TotalUC**) oder letzte Aktion des BP (**BP Last Action**) auf. **BP Last Action** gilt nur für den Buffer Pool. Entsprechende SYSBPM-Funktionen: **Display Sorted Extract** **List Objects**
USR4341N	Verwalten einer Sperrliste von Natural-Objekten, die nicht ausgeführt und in den Buffer Pool geladen werden sollen. Siehe auch die SYSBPM-Funktion **Blacklist Maintenance**.
USR4342N	■ Löschen von Objekten aus dem Message Pool. ■ Auflisten von Meldungen aus dem Message Pool. ■ Abrufen von Message Pool-Statistiken. Entsprechende SYSBPM-Funktionen: **Delete Objects** **List Objects** **Message Pool Statistics**

VI DBLOG Utility - Datenbankaufrufe protokollieren

Mit dem Dienstprogramm („Utility“) DBLOG können Sie Folgendes protokollieren:

- Adabas-Kommandos,
- SYNC/ROLB-Aufrufe,
- SQL-Statements,
- VSAM-Aufrufe.

Die Protokollierung hilft Ihnen bei der Feinabstimmung einer Anwendung (Kontrolle des Flusses von Kommandos, die auf die Datenbank zugreifen) und bei der Analyse von Fehlercodes, die von der Datenbank zurückgegeben werden können.

DBLOG ausführen
DBLOG-Menü
DBLOG Trace-Bildschirm
DBLOG Snapshot-Funktion
TEST DBLOG-Kommando

41 DBLOG ausführen

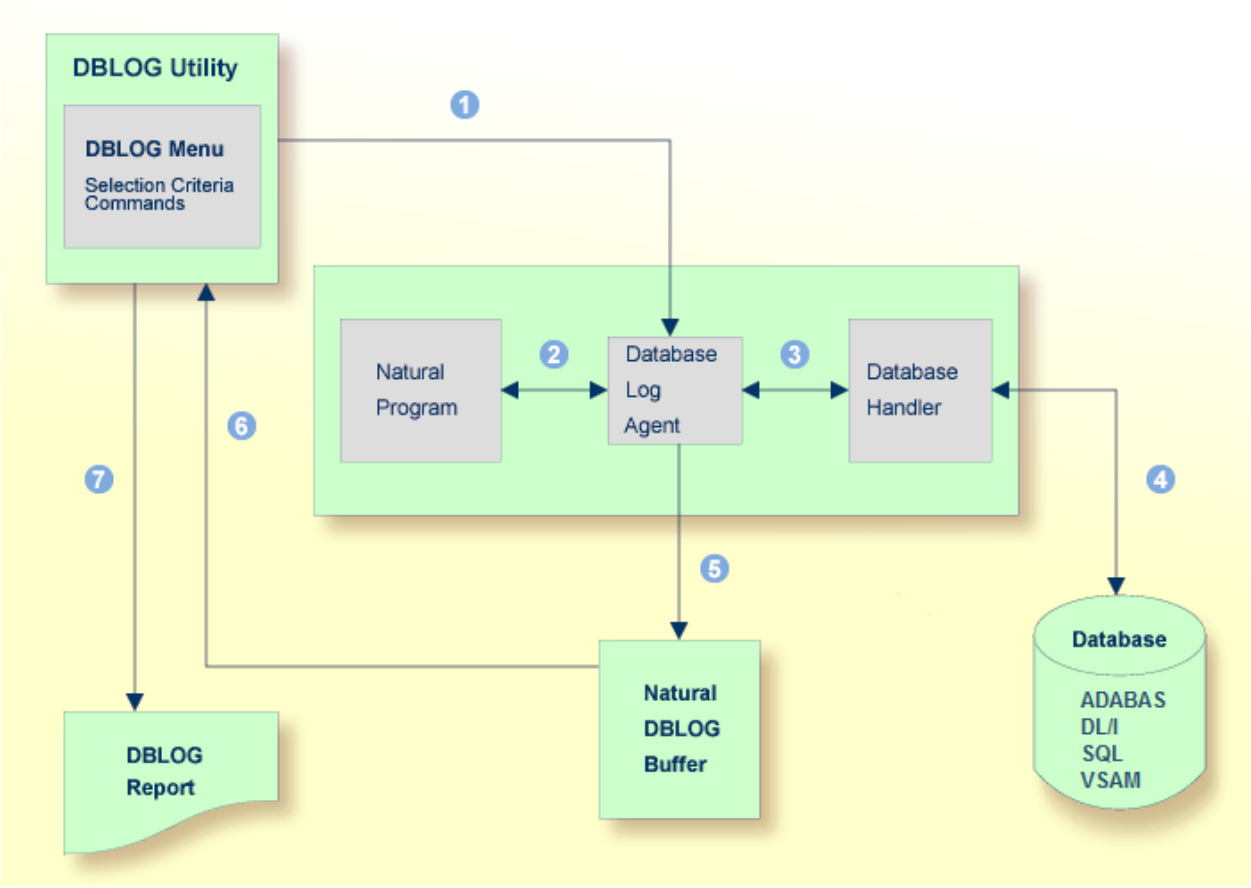
- Grundprinzipien der Datenbankprotokollierung 298
- Verarbeitung und Speicherung der Daten 299
- DBLOG aktivieren und deaktivieren 300
- DBLOG mit Auswahlkriterien benutzen 302

Die DBLOG Utility protokolliert jedes Adabas-Kommando, jeden SYNC/ROLB-Aufruf, jedes SQL-Statement oder jeden VSAM-Aufruf, nachdem ob dieses bzw. dieser durch das Datenbanksystem verarbeitet worden ist.

Folgende Themen werden behandelt:

Grundprinzipien der Datenbankprotokollierung

Die folgende Grafik veranschaulicht den Ablauf, wenn Datenbankaufrufe aus einem Natural-Programm mit der DBLOG Utility protokolliert werden:



Legende

1. Die Protokollierung von Datenbankaufrufen (Adabas-Kommando, SYNC/ROLB-Aufruf, SQL-Statement oder VSAM-Aufruf) wird aktiviert, wenn Sie die entsprechende Funktion im **DBLOG-Menü** oder das Natural-Systemkommando `TEST DBLOG` benutzen.

   Im **DBLOG-Menü** können Sie individuelle Auswahlkriterien festlegen, z.B. um die Protokollierung auf eine bestimmte Datenbankkennung (DBID) oder Dateinummer (FNR) einzuschränken.
2. Ein Natural-Programm setzt ein Statement ab, das einen Datenbankaufruf initiiert, z.B. `FIND` oder `READ`.
3. Der Database Log Agent leitet den Datenbankaufruf an den Database Handler weiter.
4. Der Database Handler passt den Datenbankaufruf an die jeweilige Datenbank an (Adabas, SQL oder VSAM), ruft die durch den Datenbankaufruf angeforderten Daten ab und gibt diese Daten an den Database Log Agent zurück.
5. Der Database Log Agent protokolliert im Natural-DBLOG-Puffer die vom Database Handler zurückgegebenen Daten leitet diese Daten an das Natural-Programm weiter.
6. Die Report-Funktion der DBLOG Utility liest die im Natural-DBLOG-Puffer aufgezeichneten Daten und selektiert Datensätze gemäß den im **DBLOG-Menü** im Schritt 1 angegebenen Auswahlkriterien.
7. Anhand der selektierten Datensätze erstellt die Report-Funktion der DBLOG Utility einen Report, der angezeigt, in eine Arbeitsdatei geschrieben oder zur Stapelverarbeitung benutzt werden kann.

Verarbeitung und Speicherung der Daten

Die von der DBLOG Utility protokollierten Daten werden im Natural-DBLOG-Puffer aufgezeichnet. Die Anfangsgröße und die maximale Größe des Pufferspeichers werden mit dem Natural-Profilparameter `DSIZE` festgelegt, siehe *Parameter-Referenz*-Dokumentation.

Falls die Anfangsgröße nicht reicht, erhöht Natural (falls nötig, öfters) automatisch die Puffergröße bis zur angegebenen maximalen Größe. Nachdem der Maximalwert erreicht ist, werden nur die aktuellsten Protokollierungsdaten im Natural-DBLOG-Puffer gehalten.

Falls die Größe des DBLOG-Puffers mangels Speicherplatz nicht weiter vergrößert werden kann, gibt Natural die Fehlermeldung NAT7545 aus, die anzeigt, dass der Speicherplatz für den DBLOG-Puffer nicht ausreicht. Alle vor dem Auftreten dieses Speicherplatzmangels protokollierten Daten werden im Natural-DBLOG-Puffer gehalten und können mit dem Kommando `TEST DBLOG` angezeigt werden.

DBLOG kann sowohl im Online- als auch im Batch-Modus benutzt werden. SYNC/ROLB-Aufrufe können unter CICS, unter IMS TM oder im Batch-Modus protokolliert werden. Weitere Informationen zum Batch-Modus siehe *Natural in Batch Mode* in der Natural-*Operations*-Dokumentation.

Die aufgezeichneten Protokollierungen werden im Bildschirm **DBLOG Trace** angezeigt.

Die DBLOG Utility verfügt über Standardeinstellungen für die Datenaufzeichnung. Wenn Sie das **DBLOG-Menü** benutzen, können Sie Auswahlkriterien für die Kommandos, Aufrufe oder Statements, die protokolliert werden sollen, und für die angezeigten Informationen angeben. Im **DBLOG-Menü** sind Funktionen zum Aktivieren oder Deaktivieren der Protokollierung vorhanden. Außerdem können Sie das Natural-Systemkommando `TEST DBLOG` benutzen, um die DBLOG-Ausführung zu steuern.

Die Felder im **DBLOG Trace-Bildschirm**, im **DBLOG-Menü** und das Natural-Systemkommando `TEST DBLOG` werden in den folgenden Abschnitten der DBLOG-Dokumentation beschrieben.

DBLOG aktivieren und deaktivieren

In diesem Abschnitt werden die Kommandos beschrieben, mit denen Sie DBLOG mit den DBLOG Utility-Standardeinstellungen aktivieren oder deaktivieren können. Weitere Informationen siehe *TEST DBLOG -Kommando*.

Um DBLOG für Adabas zu aktivieren oder zu deaktivieren:

- Geben Sie folgendes Natural-Systemkommando (Kippschalterfunktion) ein:

  ```
  TEST DBLOG
  ```

  Oder:

  Geben Sie zum Aktivieren Folgendes ein:

  ```
  TEST DBLOG ON
  ```

  oder

  ```
  TEST DBLOG START
  ```

  Geben Sie zum Deaktivieren Folgendes ein:

  ```
  TEST DBLOG OFF
  ```

  Oder:

  Geben Sie im **DBLOG-Menü** den Funktionscode `B` (zum Aktivieren) oder den Funktionscode `E` (zum Deaktivieren) ein.

Um DBLOG für SQL zu aktivieren oder zu deaktivieren:

- Geben Sie folgendes Natural-Systemkommando (Kippschalterfunktion) ein:

```
TEST DBLOG Q
```

Oder:

Geben Sie zum Aktivieren Folgendes ein:

```
TEST DBLOG Q ON
```

oder

```
TEST DBLOG Q START
```

Geben Sie zum Deaktivieren Folgendes ein:

```
TEST DBLOG Q OFF
```

Oder:

Geben Sie im **DBLOG-Menü** den Funktionscode `B` (zum Aktivieren) oder den Funktionscode `E` (zum Deaktivieren) ein.

Um DBLOG für VSAM zu aktivieren oder zu deaktivieren:

- Geben Sie folgendes Natural-Systemkommando (Kippschalterfunktion) ein:

```
TEST DBLOG V
```

Oder:

Geben Sie Folgendes zum Aktivieren ein:

```
TEST DBLOG V ON
```

oder

```
TEST DBLOG V START
```

Geben Sie Folgendes zum Deaktivieren ein:

```
TEST DBLOG OFF
```

Oder:

Geben Sie im **DBLOG-Menü** den Funktionscode `B` (zum Aktivieren) oder den Funktionscode `E` (zum Deaktivieren) ein.

DBLOG mit Auswahlkriterien benutzen

In den folgenden Beispielen wird erklärt, wie Sie im **DBLOG-Menü** Auswahlkriterien eingeben können, um Adabas-Kommandos, Aufrufe, SQL-Statements oder VSAM-Aufrufe selektiv zu protokollieren.

Um DBLOG mit Auswahlkriterien zu benutzen:

1 Rufen Sie das **DBLOG-Menü** auf, indem Sie eines der folgenden Natural-Systemkommandos eingeben:

- Für Adabas:

```
TEST DBLOG MENU
```

- Für SQL:

```
TEST DBLOG Q MENU
```

- Für VSAM:

```
TEST DBLOG V MENU
```

Das **DBLOG-Menü** erscheint.

2 Geben Sie im **DBLOG-Menü** die Einschränkungen für das Protokollieren an und aktivieren Sie die Protokollierung: Füllen Sie die Eingabefelder vollständig aus und geben Sie den Funktionscode `B` ein.

Die Meldung `DBLOG started now` erscheint.

3 Führen Sie ein Natural-Programm aus, das Adabas-Kommandos, SQL-Statements oder VSAM-Aufrufe enthält.

4 Rufen Sie den **DBLOG Trace**-Bildschirm auf und deaktivieren Sie die Protokollierung, indem Sie eines der folgenden Natural-Systemkommandos eingeben:

- Für Adabas:

```
TEST DBLOG
```

- Für SQL:

```
TEST DBLOG Q
```

- Für VSAM:

```
TEST DBLOG V
```

oder

```
TEST DBLOG V SHOW
```

Der **DBLOG Trace**-Bildschirm erscheint:

5 Löschen Sie den Inhalt des Natural-DBLOG-Puffers und deaktivieren Sie die Protokollierung, indem Sie eines der folgenden Natural-Systemkommandos eingeben:

- Für Adabas:

```
TEST DBLOG OFF
```

- Für SQL:

```
TEST DBLOG Q OFF
```

- Für VSAM:

```
TEST DBLOG V OFF
```

DBLOG wird beendet und die Eingabeaufforderung NEXT erscheint.

Weitere Informationen siehe *TEST DBLOG -Kommando*.

42 DBLOG-Menü

- DBLOG-Menü-Funktionen 306
- Einschränkungen für das Protokollieren angeben 308
- Adabas-Puffer angeben 308

Im **DBLOG-Menü** können Sie die Protokollierung aktivieren oder deaktivieren und angeben, welche Adabas-Kommandos, SQL-Statements oder VSAM-Aufrufe protokolliert werden sollen.

Um das DBLOG-Menü aufzurufen:

- Geben Sie eines der folgenden Natural-Systemkommandos ein:
  - Für Adabas:

    ```
    TEST DBLOG MENU
    ```

  - Für SQL:

    ```
    TEST DBLOG Q MENU
    ```

  - Für VSAM:

    ```
    TEST DBLOG V MENU
    ```

Folgende Themen werden behandelt:

DBLOG-Menü-Funktionen

Im **DBLOG-Menü** können Sie die nachfolgend beschriebenen Funktionen ausführen, indem Sie im Feld **Code** den Code eingeben, der der gewünschten Funktion entspricht, oder indem Sie die PF-Taste drücken, die der gewünschten Funktion entspricht.

Code oder PF-Taste	Funktion	Erklärung
B oder PF4	**Begin Logging of Adabas Commands**	Aktiviert die DBLOG-Protokollierung der Adabas-Kommandos, SQL-Statements oder VSAM-Aufrufe, die die Auswahlkriterien erfüllen. Siehe auch alternative Kommandos in *TEST DBLOG-Kommando*. Siehe auch **Optional Buffers for Codes B and W**.
E oder PF5	**End and Display Log Records**	Deaktiviert die Protokollierung. Der **DBLOG Trace**-Bildschirm mit der aktuellen Aufzeichnung wird angezeigt, wenn im Natural-DBLOG-Puffer Daten existieren. Die aktuelle Protokollierungsdaten werden im Natural-DBLOG-Puffer gehalten. Siehe auch alternative Kommandos in *TEST DBLOG-Kommando*.
S oder	**Snapshot of Specific Adabas Commands**	**Adabas:** Unterbricht ein Programm bei einem angegebenen Adabas-Kommando und zeigt ausführliche Informationen zu diesem Kommando an. Siehe *Snapshot-Funktion für Adabas-Kommandos*.

Code oder PF-Taste	Funktion	Erklärung
PF6		
W oder PF7	**Write Log Records to Work File**	Bewirkt, dass die im Natural-DBLOG-Puffer enthaltenen protokollierten Daten in eine Arbeitsdatei geschrieben werden. Die Dateistruktur (Spalten und Protokollabfolge) entspricht der Struktur des **DBLOG Trace**-Bildschirms. Beschreibung siehe Abschnitt *DBLOG Trace-Bildschirm.*. Wenn Sie diese Funktion wählen, erscheint ein Fenster, in dem Sie aufgefordert werden, folgende Informationen einzugeben: ■ **Angabe der Ziel-Arbeitsdatei** Geben Sie N (No) ein. Dies ist die Standardeinstellung. Sie bewirkt die Datenausgabe in die Natural-Arbeitsdatei 1. Oder, wenn Entire Connection installiert ist: Geben Sie Y (Yes) ein. Dies bewirkt die Datenausgabe in eine PC-Textdatei unter Verwendung der Natural-Arbeitsdatei 7. ■ **Angabe, ob die für den Adabas Control Block protokollierten Daten in die Arbeitsdatei geschrieben werden sollen:** Geben Sie N (No) ein. Dies ist die Standardeinstellung. Sie bewirkt, dass die Daten mit aufgenommen werden. Oder: Geben Sie Y (Yes) ein. Dies bewirkt, dass die Daten nicht mit aufgenommen werden. Siehe auch **Optional Buffers for Codes B and W**.
	Optional Buffers for Codes B and W	Gilt nur bei Adabas-Kommandos. Bewirkt, dass zusätzliche Adabas-Puffer zur Protokollierung ausgewählt werden, wenn Funktionscode B oder W benutzt wird. Siehe *Adabas-Puffer angeben*.
PF3		Das **DBLOG-Menü** wird verlassen. Die aktuellen Protokollierungsaufzeichnungen werden im Natural-DBLOG-Puffer gehalten.
PF12		Löscht den Inhalt des Natural-DBLOG-Puffers. Das **DBLOG-Menü** wird verlassen und die Eingabeaufforderung NEXT wird angezeigt.

Einschränkungen für das Protokollieren angeben

Das **DBLOG-Menü** enthält die folgenden Felder zur Angabe von Auswahlkriterien, mit denen Sie das Protokollieren einschränken können:

Feld	Erklärung
Skip	Gilt nur bei Funktionscode `S`. Anzahl an vom Programm abgesetzten Kommandos, Aufrufen oder Statements, die vor dem Beginn der Protokollierung übersprungen werden sollen.
Program	Einschränkung der Protokollierung auf die vom angegebenen Programm abgesetzten Kommandos, Aufrufen oder Statements.
DBID	Gilt nur bei Adabas-Kommandos. Einschränkung der Protokollierung auf Kommandos, die für die angegebene Datenbank-ID abgesetzt werden.
FNR	Gilt nur bei Adabas-Kommandos. Einschränkung der Protokollierung auf Kommandos, die für die angegebene Dateinummer abgesetzt werden.
Line from **Line to**	Einschränkung der Protokollierung auf Kommandos, Aufrufe oder Statements im Bereich der angegebenen Quellcode-Zeilennummern.
Low Resp **High Resp**	Gilt nur bei Adabas-Kommandos und VSAM-Aufrufen. Einschränkung der Protokollierung auf Kommandos, die einen Antwortcode (Adabas Response Code) oder Rückmeldecode (VSAM Return Code) im angegebenen Bereich zur Folge haben.
Low SQLC **High SQLC**	Gilt nur bei SQL-Statements. Einschränkung der Protokollierung auf Aufrufe, die einen Rückmeldecode (SQL Return Code) im angegebenen Bereich zur Folge haben.

Adabas-Puffer angeben

Gilt nur bei Adabas-Kommandos.

Der Adabas Control Block wird standardmäßig protokolliert. Dabei handelt es sich, in Abhängigkeit vom ausgeführten Kommando, entweder um den klassischen Adabas Control Block (ACB) oder den Extended Adabas Control Block (ACBX). Weitere Informationen siehe Abschnitt *Adabas Control Block Structures (ACB and ACBX)* in der *Adabas for Mainframes*-Dokumentation.

Zusätzlich zum Adabas Control Block können Sie einen oder mehrere Adabas-Puffer protokollieren, die im **DBLOG-Menü** aufgelistet sind:

FB	Format Buffer
RB	Record Buffer
SB	Search Buffer
VB	Value Buffer
IB	ISN Buffer

Um das Protokollieren dieser Puffer zu ermöglichen und den zu protokollierenden Bereich an Bytes anzugeben, können Sie die Eingabefelder unterhalb der Puffernamen benutzen:

- **Bytes 0-79**

  Markieren Sie den oder die zu protokollierenden Puffer, indem Sie ein beliebiges Zeichen neben dem bzw. den gewünschten Puffern eingeben. Wenn Sie in den Feldern **From** und **To** keine Werte eingeben, werden standardmäßig maximal 80 Bytes (von 0 bis 79) pro Puffer protokolliert.

- **From**

  Sie können eine Startnummer (z.B. `100`) eingeben, die bestimmt, ab welchem Byte der Puffer protokolliert wird.

  Wenn ein gesamter Puffer (das Maximum beträgt 32 KB) protokolliert werden soll, geben Sie hier ein `X` oder `*` (Stern) ein und lassen Sie das Feld **To** leer.

- **To**

  Sie können eine Endnummer (z.B. `200`) eingeben, die bestimmt, bis zu welchem Byte der Puffer protokolliert wird. Der Maximale Eingabewert ist `32767` (32 KB).

Falls die Länge des oder der zu protokollierenden Puffer die Gesamthöchstgrenze von 2097151 KB (2 GB - 1 Byte) übersteigt, gibt Natural eine entsprechende Meldung aus, siehe auch *Verarbeitung und Speicherung der Daten*.

Sie können sich die Protokollierungen der Puffer im **DBLOG Trace**-Bildschirm anzeigen lassen, siehe *Adabas-Puffer anzeigen*.

Anmerkung: Standardmäßig protokolliert die Snapshot-Funktion (siehe entsprechenden Abschnitt) *alle* Adabas-Puffer. Deshalb brauchen Sie keinen der optionalen Puffer markieren, bevor Sie diese Funktion ausführen.

43 DBLOG Trace-Bildschirm

- DBLOG Trace-Bildschirm für Adabas-Kommandos 312
- DBLOG Trace-Bildschirm für SQL-Statements 319
- DBLOG Trace-Bildschirm für VSAM-Aufrufe 323

Der **DBLOG Trace**-Bildschirm dient zur Anzeige der aufgezeichneten Protokollierungsdaten zu Adabas-Kommandos, SYNC/ROLB-Aufrufen, SQL-Statements oder VSAM-Aufrufen, die im Natural-DBLOG-Puffer befinden.

Folgende Themen werden behandelt:

DBLOG Trace-Bildschirm für Adabas-Kommandos

- DBLOG Trace-Funktion für Adabas-Kommandos aufrufen
- Spaltenüberschriften und Kommandos für Adabas im DBLOG Trace-Bildschirm
- Adabas-Puffer anzeigen
- Adabas-Kommandos anzeigen, die Multi-Fetch benutzen

DBLOG Trace-Funktion für Adabas-Kommandos aufrufen

Beispiel eines Aufrufs des **DBLOG Trace**-Bildschirms für Adabas-Kommandos:

1. Schreiben Sie das folgende Natural-Programm:

```
DEFINE DATA LOCAL
1 EMP-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
READ (3) EMP-VIEW BY NAME
  DISPLAY NAME
END-READ
END
```

2. Geben Sie das folgende Natural-Systemkommando ein:

```
TEST DBLOG
```

   Die Meldung `DBLOG started now` wird angezeigt.

3. Geben Sie das folgende Natural-Systemkommando ein:

```
RUN
```

   Das Natural-Programm im Source-Bereich wird ausgeführt.

4. Geben Sie erneut ein:

```
TEST DBLOG
```

   Die Protokollierung wird deaktiviert und der **DBLOG Trace**-Bildschirm erscheint (Beispiel):

```
14:14:23             ***** NATURAL TEST UTILITIES *****             2015-11-03
User SAG                        - DBLOG Trace -                   Library SAG
M _________No Cmd ___DB __FNR _Rsp ________ISN _____ISQ CID(Hex) OP_ Pgm_____ Line
_           1  S1    10  2430                          00000000     ATEST2   5470
_           2  RC    10                                00000000 F   ATEST    0220
_           3  L3    10   316             295          00500101  A  LOGTEST  0050
_           4  L3    10   316             621          00500101  A  LOGTEST  0050
_           5  L3    10   316             715          00500101  A  LOGTEST  0050
_           6  RC    10   316                          00500101 SI  LOGTEST  0050
_           7  RC    10                                00000000 F   LOGTEST  0080


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

Spaltenüberschriften und Kommandos für Adabas im DBLOG Trace-Bildschirm

Die folgende Tabelle enthält die Beschreibungen der Felder im **DBLOG Trace**-Bildschirm und die Blätterkommandos, die in dem Bildschirm oder in einem der in diesem Bildschirm aufrufbaren Fenster zur Anzeige der Puffer zur Verfügung stehen, siehe *Adabas-Puffer anzeigen*. Zur Kommandoausführung können Sie entweder eine PF-Taste drücken oder ein Direktkommando in der Kommandozeile absetzen.

Spalte	PF-Taste bzw. Direktkommando	Erklärung
M		Eingabemöglichkeit für Zeilenkommandos, mit denen zusätzliche Fenster aufgerufen werden, die ausführliche Informationen zu Puffern enthalten, siehe *Adabas-Puffer anzeigen*.
No		Folgenummer. Die Kommandos werden in der Reihenfolge angezeigt, in der sie ausgeführt wurden.
Cmd		Adabas-Kommando. Wenn vor dem Kommando ein Stern steht (z.B. ***S1**), wurde der Datenbankaufruf in **ACBX-Form** gestartet.
DB		Datenbankkennung (DBID).
FNR		Dateinummer (FNR).
Rsp		Antwortcode (Adabas Response Code).
ISN		Interne Folgenummer (Internal Sequence Number) eines Datensatzes.
ISQ		ISN Quantity (Anzahl).
CID		Kommandokennung (Command ID).
CID (Hex)		Kommandokennung (Command ID) im Hexadezimalformat.

Spalte	PF-Taste bzw. Direktkommando	Erklärung
OP		Adabas-Kommando-Optionen 1 und 2. Wenn die linke Option ein Kleiner als Zeichen ist (z.B. <A), wurde der Datensatz statt aus der Datenbank aus einem Multi-Fetch-Puffer geholt.
Pgm		Programmname.
Line		Quellcode-Zeilennummer.
	PF2	Hardcopy eines Bildschirms drucken.
	PF3	Der **DBLOG Trace**-Bildschirm wird verlassen oder ein Pufferanzeigefenster wird geschlossen. Die aktuellen Datensätze verbleiben im Natural-DBLOG-Puffer.
	PF5	Verschiebt die Protokolleinträge an den oberen Rand des Bildschirms: Positionieren Sie den Cursor in der Spalte **M** neben gewünschtes Kommando und Folgenummer in Spalte **No** und drücken Sie PF5. Die Protokolle werden neu positioniert, beginnend mit der gewählten Folgenummer.
	PF6 oder - -	An den Anfang einer Liste oder der Daten in einem Puffer-Anzeigefenster blättern.
	PF7 oder -	In einer Liste oder den Daten in einem Puffer-Anzeigefenster eine Seite nach oben blättern.
	PF8 oder +	In einer Liste oder den Daten in einem Puffer-Anzeigefenster eine Seite nach unten blättern.
	PF9 oder ++	An das Ende einer Liste oder den Daten in einem Puffer-Anzeigefenster blättern.
	PF10	Nur in einem Puffer-Anzeigefenster mit mehreren Datensatz-/Formatpuffern vorhanden. Anzeige des vorangegangenen Datensatz-/Formatpuffers.
	PF11	Nur in einem Puffer-Anzeigefenster mit mehreren Datensatz-/Formatpuffern vorhanden. Anzeige des nächsten Datensatz-/Formatpuffers.
	PF12	Löschen des Natural-DBLOG-Puffers und Deaktivieren der Protokollierung.

Adabas-Puffer anzeigen

Der Adabas Control Block wird standardmäßig aufgezeichnet. Um einen oder mehrere Adabas-Puffer aufzuzeichnen, müssen Sie den oder die gewünschten Puffer im **DBLOG-Menü** markieren, bevor Sie die Protokollierungsfunktion ausführen (siehe *Adabas-Puffer angeben*). Wenn Sie beispielsweise im **DBLOG-Menü** nur den Formatpuffer markiert haben, können Sie nur das **Format Buffer**-Fenster, aber nicht das **Record Buffer**-Fenster anzeigen.

Um den Control Block oder Puffer-Informationen anzuzeigen:

1 Geben Sie im Eingabefeld neben dem benötigten Kommando das Zeilenkommando (s.u.) ein, das dem gewünschten Puffer entspricht und drücken SieENTER:

Zeilenkommando	Angeforderter Puffer
C	Control Block
F	Format Buffer
R	Record Buffer
S	Search Buffer
V	Value Buffer
I	ISN Buffer
.	Durch Eingabe eines Punkts (.) wird der **DBLOG Trace**-Bildschirm verlassen. Die aktuellen Protokollaufzeichnungen bleiben im Natural-DBLOG-Puffer erhalten.

Es erscheint ein Fenster, in dem die protokollierten Daten des Control Buffer bzw. des angeforderten Puffers angezeigt werden. Wenn Sie mehrere Zeilenkommandos eingegeben haben, können Sie sich durch Drücken von PF3 den Control Block bzw. den Puffer des nächsten Kommandos anzeigen lassen.

Das Fenster in dem folgenden Beispiel enthält Daten eines Datensatzpuffers (Record Buffer):

```
16:50:05             ***** NATURAL TEST UTILITIES *****             2015-11-03
User SAG                        - DBLOG Trace -                   Library SAG
M _________No Cmd ___DB __FNR _Rsp ________ISN ____ISQ CID(Hex) OP_ Pgm_____ Line
_          1  S1    10  2430                          00000000     ATEST2   5470
_          2  RC    10                                00000000 F   ATEST    0220
_          3  L3    10   316             295          00500101  A  LOGTEST  0050
R          4  L3    10   316             621          00500101  A  LOGTEST  0050
_ +-----------------Page 1 of 1 (logged range:0x-0x4F)------------------+ 050
_ | _____ Seq No 4          Record Buffer (length:0x14)                  | 050
_ | 0000 * C1C6C1D5 C1E2E2C9 C5E54040 40404040 * AFANASSIEV       * 0000  | 080
_ | 0010 * 40404040 00000000 00000000 00000000 *                  * 0010  | 110
_ +-------------------------------------------------------------------------+ 270


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

Beschreibung der Felder im Fenster:

Feld	Erklärung
Page	Nummer der aktuellen Seite und die Gesamtzahl an Seiten, die für den Puffer generiert wurden (im oben gezeigten Beispiel `Page 1 of 1` (Seite 1 von 1).
logged range	Tatsächlich protokollierte Pufferlänge in Hexadezimalformat (im obigen Beispiel `0x0-0x4F`).
Seq No	Folgenummer des Kommandos. Im oben gezeigten Beispiel wurde das Kommando an dritter Stelle (`3`) ausgeführt.
buffer-type *num-current*/*num-total*	*buffer-type* kennzeichnet den Typ des angeforderten Puffers. Zusätzlich wird bei einem Format- oder Datensatzpuffer (Format oder Record Buffer) die Anzahl der Datensatz- oder Formatpuffer angezeigt. *num-current*: Bezeichnet die Nummer des aktuell angezeigten Datensatz-/Formatpuffers. *num-total*: Bezeichnet die Gesamtzahl der der protokollierten Record/Format Buffers (Datensatz-/Formatpuffer). Bei einem Datenbankaufruf unter Verwendung des **Extended Adabas Control Block** (ACBX) werden mehrere Format-/Datensatzpuffer protokolliert. Im oben gezeigten Beispiel wird der erste Datensatz von insgesamt 13 Datensätzen gezeigt (`1 / 13`). Weitere Informationen zu ACBX siehe *Adabas Control Block Structures (ACB and ACBX)* in der *Adabas for Mainframes*-Dokumentation.

Feld	Erklärung	
length	Gesamtlänge des Datensatzes in Hexadezimalformat (im obigen Beispiel `0x7A`).	
_____	Im Eingabefeld neben **Seq No** können Sie eines der folgenden Zeilenkommandos eingeben:	
	`C`	Anzeige des Control Block.

	`F`
	
	oder
	`R`
		Anzeige des Format Buffer(`F`) oder Record Buffer (`R`).
		
		Wenn Paare von Format und Record Buffers vorhanden sind, wird durch die Eingabe von `F` in einem Record Buffer oder `R` in einem Format Buffer der passende Record Buffer bzw. Format Buffer angezeigt. Wenn z. B. gerade der zweite Record Buffer angezeigt wird, ruft die Eingabe von `F` ein Fenster mit dem entsprechenden zweiten Format Buffer auf.

	`I`	Anzeige des ISN Buffer.
	`S`	Anzeige des Search Buffer.
	`V`	Anzeige des Value Buffer.
	`buffer-number`	Sie können die Nummer des Record/Format Buffer eingeben, der gezeigt werden soll. Siehe auch Schritt 2 weiter unten.
	.	Durch Eingabe eines Punkts (.) wird das aktuelle Puffer-Anzeigefenster geschlossen.

2 In einem Record/Format Buffer-Anzeigefenster, das mehrere Datensatz-/Formatpuffer enthält, können Sie eine der folgenden Methoden anwenden, um jeden einzelnen Record/Format Buffer anzuzeigen:

Drücken Sie PF10, um den vorangegangenen Record/Format Buffer anzuzeigen.

Oder:

Drücken Sie PF11, um den nächsten Record/Format Buffer anzuzeigen.

Oder:

Geben Sie im Eingabefeld _____ die Nummer ein, die dem Record/Format Buffer entspricht, den Sie sich ansehen möchten.

Adabas-Kommandos anzeigen, die Multi-Fetch benutzen

Wenn in einem `FIND`-, `READ`- oder HISTOGRAM-Statement die MULTI-FETCH-Klausel verwendet wird, greifen tatsächlich nur die Adabas-Kommandos auf die Datenbank zu, die einen Satz Records (Datensätze) abrufen. Die abgerufenen Datensätze werden in einen Multi-Fetch-Puffer verschoben, aus dem sie während der Ausführung der Datenbankabfrageschleife geholt werden. Der nächste Datenbankaufruf erfolgt erst, wenn der nächste Satz Datensätze abgerufen wird.

Weitere Informationen siehe *MULTI-FETCH-Klausel* im *Leitfaden zur Programmierung*.

Im **DBLOG Trace**-Bildschirm werden sowohl Datenbankaufrufe als auch Nicht-Datenbankaufrufe aufgelistet. Zur Kennzeichnung wird an der ersten Stelle in der Spalte **OP** ein `M` bei Datenbankaufrufen und ein Kleiner als Zeichen (`<`) bei Nicht-Datenbankaufrufen des Multi-Fetch-Puffers angezeigt. Dies wird im folgenden Beispiel veranschaulicht:

Beispiel für ein Adabas-Kommando mit Multi-Fetch

Führen Sie DBLOG für das folgende Natural-Programm mit dem Namen `MFETCH` aus:

```
DEFINE DATA LOCAL
1 EMP-VIEW VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
*
READ (5)  MULTI-FETCH OF 3  EMP-VIEW BY NAME = 'ADKINSON'
  DISPLAY *COUNTER NAME
END-READ
END
```

Der **DBLOG Trace**-Bildschirm erscheint (Beispiel):

```
10:04:46             ***** NATURAL TEST UTILITIES *****             2015-11-03
User SAG                       - DBLOG Trace -                  Library SAG
M _________No Cmd ___DB __FNR _Rsp _______ISN ____ISQ CID(Hex) OP_ Pgm_____ Line
_          1  S1    10  2430                          00000000     ATEST2   5470
_          2  RC    10                                00000000 F   ATEST    0220
_          3  L3    10   316            295          00600101 MA  MFETCH   0060
_          4  L3    10   316            621          00600101 <A  MFETCH   0060
_          5  L3    10   316            715          00600101 <A  MFETCH   0060
_          6  L3    10   316            535          00600101 MA  MFETCH   0060
_          7  L3    10   316           1038          00600101 <A  MFETCH   0060
_          8  RC    10   316                         00600101 SI  MFETCH   0060
_          9  RC    10                                00000000 F   MFETCH   0090


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

Die in der Liste unter den Folgenummern 3 und 6 aufgeführten L3-Kommandos rufen einen Satz Records (Datensätze) aus der Datenbank ab (angezeigt durch ein M in der Spalte **OP**) und geben den ersten Datensatz an das Programm zurück. Die übrigen Datensätze werden im Multi-Fetch-Puffer zwischengespeichert.

Die unter den Folgenummern 4, 5 und 7 aufgeführten L3-Kommandos rufen den Datensatz aus dem Multi-Fetch-Puffer ab (angezeigt durch < in der Spalte **OP**) und geben ihn an das Programm zurück.

Inhalt des Record Buffer bei Multi-Fetch-Datenbankaufrufen

Der Record Buffer eines Datenbankaufrufs, der Multi-Fetch benutzt, enthält die Daten aller von der Datenbank abgefragten Datensätze. Sie werden in der Reihenfolge aufgelistet, in der sie verarbeitet werden.

Beim Laden eines Satzes an Datensätzen überschreibt Adabas den Record Buffer ab dem ersten Byte bis zur Größe der Datensätze, die von der Datenbank zurückgegeben werden. Der übrig bleibende Platz im Puffer wird nicht gelöscht, sondern enthält immer noch Daten von alten Datensätzen, die während einem vorangegangenen Datenbankaufruf geladen wurden. Dies bedeutet zum Beispiel, dass, wenn ein als NAME(A20) definiertes Feld gelesen wird und ein Multi-Fetch-Faktor von 5 benutzt wird, der Record Buffer eine Länge von 100 (20 * 5) Bytes hat. Wenn nur 3 Datensätze von der Datenbank zurückgegeben werden, wird der Record Buffer nur mit den ersten 3 Datensätzen richtig gefüllt (Bytes 1 bis 60), während die letzten 2 Datensätze (Bytes 61 - 100) unverändert bleiben.

DBLOG Trace-Bildschirm für SQL-Statements

- DBLOG Trace-Funktion für SQL-Statements aufrufen
- Spaltenüberschriften und Kommandos für SQL-Statements im DBLOG Trace-Bildschirm

DBLOG Trace-Funktion für SQL-Statements aufrufen

Beispiel eines Aufrufs des **DBLOG Trace**-Bildschirms für SQL-Statements.

1. Schreiben Sie folgendes Natural-Programm:

```
DEFINE DATA LOCAL
01 EMP VIEW OF DSN8810-EMP
  02 EMPNO
  02 FIRSTNME
  02 MIDINIT
  02 LASTNAME
  02 EDLEVEL
  02 SALARY
01 EMPPROJACT VIEW OF DSN8810-EMPPROJACT
```

```
  02 EMPNO
  02 PROJNO
  02 ACTNO
  02 EMPTIME
END-DEFINE
FIND (1) EMP WITH EMPNO > '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0150)
    MOVE 0.75 TO EMPTIME
    UPDATE
  END-FIND
  ADD 1 TO EDLEVEL
  UPDATE
END-FIND
*
FIND (1) EMP WITH EMPNO > '000300'
  FIND (1) EMPPROJACT WITH EMPNO = EMPNO(0240)
    DISPLAY EMPPROJACT EMP.EDLEVEL
  END-FIND
END-FIND
ROLLBACK
END
```

2. Geben Sie das folgende Natural-Systemkommando ein:

```
TEST DBLOG Q
```

   Die Meldung `DBLOG started now` wird angezeigt.

3. Geben Sie das folgende Natural-Systemkommando ein:

```
RUN
```

   Das Natural-Programm im Source-Bereich wird ausgeführt.

4. Geben Sie erneut ein:

```
TEST DBLOG Q
```

   Die Protokollierung wird deaktiviert. Der **DBLOG Trace**-Bildschirm für SQL-Statements wird angezeigt (Beipiel):

```
11:28:58              ***** NATURAL Test Utilities *****             2008-07-28
User SAG                          - DBLOG Trace -                 Library SAG
M No   R SQL Statement (truncated)     CU SN SREF M Typ SQLC/W Program  Line LV
_    1   SELECT EMPNO,FIRSTNME,MIDINIT 01 01 0150 D DB2         LOGSQL   0150 01
_    2     FETCH CURSOR NEX            01 01 0150 D DB2         LOGSQL   0150 01
_    3   SELECT EMPNO,PROJNO,ACTNO,EMP 02 02 0160 D DB2         LOGSQL   0160 01
_    4     FETCH CURSOR NEX            02 02 0160 D DB2         LOGSQL   0160 01
_    5   UPDATE DSN8810.EMPPROJACT SET 02 03 0160 D DB2         LOGSQL   0180 01
_    6   CLOSE CURSOR                  02 02 0160 D DB2         LOGSQL   0160 01
_    7   UPDATE DSN8810.EMP SET EDLEVE 01 04 0150 D DB2         LOGSQL   0210 01
_    8   CLOSE CURSOR                  01 01 0150 D DB2         LOGSQL   0150 01
_    9   SELECT EMPNO,FIRSTNME,MIDINIT 05 05 0240 D DB2         LOGSQL   0240 01
_   10     FETCH CURSOR NEX            05 05 0240 D DB2         LOGSQL   0240 01
_   11   SELECT EMPNO,PROJNO,ACTNO,EMP 06 06 0250 D DB2         LOGSQL   0250 01
_   12     FETCH CURSOR NEX            06 06 0250 D DB2         LOGSQL   0250 01
_   13   CLOSE CURSOR                  06 06 0250 D DB2         LOGSQL   0250 01
_   14   CLOSE CURSOR                  05 05 0240 D DB2         LOGSQL   0240 01
_   15   ROLLBACK                      00 00 0000 D DB2         LOGSQL   0290 01
_
_
Command ===>


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

Spaltenüberschriften und Kommandos für SQL-Statements im DBLOG Trace-Bildschirm

In der folgenden Tabelle werden die Spaltenüberschriften im **DBLOG Trace**-Bildschirm für SQL-Statements erklärt.

Um ein Kommando auszuführen, können Sie entweder eine PF-Taste drücken oder ein Direktkommando in der Kommandozeile eingeben.

Spalte	PF-Taste / Direktkommando	Erklärung	
M		Eingabefelder für Zeilenkommandos:	
		E	Ausführung des Kommandos `EXPLAIN` zwecks Anzeige von Informationen über die Wahl der Strategie, die der Db2 Optimizer zur Ausführung von SQL-Statements gewählt hat. Siehe auch Unterabschnitt *Verwendung des Kommandos EXPLAIN bei Natural for Db2* in der Dokumentation zum Kommando *LISTSQL* in der *Systemkommandos* -Dokumentation.

Spalte	PF-Taste Direktkommando	Erklärung
		<table><tr><td></td><td></td></tr><tr><td>L</td><td>Ausführung des Kommandos LISTSQL. Dieses listet die Natural-Statements im Quellcode eines Natural-Objekts auf und zeigt die entsprechenden SQL-Kommandos, in die sie übersetzt wurden. Ein SQL Statement wird identifiziert durch den Library-Namen, den Programmnamen und die Zeilennummer, die aus dem Natural-DBLOG-Puffer genommen werden. Siehe auch *LISTSQL Command* in der *Systemkommandos*-Dokumentation.</td></tr></table> **Wichtig:** Da beide Kommandos ihre Informationen aus der Natural-Systemdatei beziehen, können unerwünschte Ergebnisse auftreten, wenn das entsprechende Natural-Programm neu katalogisiert worden ist, nachdem die Protokollierungsfunktion mit dem Kommando TEST DBLOG Q ausgeführt wurde. Diese unerwünschten Ergebnisse können durch Statements verursacht werden, die nach der Protokollierung verändert wurden.
No		Folgenummer. Die Statements werden in der Abfolge ihrer erfolgten Ausführung angezeigt.
R		Gilt nur, wenn der Natural File Server for Db2 benutzt wird. Ein Stern vor dem betreffenden Statement zeigt an, dass eine Neuselektion durchgeführt wurde, andernfalls bleibt das Feld leer. Siehe auch *Concept of the File Server* in der *Natural for Db2*-Dokumentation.
SQL Statement		Die ersten 29 Zeichen der protokollierten SQL-Statement.
CU		Cursor-Nummer.
SN		Interne Statement-Nummer.
SREF		Statement-Referenznummer.
M		Modus: D für dynamisch oder S für statisch.
Typ		Datenbanktyp: Db2 oder /DS.
SQLC/W		Entweder der Rückgabecode (SQL Return Code) im SQLCODE-Feld der SQLCA oder die Warnung im SQLWARN0-Feld der SQLCA, wenn SQLCODE 0 ist.
Pgm		Natural-Programmname.
Line		Quellcode-Zeilennummer.
LV		Programmebene (Level)
	PF2	Ausdrucken einer Hardcopy des Bildschirms.

Spalte	PF-Taste Direktkommando	Erklärung
	PF3	Beenden der **DBLOG Trace**-Funktion. Die aktuellen Protokolle werden im Natural-DBLOG-Puffer beibehalten.
	PF4	An den Listenanfang blättern.
	PF5	Protokolleinträge an den oberen Rand des Bildschirms verschieben: Positionieren Sie den Cursor in der Spalte **M** neben gewünschtes Kommando und Folgenummer in Spalte **No** und drücken Sie PF5. Die Protokolle werden neu positioniert, beginnend mit der gewählten Folgenummer.
	PF6	An das Listenende blättern.
	PF7 oder -	In der Liste eine Seite nach oben blättern.
	PF8 oder +	In der Liste eine Seite nach unten blättern.
	PF12	Löschen des Natural-DBLOG-Puffers und Deaktivieren der Protokollierung.

DBLOG Trace-Bildschirm für VSAM-Aufrufe

- Invoking DBLOG Trace for VSAM-Aufrufe
- Spaltenüberschriften im DBLOG Trace-Bildschirm

Invoking DBLOG Trace for VSAM-Aufrufe

Beispiel eines Aufrufs des **DBLOG Trace**-Bildschirms für VSAM-Aufrufe.

1. Schreiben Sie folgendes Natural-Programm:

```
DEFINE DATA LOCAL
01 EMPVSAM VIEW OF EMPLOYEES-VS
   02 NAME (A20)
   02 CITY (A20)
   02 PERSONNEL-ID (A8)
END-DEFINE
READ (10) EMPVSAM BY PERSONNEL-ID
   WRITE PERSONNEL-ID NAME CITY
END-READ
END
```

2. Geben Sie das folgende Natural-Systemkommando ein:

```
TEST DBLOG V
```

   oder

```
TEST DBLOG V START
```

   Die Meldung `DBLOG started now` wird angezeigt.
3. Geben Sie das folgende Natural-Systemkommando ein:

```
RUN
```

   Das im Source-Bereich enthaltene Natural-Programm wird ausgeführt.
4. Geben Sie erneut ein:

```
TEST DBLOG V
```

   oder

```
TEST DBLOG V SHOW
```

   Die Protokollierung wird deaktiviert und der **DBLOG Trace**-Bildschirm für VSAM-Aufrufe erscheint (Beispiel):

```
11:21:41          ***** Natural Test Utility *****                2019-10-14
User SAG                  - DBLOG V Trace -                   Library SAG

Seq Number VSAMCall Cmd Resp NRet Key (10)   RecID(10)  FileName Program  Line
---------- -------- --- ---- ---- ---------- ---------- -------- -------- ----

         1 SETL     L3                                  EMPLVS   VSAM01   0050
         2 GETNXT   L3           11100102   11100102   EMPLVS   VSAM01   0050
         3 GETNXT   L3           11100105   11100105   EMPLVS   VSAM01   0050
         4 ESETL    RC                                  EMPLVS   VSAM01   0050

                             ***** End of Log *****
```

Spaltenüberschriften im DBLOG Trace-Bildschirm

In der folgenden Tabelle werden die Spaltenüberschriften im **DBLOG Trace**-Bildschirm für VSAM-Aufrufe erklärt.

Spalte	Erklärung
Seq Number	Folgenummer. Die Aufrufe werden in der Reihenfolge angezeigt, in der sie ausgeführt wurden.
VSAMCmd	VSAM-Aufruf wird abgesetzt.
Cmd	Äquivalentes Adabas-Kommando.
Resp	Rückgabecode nach dem VSAM-Aufruf. c
NRet	Natural-Rückgabecode, falls vorhanden.
Key (10)	Erste 10 Bytes des VSAM-Schlüssels.
RecID (10)	Erste 10 Bytes der Datensatzkennung (Record ID).
FileName	VSAM-Datei, auf die zugegriffen wurde.
Program	Name des Natural-Programms.
Line	Quellcode-Zeilennummer des Natural-Programms.

44 DBLOG Snapshot-Funktion

- Snapshot-Funktion für Adabas-Kommandos 328
- Snapshot-Funktion für SQL-Statements 330
- Snapshot-Funktion für VSAM-Aufrufe 332

Die Snapshot-Funktion liefert ausführliche Informationen zu einem bestimmten Adabas-Kommando, SQL-Statement oder VSAM-Aufruf.

Folgende Themen werden behandelt:

Snapshot-Funktion für Adabas-Kommandos

Die Snapshot-Funktion unterbricht die Programmausführung nach der Ausführung des ersten Adabas-Kommandos, das die Auswahlkriterien erfüllt, die im **DBLOG-Menü** angegeben wurden. Der für das angegebene Adabas-Kommando generierte **Snapshot Report** (siehe folgenden Beispiel-Bildschirm) wird unmittelbar nach der Programmunterbrechung angezeigt.

Die Snapshot-Funktion protokolliert automatisch *alle* Adabas-Puffer. Deshalb ist vor dem Start der Snapshot-Funktion kein Markieren der optionalen Puffer im **DBLOG-Menü** nötig. Der Standard-**Snapshot Report** zeigt den Steuerblock (CB), bei dem es sich entweder um den klassischen Steuerblock (ACB) oder den erweiterten Steuerblock (ACBX) handelt.

Folgende Themen werden behandelt:

- Snapshot Report für Adabas-Kommandos aufrufen
- Puffer im Snapshot Report anzeigen

Snapshot Report für Adabas-Kommandos aufrufen

Um den Snapshot Report-Bildschirm für Adabas-Kommandos aufzurufen:

1. Geben Sie im **DBLOG-Menü** ein Adabas-Kommando und, falls gewünscht, zusätzliche Kriterien an und geben Sie den Funktionscode `S` ein.

   Die Meldung `DBLOG snapshot facility started now` erscheint und zeigt an, dass die DBLOG-Snapshot-Funktion jetzt gestartet wurde.
2. Führen Sie ein Natural-Programm aus, das das im **DBLOG-Menü** angegebene Adabas-Kommando enthält.

   Die Programmausführung wird gestoppt und ein **Snapshot Report**-Bildschirm erscheint (Beispiel):

```
16:36:39            ***** NATURAL TEST UTILITIES *****             2006-12-12
                             - Snapshot Report -

Command Code : L3           Command ID   :  ??? 00200101 File Number  : 013C
Response Code:      0       ISN          :          1300
ISN Low Limit: 00000000     ISN Quantity :             0
FB Length    : 0009         RB Length    : 0014          SB Length    : 0008
VB Length    : 0014         IB Length    : 0000          Com. Option 1:
Com. Option 2: V            Additions 1  : AE]?          Additions 2  :  ? ?
Additions 3  :              Additions 4  :
Global FID   : 0000000000000000 Command Time : 00000019 Pgm: SAGTEST  Lin: 0020
Control Block
0000 * 30D5D3F3 00200101 013C0000 00000514 * ?NL3 ?????    ?? * 0000
0010 * 00000000 00000000 00090014 00080014 *          ? ? ? ? * 0010
0020 * 000000E5 C1C5BBCA 40404040 00120014 *    VAE]?     ? ? * 0020
0030 * 00000000 00000000 00000000 00000000 *                  * 0030
0040 * 00000000 00000000 00000019 00000000 *            ?     * 0040
0050 * 00000000 00000000 00000000 00000000 *                  * 0050
0060 * 00000000 00000000 00000000 00000000 *                  * 0060
0070 * 00000000 00000000 00000000 00000000 *                  * 0070


Command ===> CB_____________________________________________________________
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  CB    FB    RB     -     +    SB    VB    IB    Canc
```

Puffer im Snapshot Report anzeigen

Standardmäßig zeigt der **Snapshot Report**-Bildschirm den Steuerblock (CB). Sie können die entsprechend belegte PF-Taste drücken oder in der Kommandozeile das entsprechende Direktkommando eingeben, um andere Adabas-Puffer anzuzeigen oder durch einen Report zu blättern (siehe unten).

Die Verfügbarkeit einer PF-Tastenfunktion ist vom zurzeit angezeigten Puffer abhängig. Wenn ein Puffer mehr als eine Bildschirmseite umfasst oder mehrere Format-/Record Buffer enthält, sind in dem Bildschirm entsprechende mit Blätterfunktionen belegte PF-Tasten vorhanden.

PF-Taste	Direktkommando	Puffer
PF4	`CB`	Anzeige des Control Block. Dies ist die Standardeinstellung.
PF5	`FB`	Anzeige des Format Buffer.

PF6	`RB`
	
	oder
	`--`
		Anzeige des Record Buffer (`RB`) oder Blättern (`--`) an den Anfang einer langen Pufferanzeige.

PF7	`-`	In einer langen Pufferanzeige eine Seite nach oben blättern.
PF8	`+`	In einer langen Pufferanzeige eine Seite nach unten blättern.

PF-Taste	Direktkommando	Puffer
PF9	SB oder ++	Anzeige des Search Buffer (SB) oder Blättern (++) an das Ende einer langen Pufferanzeige.
PF10	VB oder <	Anzeige des Wertpuffers (VB). Bei mehreren Format/Record Buffers Anzeige des vorigen (<) Format/Record Buffer.
PF11	IB oder >	Anzeige des ISN Buffer (IB). Bei mehreren Format/Record Buffers Anzeige des nächsten (>) Format/Record Buffer.

Informationen zu den Feldern, die in einem Steuerblock oder Puffer abgezeigt werden, sind im Abschnitt *Adabas-Puffer anzeigen* enthalten.

Snapshot-Funktion für SQL-Statements

Diese Snapshot-Funktion generiert den **Snapshot Report** (s. Beispiel-Bildschirm weiter unten) des ersten SQL-Statement, das die Auswahlkriterien erfüllt, die im **DBLOG-Menü** angegeben wurden. Bei einem Snapshot wird der Programmfluss nicht unterbrochen.

Im Gegensatz zu den Statements, die mit der DBLOG-Trace-Funktion angezeigt werden, zeigt der Snapshot das Statement in seiner Gesamtheit (begrenzt auf 13 Zeilen).

Die Snapshot-Daten werden im Natural-DBLOG-Puffer gehalten und werden nur dann angezeigt, wenn der Benutzer das entsprechende DBLOG-Kommando eingibt (s. unten).

Folgende Themen werden behandelt:

- Snapshot Report für SQL-Statements aufrufen
- Snapshot Report-Informationen für SQL-Statements

Snapshot Report für SQL-Statements aufrufen

Um den Snapshot Report-Bildschirm für SQL-Statements aufzurufen:

1 Geben Sie im **DBLOG-Menü** ein SQL-Statement und, falls gewünscht, zusätzliche Kriterien an und geben Sie den Funktionscode S ein.

Die Meldung `DBLOG snapshot facility started now` erscheint und zeigt an, dass die DBLOG-Snapshot-Funktion jetzt gestartet wurde.

2 Führen Sie ein Natural-Programm aus, das das im **DBLOG-Menü** angegebene SQL-Statement enthält. (Die Protokolldaten werden in den Natural-DBLOG-Puffer geschrieben.)

3 Geben Sie zur Anzeige der Snapshot-Daten das folgende Kommando ein:

```
TEST DBLOG Q
```

Oder:

Geben Sie im **DBLOG-Menü** den Funktionscode `E` ein.

Der **Snapshot Report**-Bildschirm für SQL-Statements erscheint (Beispiel):

```
10:59:28              ***** NATURAL Test Utilities *****            2006-12-12
User SAG                       - Snapshot Report -              Library SAG

CU SN M Typ R SQLC/W     Library  Program  Store Clock Value   Line LV CID(Hex)
01 01 D DB2              SAG      SAGTEST  2002/04/03 14:23:06 0150 01 01500101

SQL Statement
SELECT EMPNO,FIRSTNME,MIDINIT,LASTNAME,EDLEVEL,SALARY FROM DSN8510.EMP WHERE EM
PNO > '000300' FOR UPDATE OF EDLEVEL




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

Snapshot Report-Informationen für SQL-Statements

Der **Snapshot Report**-Bildschirm für SQL-Statements enthält folgende Informationen:

Spalte	Erklärung
CU	Cursor-Nummer.
SN	Interne Statement-Nummer.
M	Modus: `D` für dynamisch oder `S` für statisch.
Typ	Datenbanktyp: Db2.
R	Gilt nur bei Benutzung des Natural File Server for Db2. Ein Stern vor dem betreffenden Statement zeigt an, dass eine Neuselektion durchgeführt wurde, andernfalls bleibt das Feld leer. Siehe auch *Concept of the File Server* in der *Natural for Db2*-Dokumentation.
SQLC/W	Entweder der SQL-Rückgabecode im Feld SQLCODE der `SQLCA` oder die Warnung im Feld SQLWARN0 der `SQLCA`, wenn `SQLCODE 0` ist.

Spalte	Erklärung
Library	Library, in der das Natural-Programm mit dem protokollierten Statement katalogisiert wurde.
Program	Name des Natural-Programms, das das protokollierte Statement enthält.
Store Clock Value	Zeitstempel des Natural-Programms, das das protokollierte Statement enthält.
Line	Quellcode-Zeilennummer des protokollierten Statement.
LV	Aufrufebene (Level) des Natural-Programms, das das protokollierte Statement enthält.
CID (Hex)	Kommandokennung (Command ID) des protokollierten Statement im Hexadezimalformat.

Snapshot-Funktion für VSAM-Aufrufe

Diese Snapshot-Funktion generiert den **Snapshot Report** (s. Beispiel-Bildschirm weiter unten) des ersten VSAM-Aufrufs, der die Auswahlkriterien erfüllt, die im **DBLOG-Menü** angegeben wurden. Bei einem Snapshot wird der Programmfluss nicht unterbrochen.

Die Snapshot-Daten werden im Natural-DBLOG-Puffer gehalten und werden nur dann angezeigt, wenn der Benutzer das entsprechende `DBLOG`-Kommando eingibt (s. unten).

Folgende Themen werden behandelt:

- Snapshot Report für VSAM-Aufrufe aufrufen
- Snapshot Report-Informationen für VSAM-Aufrufe

Snapshot Report für VSAM-Aufrufe aufrufen

Um den Snapshot Report-Bildschirm für VSAM-Aufrufe aufzurufen:

1 Geben Sie im **DBLOG-Menü** einen VSAM-Aufruf und, falls gewünscht, zusätzliche Kriterien an und geben Sie den Funktionscode `S` ein.

Die Meldung `DBLOG snapshot facility started now` erscheint und zeigt an, dass die DBLOG-Snapshot-Funktion jetzt gestartet wurde.

2 Führen Sie ein Natural-Programm aus, das den im **DBLOG-Menü** angegebenen VSAM-Aufruf enthält. (Die Protokolldaten werden in den Natural-DBLOG-Puffer geschrieben.)

3 Geben Sie zur Anzeige der Snapshot-Daten das folgende Kommando ein:

```
TEST DBLOG V
```

oder

```
TEST DBLOG V SHOW
```

Oder:

Geben Sie im **DBLOG-Menü** den Funktionscode `E` ein.

Der **Snapshot Report**-Bildschirm für VSAM-Aufrufe (Beispiel) erscheint:

```
11:19:53            ***** Natural Test Utility *****                2019-10-14
User SAG                    - Snapshot Report -                   Library SAG

       VSAM Call..... : GETNXT        CID(Hex).... : 00500101
       Flag 02  ..... : 32            CMD Code.... : L3
       Flag 03  ..... : 00            FILENAME.... : EMPLVS
       Flag 04  ..... : 00            PATHNAME.... :
       Flag 06  ..... : 90            RETCODE..... : 0000      (Hex)
       VSAM Key ..... : 11100102      ProbCode.... : 00000000  (Hex)
       NAT Pgm  ..... : VSAM01        Key Len..... :     8
       PgmLine  ..... : 0050          Nat Resp.... :
       FileType ..... : KSDS          DBID/FNR.... :   254 /     1

 Record:  CHAR : 11100102EDGAR               PETER               SCHINDLER
         HEX  : FFFFFFFFCCCCD4444444444444444DCECD4444444444444ECCCDCDCD44444
                1110010254719000000000000000753590000000000000023895435900000

 CBA   :  CHAR :  VL3 &????               ? ? ? ?   AAANVSL3L
         HEX  : 0EDF0500F0000000000000000000000000CCCDEEDFD000000000000000000
                05330011E1000000000000000708070800011155233300000000000000000

 SB    :  CHAR : AA,8,A.
         HEX  : CC6F6C4000000000000000000000000000000000000000000000000000000
                11B8B1B000000000000000000000000000000000000000000000000000000

 FB    :  CHAR : AA,8,A.
         HEX  : CC6F6C4000000000000000000000000000000000000000000000000000000
                11B8B1B000000000000000000000000000000000000000000000000000000

 RB    :  CHAR :
         HEX  : 4444444400000000000000000000000000000000000000000000000000000
                0000000000000000000000000000000000000000000000000000000000000

 ISN   :  CHAR :
         HEX  : 0000000000000000000000000000000000000000000000000000000000000
                0000000000000000000000000000000000000000000000000000000000000

 VB    :  CHAR :
         HEX  : 0000000000000000000000000000000000000000000000000000000000000
                0000000000000000000000000000000000000000000000000000000000000
```

Snapshot Report-Informationen für VSAM-Aufrufe

Der **Snapshot Report**-Bildschirm für VSAM-Aufrufe enthält folgende Informationen:

Feld	Erklärung
VSAM Call	Ausgebener VSAM-Aufruf.
File 02	Flag 02 vom DCR Copybook.
Flag 03	Flag 03 vom DCR Copybook.
Flag 04	Flag 04 vom DCR Copybook.
Flag 06	Flag 06 vom DCR Copybook.
VSAM Key	Erste 8 Bytes des VSAM-Schlüssels.
NAT Pgm	Natural-Programmname.
PgmLine	Quellcode-Zeilennummer des Natural-Programms.
FileType	Dateityp KSDS, ESDS oder RRDS.
CID(HEX)	Kommando-ID, äquivalent zu der von ADABAS.
CMD CODE	Äquivalentes Adabas-Kommando.
FILENAME	VSAM-Datei, auf die der Zugriff erfolgt.
PATHNAME	Pfadname für die VSAM-Datei, auf die der Zugriff erfolgt.
RETCODE	Rückgabecode von dem VSAM-Aufruf.
ProbCode	Problemcode von dem VSAM-Aufruf.
Key Len	Länge des VSAM-Schlüssels.
Nat Resp	Natural-Rückgabecode, falls vorhanden.
Record	Erste 62 Bytes des VSAM-Datensatzes.
CBA	Erste 62 Bytes des CBA-Datensatzes.
SB	Erste 62 Bytes des Suchpuffers (Search Buffer).
FB	Erste 62 Bytes des Formatpuffers (Format Buffer).
RB	Erste 62 Bytes des Datensatzpuffers (Record Buffer).
ISN	Erste 62 Bytes des ISN-Puffers (ISN Buffer).
VB	Erste 62 Bytes des Wertpuffers (Value Buffer)

45 TEST DBLOG -Kommando

- Syntax-Diagramme 336
- Erklärung der Schlüsselwörter 337

Das Natural-Systemkommando `TEST DBLOG` dient dazu, die DBLOG Utility auszuführen und die zurzeit im Natural-DBLOG-Puffer gespeicherten Protokoll-Datensätze anzuzeigen oder zu löschen. Dabei ist zu beachten, dass es beim Kommando `TEST DBLOG` nicht möglich ist, Parameter zur Angabe von Auswahlkriterien anzugeben. Die Angabe von Auswahlkriterien ist nur im **DBLOG-Menü** möglich.

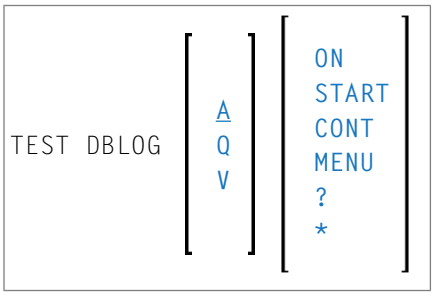
Die beim Kommando `TEST DBLOG` gültigen Schlüsselwörter werden in den Syntax-Digrammen und Tabellen weiter unten erklärt. Einige dieser Schlüsselwörter wirken wie ein Kippschalter und können sowohl zum Aktivieren als auch zum Deaktivieren benutzt werden, je nachdem ob im Natural-DBLOG-Puffer Daten gespeichert sind, siehe Abschnitt *Erklärung der Schlüsselwörter*.

Eine Beschreibung der in den Syntax-Diagrammen verwendeten Symbole befindet sich im Abschnitt *Syntax-Elemente* in der *Systemkommandos* -Dokumentation.

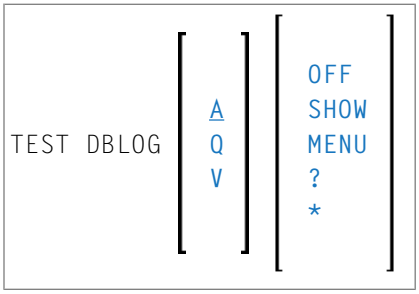
Folgende Themen werden behandelt:

Syntax-Diagramme

DBLOG aktivieren



DBLOG deaktivieren



Erklärung der Schlüsselwörter

Schlüsselwort	Funktion
A	Standardwert. Kippschalterfunktion: ■ **Aktiviert** das Protokollieren von Adabas-Kommandos, wenn im Natural-DBLOG-Puffer keine Daten existieren. ■ **Deaktiviert** das Protokollieren von Adabas-Kommandos. Der **DBLOG TraceDBLOG Trace**-Bildschirm des aktuellen Protokoll-Datensatzes wird angezeigt, wenn im Natural-DBLOG-Puffer Daten existieren.
Q	Kippschalterfunktion: ■ **Aktiviert** das Protokollieren von SQL-Statements, wenn im Natural-DBLOG-Puffer keine Daten existieren. ■ **Deaktiviert** das Protokollieren von SQL-Statements. Der **DBLOG TraceDBLOG Trace**-Bildschirm des aktuellen Protokoll-Datensatzes wird angezeigt, wenn im Natural-DBLOG-Puffer Daten existieren.
V	Kippschalterfunktion: ■ **Aktiviert** das Protokollieren von VSAM-Aufrufen, wenn im Natural-DBLOG-Puffer keine Daten existieren. ■ **Deaktiviert** das Protokollieren von VSAM-Aufrufen. Der **DBLOG Trace**-Bildschirm des aktuellen Protokoll-Datensatzes wird angezeigt, wenn im Natural-DBLOG-Puffer Daten existieren.
CONT	Aktiviert bzw. reaktiviert (durch Neustart) die Protokollierung. Ein Neustart bewirkt, dass DBLOG die Protokollierung mit dem nächsten ausgeführten Programm oder Lauf fortsetzt, nachdem die DBLOG-Ausführung gestoppt wurde, und dass die neuen Daten zu den Daten hinzugefügt werden, die von vorangegangenen Aufzeichnungen existieren.
MENU ? *	Ruft das **DBLOG-Menü** auf. Dort können Sie die Protokollierung aktivieren oder deaktivieren und die zu protokollierenden Kommandos, Aufrufe oder Statements angeben, siehe entsprechende Abschnitte.
SHOW	Deaktiviert die Protokollierung. Der **DBLOG Trace**-Bildschirm des aktuellen Protokoll-Datensatzes wird angezeigt, wenn im Natural-DBLOG-Puffer Daten existieren. Die

Schlüsselwort	Funktion
	Daten des Protokolldatensatzes werden nicht gelöscht, sondern im Natural-DBLOG-Puffer gehalten.
ON oder START	Löscht den Natural-DBLOG-Puffer und aktiviert die Protokollierung.
OFF	Löscht den Natural-DBLOG-Puffer und deaktiviert die Protokollierung.

VII INPL Utility

46 INPL Utility

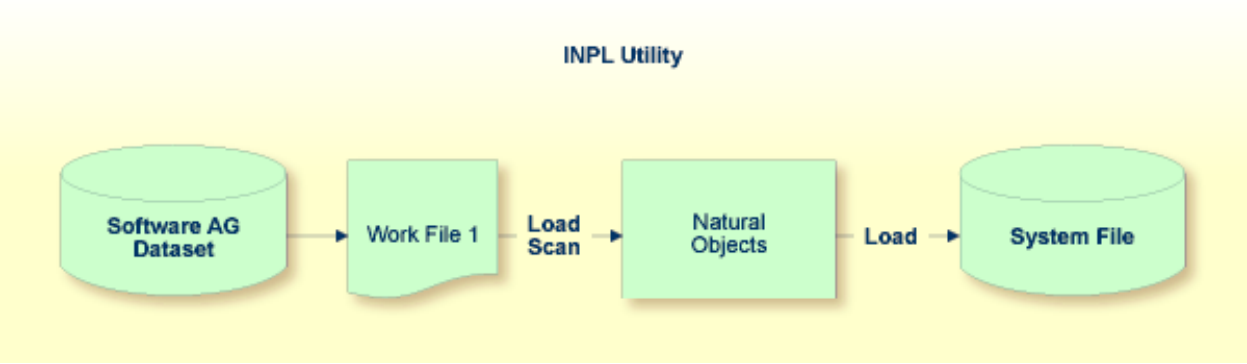
- Einführung in das Dienstprogramm INPL 342
- Load Libraries Only - Nur Libraries laden 348
- Load DDMs Only - Nur DDMs laden 349
- Load Error Messages Only - Nur Fehlermeldungen laden 349
- Load All Objects - Alle Objekte laden 350
- Replace Product Installation - Produktinstallation ersetzen 350
- Scan INPL File - INPL-Datei scannen 351
- Natural Security Recover - Initialisierung der Natural Security-Umgebung erzwingen 352
- User-Exit-Routinen 353

Mit dem Dienstprogramm INPL (Initial Natural Program Load) können Sie Natural-Objekte aus von uns gelieferten Datasets (Dateien) laden und scannen.

Einführung in das Dienstprogramm INPL

Das Dienstprogramm INPL verarbeitet die von uns gelieferten Natural-Objekte.

Das folgende Diagramm zeigt das Funktionsprinzip des Dienstprogramms INPL:



Die Natural-Objekte werden als Installations- oder Update-Datasets (Dateien) geliefert, die der Arbeitsdatei 1 (Work File 1) zugeordnet werden. Das Dienstprogramm INPL lädt die Natural-Objekte aus Arbeitsdatei 1 in Natural-Systemdateien.

Zu den Natural-Objekten gehören katalogisierte Objekte und Quellcodeobjekte, die in Libraries in den Natural-Systemdateien FNAT und FUSER oder in der Systemdatei FDIC, in der DDMs (Datendefinitionsmodule) gespeichert werden, enthalten sind.

Neben dem Laden von Natural-Objekten bietet das Dienstprogramm INPL folgende Funktionen:

- Eine Scan-Funktion zur Überprüfung des Inhalts des der Arbeitsdatei 1 zugeordneten Dataset und eine **Natural Security Recover**-Funktion, die die Initialisierung der Natural Security-Umgebung erzwingt.
- Laden von Korrekturen (Fixes) für die Natural Business Services, die Natural-Objekte, Construct-Modelle, Construct-Rahmen und Construct-Hilfetexte enthalten, wenn die Natural Business Services 5.3.1.8 oder höher installiert sind und die INPL-Funktion **Load All Objects** ausgeführt wird.

  Wenn Fixes für Natural Business Services geladen werden, wird `PRINTER(1)` für den Natural Business Services-Load Report verwendet und muss entsprechend definiert werden.

Tritt während der INPL-Ausführung ein Fehler auf, wird der INPL-Vorgang unterbrochen und mit dem Condition Code 40 vorzeitig beendet.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Restriktionen
- Sonderfälle
- Dienstprogramm INPL aufrufen
- Verfügbare Optionen
- INPL-Report
- Prüfkommandos

Restriktionen

Sie können nur Datasets (Dateien) verarbeiten, die als „SAG system INPL data set (file)“ gekennzeichnet sind.

Sonderfälle

In allen folgenden Fällen muss das INPL-Kommando über den dynamischen Natural-Profilparameter `STACK` angegeben werden:

- wenn ein INPL bei einer leeren FNAT-Systemdatei ausgeführt werden soll,
- wenn ein neues Upgrade-INPL durchgeführt werden soll,
- wenn eine bestehende Produktinstallation ersetzt werden soll,
- wenn eine INPL-Verarbeitung in einer Natural Security-Umgebung durchgeführt werden soll.

Dienstprogramm INPL aufrufen

Um das Dienstprogramm INPL aufzurufen:

1 Geben Sie das folgende Natural-Systemkommando ein:

```
INPL
```

Es wird ein INPL-Menü ähnlich dem folgenden Beispiel angezeigt:

```
10:56:15              ***** NATURAL INPL UTILITY *****            2022-07-19
User: SAG                                                    Library: SYSTEM
                     Code   Function

                       L    Load Libraries Only
                       D    Load DDMs Only
                       E    Load Error Messages Only
                       B    Load All Objects
                       P    Replace Product Installation
                       S    Scan INPL File
                       R    Natural Security Recover
                       ?    Help
                       .    Exit

         Code ........ B
         Replace ..... Y  (Y/N/P/O)   Load Except . N  (Y/N)
         DDM Name ....
         Library .....
         Object Name .                Date ........             (YYYY-MM-DD)
         Check Date .. N  (Y/N)       Number ...... 0
                                                                          ↩
```

2 Im INPL-Menü können Sie eine der folgenden Funktionen auswählen, indem Sie den entsprechenden Funktionscode in das Feld **Code** eingeben:

Code	Funktion	Beschreibung siehe:
L	Load Libraries Only	**Nur Libraries laden**
D	Load DDMs Only	**Nur DDMs laden**
E	Load Error Messages Only	**Nur Fehlermeldungen laden**
B	Load All Objects	**Alle Objekte laden**
P	Replace Product Installation	**Produktinstallation ersetzen**
S	Scan INPL File	**INPL-Datei scannen**
R	Natural Security Recover	**Natural Security Recover**

Ausführliche Informationen zu diesen Funktionen finden Sie in den angegebenen Abschnitten.

3 Ändern oder ergänzen Sie die übrigen Eingabefelder wie unter *Verfügbare Optionen* beschrieben.

4 Drücken Sie Enter, um Ihre Eingaben zu bestätigen.

Verfügbare Optionen

Der folgende Abschnitt beschreibt die Eingabefelder im INPL-Menü, in denen Sie einen oder mehrere Parameter als Objektauswahlkriterien für die im Feld **Code** angegebene INPL-Funktion angeben können. Die Verwendung eines Parameters hängt von der jeweiligen Funktion ab, wie in den entsprechenden Abschnitten in dieser Dokumentation angegeben.

Feld	Beschreibung
Replace	**Ersetzen** Gibt an, ob die zu verarbeitenden Natural-Objekte bereits in den Systemdateien vorhandene Objekte ersetzen sollen. Mögliche Einstellungen: `Y` Alle vorhandenen Objekte werden ersetzt. Dies ist die Standardeinstellung. `N` Vorhandene Objekte werden *nicht* ersetzt. `P` Alle vorhandenen Objekte werden ersetzt. Zusätzlich werden vorhandene Natural- oder Natural-Add-on-Produktinstallationen ersetzt, abhängig von den entsprechenden **Prüfkommandos** (siehe unten), die während der INPL-Verarbeitung ausgeführt werden. `O` Setzt die Eigentümerinformationen der angegebenen Objekte zurück. Gilt nur für die Funktion **Natural Security Recover**. Siehe auch **Check Date**, um nur Objekte zu ersetzen, die älter sind als die zu bearbeitenden Objekte. Wenn Sie die Funktion **Natural Security Recover** benutzen, können Sie in diesem Feld die Option `O` eingeben, um die Eigentümerinformationen der angegebenen Objekte zurückzusetzen.
DDM Name	**DDM-Name** Der Name eines Datendefinitionsmoduls (DDM) oder ein Bereich von Namen. Wenn Sie einen Wert eingeben, der mit einem Stern (*) endet, wird jedes DDM, dessen Name mit dem angegebenen Wert beginnt, verarbeitet. Wenn nur ein Stern (*) eingegeben wird oder wenn dieses Feld leer ist, werden alle DDMs verarbeitet.
Library	**Library-Name** Der Name einer Library oder ein Bereich von Namen. Wenn Sie einen Wert eingeben, der mit einem Stern (*) endet, wird jede Library mit einem Namen verarbeitet, der mit dem angegebenen Wert beginnt. Der Library-Name ist obligatorisch, wenn **Object Name** angegeben ist.
Object Name	**Objektname** Der Name eines Natural-Objekts (außer DDMs) oder ein Bereich von Namen.

Feld	Beschreibung
	Wenn der Wert mit einem Stern (*) endet, wird jedes Objekt mit einem Namen, der mit dem angegebenen Wert beginnt, verarbeitet. Wenn dieses Feld leer ist, werden alle Objekte verarbeitet, die in der im Feld **Library** angegebenen Library enthalten sind.
Check Date	**Datum prüfen** Gibt an, ob vorhandene Natural-Objekte in Abhängigkeit von ihrem Zeitstempel ersetzt werden sollen. Dieser Parameter hat keine Auswirkung, wenn **Replace** (Ersetzen) auf `N` gesetzt ist. Mögliche Einstellungen: `Y` Ein Objekt ist älter, wenn es vor dem zu ladenden Objekt gespeichert oder katalogisiert wurde. `N` Alle Objekte werden ersetzt. Dies ist die Standardeinstellung.
Load Except	**Laden außer** Gibt an, ob Natural-Objekte von der Verarbeitung ausgeschlossen werden sollen. Dieser Parameter gilt nicht bei Fehlermeldungen. Mögliche Einstellungen: `Y` Alle Objekte werden verarbeitet, außer den in den Feldern **DDM Name**, **Library** und/oder **Object Name** angegebenen Objekten. `N` Keine Ausnahmen - alle Objekte werden verarbeitet. Dies ist die Standardeinstellung. **Beispiele für Ausnahmen beim Laden:** Alle Libraries außer der Library ABC werden geladen: **Code** = `L` **Library** = `ABC` Alle DDMs mit einem anderen Präfix als XY werden geladen: **Code** = `D` **DDM Name** = `XY*` Alle Objekte, die in Libraries mit einem anderen Präfix als AB enthalten sind, und alle DDMs mit einem anderen Präfix als CD werden geladen: **Code** = `B` **Library** = `AB*` **DDM Name** = `CD*`
Date	**Datum** Schränkt die Verarbeitung auf Natural-Objekte ein, die an oder nach dem in diesem Feld eingegebenen Datum gespeichert oder katalogisiert wurden.

Feld	Beschreibung
	Das Datum muss im Format *YYYY-MM-DD* (*YYYY* = Jahr, *MM* = Monat, *DD* = Tag) eingegeben werden.
Number	**Anzahl** Begrenzt die Verarbeitung von Natural-Objekten auf eine bestimmte Anzahl. Gezählt werden alle Objekte, die gemäß den im INPL-Menü angegebenen Auswahlkriterien geladen oder gescannt werden. Wenn die Anzahl der verarbeiteten Objekte den im Feld **Number** eingegebenen Wert erreicht hat, wird die Verarbeitung mit einer entsprechenden Meldung beendet.

INPL-Report

Wenn die ausgewählte INPL-Funktion beendet ist, wird ein entsprechender INPL-Report auf dem Bildschirm angezeigt.

Prüfkommandos

Das Dienstprogramm INPL verarbeitet interne Prüfkommandos, die bei der Durchführung einer INPL-Funktion automatisch ausgeführt werden. Prüfkommandos werden verwendet, um den Lade- oder Scanvorgang zu steuern und auf bestimmte Ereignisse zu reagieren. Die Prüfkommandos, die während eines INPL-Vorgangs ausgeführt werden, werden in den INPL-Report geschrieben.

Die Parameter, mit denen das Prüfkommando auf ein Ereignis reagiert, sind z.B. `STOP`, `LOAD`, `CONTINUE`, `SKIP`, `USERLOAD` oder `USERCONTINUE`. Sie können sich alle Parameter mit der Funktion **Scan INPL File** anzeigen lassen. Diese Parameter werden zum Beispiel verwendet, um zu überprüfen, ob die Version eines zu installierenden Produkts höher ist als die Version des bereits installierten Produkts.

Sie können ein Prüfkommando nicht ändern, aber Sie können die Wirkung der Parameter `USERLOAD` und `USERCONTINUE` beeinflussen, indem Sie die Funktion **Replace Product Installation** (Funktionscode `P`) wählen oder die Option **Replace** auf `P` setzen.

`USERLOAD` (oder `USERCONTINUE`) bedeutet: Laden (bzw. Fortsetzen des Ladens) nur, wenn der Funktionscode `P` gewählt wurde oder wenn die Option **Replace** auf `P` gesetzt ist.

Beispiel für USERLOAD und USERCONTINUE

Im folgenden Beispiel für ein Prüfkommando wird die aktuell installierte Natural-Version geprüft, bevor der INPL-Vorgang durchgeführt wird:

```
CHECK VERSION NAT vrs LT USERLOAD EQ LOAD GT USERLOAD
```

wobei *vrs* die Natural-Version des zu ladenden INPL-Inhalts ist.

Liegt die installierte Natural-Version (NAT) unter (LT) oder über (GT) *vrs*, so wird die INPL-Funktion nur ausgeführt, wenn der Funktionscode P angegeben oder die Option **Replace** auf P gesetzt wurde. Andernfalls bricht das INPL-Dienstprogramm ab oder wird mit Condition Code 40 im Batch abgebrochen und eine entsprechende Meldung in den INPL-Report geschrieben.

Ist die Version gleich *vrs*, wird die INPL-Funktion immer ausgeführt, unabhängig davon, ob der Funktionscode P angegeben oder die Option **Replace** auf P gesetzt ist.

Load Libraries Only - Nur Libraries laden

Mit dieser Funktion des Dienstprogramms INPL können Sie katalogisierte Natural-Objekte und Quellcodeobjekte in die angegebenen Libraries in der Natural-Systemdatei FNAT oder FUSER laden.

Um Libraries zu laden:

1 Geben Sie im INPL-Menü den Funktionscode L ein. Sie können Parameter angeben, die bei der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (Y/N)
- Laden außer: `Load Except` (Y/N)
- Bibliothek: **Library**
- Objekt-Name: `Object Name`
- Datum: `Date` (*YYYY-MM-DD*)
- Datum prüfen: `Check Date` (Y/N)
- Anzahl: `Number`

Ausführliche Informationen zu diesen Parametern finden Sie unter *Verfügbare Optionen* im Abschnitt *Einführung in das Dienstprogramm INPL*.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt *Einführung in das Dienstprogramm INPL*) ausgegeben.

Load DDMs Only - Nur DDMs laden

Diese Funktion des Dienstprogramms INPL wird verwendet, um DDMs in die Systemdatei FDIC zu laden.

Um DDMs zu laden:

1 Geben Sie im INPL-Menü den Funktionscode `D` ein. Sie können Parameter angeben, die während der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (`Y/N`)
- Laden außer: `Load Except` (`Y/N`)
- DDM-Name: `DDM Name`
- Anzahl: `Number`

Ausführliche Informationen zu diesen Parametern finden Sie unter *Verfügbare Optionen* im Abschnitt *Einführung in das Dienstprogramm INPL*.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt *Einführung in das Dienstprogramm INPL*) ausgegeben.

Load Error Messages Only - Nur Fehlermeldungen laden

Mit dieser Funktion des Dienstprogramms INPL können benutzerdefinierte Fehlermeldungen oder Systemfehlermeldungen in bestimmte Libraries in der Natural-Systemdatei FUSER bzw. FNAT geladen werden.

Um Fehlermeldungen zu laden:

1 Geben Sie im INPL-Menü den Funktionscode `E` ein. Sie können Parameter angeben, die während der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (`Y/N`)
- Bibliothek: `Library`

Ausführliche Informationen zu diesen Parametern finden Sie unter *Verfügbare Optionen* im Abschnitt *Einführung in das Dienstprogramm INPL*.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt *Einführung in das Dienstprogramm INPL*) ausgegeben.

Load All Objects - Alle Objekte laden

Mit dieser Funktion des Dienstprogramms INPL werden alle Natural-Objekte (einschließlich Fehlermeldungen und DDMs) in die in Arbeitsdatei 1 angezeigten Libraries geladen. Die DDMs werden in die Systemdatei FDIC geladen.

Um alle Objekte zu laden:

1 Geben Sie im INPL-Menü den Funktionscode `B` ein. Sie können Parameter angeben, die während der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (`Y/N`)
- Laden außer: `Load Except` (`Y/N`)
- DDM-Name: `DDM Name`
- Bibliothek: `Library`
- Objekt-Name: `Object Name`
- Datum: `Date` (*`YYYY-MM-DD`*)
- Datum prüfen: `Check Date` (`Y/N`)
- Anzahl: `Number`

Ausführliche Informationen zu diesen Parametern finden Sie unter *Verfügbare Optionen* im Abschnitt *Einführung in das Dienstprogramm INPL*.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt *Einführung in das Dienstprogramm INPL*) ausgegeben.

Replace Product Installation - Produktinstallation ersetzen

Zusätzlich zur Funktion **Load All Object** (Alle Objekte laden) ersetzt diese Funktion alle vorhandenen Natural- oder Natural-Add-on-Produkte, die in der aktuellen Systemumgebung installiert sind. Das Ersetzen bestehender Produktinstallationen hängt von der Ausführung der entsprechenden **Prüfkommandos** ab, die im Abschnitt *Einführung in das Dienstprogramm INPL* beschrieben sind.

Um alle Objekte zu laden und bestehende Produktinstallationen zu ersetzen:

1 Geben Sie im INPL-Menü den Funktionscode `P` ein. Sie können Parameter angeben, die bei der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (`Y`/`N`)
- Laden außer: `Load Except` (`Y`/`N`)
- DDM-Name: `DDM Name`
- Bibliothek: `Library`
- Objekt-Name: `Object Name`
- Datum: `Date` (*`YYYY-MM-DD`*)
- Datum prüfen: `Check Date` (`Y`/`N`)
- Anzahl: `Number`

Ausführliche Informationen zu diesen Parametern finden Sie unter ***Verfügbare Optionen*** im Abschnitt ***Einführung in das Dienstprogramm INPL***.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt ***Einführung in das Dienstprogramm INPL***) ausgegeben.

Scan INPL File - INPL-Datei scannen

Mit dieser Funktion des Dienstprogramms INPL wird der Inhalt des der Arbeitsdatei 1 zugeordneten Dataset (Datei) gescannt.

Um eine INPL-Datei zu scannen:

1 Geben Sie im INPL-Menü den Funktionscode `S` ein. Sie können Parameter angeben, die während der Ausführung dieser Funktion gültig sein sollen:

- Ersetzen: `Replace` (`Y`/`N`)
- Laden außer: `Load Except` (`Y`/`N`)
- DDM-Name: `DDM Name`
- Bibliothek: `Library`
- Objekt-Name: `Object Name`
- Datum: `Date` (*`YYYY-MM-DD`*)
- Anzahl: `Number`

Ausführliche Informationen zu diesen Parametern finden Sie unter *Verfügbare Optionen* im Abschnitt *Einführung in das Dienstprogramm INPL*.

2 Bestätigen Sie Ihre Eingaben.

Nach Beendigung der Funktion wird ein entsprechender **INPL-Report** (siehe Abschnitt *Einführung in das Dienstprogramm INPL*) ausgegeben.

Natural Security Recover - Initialisierung der Natural Security-Umgebung erzwingen

Mit dieser Funktion des INPL-Dienstprogramms können Sie die Initialisierung der Natural Security-Umgebung erzwingen.

Die folgenden Optionen sind verfügbar:

- Reset Environment - Umgebung zurücksetzen
- Remove Owners - Eigentümer entfernen

Reset Environment - Umgebung zurücksetzen

Vorsicht: Durch die Ausführung dieser Funktion werden das Benutzerprofil DBA und das Library-Profil SYSSEC sowie die Verlinkung zwischen diesen beiden Objekten auf den Stand nach der Erstinstallation zurückgesetzt. Alle anderen Verlinkungen mit der Library SYSSEC werden aufgehoben. Andere Natural Security-Profile und Verlinkungen werden nicht geändert. Wenden Sie sich für weitere Informationen an den Support.

Um die Umgebung zurückzusetzen:

- Geben Sie im INPL-Menü den Funktionscode `R` ein.

Remove Owners - Eigentümer entfernen

Um Eigentümer zu entfernen:

- Geben Sie im INPL-Menü den Funktionscode `R` ein und tragen Sie im Feld **Replace** ein `0` ein, um die Eigentümerinformationen der angegebenen Objekte zurückzusetzen.

User-Exit-Routinen

Eine INPL-User-Exit-Routine wird als Quellcode-Objekt INPLSX*nn* in der Natural System Library SYSLIB ausgeliefert, wobei *nn* die ID (Kennung) der User-Exit-Routine angibt.

Um eine User-Exit-Routine zu aktivieren:

1 Kopieren Sie den Quellcode aus der Natural System Library SYSLIB in eine User Library.

2 Katalogisieren Sie ihn unter dem Namen `INPLUX`*nn*.

3 Kopieren Sie ihn zurück in die Natural System Library SYSLIB.

Anmerkung: Das von Ihnen möglicherweise geänderte Quellcodeobjekt und das katalogisierte Objekt der User-Exit-Routine werden umbenannt, um zu verhindern, dass sie bei einer Update-Installation überschrieben werden.

Die folgenden User-Exit-Routinen sind verfügbar:

Name	Funktion
`INPLUX01`	Verhindert, dass Fehlermeldungstexte ersetzt werden.

User-Exit-Routine `INPLUX01`

Mit diesem User-Exit können Sie Bereiche für Fehlermeldungen (benutzerdefinierte oder Natural-System-Fehlermeldungen) definieren, die während einer INPL-Sitzung nicht ersetzt werden können. Weitere Informationen finden Sie im Quellcode von `INPLSX01` in der Natural System Library SYSLIB.

VIII NATPAGE Utility

47 NATPAGE Utility

Mit dem Dienstprogramm („Utility“) NATPAGE können Sie während einer Natural-Session Bildschirmausgabedaten (Masken und Reports) aufzeichnen. Der Begriff „Bildschirm“ (Screen) meint in diesem Kontext den Inhalt des Seitenpufferspeichers (Page Buffer), d.h. die logische Seitenausgabe von Natural.

Die Bildschirmaufzeichnungen werden im Natural Scratch-Pad File gespeichert, siehe entsprechenden Abschnitt in der *Operations*-Dokumentation.

Die maximale Anzahl an Bildschirmen, die aufgezeichnet werden können, wird mit dem Session-Parameter `PD` festgelegt, siehe *Parameter-Referenz*-Dokumentation.

Das Dienstprogramm NATPAGE umfasst die folgenden Natural-Terminalkommandos:

Kommando	Funktion
`%P`	Mit diesem Terminalkommando aktivieren Sie NATPAGE, um den aktuellen Bildschirm und alle nachfolgenden Bildschirm aufzuzeichnen. Zuvor aufgezeichnete Bildschirme werden dabei gelöscht.
`%I`	Mit diesem Terminalkommando aktivieren Sie NATPAGE, um den Inhalt des aktuellen Bildschirms aufzuzeichnen.
`%O`	Mit diesem Terminalkommando deaktivieren Sie NATPAGE.
`%S`	Mit diesem Terminalkommando können Sie das Aufzeichnen von Bildschirmen durch NATPAGE wiederaufnehmen
`%E`	Mit diesem Terminalkommando können Sie sich Bildschirme anzeigen lassen, die mit NATPAGE aufgezeichnet wurden.

Ausführliche Beschreibung siehe entsprechende Abschnitte in der *Terminalkommandos*-Dokumentation.

Verwandtes Thema siehe Abschnitt *Bildschirme aufzeichnen* ebenda.

IX NATRJE Utility

48 NATRJE Utility

- Allgemeine Informationen zu NATRJE 362
- NATRJE aus einem Natural-Programm aufrufen 363
- NATRJE-Rückgabecodes 365

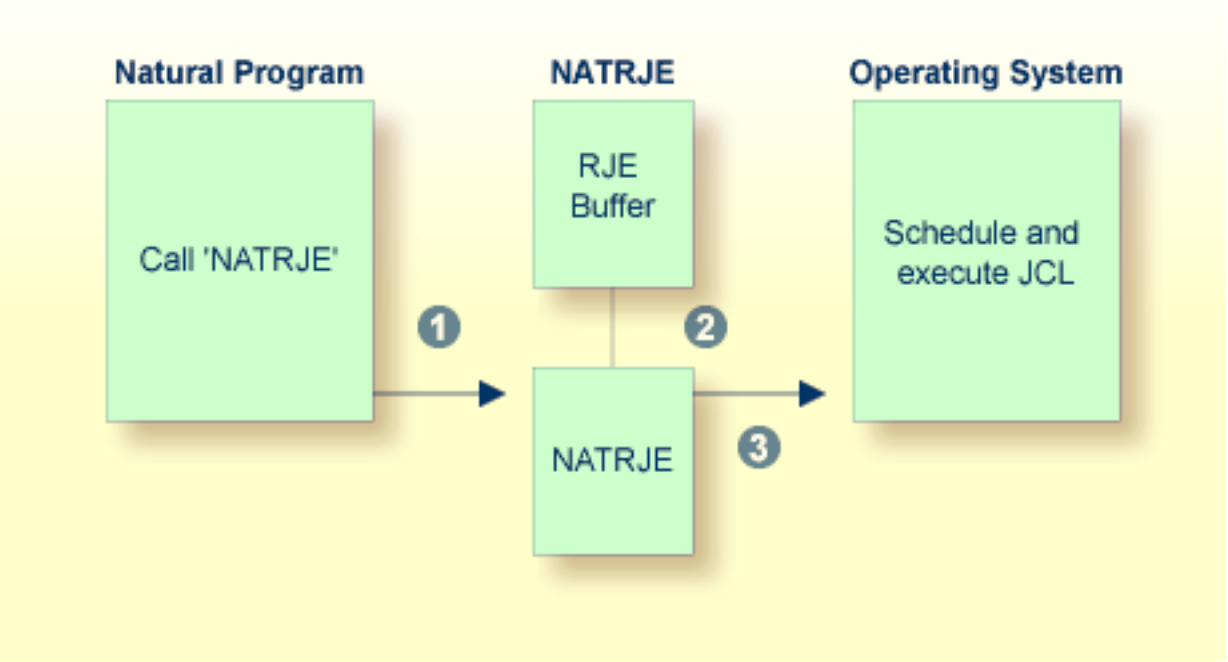
Das Dienstprogramm („Utility“) NATRJE dient zur Jobfernverarbeitung. Es kann verwendet werden, um JCL-Karten aus einem Natural-Programm an das Betriebssystem zur Zeitplanung und Ausführung zu übergeben. Es ist z.B. mit der NATRJE Utility möglich, einen Natural Batch Job zu starten.

Verwandtes Thema:

NREXPG - User Exit for NATRJE in der *Operations*-Dokumentation

Allgemeine Informationen zu NATRJE

Die folgende Grafik ist ein vereinfachtes Funktionsdiagramm der NATRJE Utility:



Legende

1. Das Natural-Programm ruft die NATRJE Utility zwecks Übergabe von JCL-Karten, die vom Betriebssystem ausgeführt werden sollen.
2. Die NATRJE Utility erfasst die JCL-Karten im RJE-Puffer solange, bis das Natural-Programm anzeigt, dass der Job vollständig ist. Im RJE-Puffer werden die JCL-Karten zwischengespeichert, bevor sie in Auftrag gegeben werden. Die Anfangsgröße des RJE-Puffers wird durch den Profilparameter `RJESIZE` festgelegt (Beschreibung siehe *Parameter-Referenz*-Dokumentation). Gegebenenfalls wird die Speichergröße automatisch erhöht, wenn ein vorliegender Job nicht in den RJE-Puffer passt. Die maximale Größe eines Jobs wird durch die Größe des Thread oder der Region bestimmt.

3 Die NATRJE Utility überträgt die Job-Karten in die interne Job-Warteschlange des Betriebssystems zur Zeitplanung und Ausführung durch das Betriebssystem.

NATRJE aus einem Natural-Programm aufrufen

Folgende Themen werden behandelt:

- NATRJE aufrufen
- Beispiel-Programm

NATRJE aufrufen

Um die NATRJE Utility aufzurufen:

- Geben Sie in dem Natural-Programm, das die NATRJE Utility aufruft, ein `CALL`-Statement mit folgender Syntax an:

```
CALL 'NATRJE' parm1 parm2 parm3 parm4
```

Die im `CALL`-Statement anzugebenden Parameter werden in der folgenden Tabelle erklärt:

Parameter	Erklärung	
parm1	Die Start-JCL-Karte der Tabelle, die eine oder mehrere zu übergebende, 80 Zeichen umfassende JCL-Karten enthält.	
parm2	Ein 4 Byte breites binäres Feld, das die Anzahl zu übergebender, 80 Zeichen umfassender JCL-Karten enthält.	
parm3	Ein 1-Byte breites alphanumerisches Feld, das dazu dient anzuzeigen, ob alle JCL-Karten übergeben worden sind.	
	`' '`	Dies ist nicht der letzte Aufruf für den aktuellen Job. Mit dem nächsten `CALL`-Statement folgt eine weitere JCL-Karte für den aktuellen Job. Die JCL-Karten werden im RJE-Puffer gesammelt.
	`B`	Letzter Aufruf für den aktuellen Job. Unter z/OS Batch und TSO, IMS TM und CICS: Der Job wird generiert, in den Reader Dataset geschrieben, aber nicht übergeben. Wird anschließend die Funktion `L` (siehe unten) aufgerufen, wird der interne Reader geschlossen und der oder die Jobs werden übergeben. Zusätzlich wird der interne Reader geschlossen und der Job wird bei einer Bildschirm-Ein-/Ausgabe (IMS TM) oder während der Beendigung einer Session (z/OS Batch, TSO und IMS TM) übergeben.

Parameter	Erklärung	
	C	Flush: Der aktuelle Job wird ignoriert. Der Job wird nicht an das System übergeben.
	L	Letzter Aufruf für den aktuellen Job. Der Job wird an das System übergeben.
parm4	Ein 2 Byte breites binäres Feld, in dem NATRJE einen Rückgabecode zurückgibt.	

Beispiel-Programm

Das folgende Programm ist ein Beispiel-Natural-Programm, das einen JCL-Eingabestrom mit drei Karten in einem Aufruf an NATRJE übergibt.

```
DEFINE DATA LOCAL
   01 COUNT  (B4)
   01 FLAG   (A1)
   01 RETHEX (B2
   01 CARDS  (A240
   01 REDEFINE CARDS
      02 CARD1 (A80)
      02 CARD2 (A80)
      02 CARD3 (A80)
  END-DEFINE
  MOVE '//JOBN JOB CLASS=G,MSGCLASS=X' TO CARD1
  MOVE '//XXX EXEC PGM=IEFBR14' TO CARD2
  MOVE '//DD1 DD DSN=NATRJE.SOURCE,DISP=SHR' TO CARD3
  MOVE 3 TO COUNT
  MOVE 'L' TO FLAG
  CALL 'NATRJE' CARDS COUNT FLAG RETHEX
  IF RETHEX = H'0000'
    WRITE 'JOB submitted successfully'
  ELSE
    WRITE 'ERROR from NATRJE' RETHEX
  END-IF
  END
```

NATRJE-Rückgabecodes

Wenn ein Natural-Programm ein `CALL`-Statement an NATRJE absetzt, wird einer der unten aufgeführten Rückgabecodes im vierten Parameter des Statements zurückgegeben. Es gibt Rückgabecodes, die für alle Umgebungen gelten, und zusätzliche Codes, die betriebssystemabhängig sind.

- Rückgabecodes, die für alle Umgebungen gelten
- Zusätzliche Rückgabecodes, die für CICS unter z/OS gelten

Rückgabecodes, die für alle Umgebungen gelten

Rückgabecode hexadezimal	Rückgabecode dezimal	Erklärung
00	00	Normale Rückgabe.
04	04	NATRJE Utility nicht verfügbar.
08	08	NATRJE Utility außer Funktion. Mögliche Ursache: Der Profilparameter `RJESIZE` ist auf `0` gesetzt (siehe *Parameter-Referenz*-Dokumentation).
0C	12	Ungültige Anzahl an JCL-Karten.
10	16	Ungültiger Funktionscode.
14	20	Kein RJE-Pufferplatz verfügbar.
18	24	Ungültige Anzahl an Parametern.
1C	28	Ein-/Ausgabefehler bei der Übergabe.
20	32	Job-Flush (Ignorierung) veranlasst durch den User Exit NREXPG (siehe *NREXPG - User Exit for NATRJE* in der *Operations*-Dokumentation).

Zusätzliche Rückgabecodes, die für CICS unter z/OS gelten

Rückgabecode	Erklärung
01*nn*	CICS WRITEQ TD Fehlschlag. Dabei ist *nn* der CICS-Rückgabecode im Hexadezimalformat.
01*nn*	CICS CLOSE TD Fehlschlag. Dabei ist *nn* der CICS-Rückgabecode im Hexadezimalformat.

X Natural Profiler und Code Coverage

Dieses Dokument beschreibt, wie Sie mit dem Natural Profiler den internen Programmablauf einer Natural-Anwendung überwachen und die Performance und Codeabdeckung (Code Coverage) der Anwendung analysieren können.

Profil-Erstellung für Natural-Anwendungen (Profiling)	Allgemeine Informationen über die Optionen, die Natural und NaturalONE zur Profil-Erstellung (Profiling) für Natural-Anwendungen zur Verfügung stellen.
Codeabdeckung von Natural-Anwendungen (Code Coverage)	Allgemeine Informationen über die Optionen, die Natural und NaturalONE für die Messung der Codeabdeckung (Code Coverage) zur Verfügung stellen.
Grundlegendes Konzept der Profiler Utility	Grundlegendes Konzept der Profiler Utility im Online- und im Batch-Modus.
Profiler Utility im Online-Modus benutzen	Profil-Erstellung (Profiling) bei Natural-Online-Programmen.
Profiler Utility im Batch-Modus benutzen	Profil-Erstellung (Profiling) und Messung der Codeabdeckung (Code Coverage) bei Natural-Batch-Programmen.
Natural Profiler Rich GUI-Schnittstelle	Auswertung der Profiler-Daten in einer grafischen, interaktiven Browser-Oberfläche.

Anmerkung: Der Funktionsumfang des NaturalONE Profiler und der NaturalONE Code Coverage wird in den relevanten Kapiteln der *NaturalONE*-Dokumentation beschrieben.

49 Profil-Erstellung für Natural-Anwendungen (Profiling)

- Einführung in die Profil-Erstellung (Profiling) 370
- Schnellstart für das Profiling 370
- Plattformspezifische Profil-Erstellung 373
- Werkzeuge zur Profil-Erstellung (Natural Profiling Tools) 373
- Natural Profiler-Auswertungen 376

Einführung in die Profil-Erstellung (Profiling)

Ein Profiler ist ein Tool für die dynamische Analyse von Programmen. Es misst die Häufigkeit und Dauer von Statements und vereinfacht so die Optimierung von Programmen.

Der Natural-Profiler wird zur Erstellung von Profilen (Profiling) für Natural-Anwendungen verwendet. Er sammelt Profiling-Daten immer dann, wenn ein definiertes Natural-Ereignis eintritt, z.B. wenn ein Programm startet oder bevor eine Datenbank aufgerufen wird. Der Natural-Profiler visualisiert die aufgezeichneten Ereignisdaten als Ereignis-Trace und die Aufrufstruktur der ausgeführten Natural-Objekte als Programm-Trace. Die vom Natural-Profiler gelieferte Performance-Auswertung zeigt den Zeitverbrauch und die Trefferzahl der ausgeführten Objekte, Natural-Statements und Programmzeilen.

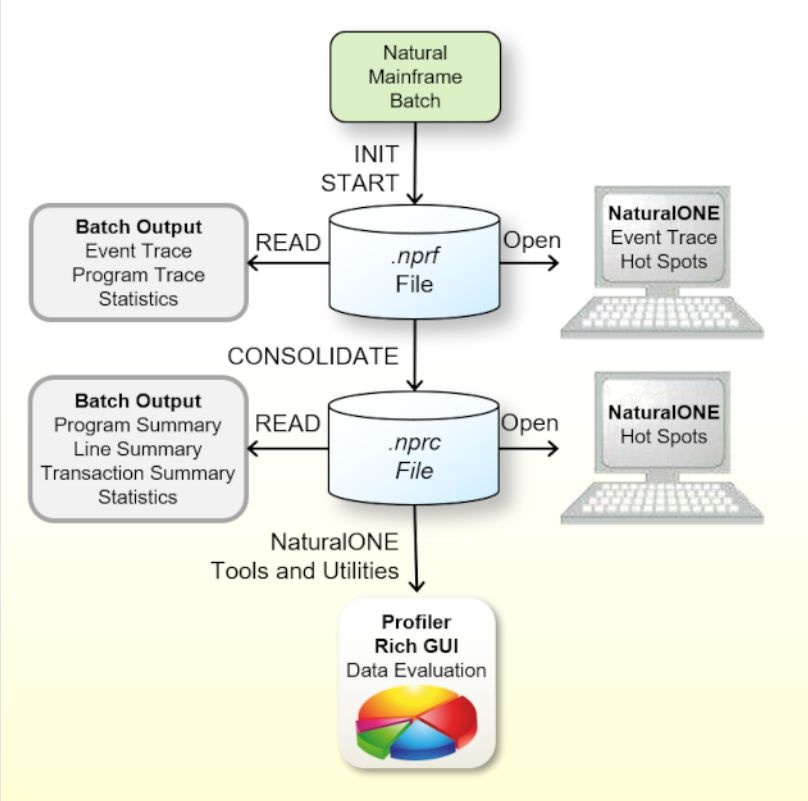
Sie können Natural Profiler-Ereignisdaten in der Ausgabe der Profiler Utility anzeigen oder die Daten im Text- oder Tabellenformat exportieren. Sie können Natural Profiler-Performance-Analysen in der Eclipse-basierten Entwicklungsumgebung NaturalONE oder mit der Natural Profiler Rich GUI-Schnittstelle (einer Natural for Ajax-Anwendung) visualisieren.

Eine Natural Profiler-Analyse dient als Grundlage für die Performance-Optimierung einer Natural-Anwendung. Der Natural Profiler verschafft Ihnen einen sehr schnellen Überblick über die zeitintensiven Teile einer Natural-Anwendung. Es sind keine Code-Änderungen erforderlich, und außerdem reichen Grundkenntnisse der Anwendung aus.

Schnellstart für das Profiling

Dieser Abschnitt beschreibt kurz die Schritte, die für die Profil-Erstellung von Natural-Batch-Anwendungen und die Anzeige der Ergebnisse erforderlich sind. Die hier aufgeführten Anweisungen können als Richtlinie dienen, wenn Sie mit der Verwendung des Natural Profilers beginnen. Detaillierte Informationen zu den Schritten finden Sie im Rest dieses Kapitels.

Die Schritte hängen, wie in der folgenden Grafik dargestellt, von der Auswertung ab, die Sie für Ihre Anwendung durchführen möchten:



1. Überprüfen Sie, ob die **Voraussetzungen** erfüllt sind.
2. Fügen Sie im Natural Batch-Job die Profiler Utility-Funktionen `INIT`, `FILTER` und `START` hinzu, um die Ereignisdatenerfassung zu starten. Im folgenden Beispiel für z/OS wird das Natural-Programm `XPROF` in der Library `PRFDEMO` profiliert:

```
//CMSYNIN  DD  *
PROFILER
FUNCTION=INIT                /* Initialize profiling
  RESOURCE-NAME='ResNam',REPLACE=YES  /* Resource name
  RESOURCE-LIB=RESLIB        /* Resource library
FILTER=EVENT                 /* Set event filter
  EVENT=ALL                  /* All events
  STATEMENT=ON               /* Collect statements
FUNCTION=START               /* Start data collection
END-PROFILER                 /* End Profiler input
LOGON PRFDEMO
XPROF
10000
FIN
```

Im obigen Beispiel werden die Profiler-Ereignisdaten in eine Ressourcendatei mit dem Namen `ResNam.nprf` in der Library `RESLIB` geschrieben. Siehe auch *Profilierung initialisieren* und *Datensammlung starten und anhalten*.

3. Öffnen Sie die NPRF-Ressource in NaturalONE, um die Hot-Spots und das Ereignis-Tracing zu sichten.
4. Starten Sie einen Natural-Batch-Job mit der `READ`-Funktion der Profiler Utility, um ein Ereignis-Tracing, ein Programm-Tracing und die Profiler-Statistiken zu drucken. Beispiel:

```
FUNCTION=READ            /* Read Profiler data
  RESOURCE-LIB=RESLIB    /* Resource library
  RESOURCE-TYPE=NPRF     /* Use resource type NPRF
  EVENT=ON               /* Print event trace
  PROGRAM=ON             /* Print program trace
  STATISTICS=ON          /* Print statistics
```

   Siehe auch *Lesefunktion (READ) der Profiler Utility*. In diesem Abschnitt wird auch beschrieben, wie Sie eine Transaktionszusammenfassung generieren.
5. Konsolidieren (aggregieren) Sie die Ereignisdaten aus der Profiler Rich GUI-Schnittstelle oder starten Sie einen Natural-Batch-Job mit der Profiler Utility-Funktion `CONSOLIDATE`. Beispiel:

```
FUNCTION=CONSOLIDATE     /* Consolidate Profiler data
  RESOURCE-LIB=RESLIB    /* Resource library
  REPLACE=YES            /* Replace resource
```

   Die konsolidierten Profiler-Ereignisdaten werden in die Ressource `ResNam.nprc` in der Library `RESLIB` geschrieben. Siehe *Ereignisdaten konsolidieren*.
6. Öffnen Sie die NPRC-Ressource in NaturalONE, um die Hot-Spots zu sichten.
7. Öffnen Sie die NPRC-Ressource mit der Profiler Rich GUI-Schnittstellenfunktion **Data Evaluation** oder **Program Analysis**, um die Profiler-Daten in einer grafischen, interaktiven Browser-Oberfläche zu visualisieren.
8. Starten Sie einen Natural-Batch-Job mit der Profiler Utility-Funktion `READ`, um eine Programmzusammenfassung, eine Zeilenzusammenfassung und die Profiler-Statistiken zu generieren. Beispiel:

```
FUNCTION=READ            /* Read Profiler Data
  RESOURCE-LIB=RESLIB    /* Resource library
  RESOURCE-TYPE=NPRC     /* Use resource type NPRC
  PROGRAM=ON             /* Print program summary
  LINE=ON                /* Print line summary
  STATISTICS=ON          /* Print statistics
```

   Siehe auch *Lesefunktion (READ) der Profiler Utility*. In diesem Abschnitt wird auch beschrieben, wie Sie eine Transaktionszusammenfassung generieren.



Anmerkungen:

1. Wenn der Ressourcenname nicht explizit in der `READ`- oder `CONSOLIDATE`-Funktion der Profiler Utility angegeben wird, wird die zuletzt erstellte NPRF- oder NPRC-Ressource in der Library verwendet.
2. Wenn Sie vorhaben, ein Profil einer langlaufenden Batch-Anwendung zu erstellen, lesen Sie den Abschnitt ***Langlaufende Anwendungen profilieren***. Er behandelt Strategien, wie die Anzahl der zu überwachenden Ereignisse minimiert werden kann.
3. Der NaturalONE Profiler und die Profiler Rich GUI-Schnittstelle werden in der *NaturalONE*-Dokumentation ausführlich beschrieben.

Plattformspezifische Profil-Erstellung

Sie können den Natural-Profiler für die Profil-Erstellung von Natural-Anwendungen auf Linux-, Windows- und z/OS-Plattformen verwenden. Wie die Profil-Erstellung erfolgt, hängt von der Plattform und dem verwendeten Verarbeitungsmodus der Anwendung ab:

z/OS

- Interaktive z/OS-Anwendungen werden mit dem NaturalONE Profiler oder der Natural Profiler Utility im Online-Modus profiliert.
- Interaktive z/OS-Anwendungen, die remote über Natural Studio oder Natural RPC ausgeführt werden, werden mit der Natural Profiler Utility im Batch-Modus profiliert.
- z/OS-Batch-Anwendungen werden mit der Natural Profiler-Utility im Batch-Modus profiliert.

Linux und Windows

- Interaktive Linux- und Windows-Anwendungen werden mit dem NaturalONE Profiler bzw. dem Natural Profiler for Linux bzw. Windows profiliert.
- Linux- und Windows-Batch-Anwendungen werden mit dem Natural Profiler für Linux bzw. Windows profiliert.

Werkzeuge zur Profil-Erstellung (Natural Profiling Tools)

In diesem Abschnitt werden die wichtigsten Funktionen der Natural Profiling Tools zusammengefasst:

- Funktionsumfang des NaturalONE Profiler
- Funktionsumfang des Natural Profiler für Linux und Windows
- Funktionsumfang der Profiler Utility

- Funktionsumfang des Natural Profiler Rich GUI-Schnittstelle

Funktionsumfang des NaturalONE Profiler

- Erstellt Profile für interaktive Natural-Anwendungen auf Linux-, Windows- oder z/OS-Plattformen in einer Eclipse-basierten Entwicklungsumgebung.
- Liest und analysiert Profiler-Ressourcendateien mit Ereignisdaten, die von der z/OS-Profiler Utility im Batch-Modus oder vom Natural Profiler für Linux und Windows gesammelt wurden.
- Bietet Funktionen für die Verarbeitung großer Datenmengen:
  - Ereignisfilter,
  - Stichprobentechnik,
  - Datenkonsolidierung.
- Performance-Analysen von Programmen, Statements und Programmzeilen:
  - CPU-Zeit,
  - verstrichene (elapsed) Zeit,
  - Trefferquote.
- Zeigt ein Ereignis-Trace an.
- Gestattet die direkte Navigation von einer profilierten Programmzeile zum entsprechenden Quellcode.
- Speichert die Profiler-Daten als XML-formatierte Datei und lädt die Daten wieder zurück.

Funktionsumfang des Natural Profiler für Linux und Windows

- Erstellt Profile für interaktive oder Natural-Batch-Anwendungen von einer Linux- oder Windows-Plattform aus.
- Bietet Funktionen für die Verarbeitung großer Datenmengen:
  - Ereignisfilter,
  - Stichprobentechnik,
  - Datenkonsolidierung.
- Speichert die Profiler-Daten als Profiler-Ressourcendatei.

Funktionsumfang der Profiler Utility

Online-Modus (z/OS)

- Erstellt Profile für interaktive Natural-Anwendungen auf z/OS-Plattformen.
- Bietet einen Ereignisfilter.
- Zeigt ein Ereignis-Trace an.
- Speichert die Profiler-Daten in einem Tabellenformat.
- Speichert die Profiler-Daten als Profiler-Ressourcendatei.



Anmerkung: Die Menge der von der Profiler Utility im Online-Modus gesammelten Daten ist durch die relativ geringe Größe des Natural Data Collector-Puffers begrenzt, der in einem Wrap-Around-Modus arbeitet. Außerdem wird bei der Ausführung unter CICS oder Com-plete die CPU-Zeit nicht bereitgestellt. Generell empfehlen wir Ihnen, den NaturalONE Profiler für das Profiling interaktiver Natural z/OS-Anwendungen zu verwenden, weil der NaturalONE Profiler keine Größenbeschränkungen hat und CPU-Performance-Analysen unterstützt.

Batch-Modus (z/OS)

- Erstellt Profile für Natural-Batch- und Natural-RPC-Anwendungen auf z/OS-Plattformen.
- Erstellt Profile für interaktive z/OS-Anwendungen, die remote von Natural Studio ausgeführt werden.
- Bietet Funktionen für die Verarbeitung großer Datenmengen:
  - Ereignis-, Programm-, Zähl- und Zeitfilter,
  - Stichprobentechnik,
  - Datenkonsolidierung.
- Speichert Profiler-Daten als Profiler-Ressourcendatei.
- Liest und analysiert Profiler-Ressourcendateien.
- Druckt Programm- und Ereignis-Traces aus.
- Analysiert die Programm-Performance.
- Bewertet die Antwortzeiten von Transaktionen.
- Sammelt und zeigt Profiler-Eigenschaften und Statistiken an.
- Bietet eine Funktion zum Löschen einer Natural Profiler-Ressourcendatei.

Batch-Modus (Linux und Windows)

- Liest und analysiert Profiler-Ressourcendateien.
- Bietet Funktionen für die Verarbeitung großer Datenmengen:

  Datenkonsolidierung

- Speichert konsolidierte Profiler-Daten als Profiler-Ressourcendatei.
- Druckt Programm- und Ereignis-Traces aus.
- Analysiert die Programm-Performance.
- Zeigt Profiler-Eigenschaften und Statistiken an.

Funktionsumfang des Natural Profiler Rich GUI-Schnittstelle

- Listet die Profiler-Ressourcen einer bestimmten Library auf.
- Zeigt Profiler-Eigenschaften und Statistiken an.
- Bietet eine Funktion zur Konsolidierung von Profiler-Daten.
- Bietet eine Funktion zum Löschen einer Natural Profiler-Ressourcendatei.
- Visualisiert Profiler-Daten auf einer grafischen, interaktiven Browser-Oberfläche.
- Analysiert die Anwendungs-Performance anhand von Auswahlkriterien wie Library, Programm, Programmzeile oder Benutzer für verschiedene Leistungskennzahlen (KPIs, Key Performance Indicators):
  - CPU-Zeit,
  - verstrichene (elapsed) Zeit,
  - Adabas-Kommandozeit,
  - Trefferzahlen wie Datenbankaufrufe oder Laufzeitfehler.
- Kombiniert den Programmquellcode mit den Profiler-Daten zur Programmanalyse.

Natural Profiler-Auswertungen

Die folgende Tabelle enthält eine Zusammenfassung der Auswertungskriterien, die von den Natural Profiling Tools zur Verfügung gestellt werden.

Auswertung	Profiling-Tool	Beschreibung
Program Summary	Profiler Utility (Batch)	**Programmzusammenfassung** Zeigt die CPU-Zeit, die für jedes ausgeführte Natural-Objekt aufgewendet wurde, und die Natural-Ereignisse, die in einem Objekt aufgetreten sind. Siehe auch *Beispiel für eine Programmzusammenfassung*.
Line Summary	Profiler Utility (Batch)	**Zeilenzusammenfassung** Zeigt die während der Natural-Programmausführung verbrauchte CPU-Zeit und die verstrichene (elapsed) Zeit für jede einzelne

Auswertung	Profiling-Tool	Beschreibung
		Quellcodezeile sowie die Anzahl der Natural-Ereignisse, die in der Zeile aufgetreten sind. Siehe auch *Beispiel für eine Zeilenzusammenfassung*.
Transaction Summary	Profiler Utility (Batch)	**Transaktionszusammenfassung** Zeigt die für jede einzelne Transaktion verbrauchte Antwortzeit und die Anzahl der Natural-Ereignisse, die während der Transaktionsausführung aufgetreten sind. Weitere Informationen siehe *Transaktionszusammenfassung*. Siehe auch *Beispiel für eine Transaktionszusammenfassung*.
Hot Spots	NaturalONE Profiler	**Brennpunkte** Zeigt die von Natural-Objekten, Statements und Programmzeilen verbrauchte CPU-Zeit und die verstrichene (elapsed) Zeit sowie die Häufigkeit der Ausführung eines Objekts oder Statements. Von einer profilierten Programmzeile aus können Sie direkt zu der entsprechenden Quellcode-Zeile navigieren. Siehe auch die entsprechende Beschreibung der Hot Spots unter *Using the Natural Profiler* in der *NaturalONE*-Dokumentation.
Data Evaluation	Natural Profiler Rich GUI-Schnittstelle	**Datenauswertung** Visualisiert die Profiler-Daten auf einer interaktiven Rich-GUI-Benutzeroberfläche. Sie können die Verteilung der CPU- und verstrichenen (elapsed) Zeit, die Adabas-Befehlszeit oder verschiedene Trefferquoten (Hit-Counts) auswerten und Kriterien für die Verteilung auswählen. Siehe auch das Beispiel für eine Datenauswertung im Abschnitt *Natural Profiler Rich GUI-Schnittstelle*.
Program Analysis	Natural Profiler Rich GUI-Schnittstelle	**Programmanalyse** Kombiniert den Quellcode der überwachten Programme mit den Profiler-Daten, wobei der Quellcode entsprechend dem Wert des ausgewählten KPIs eingefärbt wird. Siehe auch das Beispiel für eine Programmanalyse im Abschnitt *Natural Profiler Rich GUI-Schnittstelle*.
Program Trace	Profiler Utility (Batch)	**Programm-Trace** Zeigt den Programmablauf der profilierten Anwendung in der Aufrufhierarchie und die Anzahl der aufgetretenen Ereignisse. Siehe auch *Beispiel für einen Programm-Trace*.

Auswertung	Profiling-Tool	Beschreibung
Event Trace	NaturalONE Profiler, Profiler Utility	**Ereignis-Trace** Listet die aufgezeichneten Ereignisdaten in chronologischer Reihenfolge auf. Siehe auch ***Beispiel für ein Ereignis-Trace***, ***Beispiel für Trace-Datensätze auflisten***, und die entsprechende Beschreibung in der *NaturalONE*-Dokumentation unter *Using the Natural Profiler*.
Profiler Properties, **Profiler Statistics**	Natural Profiler Rich GUI-Schnittstelle, Profiler Utility (Batch)	**Profiler-Eigenschaften** Listet Profiler-Eigenschaften wie die Profiler-Revision und Statistiken der überwachten Anwendung auf, die z. B. die gesamte CPU- und die verstrichene (elapsed) Zeit anzeigen. Siehe auch das Beispiel mit Eigenschaften und Statistiken im Abschnitt ***Natural Profiler Rich GUI-Schnittstelle*** und Profiler ***Statistiken*** (Batch).

50 Codeabdeckung von Natural-Anwendungen (Code Coverage)

- Einführung in die Codeabdeckung 380
- Schnellstart für die Codeabdeckung 381
- Grundlegende Konzepte für die Codeabdeckung 383
- Plattformspezifische Codeabdeckung 384
- Tools zur Messung der Codeabdeckung 385
- Natural Code Coverage-Auswertungen 388

Dieses Dokument enthält allgemeine Informationen über die Codeabdeckung (Code Coverage) von Natural-Anwendungen.

Einführung in die Codeabdeckung

Normalerweise misst die Codeabdeckung den Grad, in dem der Quellcode eines Programms ausgeführt wird. Sie wird häufig für systematische Softwaretests verwendet. Je höher der Prozentsatz der Codeabdeckung ist, desto geringer ist die Wahrscheinlichkeit, dass der Code unentdeckte Softwarefehler in nicht ausgeführtem Code enthält.

Die Natural-Codeabdeckung (Code Coverage) wird zur Überwachung der ausgeführten Statements einer Natural-Anwendung verwendet. Sie sammelt die Abdeckungsdaten, während die Anwendung ausgeführt wird, und stellt Tools bereit, um die gesammelten Daten anschließend zu analysieren.

Die **Code Coverage**-Ansicht der Eclipse-basierten Entwicklungsumgebung NaturalONE und die **Program Coverage**-Tabelle der Profiler Utility zeigen - ausgedrückt in Prozent - wie viele der Statements der Natural-Objekte ausgeführt wurden. Das **Natural Coverage Plugin** für Jenkins visualisiert das Ergebnis eines Natural-Coverage-Zyklus direkt auf den Ergebnisseiten des Jenkins-Jobs.

Im NaturalONE-Editor und in der Tabelle **Statement Coverage** der Profiler Utility können Sie sehen, welche einzelnen Statement-Zeilen eines Natural-Objekts ausgeführt wurden. Hier können Sie auch sehen, welche Statement-Zeilen nicht oder nur teilweise abgedeckt wurden.

Wenn sich ein Statement im Quellcode über mehrere Zeilen erstreckt, wird in den Abdeckungsreports nur die Zeile erwähnt, in der das Statement beginnt.

Sie können die Abdeckungsdaten mit der Ausgabe der Profiler Utility im Text- oder Tabellenformat (CSV) exportieren. Die CSV-Tabelle kann mit einer Tabellenkalkulationssoftware wie Microsoft Excel ausgewertet werden.

GP- und Quellcode-Abdeckung

Wenn ein Natural-Quellcode `INCLUDE`-Statements enthält, wird der entsprechende Copycode in das generierte Objekt (das **GP** = Generated Program, das vom Compiler generierte, ausführbare (katalogisierte) Objekt) aufgenommen. Hinsichtlich der Abdeckung können beim Natural Profiler zwei Statement-Zahlen überwacht werden:

1. Die Anzahl der Statements im GP, das alle Copycodes rekursiv einschließt (ein Copycode kann weitere Copycodes einschließen).
2. Die Anzahl der Statements im Quellcode (Source), die die Copycodes nicht enthalten.

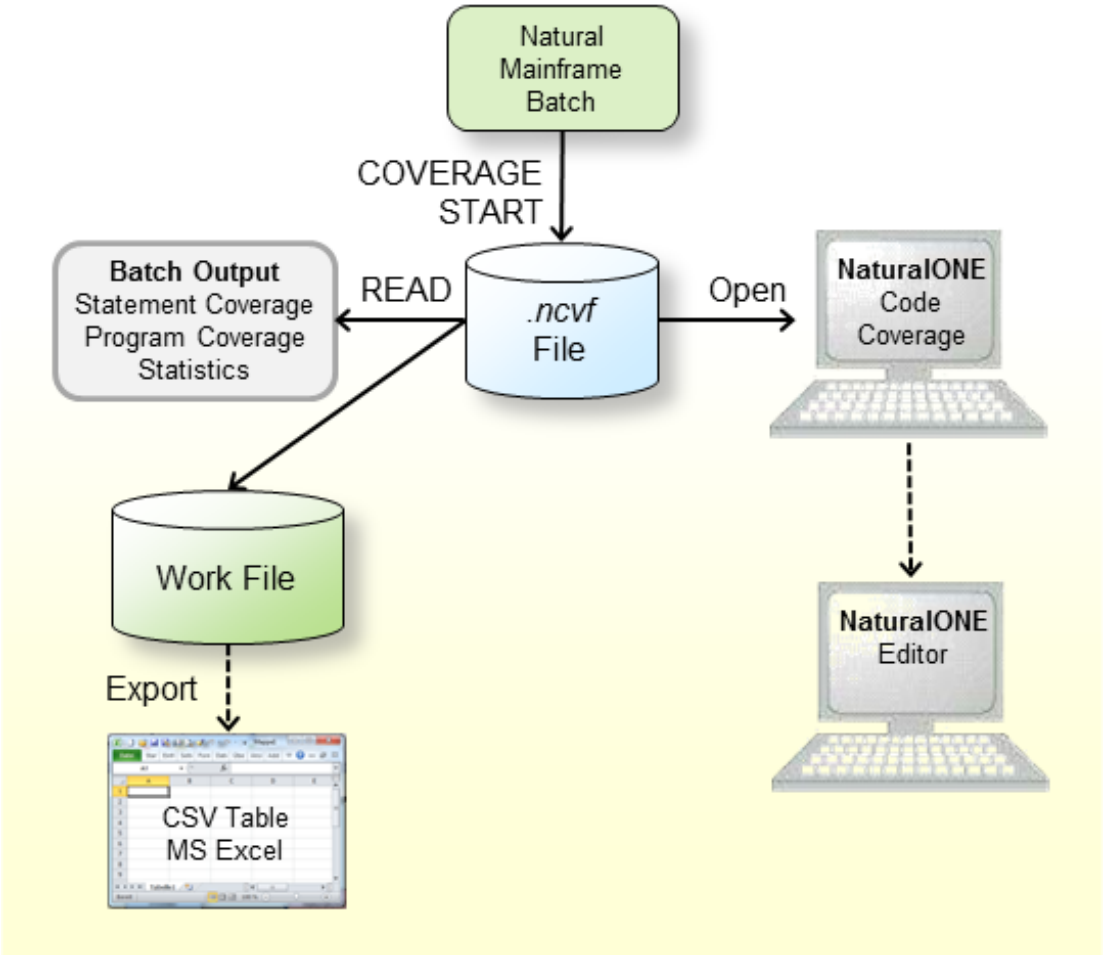
Die GP-Abdeckung (GP Coverage) spiegelt den Prozentsatz der abgedeckten Statements in der GP wider mit Berücksichtigung der Copycode Statements, während die Quellcode-Abdeckung

(Source Coverage) den Prozentsatz der abgedeckten Statements im Quellcode widerspiegelt, ohne die Copycodes zu berücksichtigen.

Schnellstart für die Codeabdeckung

Dieser Abschnitt beschreibt kurz die Schritte, die erforderlich sind, um die Messung der Codeabdeckung einer Natural-Batch-Anwendung durchzuführen und die Ergebnisse zu sichten. Die hier bereitgestellten Anweisungen können als Richtlinie dienen, wenn Sie mit der Benutzung der **Natural Code Coverage**-Funktion beginnen. Detaillierte Informationen zu den Schritten finden Sie weiter unten in diesem Kapitel.

Die Schritte hängen, wie in der folgenden Grafik dargestellt, von der Bewertung ab, die Sie für Ihre Anwendung durchführen möchten:



1. Überprüfen Sie, ob die Voraussetzungen erfüllt sind.

2. Fügen Sie in dem Natural Batch-Job die Profiler Utility-Funktionen `COVERAGE` und `START` hinzu, um die Sammlung von Code-Abdeckungsdaten zu starten.

   Im folgenden Beispiel für z/OS wird das Natural-Programm `TESTCOVP` in der Library `COVDEMO` behandelt.

```
//CMSYNIN  DD  *
PROFILER
FUNCTION=COVERAGE            /* Initialize coverage
  RESOURCE-NAME='ResNam'     /* Resource name
  REPLACE=YES                /* Replace the resource
  RESOURCE-LIB=RESLIB        /* Resource library
FUNCTION=START               /* Start data collection
END-PROFILER                 /* End Profiler input
LOGON COVDEMO
TESTCOVP
FIN
```

   Im obigen Beispiel werden die Profiler-Abdeckungsdaten in eine Ressourcendatei mit dem Namen `ResNam.ncvf` in der Library `RESLIB` geschrieben. Siehe auch ***Initialisierung der Codeabdeckung (Code Coverage)*** und ***Datensammlung starten und anhalten***.

3. Öffnen Sie die NCVF-Ressource in NaturalONE, um die **Code Coverage**-Ansicht zu erhalten.
4. Von der NaturalONE **Code Coverage**-Ansicht aus können Sie den Quellcode direkt bearbeiten. Der Editor zeigt alle Zeilen, die abgedeckte Statements enthalten, grün hinterlegt an.
5. Starten Sie einen Natural-Batch-Job mit der Profiler-Utility-Funktion `READ`, um die Programm- und Statement-Abdeckung und die Profiler-Statistiken zu drucken.

   Beispiel:

```
FUNCTION=READ            /* Read Profiler data
  RESOURCE-LIB=RESLIB    /* Resource library
  RESOURCE-TYPE=NCVF     /* Use resource type
  EVENT=ON               /* Print statement co
  PROGRAM=ON             /* Print program cove
  STATISTICS=ON          /* Print statistics
  EXPORT=ON              /* write to work 7
  FORMAT=C               /* Semicolon/Comma/Text
```

   Wenn das Schlüsselwort `EXPORT` der `READ`-Funktion der Profiler-Utility aktiviert ist, wird die Ausgabe in Arbeitsdatei 7 (Work File 7) geschrieben. Wenn `FORMAT` als `C` oder `S` angegeben wird, wird das Ergebnis als kommagetrennte Werte (CSV) geschrieben, wobei ein Komma bzw. ein Semikolon als Trennzeichen verwendet wird.

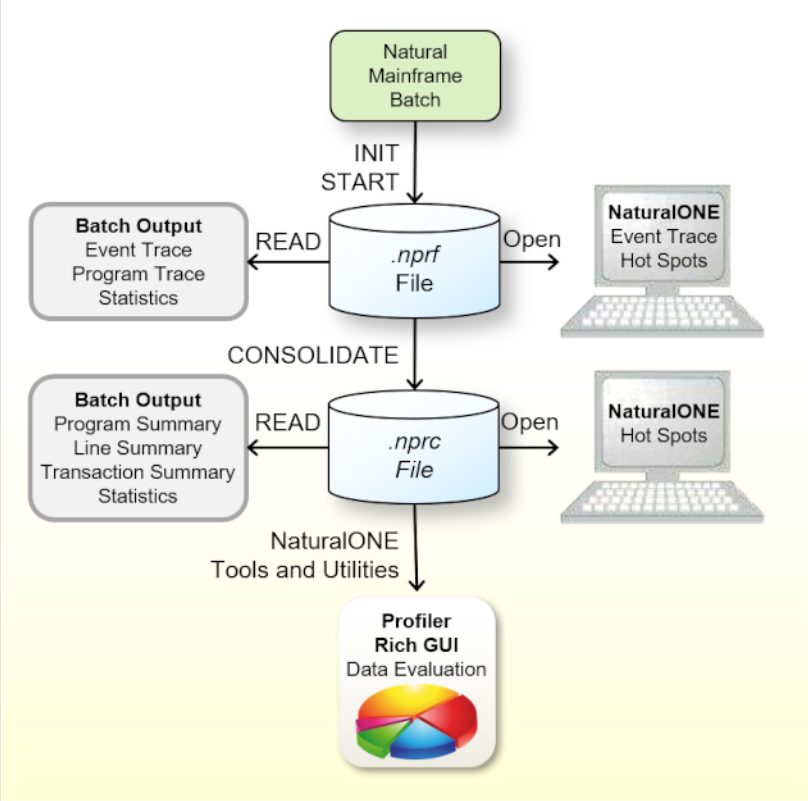
6. Exportieren Sie die Daten von Arbeitsdatei 7 mit einem beliebigen Tool (z. B. FTP) als CSV-formatierte Datei in eine Windows-Umgebung, wenn Sie sie in Microsoft Excel weiterverarbeiten möchten.

Anmerkungen:

1. Wenn der Ressourcenname nicht explizit in der READ-Funktion der Profiler Utility angegeben wird, wird die zuletzt erstellte NCVF-Ressource in der Library verwendet.
2. Die NaturalONE **Code Coverage**-Ansicht und der NaturalONE Editor sind in der NaturalONE-Dokumentation beschrieben.

Grundlegende Konzepte für die Codeabdeckung

Während eine Natural-Anwendung ausgeführt wird, können Abdeckungsdaten gesammelt und in eine NCVF-Ressourcendatei geschrieben werden.



- Auf dem Großrechner wird die Datensammlung durch die Profiler Utility im Batch-Modus durchgeführt. Daher ist die Codeabdeckung von Großrechneranwendungen nur verfügbar, wenn die Profiler Utility im Batch-Modus gestartet werden kann. Dies gilt für Großrechner-Batch-Anwendungen, aber auch für interaktive Großrechner-Anwendungen, die über Natural Studio oder RPC remote ausgeführt werden, sofern die entsprechenden Einstellungen vorgenommen wurden.

- Unter Windows und Linux wird die Datensammlung von Natural durchgeführt, wenn der Parameter `COVERAGE` angegeben ist. Dadurch ist Code Coverage immer möglich.
- Anders als beim Profiling gibt es unter NaturalONE keine direkte Codeabdeckung, bei der die Coverage ausgelöst und sofort von NaturalONE verarbeitet wird.
- Wenn NaturalONE gegen ein Natural auf Windows oder Unix läuft, kann die Sammlung von Code-Coverage-Daten in eine NCVF-Ressourcendatei durch Setzen des Natural-Parameters `COVERAGE` ausgelöst werden.
- NaturalONE verarbeitet die Abdeckungsdaten einer NCVF-Ressourcendatei und zeigt die Ergebnisse in der Natural **Code Coverage**-Ansicht und im Natural Source Editor an.
- Die Profiler Utility im Batch-Modus verarbeitet die Abdeckungsdaten einer NCVF-Ressourcendatei und liefert Ergebnisse zur Statement- und Programmabdeckung. Die Daten können auch im CSV-Format (kommagetrennte Werte) in eine Datei geschrieben werden, die mit einer Tabellenkalkulationssoftware (z.B. Microsoft Excel) weiter analysiert werden kann.

Plattformspezifische Codeabdeckung

Generell können Sie die Natural Code Coverage auf Linux-, Windows- und z/OS-Plattformen durchführen. Wie Sie dabei vorgehen, hängt von der jeweiligen Plattform und der Art der Anwendungsverarbeitung ab:

z/OS

- Die Messung der Codeabdeckung bei interaktiven z/OS-Anwendungen, die remote von Natural Studio oder RPC ausgeführt werden, wird mit der Profiler Utility im Batch-Modus durchgeführt.
- Die Messung der Codeabdeckung bei z/OS-Batch-Anwendungen wird mit der Profiler Utility im Batch-Modus durchgeführt.



Anmerkung: Die Messung der Codeabdeckung ist bei interaktiven z/OSanwendungen, die lokal auf z/OS oder remote von NaturalONE ausgeführt werden, nicht verfügbar.

Linux und Windows

- Die Codeabdeckung von interaktiven Linux- und Windows-Anwendungen wird mit der NaturalONE-Code Coverage oder der Natural-Code Coverage für Linux bzw. Windows gemessen.
- Die Messung der Codeabdeckung bei Linux- und Windows-Batch-Anwendungen erfolgt mit der Natural Code Coverage für Linux bzw. Windows.

Tools zur Messung der Codeabdeckung

Der folgende Abschnitt enthält eine Zusammenfassung der Merkmale der Natural Profiling Tools.

- Funktionsumfang von NaturalONE Code Coverage
- Funktionsumfang der Natural Code Coverage für Linux und Windows
- Funktionsumfang der Profiler Utility
- Merkmale des Natural Code Coverage Spreadsheet

Funktionsumfang von NaturalONE Code Coverage

- Liest und analysiert Natural Code Coverage-Ressourcendateien, die Abdeckungsdaten enthalten, die von der z/OS-Profiler Utility im Batch-Modus oder von der Natural Code Coverage für Linux oder Windows gesammelt wurden.
- Die Codeabdeckung bei interaktiven Natural-Anwendungen auf Linux oder Windows kann gemessen werden, indem die Natural-Codeabdeckung für Linux oder Windows aktiviert und die entsprechende Natural Code Coverage-Ressourcendatei gelesen wird.
- Die Natural Code Coverage-Ansicht zeigt, wie viel Prozent der Statements der Natural-Objekte ausgeführt wurden.
- Von der Natural Code Coverage-Ansicht aus können die beteiligten Natural-Objekte bearbeitet werden. Der NaturalONE-Editor zeigt alle abgedeckten Zeilen mit grünem Hintergrund an.



Anmerkung: Die interaktive Codeabdeckung von Natural-Anwendungen auf z/OS-Plattformen wird derzeit nicht unterstützt.

Funktionsumfang der Natural Code Coverage für Linux und Windows

- Codeabdeckung von interaktiven oder Natural-Batch-Anwendungen auf Linux- oder Windows-Plattformen.
- Bietet Funktionen für die Behandlung großer Datenmengen:
  - automatisches Ereignisfilter,
  - automatische Datenkonsolidierung.
- Speichert die Code Coverage-Daten als Natural Code Coverage-Ressourcendatei.

Funktionsumfang der Profiler Utility

Batch-Modus (z/OS)

- Codeabdeckung von Natural-Batch-Anwendungen auf z/OS-Plattformen.
- Codeabdeckung von interaktiven z/OS-Anwendungen, die remote von Natural Studio oder gegen einen Natural RPC-Server ausgeführt werden.
- Bietet Funktionen für die Behandlung großer Datenmengen:
  - Programm-, Zähl- und Zeitfilter,
  - automatisches Ereignisfilter,
  - automatische Datenkonsolidierung.
- Speichert Code-Coverage-Daten als Natural-Code-Coverage-Ressourcendatei.
- Liest und analysiert Natural-Code-Coverage-Ressourcendateien.
- Listet die **Program Coverage**-Tabelle für alle aufgerufenen Natural-Objekte mit
  - Prozentsatz der abgedeckten Statements,
  - Anzahl der abgedeckten Statements,
  - Anzahl der nicht abgedeckten („missed“) Statements und
  - Gesamtzahl der Statements des Objekts.
- Die Statement-Abdeckung (**Statement Coverage**) listet den Quellcode jedes Natural-Objekts, auf das zugegriffenen wurde, und zeigt für jede Zeile den Prozentsatz der abgedeckten Statements an.
- Exportiert Natural-Code-Coverage-Daten im CSV-Format (kommagetrennte Werte), die mit einer Tabellenkalkulationssoftware (z. B. Microsoft Excel) weiter analysiert werden können.
- Sammelt und zeigt Profiler- und Codeabdeckungseigenschaften und -statistiken an.



Anmerkung: Auf z/OS gibt es keine Eins-zu-Eins-Beziehung zwischen einem Natural-Quellcode-Statement und dem entsprechenden Objektcode im katalogisierten Objekt. Die Natural-Codeabdeckung auf z/OS überwacht den Objektcode und nicht den Natural-Quellcode. Daher können mehrere Natural-Statements in einem Coverage-Eintrag zusammengeführt werden und umgekehrt kann ein Natural-Statement mehrere Coverage-Einträge abdecken.

Batch-Modus (Linux und Windows)

- Liest und analysiert Natural-Code-Coverage-Ressourcendateien.
- Listet die **Program Coverage**-Tabelle für alle Natural-Objekte, auf die zugegriffenen wurde, mit folgenden Angaben:
  - Prozentsatz der abgedeckten Statements,
  - Anzahl der abgedeckten Statements,

  - Anzahl der nicht abgedeckten („missed“) Statements und
  - Gesamtzahl der Statements des Objekts.
- Die Statement-Abdeckung (**Statement Coverage**) listet den Quellcode jedes Natural-Objekts, auf das zugegriffenen wurde, und zeigt für jede Zeile den Prozentsatz der abgedeckten Statements an.
- Exportiert Natural-Code-Coverage-Daten im CSV-Format (kommagetrennte Werte), die mit einer Tabellenkalkulationssoftware (z. B. Microsoft Excel) weiter analysiert werden können.



Anmerkung: Unter Windows und Linux werden nicht abgedeckte („missed“) Statements nicht erfasst. Daher kann die Statement Coverage nur Zeilen markieren, die abgedeckte Statements enthalten, und die Abdeckung dieser Zeilen beträgt immer 100%.

Merkmale des Natural Code Coverage Spreadsheet

- Vorlage zum Einfärben der Natural-Codeabdeckungsdaten, die von der Natural Profiler Utility im CSV-Format (kommagetrennte Werte) exportiert werden.
- Programm- und Copycode-Abdeckung mit Quellcode- und GP-Zählern für
  - Prozentsatz der abgedeckten Statements,
  - Anzahl der abgedeckten Statements,
  - Anzahl der „missed“ (nicht abgedeckten) Statements und
  - Gesamtzahl der Statements des Objekts.
- Statement-Abdeckung des Objekt-Quellcodes, wobei die Zeilen wie folgt eingefärbt werden
  - grün – wenn alle Statements der Zeile abgedeckt sind,
  - gelb – wenn die Statements der Zeile teilweise abgedeckt sind,
  - rot – wenn alle Statements der Zeile fehlen,
  - grau – wenn die Zeile leer ist oder nur Kommentare enthält.
- Profiler- und Code-Abdeckungseigenschaften und Statistiken (für z/OS-Daten).



Anmerkung: Eine Microsoft Excel-Tabellenvorlage für die Natural-Codeabdeckung ist als Ressource in der Natural-Profiler-Bibliothek `SYSPRFLR` auf Linux und Windows verfügbar.

Natural Code Coverage-Auswertungen

Dieser Abschnitt beschreibt die Programmabdeckungsauswertungen, die von den Natural Code Coverage Tools bereitgestellt werden:

- Programmabdeckung (Program Coverage)
- Zeilen- und Statement-Abdeckung
- Profiler-Eigenschaften und Statistikdaten

Programmabdeckung (Program Coverage)

Die **Program Coverage** gibt Ihnen einen Überblick über die ausgeführten Programme und den Umfang des Codes, der von der Anwendung abgedeckt wurde.

Programmabdeckungsreport

Der Programmabdeckungsreport der Profiler Utility zeigt die Abdeckung (in Prozent der Gesamtzahl der Statements) jedes ausgeführten Natural-Objekts an. Er zeigt für jedes Objekt an, wie viele Statements abgedeckt wurden bzw. nicht abgedeckt wurden („missed“) und wie viele Statements insgesamt enthalten sind. Außerdem fasst er die Werte für alle Objekte in einer Library und die Gesamtwerte für alle Librairies zusammen.

Wird die Ausgabe im Textformat geschrieben, wird nur die GP-Abdeckung angegeben. Werden die Daten im CSV-Format (kommagetrennte Werte) exportiert, wird auch die Quellcodeabdeckung angegeben. Zusätzlich werden die Zähler für alle inkludierten Copycodes gedruckt.

Im Folgenden finden Sie ein Beispiel für das Textformat:

```
Program Coverage
----------------
Library  Object   Ty Coverage%  Covered    Missed    Total
COVDEMO  TESTCOVN N      84.0%       37         7       44
COVDEMO  TESTCOVP P      69.2%        9         4       13
COVDEMO  -------- --      80.7%       46        11       57
Totals   -------- --      80.7%       46        11       57
```

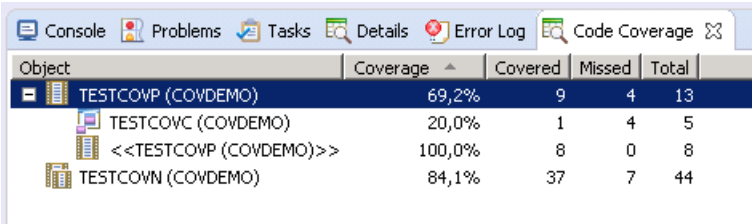
Im obigen Beispiel für die Programmabdeckung wurden `69,2 %` von `13` Statements im Programm `TESTCOVP` abgedeckt, was `9` abgedeckten und 4 nicht abgedeckten („missed“) Statements entspricht. `80,7` % der Statements der Objekte, auf die in der Library `COVDEMO` zugegriffen wurde, wurden abgedeckt, was auch dem Gesamtwert für den gesamten Anwendungslauf entspricht.

Code Coverage-Ansicht

Die **Code Coverage**-Ansicht von NaturalONE zeigt die Abdeckung (in Prozent der Gesamtzahl der Statements) jedes ausgeführten Natural-Objekts an. Sie zeigt für jedes Objekt an, wie viele Statements abgedeckt wurden oder nicht abgedeckt wurden („missed“) und wie viele Statements insgesamt enthalten sind. Wenn Copycodes enthalten sind, kann der Objektknoten geöffnet werden, um die Abdeckung des Copycodes anzuzeigen. Generell spiegeln die Zähler die GP-Abdeckung wider (Copycodes enthalten). Die Quellcodeabdeckung (Copycodes nicht enthalten) wird in der Zeile angezeigt, in der der Objektname in spitzen Klammern `<< >>` eingeschlossen ist.

Von jeder Zeile aus können Sie direkt zu dem entsprechenden Quellcode navigieren, um die Statement-Abdeckung zu sehen.

Beispiel:



Im obigen Beispiel hat das Programm `TESTCOVP` eine GP-Abdeckung von `69,2` Prozent, wobei im Programm selbst alle `8` Statements abgedeckt sind (100% Quellcodeabdeckung) und im enthaltenen Copycode `TESTCOVC` nur `1` von `5` Statements abgedeckt wurde.

Zeilen- und Statement-Abdeckung

Die **Statement Coverage** zeigt, welche Zeilen des Programms ausgeführt wurden. Bei z/OS-Daten zeigt die Profiler Utility außerdem an, welche Zeilen mit Statements nicht oder nur teilweise ausgeführt wurden.

Statement Coverage-Report

Der **Statement Coverage**-Report der Profiler Utility zeigt für jede Quellcodezeile die Abdeckung der Statements in der Zeile an.

Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, werden auch die Anzahl der abgedeckten oder nicht abgedeckten („missed“) Statements und die Gesamtzahl der Statements in der Zeile ausgedruckt. Die Microsoft Excel-Tabellenvorlage, die mit Natural unter Linux und Windows geliefert wird, kann verwendet werden, um die Zeilen entsprechend der Abdeckung einzufärben.

Wenn eine Quelle ein `INCLUDE`-Statement enthält, wird der entsprechende Copycode-Quellcode direkt nach dem `INCLUDE`-Statement in den Report aufgenommen.

Im Folgenden finden Sie ein Beispiel für einen Export im CSV-Format, der mit Hilfe einer Microsoft Excel-Tabelle eingefärbt wurde:

Line	Source	Coverage%	Covered	Missed	Total
10	* Test function Coverage		0	0	0
20	* Subprogram TESTCOVN		0	0	0
30	DEFINE DATA		0	0	0
40	PARAMETER		0	0	0
50	1 FUNC (I2) /* function		0	0	0
60	1 RET-CODE (I4) /* Return code		0	0	0
70	END-DEFINE		0	0	0
80	*		0	0	0
90	/* Return 0 by default		0	0	0
100	RESET RET-CODE	100	1	0	1
110	*		0	0	0
120	DECIDE ON FIRST VALUE OF FUNC	100	1	0	1
130	VALUE 0	50	1	1	2
140	PRINT 'Test function 0'	0	0	1	1
150	VALUE 1	66	2	1	3
160	PRINT 'Test function 1'	100	1	0	1
170	VALUE 2	100	3	0	3
180	PRINT 'Test function 2'	100	1	0	1
190	VALUE 3	100	3	0	3
200	PRINT 'Test function 3'	100	1	0	1
210	VALUE 4	100	3	0	3
220	PRINT 'Test function 4'	100	1	0	1
230	VALUE 5	100	3	0	3
240	PRINT 'Test function 5'	100	1	0	1
250	VALUE 6	100	3	0	3
260	PRINT 'Test function 6'	100	1	0	1
270	VALUE 7	100	3	0	3
280	PRINT 'Test function 7'	100	1	0	1
290	VALUE 8	100	3	0	3
300	PRINT 'Test function 8'	100	1	0	1
310	VALUE 9	33	1	2	3
320	PRINT 'New test function 9'	0	0	1	1
330	NONE VALUE	100	1	0	1
340	RET-CODE := 1 /* Unsupported function	0	0	1	1
350	END-DECIDE		0	0	0
360	*		0	0	0
370	END	100	1	0	1

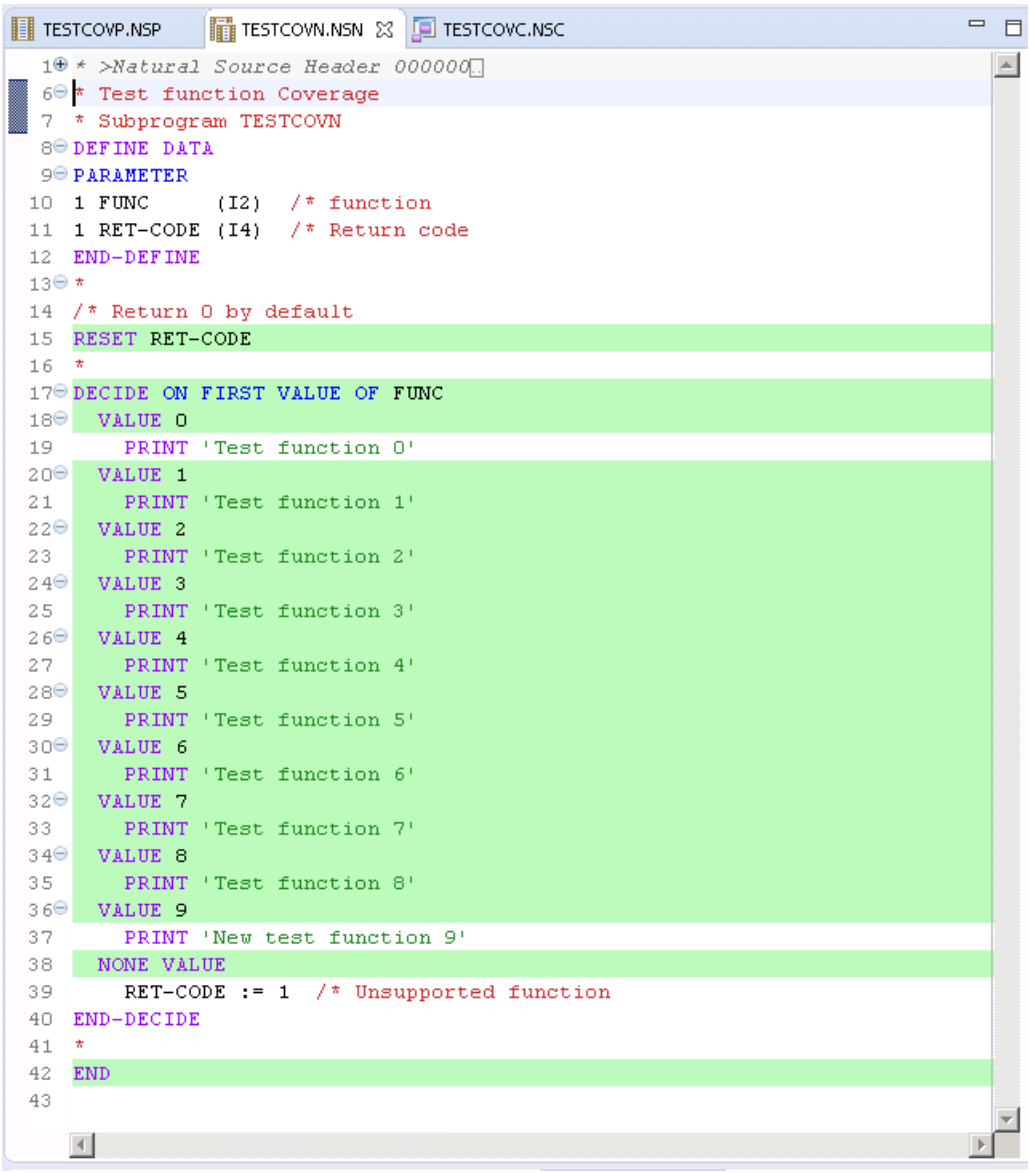
Die drei roten Zeilen des Unterprogramms `TESTCOVN` sind nicht ausgeführt worden. Der Testlauf deckt also nicht die neue Testfunktion `9` ab. Er deckt außerdem weder die (alte) Funktion `0` noch den Fall ab, dass das Subrogramm mit einer nicht unterstützten Funktion aufgerufen wird.

Die Daten stammen vom Großrechner. Daher beziehen sich die Zählungen auf Objektcode-Statements und nicht auf Natural-Statements. Einem Natural-`VALUE`-Statement können bis zu 3 Objektcode-Statements entsprechen. Die gelben Zeilen beziehen sich auf `VALUE`-Statements, bei denen ein Teil des Objektcodes abgedeckt wurde und ein anderer nicht.

NaturalONE Source Editor (Quellcode-Editor)

Wenn der Source Editor aus der **Code Coverage**-Ansicht in NaturalONE geöffnet wird, wird der Quellcode entsprechend der Codeabdeckung eingefärbt. Jede Zeile, in der ein oder mehrere Statements abgedeckt sind, ist grün hinterlegt.

Beispiel:



Der **Source Editor** zeigt alle Zeilen, in denen mindestens ein Statement ausgeführt worden ist, grün hinterlegt an. Also sind alle Zeilen des `DECIDE`-Statement, mit Ausnahme der Zeilen `19`, `37` und `39`, ausgeführt worden.

Profiler-Eigenschaften und Statistikdaten

Die Anzeige **Profiler Properties and Statistics** steht sowohl bei der Natural Profiler Utility als auch bei der Profiler Rich GUI-Schnittstelle zur Verfügung. Sie liefert eine Auflistung der Profiler-Eigenschaften, z.B. die Profiler-Revisionsnummer. Darüber hinaus zeigt sie Statistikdaten der überwachten Anwendung, z.B. die CPU-Gesamtzeit und die verstrichene (elapsed) Zeit. Zu einem Codeabdeckungslauf zeigt sie die Statistikdaten der Abdeckung.

Beispiel:

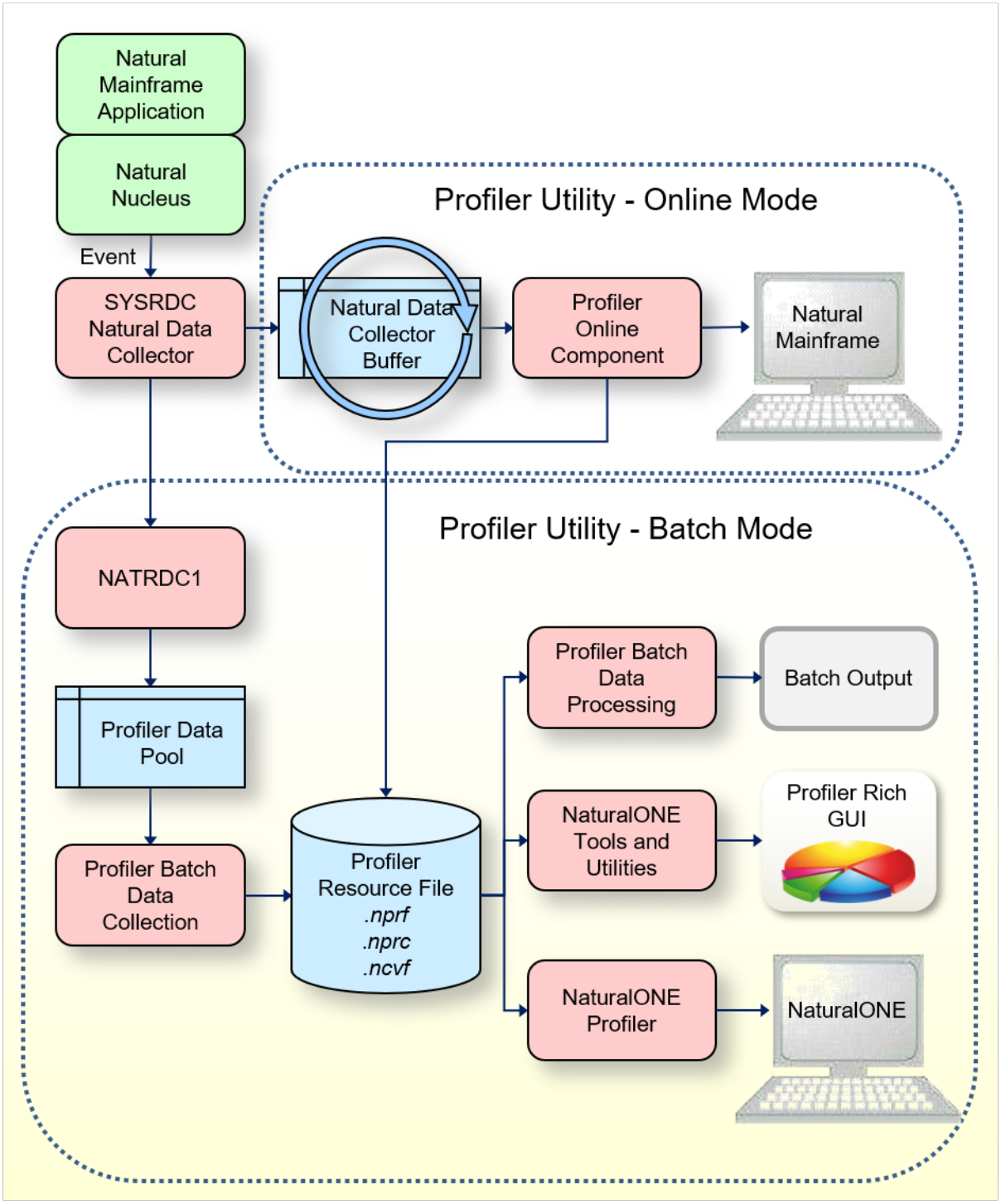
```
************************************************************************
* 13:30:48          ***** NATURAL PROFILER UTILITY *****         2017-09-04
* User SAG                      - Statistics -                      COVREAD
*
* General Info
* Machine class ..................... MAINFRAME
* Environment ....................... Batch
...
* Coverage
* Coverage .......................... ON
* Missed statements recorded ......... ON
* Coverage records ................... 60
* Program information records ........ 3
* Coverage records/block ............. 60
* Bytes/coverage record .............. 10.3
* Programs covered ................... 2
* Statement coverage (percent) ....... 80.7
* Statements covered ................. 46
* Statements missed .................. 11
* Statements total ................... 57
```

51 Grundlegendes Konzept der Profiler Utility

- Profiler Utility - Übersicht 394
- Datensammlung (Data Collection) im Batch-Modus 398
- Datenkonsolidierung, Codeabdeckung und Datenverarbeitung 399
- Stichprobentechnik (Natural Profiler Sampling) 403
- Profiler-Performance im Batch-Modus 406
- Langlaufende Anwendungen profilieren 407
- Verwandte Themen 411

Die Profiler-Utility ist für interaktive (Online-) und Batch-Natural-Anwendungen verfügbar. Interaktive Natural-Anwendungen können auch von NaturalONE profiliert werden, was in der NaturalONE-Dokumentation detailliert beschrieben wird.

Profiler Utility - Übersicht



Die Grafik veranschaulicht den Prozessablauf, wenn die Profiler Utility im Online- oder im Batch-Modus Daten verfolgt (Profiler-Tracing).

Die Profiler-Utility basiert, wie in der Grafik dargestellt, auf der Natural Data Collector-Technik. Wenn in einer Natural-Großrechner-Anwendung ein Ereignis wie z. B. ein Programmstart auftritt, ruft der Natural-Nukleus den Natural Data Collector der SYSRDC Utility auf, der die Natural-

Ereignisdaten im Natural Data Collector Buffer sammelt und die Ereignisdaten an die User-Exits des Natural Data Collector weitergibt. Die Art und Weise, wie die Ereignisdaten weiterverarbeitet werden, hängt davon ab, in welchem Modus Sie die Profiler Utility ausführen.

Die Grafik wird im folgenden Abschnitt erläutert:

Profiler Utility im Online-Modus

Die Profiler Utility im Online-Modus ist menübasiert. Sie dient dazu, einen schnellen Überblick über die letzten Aktionen einer interaktiven Natural-Anwendung zu erhalten.

- Der Natural Data Collector-Pufferspeicher (**Natural Data Collector Buffer**) hat eine maximale Größe von 128 KB und arbeitet im Wrap-around-Modus. Er liefert die jüngsten Ereignisse, während die ältesten Daten überschrieben werden, wenn der Puffer voll ist. Die Datensammlung wird gestoppt, wenn die Daten gelesen werden, und der Puffer wird geleert, wenn die Sammlung neu gestartet wird.
- Die Profiler-Online-Komponente (**Profiler Online Component**) liest die Ereignisdaten aus dem Natural Data Collector Buffer. Sie bietet Funktionen zur Steuerung des Profiler-Tracing, zur Auswahl der gewünschten Ereignistypen, zur Pflege und Anzeige von Trace-Datensätzen, zum Herunterladen der Ereignisdaten auf den PC und zum Speichern der Ereignisdaten als Profiler-Ressourcendatei. Die Profiler-Ressourcendatei kann entweder mit den Profiler-Batch-Datenverarbeitungsfunktionen, NaturalONE oder der Profiler Rich GUI-Schnittstelle verarbeitet werden.

Profiler Utility im Batch-Modus

Die Profiler-Utility im Batch-Modus wird durch JCL-Eingabe gesteuert. Sie wurde für die Analyse von Natural-Batch-Anwendungen entwickelt.

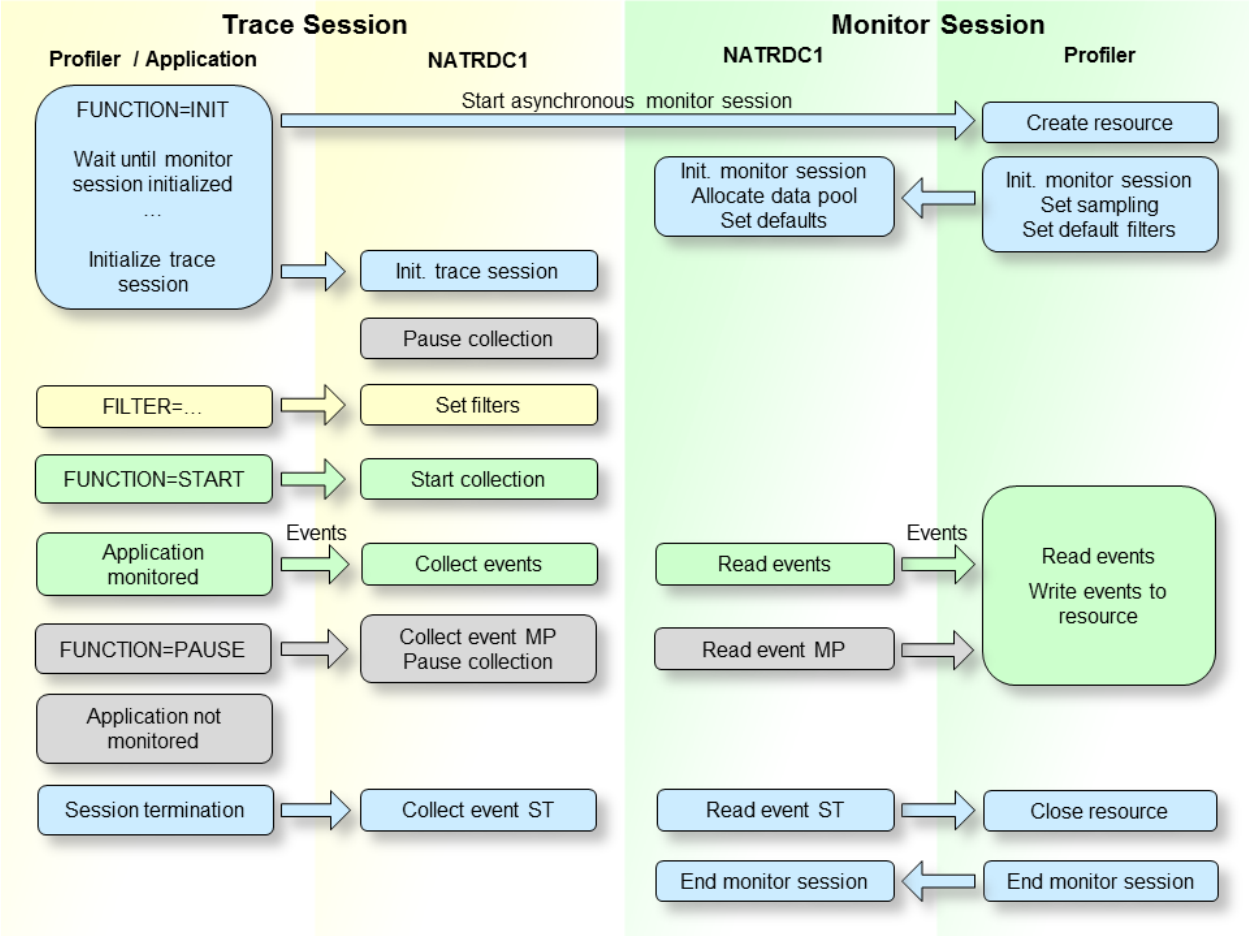
- Der User Exit **NATRDC1** des Natural Data Collector sammelt die Natural-Ereignisdaten im Profiler-Daten-Pool. Er erweitert den Natural Data Collector-Ereignisdatensatz um zusätzliche Ereignisinformationen und führt spezielle Funktionen wie Filtern oder Sampling (Stichprobentechnik) aus.
- Der Profiler-Daten-Pool (**Profiler Data Pool**) hat eine maximale Größe von 2 GB und sammelt die Natural-Ereignisdaten für die Profiler Utility im Batch-Modus und für den NaturalONE Profiler. Eine spezielle Logik sorgt dafür, dass keine Daten verloren gehen, auch wenn der Pool voll ist.
- Die Profiler-Batch-Datensammlung (**Profiler Batch Data Collection**) liest die Ereignisdaten asynchron aus dem Profiler-Daten-Pool, während die Anwendung und die Ablaufverfolgung (Tracing) weiterlaufen. Sie bietet Funktionen zur Steuerung des Profiler-Tracing, zur Auswahl erforderlicher Ereignistypen, zum Filtern, Entnehmen von Stichproben (Sampling) oder Konsolidieren der Daten, zum Messen der Natural-Codeabdeckung und zum Schreiben der resultierenden Ereignisse in eine Profiler-Ressourcendatei. Allgemeine Statistiken und ein Ereignis-Trace mit den wichtigsten Daten können in die Standardausgabe geschrieben werden.
- Die Profiler-Ressourcendatei (**Profiler Resource File**) hat die Namenserweiterung `.nprf`, `.nprc` oder `.ncvf` und ist eine Ressource, die abhängig von der Ressource-Library-Auswahl in der Natural-Systemdatei FNAT oder FUSER zugeordnet wird. Sie enthält die Ereignisdaten

in komprimiertem Format mit einem um bis zu 80 % verringerten Speicherbedarf. Für eine optimierte Übertragung nach NaturalONE werden die Daten in Datenblöcken zusammengefasst.

- Während der Profiler-Batch-Datenverarbeitung (**Batch Data Processing**) liest und verarbeitet der Profiler die Ereignisdaten aus den Profiler-Ressourcendateien. Dabei stehen Funktionen für Datenkonsolidierung (Aggregation), Ereignis-Tracing und Programm-Tracing zur Verfügung. Der Profiler bietet außerdem eine Programmzusammenfassung, eine Zeilenzusammenfassung und eine Transaktionszusammenfassung und zeigt Profiler-Eigenschaften und -Statistiken an. Für die Natural-Codeabdeckungsdaten werden Programm- und Statement-Abdeckungsreports bereitgestellt. Die resultierenden Daten können in eine Datei im Text- oder CSV-Format (kommagetrennte Werte) exportiert werden.
- In der **Natural Server**-Ansicht von NaturalONE werden die Profiler-Ressourcendateien als NPRF- oder NPRC-Ressourcen aufgelistet. Die Kontextmenüfunktion **Open with Natural Profiler** liest die Ressourcendaten in den NaturalONE Profiler. Der NaturalONE Profiler bietet die allgemeine Analyse der Ereignisdaten. Sie zeigt, wie sich die CPU-Zeit oder die verstrichene (elapsed) Zeit auf die Programme, Statements und sogar Programmzeilen der Anwendung verteilt und wie oft ein Statement ausgeführt wurde. Zusätzlich wird der vollständige Ereignis-Trace bereitgestellt.
- Codeabdeckungs-Ressourcedateien (Coverage Resource Files, NCVF) können aus der NaturalONE Server-Ansicht zu einem Projekt im NaturalONE-Arbeitsbereich hinzugefügt werden. Die Kontextmenüfunktion **Open with Natural Code Coverage** liest die Ressourcendaten und zeigt sie in der **Code Coverage**-Ansicht an.

  Von der NaturalONE **Code Coverage**-Ansicht aus kann der **Natural Source Editor** geöffnet werden. Er zeigt alle Quellcodezeilen mit einem oder mehreren abgedeckten Statements mit grünem Hintergrund.
- Über die **NaturalONE Tools and Utilities** kann die Profiler Rich GUI-Schnittstelle gestartet werden. Sie listet alle Profiler-Ressourcen der gegebenen Library auf. Für eine ausgewählte Profiler-Ressource werden die Eigenschaften und Statistiken der Profil-Erstellung angezeigt. Das Tool bietet Funktionen zur Datenkonsolidierung und -auswertung, zur Analyse überwachter Programme oder zum Löschen einer Profiler-Ressourcendatei. Für die Auswertung werden Kreisdiagramme verwendet, um die Verteilung der Profiler-Leistungskennzahlen (KPIs, Key Performance Indicators) für ausgewählte Kriterien wie die Verteilung der CPU-Zeit für Programme anzuzeigen. Zusammengefasste Summen der KPI-Werte werden ebenfalls angezeigt. Für die Programmanalyse wird der Quellcode der überwachten Programme mit den Profiler-Daten kombiniert, wobei der Quellcode entsprechend dem Wert des ausgewählten KPI farblich hinterlegt wird.

Datensammlung (Data Collection) im Batch-Modus



Diese Grafik zeigt, welche Aktionen die Profiler Utility bei der Profil-Erstellung einer Natural-Anwendung im Batch-Modus ausführt.

- Die Natural-Trace-Session ist die Sitzung, in der die Natural Batch-Anwendung ausgeführt und die Profiler-Trace-Daten generiert und gesammelt werden.
- Wenn die Funktion `INIT` der Profiler Utility ausgeführt wird, wird eine neue Natural-Session als asynchroner Subtask gestartet. Diese Session wird als Monitor-Session bezeichnet, weil sie die Ereignisse überwacht. In beiden Sessions läuft eine Natural-Nukleus-Instanz mit einem verlinkten NATRDC1-Exit. Die `INIT`-Funktion löst die Ausführung der Profiler Utility in der Monitor-Session aus und leitet die `INIT`-spezifischen Schlüsselwörter an sie weiter.

  In der Monitor-Session wird die Profiler-Ressourcendatei erstellt, in der die Ereignisse später gespeichert werden. Dann wird der Exit NATRDC1 aufgerufen, um die Monitor-Session zu initialisieren. NATRDC1 ordnet den Profiler-Daten-Pool zu, der verwendet wird, um die

Ereignisdaten von der Trace-Session in die Monitor-Session zu übertragen. Er initialisiert auch die Entnahme von Stichproben (Sampling) und legt Standardwerte für Filter fest.

- Wenn der Profiler in der Trace-Session feststellt, dass die Monitor-Session erfolgreich initialisiert wurde, ruft er den NATRDC1-Exit auf, um die Trace-Session zu initialisieren. Standardmäßig wird die Datensammlung nach der Initialisierung angehalten.
- Nach der Initialisierung können die Profiler-Filter eingestellt werden. Dies sollte durchgeführt werden, bevor die Datensammlung gestartet wird, damit die Filter gleich zu Beginn einsatzbereit sind.
- Die Datensammlung beginnt mit der `START`-Funktion. Der Profiler in der Trace-Session sendet eine Startanforderung an den Exit NATRDC1, der die Ereignisse der nachfolgenden Natural-Anwendungen im Profiler-Daten-Pool sammelt. Gleichzeitig ruft der Profiler in der Monitor-Session den Exit NATRDC1 auf, um die Ereignisdaten aus dem Profiler-Daten-Pool zu lesen. Der Speicherplatz der Ereignisdaten wird sofort freigegeben, so dass die Trace-Session ihn wiederverwenden kann. Der Profiler in der Monitor-Session komprimiert die Ereignisdaten und schreibt sie in die Profiler-Ressourcendatei.
- Mit der `PAUSE`-Funktion kann die Datensammlung angehalten werden. Der Profiler sendet in der Trace-Session eine Pausenanforderung an den Exit NATRDC1, der ein `MP`-Ereignis (Monitor Pause) in den Profiler-Daten-Pool schreibt und die Datensammlung aussetzt, d.h. er lehnt alle Ereignisse ab, bis eine Startanforderung empfangen wird oder die Trace-Session endet.
- Am Ende der Anwendung wird ein `ST`-Ereignis (Session Termination) in den Profiler-Daten-Pool geschrieben. Die Trace-Session wird jedoch nicht sofort beendet (was den Subtask der Monitor-Session abbrechen würde). Stattdessen wartet sie, bis die Monitor-Session die verbleibenden Daten aus dem Profiler-Daten-Pool liest. Wenn der Profiler in der Monitor-Session das `ST`-Ereignis im Profiler-Daten-Pool findet, schließt er die Profiler-Ressourcendatei, schreibt die Statistikdaten und sendet eine Beendigungsanforderung an den Exit NATRDC1, der die Monitor-Session beendet. Zuletzt wird auch die Trace-Session beendet.

Datenkonsolidierung, Codeabdeckung und Datenverarbeitung

Dieser Abschnitt behandelt die folgenden Themen:

- Datenkonsolidierung
- Natural-Codeabdeckung (Code Coverage)

- Datenverarbeitung

Datenkonsolidierung

Wenn eine Natural-Anwendung profiliert wird, sammelt der Natural Profiler für jedes Ereignis einen Datensatz. Je nach Anwendung kann dies riesige Datenmengen erzeugen, insbesondere wenn Natural-Statements überwacht werden. Je mehr Daten der Profiler generiert, desto mehr Zeit wird benötigt, um die Daten vom Server zum NaturalONE-Client zu transportieren.

Die Profiler Utility und die Profiler Rich GUI-Schnittstelle bieten eine serverseitige Datenkonsolidierung, die die Datenmenge erheblich reduziert und gleichzeitig den Transportfluss erhöht. Die Profiler-Datenkonsolidierung kombiniert ähnliche Datensätze in einem konsolidierten Datensatz mit aggregierten Zeitwerten und einem Trefferzähler. Die konsolidierten Daten werden in eine Ressourcendatei geschrieben, die den gleichen Namen wie die entsprechende nicht konsolidierte Ressourcendatei hat, aber die Erweiterung `.nprc` (Natural Profiler resource consolidated).

Während der Profil-Erstellung können die Daten sofort konsolidiert werden, indem das Schlüsselwort `CONSOLIDATE` der `INIT`-Funktion der Profiler Utility aktiviert wird. Nicht konsolidierte Daten einer NPRF-Datei können später mit der Profiler Utility oder der Profiler Rich GUI-Schnittstelle `CONSOLIDATE`-Funktion konsolidiert werden.

Beispiel:

Ein Natural-Statement wird 1000 Mal in einer `FOR`-Schleife ausgeführt. Die nicht konsolidierten Daten enthalten 1000 Datensätze für jede Ausführung des Statement. Jeder Datensatz enthält neben anderen Informationen die Ereigniszeit und den CPU-Zeitstempel. Die Profiler-Konsolidierung fasst diese 1000 Datensätze zu einem konsolidierten Datensatz zusammen. Alle gemeinsamen Informationen (z.B. die Library oder der Programmname) werden beibehalten, die verstrichene (elapsed) Zeit und die CPU-Zeit jeder Ausführung des Statement werden bestimmt, zusammengefasst und im Konsolidierungsdatensatz gespeichert. Darüber hinaus wird eine Trefferzahl von 1000 aufgezeichnet.



Anmerkungen:

1. Eine NPRC-Ressourcendatei, die serverseitig konsolidiert wurde, enthält die gleichen Hot-Spot-Werte wie die entsprechende nicht konsolidierte NPRF-Ressource, wird aber mit NaturalONE viel schneller geöffnet.
2. Die konsolidierten Daten enthalten nicht den Ereignis-Trace (Zeitstempel). Daher ist es nicht möglich, die Ereignisverfolgung anzuzeigen, wenn man eine NPRC-Ressource in NaturalONE öffnet.
3. Die Datenkonsolidierung ist eine Voraussetzung, wenn man die Ereignisdaten über die Profiler Rich GUI-Schnittstelle auswerten möchte.

Natural-Codeabdeckung (Code Coverage)

Die Natural-Codeabdeckung wird verwendet, um ausgeführte und nicht ausgeführte Statements einer Natural-Anwendung zu überwachen. Sie wird durch Einschalten des Schlüsselworts `COVERAGE` der `INIT`-Funktion der Profiler Utility gestartet.

Für die Codeabdeckung verwendet der Profiler automatisch einen Ereignisfilter, so dass nur die Ereignisse „Programmstart“ (`PS`) und „Natural Statement“ (`NS`) gesammelt werden. Darüber hinaus enthalten die Statement-Ereignisse in diesem Fall einen GP-Offset, der benötigt wird, um das Statement eindeutig zu identifizieren.

Für jedes aufgerufene Programm wird das entsprechende katalogisierte Programm (GP = Generated Program) gelesen und die Statement-Tabelle gebaut, die die Zeilennummer, den GP-Offset, den Objektcode und die Copycode-Informationen des Statement enthält. Darüber hinaus wird eine invertierte Liste von GP-Offsets für die schnelle Suche nach dem Offset erstellt.

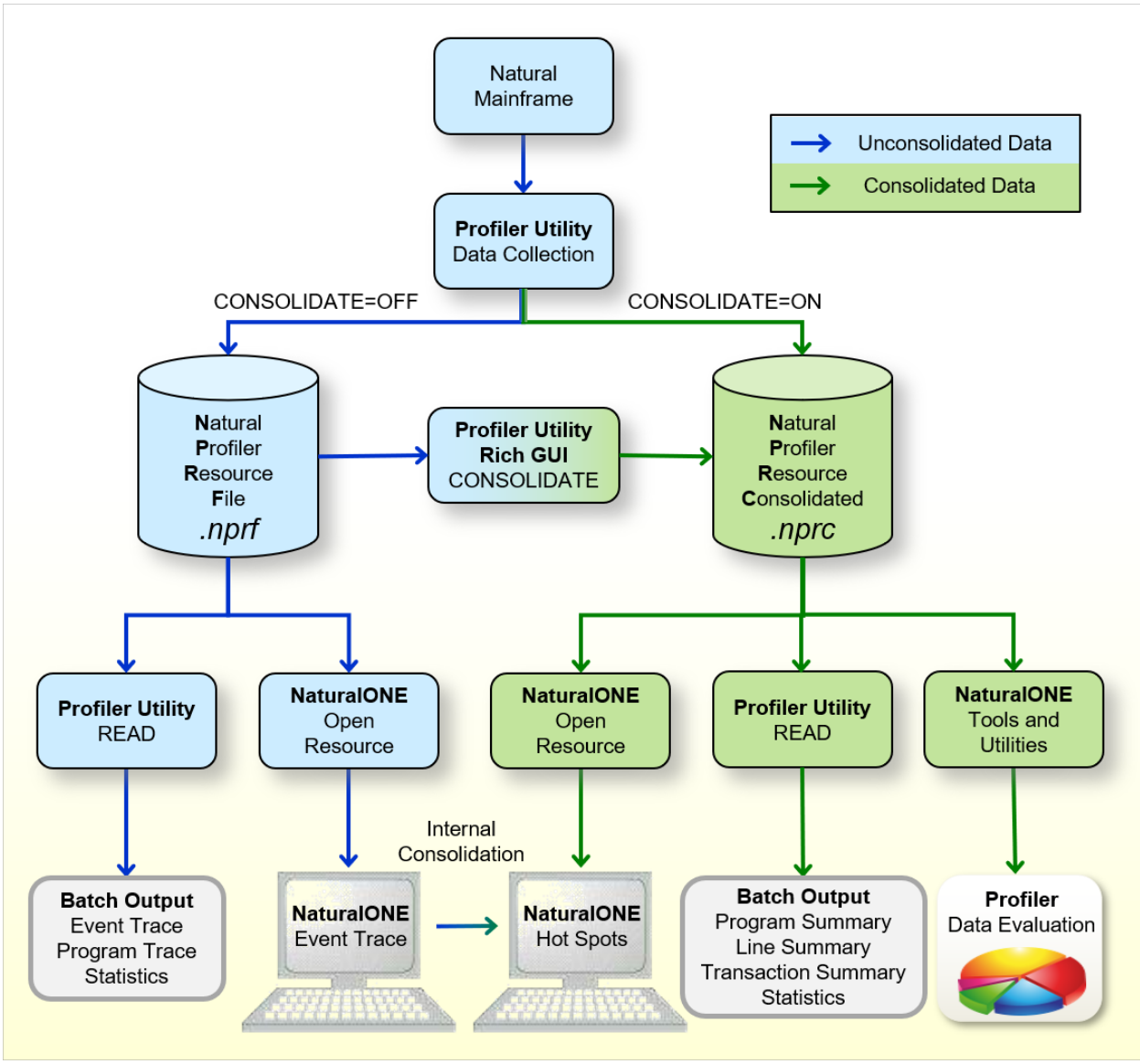
Während die Anwendung läuft, wird der GP-Offset zu jedem ausgeführten Statement in der Statement-Tabelle gesucht und als abgedeckt markiert. Schließlich werden die abgedeckten Statements und die nicht abgedeckten Statements (die nicht markiert worden sind) in eine Natural-NCVF-Ressourcendatei geschrieben.

Wenn die NCVF-Abdeckungsressourcendatei mit der Profiler `READ`-Funktion analysiert wird, wird der Quellcode der überwachten Programme gelesen und die Zeilen werden entsprechend der Abdeckung der Statements in der Zeile markiert.

Anmerkung: Die Natural-Codeabdeckung ist nicht für Programme vorgesehen, die mit dem Natural Optimizer Compiler (NOC) katalogisiert wurden.

Datenverarbeitung

Die folgende Grafik zeigt, wie die Profiler Utility nicht konsolidierte und konsolidierte Daten verarbeitet:



Die Grafik wird im folgenden Abschnitt erläutert:

- Wenn eine Großrechner-Natural-Batch-Anwendung mit der Datensammlung der Profiler Utility profiliert wird, werden die resultierenden Ereignisdaten, abhängig von der Einstellung des `CONSOLIDATE`-Schlüsselworts der `INIT`-Funktion der Profiler Utility, in eine Natural Profiler-Ressourcendatei (NPRF) oder eine konsolidierte Natural Profiler-Ressourcendatei (NPRC) geschrieben.
- Die Natural Profiler-Ressourcendatei (Erweiterung `.nprf`) enthält die Ereignisdaten in einem nicht konsolidierten Format, was bedeutet, dass es für jedes Ereignis einen Datensatz gibt.
- Die `READ`-Funktion der Profiler Utility liest die Ereignisdaten aus der NPRF-Ressourcendatei. Sie bietet ein Ereignis-Trace, ein Programm-Trace und die Profiler-Statistiken. Die resultierenden Daten können in eine Datei im Text- oder CSV-Format (kommagetrennte Werte) exportiert werden.

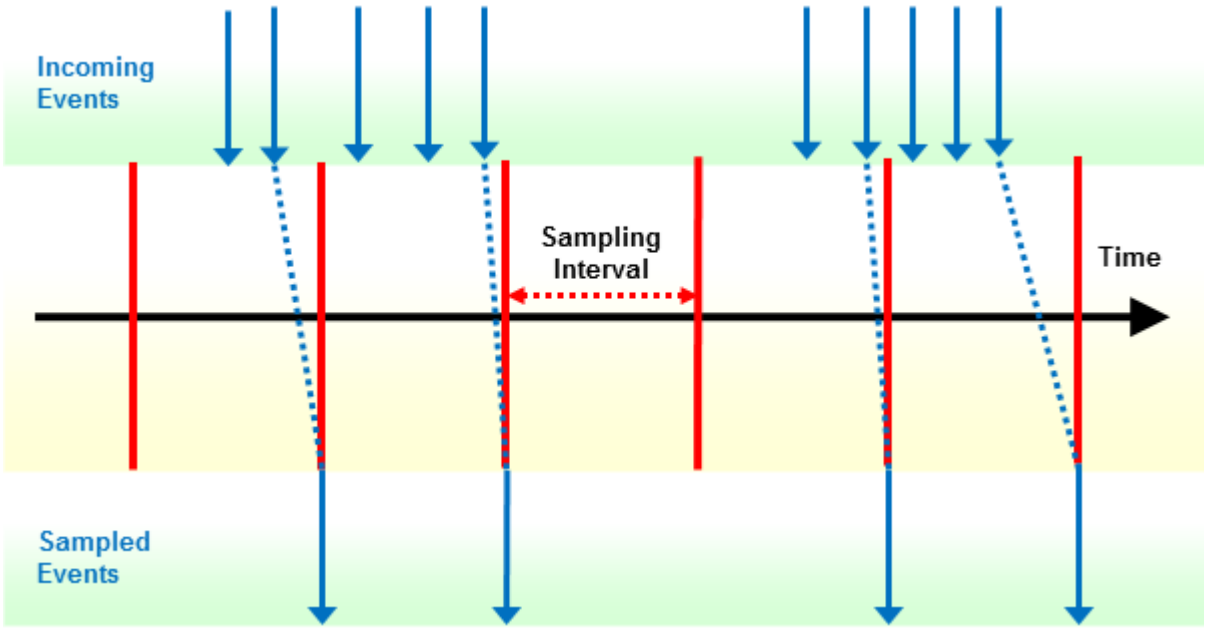
- Wenn die NPRF-Ressourcendatei von NaturalONE aus geöffnet wird, werden die nicht konsolidierten Ereignisdaten auf der NaturalONE **Event Trace**-Seite aufgelistet.
- Die NaturalONE **Hot Spots**-Seite zeigt die Ereignisdaten in konsolidierter Form an. Wenn die Daten aus einer NPRF-Ressourcendatei stammen, konsolidiert NaturalONE die Daten intern.
- Die `CONSOLIDATE`-Funktion der Profiler Utility und die Profiler Rich GUI-Schnittstelle liest die Ereignisdaten aus der NPRF-Ressourcendatei, konsolidiert sie und schreibt sie in eine NPRC-Ressourcendatei.
- Die konsolidierte Natural Profiler-Ressourcendatei (Erweiterung `.nprc`) enthält die Ereignisdaten in einem konsolidierten Format, was bedeutet, dass ähnliche Datensätze in einem konsolidierten Datensatz aggregiert werden. Im Allgemeinen ist eine NPRC-Ressourcendatei viel kleiner als die entsprechende NPRF-Ressourcendatei und daher viel schneller zu verarbeiten.
- Wenn die NPRC-Ressourcendatei von NaturalONE aus geöffnet wird, werden die konsolidierten Ereignisdaten auf der **Hot Spots**-Seite angezeigt. Es ist nicht möglich, das Ereignis-Trace anzuzeigen, da die NPRC-Ressourcendatei nicht die Daten jedes einzelnen Ereignisses enthält.
- Die READ-Funktion der Profiler-Utility liest die Ereignisdaten aus der NPRC-Ressourcendatei. Sie liefert ein Trace der konsolidierten Datensätze, eine Programmzusammenfassung, eine Zeilenzusammenfassung, eine Transaktionszusammenfassung und Profiler-Statistikdaten. Die resultierenden Daten können in eine Datei im Text- oder CSV-Format (kommagetrennte Werte) exportiert werden.
- Die Profiler Rich GUI-Schnittstelle (gestartet mit NaturalONE **Tools and Utilities**) visualisiert die Profiler-Ereignisdaten und Statistiken in einer grafischen, interaktiven Browser-Oberfläche.

Stichprobentechnik (Natural Profiler Sampling)

Allgemein werden Profiler in *ereignisbasierte* oder *statistische* Profiler eingeteilt. Statistische Profiler, die mittels Stichproben (Sampling) arbeiten, unterbrechen das Betriebssystem in regelmäßigen Abständen, um die Profiling-Daten zu erhalten. Die resultierenden Daten sind nicht genau, sondern eine statistische Annäherung.

Der Natural Profiler ist ein *ereignisbasierter* Profiler. Er erhält die Kontrolle und sammelt die Profiling-Daten, wenn ein Natural-Ereignis eintritt. Obwohl der Natural Profiler das Betriebssystem nicht unterbricht, bietet er eine Stichprobentechnik, die die gleichen Profilierungsdaten wie statistische Profiler generiert.

Das Natural Profiler Sampling funktioniert wie ein Filter: Es eliminiert alle Ereignisse außer dem letzten in einem Stichprobenintervall. Darüber hinaus ersetzt es den CPU-Zeitstempel des Ereignisses durch die nachfolgende Abtastzeit. Auf diese Weise sammelt der Natural Profiler nur die Ereignisse, die zu Beginn eines Stichprobenintervalls aktiv waren.



Bei Verwendung des Natural Profiler Sampling ist Folgendes zu beachten:

- Die Stichprobentechnik des Natural Profiler Sampling bietet eine gute Schätzung der verbrauchten CPU-Zeit. Es liefert keine anderen Schätzungen wie Trefferzahlen, verstrichene (elapsed) Zeiten und Adabas-Zeiten.
- Das Natural Profiler Sampling ist ein statistischer Ansatz, der die Anzahl der Ereignisse mit fast den gleichen CPU-Zeitergebnissen stark reduziert.
- Je kleiner das Stichprobenintervall, desto genauer das Ergebnis.
- Je größer das Stichprobenintervall, desto weniger Daten werden produziert.
- Die resultierende Ereignisdauer ist ein Vielfaches des Stichprobenintervalls.
- Die Stichprobentechnik generiert höchstens einen Datensatz pro Stichprobenintervall.
- Ereignisse, die mehr Zeit als ein Stichprobenintervall verbraucht haben, benötigen nur einen Datensatz.
- Das Ereignis der Sitzungsbeendigung (`ST`, Session Termination) wird unverändert aufgezeichnet.

Wenn die gesamte CPU-Zeit der Anwendung bekannt ist und Stichprobentechnik verwendet wird, kann die Anzahl der Ereignisse geschätzt werden:

$$\text{Anzahl der Ereignisse} \approx \frac{\text{Gesamte CPU-Zeit in Mikrosekunden}}{\text{Stichprobenintervall}}$$

Beispiel:

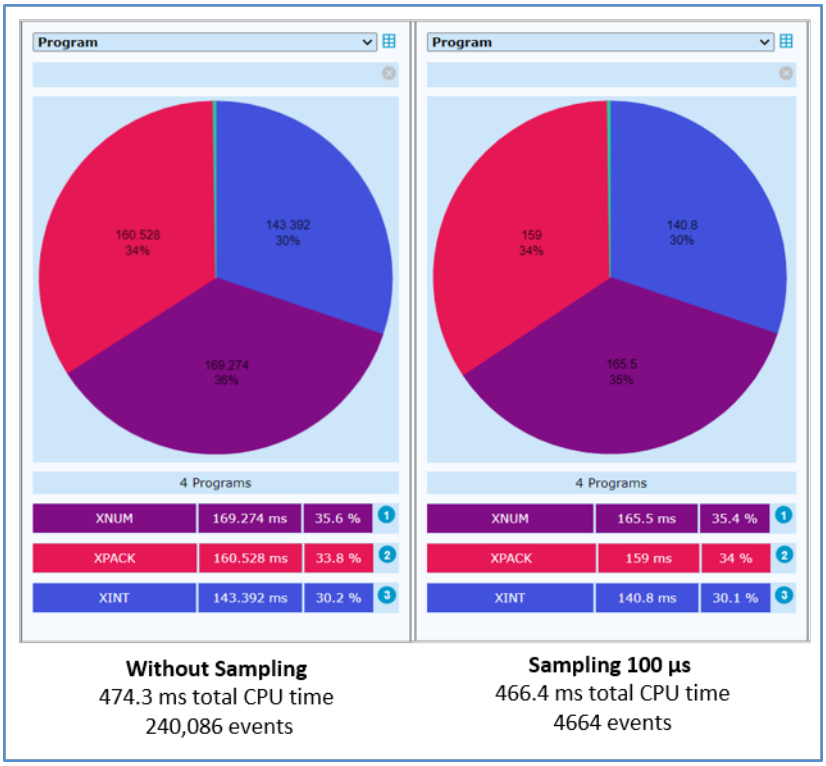
In der folgenden Beispielanwendung ruft das Programm `XPROF` drei Subprogramme auf. Die Anwendung wird zweimal profiliert:

1. Ohne Stichprobennahme.
2. Mit Stichprobennahme, wobei ein Stichprobenintervall von 100 Mikrosekunden verwendet wird.

Für die Stichprobennahme werden die folgenden Schlüsselwörter mit der `INIT`-Funktion der Profiler Utility angegeben:

```
FUNCTION=INIT            /* Initialize Profiling
  SAMPLING=ON            /* Use sampling
  INTERVAL=100           /* Microseconds
```

Die unten aufgeführten, mit der Natural Profiler Rich GUI-Schnittstelle erzeugten Kreisdiagramme zeigen für jedes Programm den Namen des Programms, die aufgewendete CPU-Zeit (Einheit: Millisekunden) und den CPU-Zeitprozentsatz in Bezug auf die gesamte CPU-Zeit. Das linke Diagramm spiegelt den Lauf ohne Stichprobennahme (**Without Sampling**) und das rechte Diagramm den Lauf mit Stichprobennahme (**Sampling**) wider. Obwohl das Sampling die Anzahl der Ereignisse von 240.086 auf 4.664 reduzierte, sind die resultierende CPU-Zeit und -verteilung fast gleich.



Profiler-Performance im Batch-Modus

Das Profilieren einer Anwendung wirkt sich in der Regel auf die Performance (Laufzeitverhalten) der Anwendung aus. Die Auswirkungen können die Messung und die Gesamtdauer des Jobs betreffen. Im Natural Profiler sind mehrere Funktionen implementiert, um die Messung so genau und den Performance-Verlust so gering wie möglich zu halten.

- Die Profiler-Monitor-Session, die die Daten aus dem Profiler-Daten-Pool liest und in die Profiler-Ressourcendatei schreibt, läuft als asynchroner Task. Die für die Monitor-Session aufgewendete Zeit wird daher bei der Anwendungszeitmessung nicht berücksichtigt. Da die Profiler-Trace-Session jedoch am Ende warten muss, bis die Monitor-Session beendet ist, kann sie sich auf die Gesamtdauer des Jobs auswirken.
- Der NATRDC1-Exit, der in der Profiler-Trace-Session läuft, misst die für das Zusammenstellen des Trace-Datensatzes benötigte CPU-Zeit separat und subtrahiert sie von der CPU-Zeit der Session.
- Wenn Natural-Statement-Ereignisse mit der Ereignisfiltereinstellung `STATEMENT=ON` gesammelt werden, deaktiviert der Profiler die Generierung von Statement-Ereignissen im Natural-Nukleus, solange ein Blockfilter (Library, Programm, Zeile, FNAT, Ereignisanzahl oder Zeitfilter) aktiv ist. Dies reduziert die Belastung des Natural-Nukleus, des SYSRDC Data Collector und schließlich des Exit NATRDC1, der andernfalls das Ereignis ablehnen würde.

- Der Profiler komprimiert die Ereignisdaten, bevor er sie in die Ressource schreibt. Durch die Komprimierung kann bis zu 80 Prozent des Bedarfs an Datenspeicherplatz eingespart werden, was die Anzahl der E/A-Vorgänge drastisch reduziert. Die Ereignisdaten werden auch von NaturalONE im komprimierten Format gelesen, was die Transportflussrate erhöht.
- Bei der Ausführung auf einem z/OS-Rechner mit zIIP (IBM System z Integrated Information Processor) geht Zeit verloren, wenn die Ausführung vom Allzweckprozessor (GCP, General Central Processor) zum zIIP wechselt und umgekehrt. Wenn Sie Natural for zIIP mit dem Profiler verwenden, läuft der NATRDC1-Exit auf dem zIIP mit einer minimalen Anzahl von Umschaltungen.

Wahl der Profiler-Daten-Poolgröße

Wenn der Profiler-Daten-Pool voll ist, wartet die Profiler-Trace-Session eine Sekunde, damit die Profiler-Monitor-Session etwas Speicherplatz lesen und freigeben kann. Wenn der Daten-Pool zu klein ist, kann es vorkommen, dass die Profiler-Monitor-Session alle Daten liest, bevor die Trace-Session neu gestartet wird. Wenn die Monitor-Session keine Daten im Daten-Pool findet, wartet sie eine Sekunde auf neue Daten. Jetzt warten beide Sitzungen abwechselnd, was die Gesamtdauer des Jobs stark erhöht.

Die Eigenschaft **Data pool empty after full** (Daten-Pool leer nach voll) in der Kategorie **Trace Session** der Profiler-Statistik zeigt solche abwechselnden Wartezeiten an. Wenn der Wert dieser Eigenschaft größer als Null (`0`) ist, müssen Sie den Natural-Profilparameter `PDPSIZE` auf einen entsprechenden Wert erhöhen.

Beispiel:

```
PDPSIZE=50000
```

Langlaufende Anwendungen profilieren

Das Profilieren einer langlaufenden Batch-Anwendung kann eine riesige Datenmenge erzeugen, insbesondere wenn Natural-Statements überwacht werden.

In diesem Abschnitt wird beschrieben, wie Sie die Anzahl der zu überwachenden Ereignisse minimieren und dabei wesentliche Informationen beibehalten können:

- Profil-Erstellung starten und pausieren
- Filter einstellen
- Stichprobennahme (Sampling) für die CPU-Analyse verwenden

- Serverseitige Datenkonsolidierung verwenden

Profil-Erstellung starten und pausieren

- Wenn der Batch-Job mehrere Natural-Anwendungen startet, initialisieren und starten Sie den Profiler unmittelbar vor der ersten Anwendung von Belang. Sobald der Profiler initialisiert ist, wirkt er sich auf die Performance aus, auch wenn keine Ereignisse gesammelt werden.
- Pausieren Sie den Profiler für Anwendungen, die nicht von Belang sind, und starten Sie ihn für die nächste Anwendung von Belang neu.
- Verwenden Sie eventuell die Anwendungsprogrammierschnittstelle (API) USR8210N, um die Profil-Erstellung an bestimmten Stellen in der Anwendung zu starten und anzuhalten. Siehe auch *Anwendungsprogrammierschnittstelle USR8210N benutzen*.

Beispiel:

Ein Job führt drei Natural-Anwendungen aus. Von diesen drei Anwendungen ist nur die zweite für eine Profiler-Analyse von Belang.

Initialisieren und starten Sie die Profil-Erstellung unmittelbar vor Beginn der Ausführung der zweiten Anwendung und pausieren Sie die Profil-Erstellung direkt nach der Ausführung, wie im folgenden Beispiel:

```
APP-01
PROFILER
FUNCTION=INIT,...          /* Initialize profiling
FUNCTION=START             /* Start data collection
END-PROFILER               /* End Profiler input
APP-02
PROFILER
FUNCTION=PAUSE             /* Pause data collection
END-PROFILER               /* End Profiler input
APP-03
FIN
/*
```

Auf diese Weise hat die Profil-Erstellung keinen Einfluss auf die Performance der anderen Anwendungen.

Filter einstellen

- Verwenden Sie den Ereignisfilter `FNAT=OFF`, um die Überwachung von Natural-Systemprogrammen zu vermeiden, oder geben Sie das Schlüsselwort `FNAT` überhaupt nicht an.
- Statement-Ereignisse haben den größten Einfluss auf die Performance und Quantität. Die anderen Ereignisse haben nur einen geringen Einfluss auf die Performance, vergrößern aber die Menge. Überwachen Sie Statement-Ereignisse nur, wenn Sie sie wirklich benötigen. Überwachen Sie von den Nicht-Statement-Ereignissen nur diejenigen, die Sie analysieren möchten.

  Wenn Sie beispielsweise in NaturalONE die Programm-Hot-Spots, aber weder das Statement noch die Zeilen-Hot-Spots anzeigen möchten, ist die folgende Profiler-Ereignisfiltereinstellung ausreichend:

```
FILTER=EVENT            /* Set event filter
  EVENT=P               /* Program events
```

  Mit dieser Einstellung werden nur die Programm- und Session-Ereignisse überwacht, die für die Programm-Hot-Spots benötigt werden, während Statement- und FNAT-Sammlung standardmäßig deaktiviert sind.
- Überwachen Sie nur die Libraries und Programme, die Sie analysieren möchten. Verwenden Sie den Programmfilter, um die Profil-Erstellung einzuschränken.

  Wenn beispielsweise ein (erster) Profiler ohne Statement-Sammlung gezeigt hat, dass die meiste CPU-Zeit im Programm `HIGHCPU` verbracht wurde, dann möchten Sie vielleicht nur wissen, in welcher Zeile dieses Programms die meiste Zeit verbracht wurde und welche anderen Ereignisse (Datenbankaufrufe, externe Programmaufrufe usw.) ausgeführt werden:

```
FILTER=PROGRAM          /* Set program filter
  LIBRARY=PRFDEMO       /* Monitored library
  PROGRAM=HIGHCPU       /* Monitored program
FILTER=EVENT            /* Set event filter
  EVENT=ALL             /* All events
  STATEMENT=ON          /* Collect statements (no count)
```

Stichprobennahme (Sampling) für die CPU-Analyse verwenden

Für die CPU-Analyse einer langlaufenden Anwendung empfehlen wir die Stichprobennahme (**Stichprobentechnik (Natural Profiler Sampling)**). Wenn Sie bereits Filtereinstellungen verwenden, um die Anzahl der Ereignisse zu reduzieren, können Sie zusätzlich das Sampling aktivieren, um die Anzahl der Ereignisse weiter zu reduzieren.

Die meisten Ereignisdaten werden generiert, wenn Statements gesammelt werden. Daher wird die Stichprobennahme oft in Verbindung mit der Statement-Sammlung verwendet. Für sehr lange laufende Anwendungen kann es jedoch hilfreich sein, Stichproben zu verwenden, auch wenn keine Statements gesammelt werden. Wenn Sie Stichproben ohne Statement-Sammlung verwenden,

empfehlen wir ein Stichprobenintervall, das höher ist als das, das bei der Statement-Sammlung angegeben wurde.

Die Stichprobennahme hat nur eingeschränkte Auswirkungen auf die Profiler-Performance, kann aber die Datenmenge drastisch reduzieren. Die Formel im Abschnitt ***Stichprobentechnik (Natural Profiler Sampling)*** (Probenahme), die hier neu angeordnet ist, kann verwendet werden, um ein Stichprobenintervall so auszuwählen, dass die Anzahl der Ereignisse gleich oder kleiner als ein Näherungswert ist:

$$\text{Stichprobenintervall} \geq \frac{\text{CPU-Gesamtzeit in Mikrosekunden}}{\text{Ungefähre Anzahl der Ereignisse}}$$

Zum Beispiel benötigt eine Batch-Anwendung 40 Minuten CPU-Zeit (2.400.000.000 µs). Die Stichprobennahme sollte die Anzahl der Ereignisse auf höchstens 500.000 Ereignisse beschränken. Das entsprechende Probenahmeintervall kann mit der obigen Formel berechnet werden.

$$\text{Stichprobenintervall} \geq \frac{2.400.000.000}{500.000} = 4.800$$

Geben Sie die folgende Stichprobeneinstellung bei der INIT-Funktion der Profiler Utility an:

```
SAMPLING=ON
INTERVAL=4800
```

Serverseitige Datenkonsolidierung verwenden

Wenn Sie die Performance der Ereignisdaten analysieren möchten und kein Ereignis- oder Programm-Trace benötigen, empfehlen wir Ihnen, die Ereignisdaten serverseitig zu konsolidieren. Die Profiler-Datenkonsolidierung kombiniert ähnliche Datensätze in einem konsolidierten Datensatz mit aggregierten Zeitwerten und einem Trefferzähler.

Die Ereignisdaten können während der Datensammlung mit dem Schlüsselwort `CONSOLIDATE` in der INIT-Funktion der Profiler-Utility konsolidiert werden, wie im Abschnitt ***Profil-Erstellung initialisieren*** beschrieben.

Nicht konsolidierte Ereignisdaten einer NPRF-Ressourcendatei (Natural Profiler Resource File) können mit der CONSOLIDATE-Funktion der Profiler Utility konsolidiert werden, wie im Abschnitt ***Ereignisdaten konsolidieren*** beschrieben, oder mit der CONSOLIDATE-Funktion der Natural Profiler Rich GUI-Schnittstelle, wie in der Dokumentation von NaturalONE beschrieben, die unter **documentation.softwareag.com** verfügbar ist. Siehe *Using NaturalONE > Using Natural Tools and Utilities > Rich GUI Interface of the Natural Profiler*.

Konsolidierte Daten werden in eine NPRC-Ressourcendatei (Natural Profiler Resource Consolidated) geschrieben, die allgemein deutlich kleiner ist als die entsprechende NPRF-Ressourcendatei. Sie lässt sich viel schneller von NaturalONE aus öffnen und bietet die gleichen Hot-Spots wie die NPRF-Ressourcendatei.

Anmerkung: Natural-Codeabdeckungsdaten, die in eine NCVF-Ressourcendatei geschrieben werden, werden automatisch durch Natural-Codeabdeckung konsolidiert.

Verwandte Themen

- Mit dem Profilparameter `RDC` wird der Natural Data Collector konfiguriert, der von der Profiler Utility und der SYSRDC Utility benutzt wird. Siehe *RDC - Konfiguration des Natural Data Collector* in der *Parameter-Referenz*-Dokumentation.

  Die CMRDC-Schnittstelle steuert die im Natural Data Collector-Puffer aufgezeichneten Daten. Siehe *Aufruf des CMRDC-Interface* im Dokument *SYSRDC Utility* in der *Debugger und AE-Dienstprogramme*-Dokumentation.
- Die Verwendung der Profiler Utility kann von Natural Security kontrolliert werden, siehe *Protecting Utilities* in der *Natural Security*-Dokumentation.
- Die Verwendung des NaturalONE Profilers und der NaturalONE-Codeabdeckung ist in der *NaturalONE*-Dokumentation beschrieben.
- Die NaturalONE Tools und Utilities einschließlich der Natural Profiler Rich GUI-Schnittstelle sind in der *NaturalONE*-Dokumentation beschrieben.

52 Profiler Utility im Online-Modus benutzen

- Voraussetzungen 414
- Profiler Utility im Online-Modus aufrufen und beenden 414
- Ereignisse (Events) 415
- Funktionen 416

Die Profiler Utility im Online-Modus dient dazu, einen schnellen Überblick über die letzten Aktionen einer Natural-Online-Anwendung zu erhalten. Die Utility ist menübasiert und bietet Funktionen zur Steuerung des Profiler-Tracing, zur Auswahl der erforderlichen Ereignistypen, zur Pflege und Anzeige von Trace-Datensätzen, zum Herunterladen der Ereignisdaten auf den PC und zum Speichern der Ereignisdaten als Profiler-Ressourcendatei.

Mit der Profiler Utility können Sie den logischen Ablauf von Natural-Anwendungen analysieren und die Nutzung von Ressourcen verfolgen.

Voraussetzungen

Damit Sie die Profiler Utility im Online-Modus verwenden können, muss der Schlüsselwort-Subparameter `SIZE` des Profilparameters `RDC` auf einen Wert größer als 2 gesetzt werden, siehe *RDC - Konfiguration des Natural Data Collector* in der *Parameter-Referenz*-Dokumentation.

Darüber hinaus steuert der `RDC`-Parameter das folgende Standardverhalten der Profiler Utility:

- Durch die Einstellung `RDC=ON` ist das Profiler-Tracing bereits aktiv, wenn die Profiler Utility gestartet wird. Standardmäßig ist `RDC` auf `OFF` gesetzt und das Profiler-Tracing muss separat aktiviert werden, siehe ***Profiler Tracing starten/stoppen***.
- Der Schlüsselwort-Subparameter `EVENT` bestimmt, welche Ereignistypen für das Profiler-Tracing ausgewählt werden. Standardmäßig ist `EVENT` auf `ALL` gesetzt, was alle verfügbaren Ereignistypen abdeckt (siehe *Datenerfassung auslösende Ereignisse* in der *SYSRDC Utility*-Dokumentation).
- Informationen zu Ereignistypen, die von der Profiler Utility abgedeckt werden, finden Sie unter ***Ereignisse***.

Profiler Utility im Online-Modus aufrufen und beenden

Dieser Abschnitt enthält eine Anleitung zum Aufrufen und Beenden der Profiler Utility im Online-Modus.

Um die Profiler Utility aufzurufen:

- Geben Sie das folgende Systemkommando ein:

  ```
  PROFILER
  ```

  Das Hauptmenü der Profiler Utility (Beispiel siehe unten) erscheint. In der Kopfzeile des Menüs wird der aktuelle Status in Bezug auf das Profiler-Tracing angezeigt (`Trace started`/`Trace stopped`). Wenn das Profiler-Tracing gestoppt wird, wird auch die Anzahl der Trace-Datensätze angezeigt, die im Puffer des Natural Data Collector gesammelt wurden.

```
12:30:57              ***** NATURAL PROFILER UTILITY *****           2016-03-07
User SAG                           - Main Menu -                    1035 records
                                                                   Trace stopped

                      Code  Function

                       S    Select Profiler events
                       L    List trace records
                       D    Display trace record
                       T    Start/Stop Profiler tracing
                       P    Print trace records
                       W    Download trace records
                       B    Save data as resource
                       ?    Help
                       .    Exit

                Code .. _    Record from .. 1_____  to .. 1035__



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

Um die Profiler Utility zu beenden:

- Drücken Sie PF3 oder PF12.

  Oder:

  Geben Sie einen Punkt (.) in das Feld **Code** ein.

  Oder:

  Geben Sie `EXIT` in der Kommandozeile **Command** ein.

Ereignisse (Events)

Während einer Natural-Session können verschiedene Arten von Ereignissen auftreten, z.B. ein Programmstart. Daten, die für ein Ereignis spezifisch sind, können in einem Trace-Datensatz gesammelt werden. Jedes Ereignis ist mit einem Ereignistyp verknüpft, d.h. einem ein- oder zweibuchstabigen Code. Folgende Ereignisse und Ereignistypen stehen zur Verfügung:

Ereignis	Ereignistyp	Auftreten des Ereignisses
Program **L**oad	PL	Wenn ein Programm (Natural-Objekt) geladen wird oder wenn es sich bereits im Bufferpool befindet.
Program **S**tart	PS	Wenn ein Programm (Natural-Objekt) gestartet wird.
Program **T**ermination	PT	Wenn ein Programm (Natural-Objekt) beendet wird.
Before **D**atabase Call	DB	Bevor ein Datenbankaufruf ausgeführt wird.
After **D**atabase Call	DA	Nachdem ein Datenbankaufruf ausgeführt worden ist.
Before Terminal **I**/O	IB	Bevor eine Terminal-Ein-/Ausgabe ausgeführt wird.
After Terminal **I**/O	IA	Nachdem eine Terminal-Ein-/Ausgabe ausgeführt worden ist.
Before External Program Call	CB	Bevor ein externer Programmaufruf (`CALL`-Statement) ausgeführt wird.
After External Program Call	CA	Nachdem ein externer Programmaufruf (`CALL`-Statement) ausgeführt worden ist.
Runtime **E**rror	E	Wenn ein Natural-Laufzeitfehler aufgetreten ist.
Natural **S**tatement	NS	Wenn ein Natural-Statement ausgeführt wird. Aus technischen Gründen gibt es keine Eins-zu-Eins-Beziehung zwischen einem Natural-Quellcode-Statement und einem Objektcode im katalogisierten Objekt. Daher können mehrere Natural-Statement zu einem `NS`-Ereignis zusammengeführt werden, und umgekehrt kann ein Natural-Statement mehrere `NS`-Ereignisse abdecken.
User-Defined Event	U	Wenn ein benutzerdefiniertes Ereignis mit dem Natural-Statement `CALL 'CMRDC' 'U'` generiert wird (siehe *Benutzerdefinierte Ereignisse* in der *SYSRDC Utility*-Dokumentation). Das erste Byte der Benutzerdaten wird als Subtyp interpretiert. Daher wird ein zweibuchstabiger Code für ein benutzerdefiniertes Ereignis angezeigt, wenn die Trace-Datensätze aufgelistet werden.

Funktionen

Die Funktionen, die von der Profiler Utility im Online-Modus abgedeckt werden, können in drei Kategorien unterteilt werden:

- Funktion **Select Profiler Events**, um Profiler-Ereignisse auszuwählen.
- Die Anzeigefunktionen **List Trace Records**, **Display Trace Record**, **Print Trace Records**, **Download Trace Records** und **Save Data as Resource**, um Trace-Datensätze aufzulisten, anzuzeigen, zu drucken, herunterzuladen bzw. zu speichern.
- Funktion **Start/Stop Profiler Tracing**, um das Profiler-Tracing zu starten oder zu stoppen.

Alle Funktionen der Profiler Utility können über das **Hauptmenü** aufgerufen werden.

Um eine Funktion aufzurufen:

- Geben Sie einen Funktionscode in das Feld **Code** ein.

  Anmerkung: Die Eingabefelder **Record from** und **Record to** können geändert werden, um einen Bereich von Datensätzen zu definieren, der zuerst von den Funktionen **List Trace Records** und **Display Trace Record** aufgelistet bzw. angezeigt wird, oder um Datensätze für die Funktionen **Print Trace Records** und **Download Trace Records** zu filtern.

Dieser Abschnitt behandelt die folgenden Funktionen:

- Profiler-Ereignisse auswählen (Select Profiler Events)
- Trace-Datensätze auflisten (List Trace Records)
- Trace-Datensatz anzeigen (Display Trace Record)
- Profiler-Tracing starten/stoppen
- Trace-Datensätze drucken
- Trace Records herunterladen (Download Trace Records)
- Daten als Ressource speichern (Save Data as Resource)

Profiler-Ereignisse auswählen (Select Profiler Events)

Diese Funktion wird verwendet, um Ereignistypen für das Profiler-Tracing in der Spalte **Collect** (Sammeln) und als Filter für die Anzeigefunktionen in der Spalte **Display** (Anzeigen) auszuwählen.

Um einen Ereignistyp für das Profiler-Tracing auszuwählen:

- Geben Sie einen beliebigen Schlüssel in das entsprechende Eingabefeld in der Spalte **Collect** ein. Sie müssen mindestens einen Ereignistyp auswählen.

  Infolgedessen wird der Natural Data Collector dynamisch entsprechend Ihrer Auswahl aktualisiert. Wenn das Profiler-Tracing aktiv ist, wird es mit den von Ihnen getroffenen Auswahlen fortgesetzt.

Anmerkungen:

1. Um das Profiler-Tracing für den Ereignistyp **Natural Statement** (**NS**) zu aktivieren, müssen der Profilparameter `TRACE` auf `NATPROX` und der der Profilparameter `ITRACE` auf `ON` gesetzt sein, siehe *Parameter-Referenz*-Dokumentation. Wenn Sie diese Werte nicht angeben, bevor Sie Ihre Natural-Sitzung starten, werden Sie dazu aufgefordert, wenn Sie diesen Ereignistyp auswählen.
2. Zu Beginn der Sitzung werden die für das Profiler-Tracing ausgewählten Ereignistypen durch den Schlüsselwort-Subparameter `EVENT` des Profilparameters `RDC` angegeben. Siehe *EVENT - Natural Data Collector-Ereignisse, die aufgezeichnet werden sollen* in der Parameter Referenz-Dokumentation. Jede Änderung dieser Standardauswahl gilt für die gesamte Natural-Session.

Um einen Ereignistyp als Filter für Anzeigefunktionen auszuwählen:

- Geben Sie einen beliebigen Schlüssel in das entsprechende Eingabefeld in der Spalte **Display** ein. Sie müssen mindestens einen Ereignistyp auswählen.



Anmerkung: Beim Start der Session werden die für Anzeigefunktionen ausgewählten Ereignistypen durch den Schlüsselwort-Subparameter `EVENT` des Profilparameters `RDC` (siehe oben) angegeben. Jede Änderung dieser Standardauswahl ist bis zum nächsten `LOGON` gültig. Danach ist der Satz der ausgewählten Ereignistypen gleich dem Satz der Ereignistypen, die für das Profiler-Tracing ausgewählt wurden (siehe oben).

PF-Tasten

Folgende PF-Tasten stehen zur Verfügung:

PF-Taste	Name	Funktion
PF1	Help (Hilfe)	Zeigt die kontextsensitive Hilfe an. Für jedes Eingabefeld gibt es einen spezifischen Hilfetext. In anderen Kontexten, z. B. in der Kommandozeile, wird ein allgemeiner Hilfetext angezeigt.
PF3	Exit (Verlassen)	Aktuelles Menü verlassen.
PF8	Reset (Zurücksetzen)	Auswahl aller Ereignistypen rückgängig machen.
PF9	All (Alle)	Alle Ereignistypen auswählen.
PF12	Canc (Abbruch)	Aktuelles Menü verlassen.

Beispiel für die Auswahl von Ereignistypen:

Im folgenden Beispiel-Bildschirm werden die folgenden Ereignistypen für das Profiler-Tracing ausgewählt: `PL`, `PS`, `PT`, `DB`, `DA`, `CB`, `CA`, `NS` und `U`. Die Ereignistypen `PL`, `PS`, `PT`, `DA`, `CB`, `CA` und `U` werden als Filter für Anzeigefunktionen ausgewählt.

```
10:13:24              ***** NATURAL PROFILER UTILITY *****           2011-03-29
User SAG                    - Select Profiler Events -               79 records
                                                                   Trace stopped

             Collect Display Type Profiler Event
             ------- ------- ---- ----------------------------
                X       X     PL  Program load
                X       X     PS  Program start
                X       X     PT  Program termination
                X       _     DB  Before database call
                X       X     DA  After database call
                _       _     IB  Before terminal I/O
                _       _     IA  After terminal I/O
                X       X     CB  Before external program call
                X       X     CA  After external program call
                _       _     E   Runtime error
                X       _     NS  Natural statement
                X       X     U   User-defined event



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

Trace-Datensätze auflisten (List Trace Records)

Die Funktion **List Trace Records** wird verwendet, um alle Trace-Datensätze aufzulisten, die im Natural Data Collector-Puffer gesammelt wurden. Zu diesem Zweck wird das Profiler-Tracing gestoppt. Ein Trace-Datensatz kann nur angezeigt werden, wenn der entsprechende Ereignistyp in der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) für die Anzeigefunktionen ausgewählt wurde.

Wenn die Funktion aufgerufen wird, wird der zuerst angezeigte Trace-Datensatz durch das Feld **Record from** angegeben, siehe **Hauptmenü**. Zusätzlich ist dieser Trace-Datensatz auch mit einem `X` und der Trace-Datensatz, der durch das Feld **Record to** angegeben wird, mit einem `Y` markiert.

Dieser Abschnitt behandelt die folgenden Themen:

- Struktur der angezeigten Trace-Datensätze
- Navigation
- Auswahl der anzuzeigende Ereignistypen aktualisieren
- Zeilenkommandos
- Lokale Kommandos
- PF-Tasten

- Beispiel für Trace-Datensätze auflisten

Struktur der angezeigten Trace-Datensätze

Jeder Trace-Datensatz wird in einer separaten Zeile angezeigt, die von einem Eingabefeld für Zeilenbefehle (`C`) eingeleitet wird. Der Datensatz selbst enthält allgemeine und ereignisspezifische Daten:

- Allgemeine Daten umfassen Merkmale, die allen Datensätzen gemeinsam sind.

  Zum Beispiel Datensatznummer (**Record**), Ereignistyp (**Ev**) oder Ereigniszeit (**Time**).
- Ereignisspezifische Daten umfassen Merkmale, die für eine Ansicht spezifisch sind, d.h. eine Gruppe verwandter Ereignisse.

  Beispielsweise bilden die Ereignistypen `PL`, `PS` und `PT` die Ansicht `Program Load/Start/Termination`. Die Ansicht `General Information` umfasst alle Ereignistypen und deckt die allgemeinen Daten ab.

Die folgende Tabelle gibt einen Überblick über die verfügbaren Ansichten und die damit verbundenen Ereignistypen:

Ansicht	Code	Ereignistypen
General Information	G	Alle
Program Load/Start/Termination	P	PL, PS, PT
Before/After **D**atabase Call	D	DB, DA
Before/After Terminal **I**/O	I	IB, IA
Before/After External Program **C**all	C	CB, CA
Runtime **E**rror	E	E
Natural **S**tatement	N	NS
User-Defined Event	U	U

Anmerkungen:

1. **General Information** (G) ist standardmäßig die aktuelle Ansicht.
2. Wenn es eine Ansicht gibt, für die kein anzuzeigender Ereignistyp ausgewählt wurde, wird der Buchstabencode für diese Ansicht durch ein Minuszeichen (**-**) ersetzt.

Navigation

Die Liste der Trace-Datensätze, die von der Funktion **List Trace Records** (Datensätze auflisten) angezeigt werden, kann hinsichtlich Größe der angezeigten Datensätze und Anzahl der aufgelisteten Datensätze recht umfangreich sein.

Dieser Abschnitt behandelt die folgenden Optionen:

- Navigation innerhalb eines Datensatzes
- Innerhalb einer Liste von Datensätzen navigieren

Navigation innerhalb eines Datensatzes

Um innerhalb eines Datensatzes zu navigieren, müssen Sie die aktuelle Ansicht ändern.

Um die aktuelle Ansicht zu ändern:

- Markieren Sie mit dem Cursor in **View** einen Buchstabencode und drücken Sie Enter. In der obigen Tabelle finden Sie verfügbare Buchstabencodes.

  Oder:

  Drücken Sie PF10 (**<**), um die benachbarte Ansicht auf der linken Seite auszuwählen. Drücken Sie PF11 (**>**), um die benachbarte Ansicht auf der rechten Seite auszuwählen.

Innerhalb einer Liste von Datensätzen navigieren

Um innerhalb einer Liste von Datensätzen zu navigieren:

- Geben Sie eine Datensatznummer in das Eingabefeld **Record** ein, um auf dem entsprechenden Trace-Datensatz zu positionieren. Beachten Sie, dass der ausgewählte Trace-Datensatz einen Ereignistyp haben muss, der in der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) für die Anzeigefunktionen ausgewählt wurde.

  Oder:

  Geben Sie einen Ereignistyp in das Eingabefeld **Ev** ein, um zum nächsten Trace-Datensatz zu gelangen, der mit einem solchen Ereignistyp markiert ist. Drücken Sie von dieser Position aus weiterhin Enter, um zum nächsten übereinstimmenden Vorkommen zu gelangen. Verwenden Sie Stern-Notation (*), um eine Gruppe von Ereignistypen zu wählen, z.B. `D*` für alle Datenbankaufrufe.

  Oder:

  Drücken Sie PF6 (**--**), PF7 (**-**), PF8 (**+**) oder PF9 (**++**), um in der Liste zu blättern.

Auswahl der anzuzeigende Ereignistypen aktualisieren

Eine Auswahl von Ereignistypen für Anzeigefunktionen kann durch Drücken von PF4 (DiEv) aktualisiert werden. Dies öffnet ein Menü ähnlich dem Menü, das bei der Auswahl der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) geöffnet wird, mit der Ausnahme, dass Ereignistypen nur für Anzeigefunktionen ausgewählt werden können. Siehe ***PF-Tasten*** in *Profiler-Ereignisse auswählen*.

Zeilenkommandos

Sie können ein Zeilenkommando in der Spalte **C** neben dem von Ihnen ausgewählten Trace-Datensatz eingeben. Folgende Zeilenkommandos stehen zur Verfügung:

Lokale Kommandos	Funktion
D	Alle für den ausgewählten Datensatz verfügbaren Informationen anzeigen. Siehe ***Trace-Datensätze anzeigen***.
X	Wenn Sie die Zeile mit einem **X** markieren, wird der Inhalt des Feldes **Record from** im **Hauptmenü** entsprechend aktualisiert.
Y	Wenn Sie die Zeile mit einem **Y** markieren, wird der Inhalt des Feldes **Record to** im **Hauptmenü** entsprechend aktualisiert.

Lokale Kommandos

Sie können ein lokales Kommando in der Kommandozeile **Command** eingeben. Die folgenden lokalen Kommandos stehen zur Verfügung:

Lokales Kommando	Funktion
X	Blättern bis zur Zeile, die mit einem **X** markiert ist (und im Feld **Record from** angezeigt wird).
Y	Blättern bis zur Zeile, die mit einem **Y** markiert ist (und im Feld **Record from** angezeigt wird).

PF-Tasten

Folgende PF-Tasten stehen zur Verfügung:

PF-Taste	Name	Funktion
PF1	Help (Hilfe)	Kontextsensitive Hilfe anzeigen. Siehe **oben**.
PF2	Disp (Anzeigen)	Alle Informationen anzeigen, die für den ausgewählten Trace-Datensatz verfügbar sind. Siehe auch ***Trace-Datensätze anzeigen***.
PF3	Exit (Verlassen)	Aktuelles Menü verlassen.
PF4	DiEv (Ändern)	Auswahl der anzuzeigenden Ereignistypen ändern. Siehe Abschnitt ***Auswahl der anzuzeigenden Ereignistypen aktualisieren***.

PF-Taste	Name	Funktion
PF5	Hex (Hexadezimal)	Benutzerdaten im Hexadezimalmodus anzeigen (nur für Ereignistyp **U**).
	Struc (Struktur)	Programmstruktur anzeigen (nur für die Ereignistypen **PL**, **PS** und **PT**).
PF6	--	Zum Anfang der Liste blättern.
PF7	-	Eine Seite nach oben blättern.
PF8	+	Eine Seite nach unten blättern.
PF9	++	Zum Ende der Liste blättern.
PF10	<	Ansicht auf der linken Seite der aktuellen Ansicht auswählen.
PF11	>	Ansicht auf der rechten Seite der aktuellen Ansicht auswählen.
PF12	Canc (Abbruch)	Aktuelles Menü verlassen.

Beispiel für Trace-Datensätze auflisten

Das folgende Menü listet Trace-Datensätze für die Ereignistypen `PL`, `PS`, `PT` und `DA` auf, die im Beispiel für *Beispiel für Ereignistypauswahl* ausgewählt wurden, und zeigt Trace-Informationen für die Ansicht `P` an, die die ereignisspezifischen Daten der Ereignistypen `PL`, `PS` und `PT` abdecken.

```
10:13:39              ***** NATURAL PROFILER UTILITY *****           2011-03-29
User SAG                    - List Trace Records -                  79 records
                      View:  G P D - C - - U
                             PL/PS/PT - Program Load/Start/Termination

  C Record Ev Library  Program  Type DBID  FNR
  - ______ __ -------- -------- ---- ----- -----
X _ 000001 PL SYSEXPG  DYNAMX06
  _ 000002 PS SYSEXPG  DYNAMX06  P      10  2430
  _ 000010 PL SYSEXPG  DYNAMX05
  _ 000011 PS SYSEXPG  DYNAMX05  N      10  2430
  _ 000018 PT SYSEXPG  DYNAMX05  N      10  2430
  _ 000025 PT SYSEXPG  DYNAMX06  P      10  2430
  _ 000026 PL SYSEXPG  EDITMX02
  _ 000027 PS SYSEXPG  EDITMX02  P      10  2430
  _ 000031 DA
  _ 000040 DA
  _ 000047 DA
  _ 000054 DA
  _ 000057 PT SYSEXPG  EDITMX02  P      10  2430
Use PF10/PF11 to show event-specific data of the trace records.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Disp  Exit  DiEv  Struc --    -     +     ++    <     >     Canc
```

Trace-Datensatz anzeigen (Display Trace Record)

Die Funktion **Display Trace Record** wird verwendet, um alle Tracing-Informationen für jeden Trace-Datensatz separat anzuzeigen. Zu diesem Zweck wird das Profiler-Tracing gestoppt. Wenn die Funktion aufgerufen wird, wird der zuerst angezeigte Trace-Datensatz durch das Feld **Record from** angegeben, siehe **Hauptmenü**.

Dieser Abschnitt behandelt die folgenden Themen:

- Navigation
- PF-Tasten
- Beispiel:

Navigation

Um zu einem anderen Trace-Datensatz zu navigieren:

- Geben Sie eine Datensatznummer in das Eingabefeld **Record** ein, um auf dem entsprechenden Trace-Datensatz zu positionieren. Beachten Sie, dass der ausgewählte Trace-Datensatz einen Ereignistyp haben muss, der von der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) für die Anzeigefunktionen ausgewählt wurde.

  Oder:

  Drücken Sie PF6 (**--**), PF7 (**-**), PF8 (**+**) oder PF9 (**++**), um durch die angezeigten Trace-Datensätze zu blättern.

PF-Tasten

Folgende PF-Tasten stehen zur Verfügung:

PF-Taste	Name	Funktion
PF1	Help	Kontextsensitive Hilfe anzeigen. Siehe **oben**.
PF3	Exit	Aktuelles Menü verlassen.
PF5	Hex	Benutzerdaten im Hexadezimalmodus anzeigen (nur für Ereignistyp **U**).
PF6	--	Ersten Trace-Datensatz anzeigen.
PF7	-	Vorherigen Trace-Datensatz anzeigen.
PF8	+	Nachfolgenden Trace-Datensatz anzeigen.
PF9	++	Letzten Trace-Datensatz anzeigen.
PF10	LongE	Langtext einer Fehlermeldung anzeigen (nur für den Ereignistyp **E** definiert).
PF12	Canc	Aktuelles Menü verlassen.

Beispiel:

Im folgenden Beispiel wird der Trace-Datensatz `10`, der bereits im *Beispiel für Trace-Datensätze auflisten* aufgeführt wurde, mit allen Trace-Informationen angezeigt.

```
10:14:01              ***** NATURAL PROFILER UTILITY *****           2011-03-29
User SAG                    - Display Trace Record -                79 records


Record / Event .... 10____  /  PL - Program load

Event time ........ 10:12:14.130696    Elapsed time ....... 0.000157
CPU time .......... 0
Current user ID ... SAG                Current group ID ...
Current program ... DYNAMX06           Library ............ SYSEXPG
Program level .....                    Program line ....... 0160

Loaded program .... DYNAMX05           Invocation type .... CALLNAT
Library ........... SYSEXPG









Enter required record number, or use PF6 - PF9 for paging.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit              --    -     +     ++                Canc
```

Profiler-Tracing starten/stoppen

Diese Funktion wird verwendet, um die Profil-Erstellung ein- oder auszuschalten. Wenn das Profiler-Tracing aktiv war, wird es gestoppt. Umgekehrt, wenn es inaktiv war, wird es gestartet. Der neue Status wird in der Kopfzeile des Hauptmenüs angezeigt.

Um das Profiler-Tracing zu starten oder zu stoppen:

- Geben Sie `T` in das Feld **Code** ein und drücken Sie Enter.

Anmerkung: Alternativ können Sie auch die im Abschnitt *Trace Recording-Funktion* der SYSRDC Utility-Dokumentation beschriebenen Kommandos verwenden.

Trace-Datensätze drucken

Die Funktion **Print Trace Records** wird verwendet, um die Trace-Datensätze innerhalb des Bereichs zu drucken, der durch die Eingabefelder **Record from** und **Record to** definiert ist. Zu diesem Zweck wird das Profiler-Tracing gestoppt. Beachten Sie, dass ein Trace-Datensatz nur gedruckt werden kann, wenn der entsprechende Ereignistyp für die Anzeigefunktionen in der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) ausgewählt wurde.

Um Trace-Datensätze zu drucken:

1 Geben Sie `P` in das Feld **Code** ein.

Es wird ein separates Fenster geöffnet, in dem Sie den (Standard-) Druckernamen oder den Bereich der zu druckenden Trace-Datensätze ändern können. Sie können auch angeben, ob die Trace-Datensätze mit Header-Informationen gedruckt werden. Der verwendete Drucker wird durch die Arbeitsdatei 1 (`print file 1`) identifiziert.

2 Drücken Sie Enter.

Trace Records herunterladen (Download Trace Records)

Die Funktion **Download Trace Records** dient dazu, die Trace-Datensätze innerhalb des Bereichs herunterzuladen, der durch die Eingabefelder **Record from** und **Record to** definiert ist. Dazu wird das Profiler-Tracing gestoppt. Beachten Sie, dass ein Trace-Datensatz nur heruntergeladen werden kann, wenn der entsprechende Ereignistyp in der Funktion **Select Profiler Events** (Profiler-Ereignisse auswählen) für die Anzeigefunktionen ausgewählt wurde.

Um Trace-Datensätze herunterzuladen:

1 Geben Sie `W` in das Feld **Code** ein.

Es wird ein separates Fenster geöffnet, in dem Sie den Bereich der herunterzuladenden Trace-Datensätze ändern können. Die verwendete Download-Datei wird durch die Arbeitsdatei 7 (`work file 7`) identifiziert. Zum Herunterladen auf den PC empfehlen wir Ihnen, Natural Connection zu verwenden.

2 Drücken Sie Enter.

Anmerkungen:

1. Geben Sie `.htm` als Erweiterung der heruntergeladenen Datei an.
2. Wir empfehlen Ihnen, MS Excel zu verwenden, um die heruntergeladenen Trace-Datensätze anzuzeigen. Dies erfordert, dass MS Excel und Natural kompatible Einstellungen haben, z. B. in Bezug auf das verwendete Dezimalzeichen.

Daten als Ressource speichern (Save Data as Resource)

Die Funktion **Save Data as Resource** dient dazu, alle Trace-Datensätze in einer Profiler-Ressourcendatei (Erweiterung .nprf) zu speichern. Dazu wird die Profiler-Nachverfolgung gestoppt.

Die Profiler-Ressourcendatei kann aus NaturalONE von der Profiler Rich GUI-Schnittstelle und von der Profiler Utility im Batch-Modus gelesen werden. Die Tools bieten eine Performance-Analyse (Hot Spots) und andere Auswertungen. Weitere Informationen finden Sie in der *NaturalONE*-Dokumentation und unter ***Profiler Utility im Batch-Modus benutzen***.



Anmerkungen:

1. Wenn die Profiler Utility unter CICS oder Com-plete ausgeführt wird, wird nur die verstrichene (elapsed) Zeit angegeben, aber nicht die CPU-Zeit.
2. Die von der Profiler Utility im Online-Modus gesammelten Ereignisdaten enthalten keine Copycode-Informationen.

Um die Daten als Ressource zu speichern:

1 Geben Sie B in das Feld **Code** ein.

Es wird ein separates Fenster geöffnet, in dem Sie den (Standard-)Ressourcennamen und die Library ändern können. Sie können außerdem angeben, ob die Ressource ersetzt werden soll, falls sie bereits vorhanden ist.

Der Standard-Ressourcenname ist `*INIT-USER_yyyymmdd_hhiiss`, wobei `*INIT-USER` die Benutzerkennung ist, unter der die Natural-Session ausgeführt wird, und *`yyyymmdd_hhiiss`* das aktuelle Datum und die aktuelle Uhrzeit ist. Wenn die Erweiterung `.nprf` (Natural Profiler-Ressourcendatei) nicht mit dem Ressourcennamen angegeben wurde, wird sie automatisch hinzugefügt.

Die Standard-Library ist die aktuelle Library.

2 Drücken Sie Enter.

53 Profiler Utility im Batch-Modus benutzen

- Voraussetzungen für Profiler Utility im Batch-Modus 430
- Profiler Utility aufrufen und beenden 432
- Syntax und Schlüsselwörter 433
- Gesammelte Ereignisse und Daten 437
- Profilierung initialisieren 442
- Initialisierung der Codeabdeckung (Code Coverage) 447
- Datensammlung starten und anhalten 450
- Filter zur Begrenzung der gesammelten Daten benutzen 453
- Stichprobennahme (Sampling) aktivieren 459
- Benutzerdefinierte Ereignisse schreiben 460
- Monitor-Session CMPRMIN 461
- Monitor-Session CMSYNIN 463
- Batch-Natural RPC-Server profilieren 464
- Profil-Erstellung einer Großrechner-Session aus Natural Studio 466
- Ereignisdaten konsolidieren 467
- Ereignisdaten auswerten 471
- Profiler-Ressourcendateien verwalten 492
- Profiler-Eingaben aus Natural-Textobjekten einfügen 496
- Ereignisverfolgung (Event-Trace) 497
- Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung 499
- Interner Trace 502
- Profiler-Statistikdaten 505

Der Natural Profiler wird verwendet, um den internen Prozessfluss einer Natural Batch-Anwendung zu überwachen und die Performance und die Codeabdeckung der Anwendung zu analysieren.

Die Profiler Utility wird durch JCL-Eingabe gesteuert und bietet Funktionen zur Datensammlung und Datenverarbeitung:

1. Die **Datensammlungsfunktionen** steuern das Profiler-Tracing, wählen erforderliche Ereignistypen aus, filtern, konsolidieren (aggregieren) oder nehmen Datenproben (Sampling) und schreiben die resultierenden Ereignisse in die Profiler-Ressourcendatei.
2. Die **Datenverarbeitungsfunktionen** lesen und verarbeiten die Ereignisdaten aus der Profiler-Ressourcendatei. Nicht konsolidierte Ereignisdaten können konsolidiert werden.

   Sie können Statistiken, eine Programmzusammenfassung, eine Zeilenzusammenfassung und eine Transaktionszusammenfassung, ein Programm-Trace (Ablaufverfolgung), ein Ereignis-Trace mit den wichtigsten Daten sowie Reports über die Programm- und Statement-Abdeckung ausgeben. Sie können die resultierenden Daten im Text- oder CSV-Format (kommatrennte Werte) exportieren.

Die folgenden Tools bieten außerdem Funktionen zum Lesen von Profiler-Ressourcendateien und zum Analysieren der Daten:

- NaturalONE zeigt das vollständige Ereignis-Tracing an und bietet eine Performance-Analyse (Hot-Spots) der Natural Batch-Anwendung. Abdeckungsdaten können in der NaturalONE **Coverage**-Ansicht und im NaturalONE **Source Editor** überprüft werden.
- Die Profiler Rich GUI-Schnittstelle, das von den NaturalONE **Tools and Utilities** gestartet wird, bietet Funktionen zum Auswerten oder Konsolidieren von Profiler-Daten oder zum Löschen einer Ressourcendatei. Interaktive Kreisdiagramme zeigen die Verteilung der Profiler-Leistungskennzahlen (KPIs, Key Performance Indicators) für ausgewählte Kriterien, z.B. die Verteilung der CPU-Zeit für Programme. Bei überwachten Programmen wird der Quellcode mit den Profiler-Daten kombiniert, wobei der Quellcode entsprechend dem Wert des ausgewählten KPI eingefärbt wird.

Voraussetzungen für Profiler Utility im Batch-Modus

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie die Profiler Utility im Batch-Modus verwenden können:

- Natural-Parametereinstellungen
- Natural-Nukleus-Verlinkung

- Anzeige von Ressourcendateien in Natural Development Server-Umgebungen

Natural-Parametereinstellungen

Für die Natural Profiler-Datenerfassungsfunktionen (`INIT`, `COVERAGE`, `START`, `PAUSE` und `TEXT`) müssen Sie die folgenden Natural-Profilparameter definieren:

- `RDCSIZE` - Um den Natural Data Collector (ohne Aufzeichnungsdaten im Puffer des Natural Data Collector) zu aktivieren, geben Sie an:

```
RDCSIZE=2
```

- `RDCEXIT` - Um NATRDC1 als User Exit für den Natural Data Collector zu definieren, geben Sie an:

```
RDCEXIT=NATRDC1
```

- `PDPSIZE` - Optionaler Parameter, den Sie zusätzlich angeben können, um die Profiler-Daten-Pool-Größe zu bestimmen, zum Beispiel:

```
PDPSIZE=1000
```

Die Datenverarbeitungsfunktionen der Profiler Utility (`CONSOLIDATE`, `READ`, `LIST` und `DELETE`) können nicht ausgeführt werden, wenn die Profil-Erstellung aktiv ist. Aus Performance-Gründen empfehlen wir, auch den Natural Data Collector für diese Funktionen mit der folgenden (Standard-)Parametereinstellung zu deaktivieren:

```
RDCSIZE=0
```

Einzelheiten zu den oben genannten Natural-Profilparametern finden Sie in den entsprechenden Abschnitten in der *Parameter-Referenz*-Dokumentation.

Programme, die mit dem Natural Optimizer kompiliert werden, erfordern die folgende Natural Optimizer-Optionseinstellung:

```
NODBG=OFF
```

(Siehe auch *Liste der Optionen* in der *Natural Optimizer Compiler*-Dokumentation.)

Natural-Nukleus-Verlinkung

Der Natural-Nukleus muss „reentrant“ und „reusable“ sein. Verwenden Sie die folgenden Linkage-Optionen:

```
RENT,REUS
```

Anzeige von Ressourcendateien in Natural Development Server-Umgebungen

Standardmäßig werden Natural-Großrechner-Ressourcendateien in Natural Development Server (NDV) -Umgebungen wie NaturalONE oder Natural Studio nicht angezeigt. Für die Analyse der Profiler-Ereignisdaten muss von NaturalONE aus auf die Ressourcendatei zugegriffen werden. Dazu müssen Sie das NDV-Verhalten ändern.

Um Ressourcendateien in einer NDV-Umgebung anzuzeigen:

1. Kopieren Sie den Quellcode des NDV-User Exit `NDV-SX03` aus der Natural-System-Library `SYSLIB` in eine Benutzer-Library.
2. Bearbeiten Sie das Member. Passen Sie den Code folgendermaßen an:

   ```
   DISPLAY-RESOURCES := 'Y'   /* Display resources in NaturalONE/Studio
   ```

3. Katalogisieren Sie das Member unter dem Namen `NDV-UX03`.
4. Kopieren Sie es zurück in die System-Library `SYSLIB` oder in die Library `SYSLIBS` oder `SYSTEM`.

Profiler Utility aufrufen und beenden

Dieser Abschnitt beschreibt, wie Sie die Profiler Utility im Batch-Modus aufrufen und beenden.

Um die Profiler Utility aufzurufen:

- Geben Sie das folgende Systemkommando in die primäre Batch-Eingabedatei für Kommandos `CMSYNIN` ein:

  ```
  PROFILER
  ```

  Anmerkung: Nach dem Systemkommando `PROFILER` erwartet der Profiler eine oder mehrere Zeilen mit Profiler-Schlüsselworteinträgen.

Um die Profiler Utility zu beenden:

- Geben Sie das folgende Profiler-Schlüsselwort in die primäre Batch-Eingabedatei für Kommandos `CMSYNIN` ein:

```
END-PROFILER
```

Oder:

```
END
```

Oder:

```
.
```

Syntax und Schlüsselwörter

Die Profiler Utility im Batch-Modus liest die Profiler-Schlüsselwörter, die die Profil-Erstellung aus der primären Batch-Eingabedatei für Kommandos CMSYNIN steuern. Der Profiler liest die Eingabezeilen, bis er das Schlüsselwort `END-PROFILER` (oder `END` oder `.`) erreicht.

Dieser Abschnitt behandelt die folgenden Themen:

- Profiler Utility-Syntax
- Profiler Utility-Schlüsselwörter

Profiler Utility-Syntax

Die Symbole, die in den in diesem Abschnitt gezeigten Syntaxdiagrammen verwendet werden, werden in der *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Geben Sie ein Profiler Utility-Kommando mit einem der folgenden Syntaxformate ein:

keyword[=*value*][,*keyword*[=*value*]]...

Oder:

```
keyword
[value]
...
```

Anmerkungen:

1. Wenn ein Wert mit einem Schlüsselwort verknüpft ist, aber kein Gleichheitszeichen gefunden wird, erwartet der Profiler den Wert in einer separaten Eingabezeile ohne anderes Schlüsselwort (zweites Syntaxformat).
2. Das erste Syntaxformat erwartet die Eingabe im Delimiter-Modus (`IM=D`).
3. Das zweite Syntaxformat kann verwendet werden, wenn der Profiler mit dem Natural-Profilparameter `STACK` ausgeführt werden soll oder wenn die Daten im Forms-Modus (IM=F) eingegeben werden.

Es gelten die folgenden Regeln:

- Leerzeilen und Zeilen, die mit einem Stern (`*`) beginnen, werden ignoriert.
- Alle Zeichen in einer Zeile von `/*` bis `*/` oder bis zum Ende der Zeile werden ignoriert.
- Einige Schlüsselwörter haben keinen zugeordneten Wert.
- Leerzeichen können vor oder nach dem Schlüsselwort oder Wert hinzugefügt werden.
- Mehrere Schlüsselwörter in einer Zeile werden durch Kommas getrennt (gilt nur für das erste Syntaxformat).
- Ein Wert kann in Apostrophe (`'value'`) eingeschlossen werden.
- Ein Wert darf kein Komma enthalten.
- Schlüsselwörter und Werte können in Groß- oder Kleinbuchstaben angegeben werden.
- Die maximale Länge der Eingabezeile beträgt 78 Zeichen.

Die Profiler Utility kann mehrmals in einer Natural-Session ausgeführt werden. Zum Beispiel wird sie zuerst mit den Funktionen `INIT` und `START` und dann, nach der Ausführung eines Benutzerprogramms, mit der `PAUSE`-Funktion ausgeführt.

Beispiel:

Das folgende Natural-Batch-Beispiel (auf z/OS) zeigt die ursprüngliche JCL, die das `XPROF`-Programm in der Library `PRFDEMO` (Zeilen in normaler Schrift) ausführt, und die Eingabezeilen der Profiler Utility, die zum Profilieren des `XPROF`-Programms verwendet werden (Zeilen fett gedruckt).

```
//CMSYNIN  DD  *
PROFILER
FUNCTION=INIT                /* Initialize profiling
  RESOURCE=ON                /* Write to resource
  RESOURCE-NAME='Demo01',REPLACE=YES  /* Resource name
  RESOURCE-LIB=PRFDATA       /* Resource library
FUNCTION=START               /* Start data collection
END-PROFILER                 /* End Profiler input
LOGON PRFDEMO
XPROF
10000
FIN
```

Das folgende Natural Batch-Beispiel (auf z/OS) zeigt, wie der Profiler mit dem Natural-Profilparameter `STACK` ausgeführt werden soll.

```
STACK=(
 PROFILER FUNCTION:INIT:
   RESOURCE:ON:
   RESOURCE-LIB:PRFDATA:
   REPLACE:YES:
 FUNCTION:START:
 END-PROFILER;
 LOGON PRFDEMO
 )
```

Profiler Utility-Schlüsselwörter

Die wichtigsten Schlüsselwörter, die in der Syntax der Profiler Utility im Batch-Modus verwendet werden, werden in der folgenden Tabelle beschrieben. Alle zusätzlichen (untergeordneten) Schlüsselwörter, die bei einem Hauptschlüsselwort verfügbar sind, werden in den Abschnitten beschrieben, auf die in der Tabelle verwiesen wird. Im Allgemeinen muss ein untergeordneter Schlüsselwortwert dem Hauptschlüsselwortwert folgen, zum Beispiel:

```
FUNCTION=READ
  PRINT=ON
```

Ein untergeordnetes Schlüsselwort, das vor dem ersten `FUNCTION`- oder `FILTER`-Schlüsselwort angegeben wurde, wird als untergeordnetes Schlüsselwort des ersten `FUNCTION`- oder `FILTER`-Schlüsselworts behandelt.

Die folgenden Hauptschlüsselwörter stehen zur Verfügung:

Schlüsselwort	Wert	Profiler Utility-Funktion ausführen
FUNCTION	CONSOLIDATE	Ressourcendaten konsolidieren (aggregieren). Siehe *Ereignisdaten konsolidieren*.
	COVERAGE	Natural-Codeabdeckung initialisieren. Diese Funktion ist für die Sammlung von Code-Abdeckungsdaten obligatorisch. Siehe *Initialisierung der Codeabdeckung (Code Coverage)*.
	DELETE	Eine Natural Profiler-Ressourcendatei löschen. Siehe *Profiler-Ressourcendateien verwalten*.
	INIT	Profil-Erstellung initialisieren. Diese Funktion ist für die Profil-Erstellungsdatenerfassung obligatorisch. Siehe *Profilierung initialisieren*.
	LIST	Profiler-Ressourcen auflisten. Siehe *Profiler-Ressourcendateien auflisten* unter *Maintaining Profiler Resource Files*.
	PAUSE	Datenerfassung pausieren. Siehe *Datensammlung starten und anhalten*.
	READ	Ressourcendaten lesen und bewerten. Siehe *Ereignisdaten auswerten*.
	START	Datensammlung starten bzw. erneut starten. Siehe *Datensammlung starten und anhalten*.
	TEXT	Zu den Ereignisdaten ein Benutzerereignis hinzufügen. Siehe *Benutzerdefinierter Ereignisse schreiben*.
FILTER		Profiler-Filter definieren, um die Menge der Ereignisdaten zu reduzieren. Siehe *Filtern zur Begrenzung der gesammelten Daten benutzen*.
	COUNT	Filter für die Ereignisanzahl setzen.
	EVENT	Ereignis-, FNAT- und Statement-Filter setzen.
	PROGRAM	Library-, Programm und Zeilen-Filter setzen.
	TIME	CPU-Zeitfilter setzen.
RPC		Einen Batch-Natural RPC-Server profilieren. Siehe *Batch-Natural RPC-Server profilieren*.
ON-ERROR		Behandlung von Profiler-Fehlersituationen festlegen. Standardwert: TERMINATE
	CONTINUE	Profiling stoppen, aber Natural-Session fortsetzen.
	TERMINATE	Mit dem Natural Profiler eine Beendigung der Natural Session erzwingen.
TRACE	0 - 10	Ebene der internen Ablaufverfolgung der Profiler-Trace-Sitzung festlegen. Das interne Trace enthält Informationen wie Profiler-Fehler und wird in die Standardausgabe der Trace-Sitzung (CMPRINT-Datensatz) geschrieben. Siehe *Interner Trace*. Standardwert: 2 (Warnung).
HELP		Hilfe benutzen. Es wird eine zusammengefasste Beschreibung der Profiler-Schlüsselwörter in die Standardausgabe geschrieben.

Schlüsselwort	Wert	Profiler Utility-Funktion ausführen
INCLUDE	*object-name*	Der Name des Natural-Textobjekts, das Profiler-Eingabedaten enthält. Siehe auch ***Profiler-Eingaben aus Natural-Textobjekten einfügen***.
INCLUDE-LIB	*library-name*	Name der Natural-Library, die das mit dem INCLUDE-Schlüsselwort angegebene Textobjekt enthält. Wenn die Natural-Systemvariable *LIBRARY-ID angegeben wird, wird der Name der aktuellen Bibliothek verwendet. Der Library-Name wird für alle folgenden INCLUDE-Schlüsselwörter verwendet. Standardeinstellung: Wenn INCLUDE-LIB nicht vor einem INCLUDE-Schlüsselwort angegeben ist, wird standardmäßig die Natural-System-Library SYSPRFLR verwendet. Siehe auch ***Profiler-Eingaben aus Natural-Textobjekten einfügen***.
END-PROFILER oder END oder .		Profiler-Eingabe beenden. Das Schlüsselwort END-PROFILER, END oder ein Punkt (.) gibt das Ende der Profiler-Eingabe an.
ONLINE		Die Online Profiler Utility in einem Batch-Lauf ausführen. Diese Option dient der Kompatibilität mit früheren Natural-Versionen, als die Profiler Utility im Batch-Modus das Profiler-Online-Menü startete. Wenn diese Funktionalität weiterhin erforderlich ist, verwenden Sie ONLINE als erstes Schlüsselwort. Alle nachfolgenden Eingaben werden über die Profiler-Online-Menüs getätigt.

Gesammelte Ereignisse und Daten

Dieser Abschnitt beschreibt die Ereignisse und Daten, die von der Profiler Utility im Batch-Modus verarbeitet werden.

- Ereignisse

- Gesammelte Daten

Ereignisse

Während einer Natural-Session können verschiedene Arten von Ereignissen auftreten (z. B. ein Programmstart), bei denen der Profiler ereignisspezifische Daten in einem Trace-Datensatz sammelt. Jedes Ereignis ist mit einem Ereignistyp verknüpft, d. h. einem Code mit einem oder zwei Buchstaben. Verwandte Ereignistypen werden zu einer Ereignisgruppe zusammengefasst, die durch einen Ein-Buchstaben-Code gekennzeichnet ist.

Folgende Ereignisse, Ereignistypen und Ereignisgruppen stehen zur Verfügung:

Ereignis	Ereignistyp	Ereignisgruppe	Zeitpunkt des Auftretens
Session **I**nitialization	SI	S	Wenn eine Natural-Batch-Session initialisiert wird. Da die Profiler-Monitor-Session nach der Trace-Session gestartet wird, kann dieses Ereignis nicht überwacht werden.
Session **T**ermination	ST	S	Wenn eine Natural-Batch-Session beendet wird. Der Profiler überwacht dieses Ereignis immer.
Program **L**oad	PL	P	Wenn ein Programm (Natural-Objekt) geladen wird oder wenn es sich bereits im Buffer Pool befindet.
Program **S**tart	PS	P	Wenn ein Programm (Natural-Objekt) gestartet wird.
Program **T**ermination	PT	P	Wenn ein Programm (Natural-Objekt) beendet wird.
Program **R**esume	PR	P	Wenn ein Programm (Natural-Objekt) die Kontrolle wieder übernimmt, nachdem ein anderes Natural-Objekt ausgeführt wurde, oder wenn die Kontrolle zur Ebene 0 zurückkehrt (kein Programm aktiv).
Program **I**nformation	PI	P	Wenn auf ein Programm (Natural-Objekt) zum ersten Mal zugegriffen wird. Dieses Ereignis wird nur bei der Natural-Codeabdeckung ausgelöst.
Before **D**atabase Call	DB	D	Bevor ein Datenbankaufruf ausgeführt wird.
After **D**atabase Call	DA	D	Nachdem ein Datenbankaufruf ausgeführt worden ist.
Before Terminal **I**/O	IB	I	Bevor eine Terminal-Ein-/Ausgabe ausgeführt wird.
After Terminal **I**/O	IA	I	Nachdem eine Terminal-Ein-/Ausgabe ausgeführt worden ist.
Before External Program **C**all	CB	C	Bevor ein externer Programmaufruf (CALL-Statement) ausgeführt wird.
After External Program **C**all	CA	C	Nachdem ein externer Programmaufruf (CALL-Statement) ausgeführt worden ist.
Runtime **E**rror	E	E	Wenn ein Natural-Laufzeitfehler aufgetreten ist.
Natural **S**tatement	NS	N	Wenn ein Natural-Statement ausgeführt wird. Aus technischen Gründen gibt es keine Eins-zu-Eins-Beziehung zwischen einem

Ereignis	Ereignistyp	Ereignisgruppe	Zeitpunkt des Auftretens
			Natural-Quellcode-Statement und dem entsprechenden Objektcode im katalogisierten Objekt. Daher können mehrere Natural-Statements zu einem NS-Ereignis zusammengeführt werden, und umgekehrt kann ein Natural-Statement mehrere NS-Ereignisse abdecken.
Inbound **R**PC Message	RI	R	Wenn das Natural RPC Server Layer die Client-Anforderung empfängt.
Start of **R**PC Request Execution	RS	R	Wenn das Natural RPC Server Layer das Natural-Server-Programm aufruft.
Outbound **R**PC Message	RO	R	Wenn der Natural RPC Server das Ergebnis an den Client zurückgibt.
RPC **W**ait for Client	RW	R	Wenn der Natural RPC Server auf die nächste Nachricht vom Client wartet.
User-Defined **E**vent	U	U	Wenn ein benutzerdefiniertes Ereignis generiert wurde. Siehe *Benutzerdefinierter Ereignisse schreiben*.
Monitor **P**ause	MP	M	Wenn die Datensammlung angehalten wird. Ein Pausenereignis kann durch eine explizite Pausenanforderung, zu Beginn eines Blockfilters oder wenn der Datenpool voll ist, verursacht werden. Die Dauer einer Pause wird bei der Analyse der Anwendungs-Performance nicht berücksichtigt.

Mit jedem gesammelten Ereignis werden ein CPU- und ein Ereigniszeitstempel aufgezeichnet. Im Allgemeinen wird zu Beginn eines Ereignisses ein Zeitstempel genommen. Die Dauer eines Ereignisses entspricht daher der Zeit, die zwischen dem Zeitstempel des Ereignisses und dem Zeitstempel des folgenden Ereignisses vergeht.

Gesammelte Daten

Dieser Abschnitt beschreibt die von der Profiler Utility gesammelten Daten:

Allgemeine Daten

Die folgenden Datenelemente werden bei jedem Ereignis gesammelt:

- Ereigniszähler
- Ereignistyp
- Ereigniszeit in Mikrosekunden
- Session-CPU-Zeit in Mikrosekunden
- Trace-Session-Kennung
- Natural Security-Benutzergruppenkennung
- Natural-Benutzerkennung

- Natural-Anwendungsname
- Programm-Library
- Programmname
- Programmebene
- Copycode-Library
- Copycode-Name
- Statement-Zeilennummer
- Statement-Op-Code
- Coverage Flag (für Natural-Codeabdeckung)

Anmerkungen:

1. Die für die Datensammlung aufgewendete Zeit (Monitor-CPU-Zeit) wird separat gemessen und aus der CPU-Zeit der Session herausgenommen.
2. Die Ereignisse werden gezählt, bevor eine Filterung oder Probenahme (Sampling) durchgeführt wird. Daher erhalten die Ereignisse im Allgemeinen die gleichen Zählergebnisse, unabhängig davon, welche Filterung oder Probenahme verwendet wird.
3. Statement-Ereignisse werden nur gezählt, wenn das Ereignisfilter-Schlüsselwort `STATEMENT=COUNT` verwendet wird.
4. Monitor Pause-Ereignisse werden nicht gezählt.
5. Natural-Codeabdeckung sammelt nur `NS`- und `PI`-Ereignisse
6. Natural-Codeabdeckung sammelt keine Zeitwerte.
7. Für jedes Objekt, auf das zugegriffen wird, und für alle (rekursiv) im Objekt enthaltenen Copycodes wird ein `PI`-Ereignis gesammelt.

Ereignisspezifische Daten:

Die folgenden Daten werden nur bei folgenden Ereignissen erhoben:

Ereignis	Datenelemente
Session Initialization (Session-Initialisierung)	Keines
Session Termination (Session-Beendigung)	Beendigung-Rückgabecode Natural-Beendigung-Meldungscode NAT99*nn* Name des Back-End-Programms Monitor-CPU-Zeit im Mikrosekunden
Program Load (Laden eines Programms)	Name des zu ladenden Programms Name der Load Library Aufruftyp
Program Resume	Keines

Ereignis	Datenelemente
(Programm-Wiederaufnahme)	
Program Start/Termination (Programm-Start/Beendigung)	Programmtyp Datenbankkennung der Programm-Library Dateinummer der Programm-Library
Program Information (Programminformationen)	Programmtyp Anzahl der Statements in dem Programm oder im Copycode Erstes Statement Item `INCLUDE`-Zeilennummer Parent-Copycode-Kennung
Database Call (Datenbankaufruf)	Datenbanktyp Kommandocode Kommandokennung Datenbankkennung DBID Dateinummer Antwortcode (Ereignistyp `DA`) Fehler-Subcode (Ereignistyp `DA`) Adabas-Kommandozeit (Ereignistyp `DA`)
Terminal I/O (Terminal-Ein-/Ausgabe)	Anzahl gesendeter Bytes Anzahl gelesener Bytes Zugeordneter Gesamtspeicher Länge des komprimierten Session-Speichers
External Program Call (externer Programmaufruf)	Name des aufgerufenen Programms Aufrufmodus, z.B. dynamischer oder statischer Modus Programm-Link-Ort Parametertyp, z.B. Referenz oder Wert Rückmeldecode (Ereignistyp `CA`)
Runtime Error (Laufzeitfehler)	Natural-Systemfehlermeldungscode Externer Abend Code Name des Fehlerbehandlungsprogramms
Natural Statement	Profiling: Keines Natural-Codeabdeckung: Statement Item Identifier (GP Offset)
Start of RPC Request Execution (Beginn der RPC-Anforderungsausführung)	Umgebung (`C` = Client, `S` = Server) Name des Subprogramms Adabas-Benutzerkennung (ETID) Status der Konversation Anmelde-Indikator (`Y` = Anmeldung durchgeführt) Impersonierung-Indikator der RPC-Anforderung (`Y` = Impersonierung durchgeführt)
Outbound/Inbound RPC Message / RPC Wait for Client (Ausgehende/eingehende RPC-Nachricht / RPC Warten auf Client)	Umgebung (`C` = Client, `S` = Server) Transportprotokoll RPC-Funktion Typ der Client-Benutzerkennung Länge der Nachricht RPC-Rückgabecode

Ereignis	Datenelemente	
	Externe Konversationskennung Client-Benutzerkennung Serverknoten (Ereignistypen RO und RW) Servername (Ereignistypen RO und RW)	
User-Defined Event Benutzerdefiniertes Ereignis	Subtyp des benutzerdefinierten Ereignisses Bis zu 249 Bytes benutzerdefinierte Informationen	
Monitor Pause	Art der Monitorpause Mögliche Werte:	
	R	Monitorpause angefordert. Dieser Wert wird auch gesetzt, wenn die Sitzung mit der Pause-Option initialisiert wird.
	F	Beginn eines Blocks von herausgefilterten Ereignissen. Blockfilter sind: Library, Programm, Zeile, FNAT, Ereignisanzahl oder Zeitfilter.
	W	Die Trace-Session wartet, weil der Daten-Pool voll ist.

Profilierung initialisieren

Die INIT-Funktion der Profiler Utility initialisiert die Profil-Erstellung. Die INIT-Funktion muss vor einer der Profiler Utility-Funktionen START, PAUSE oder TEXT oder einem Profiler-Filter ausgeführt werden.

Syntax der INIT-Funktion:

```
FUNCTION=INIT
[RESOURCE={ON|OFF}]
[RESOURCE-NAME=resource-name]
[RESOURCE-LIB=library-name]
[REPLACE={YES|NO}]
[SAMPLING={ON|OFF}]
[INTERVAL={100|interval-time}]
[CONSOLIDATE={ON|OFF}]
[TRANSACTION={ON|OFF}]
[WAIT-FULL={60|wait-full-time}]
```

```
[WAIT-EMPTY={60|wait-empty-time}]
[CMPRMIN=data-set-name]
[CMSYNIN=data-set-name]
[TRACE-EVENT={ON|OFF}]
[TRACE-MONITOR={3|trace-level}]
[TRACE-CONSOLIDATE={ON|OFF}]
```

Syntax-Beschreibung:

Schlüsselwort für INIT	Wert	Beschreibung
RESOURCE		Gibt an, ob die Ereignisdaten in eine Natural Profiler-Ressourcendatei (NPRF oder NPRC) geschrieben werden sollen.
	ON	Die Ereignisdaten werden in eine Natural Profiler-Ressourcendatei geschrieben und können mit NaturalONE oder mit den Datenverarbeitungsfunktionen der Natural Profiler Utility analysiert werden.
	OFF	Die Ereignisdaten werden nicht in eine Natural Profiler-Ressourcendatei geschrieben. Diese Option kann verwendet werden, wenn nur der Profiler-Ereignis-Trace oder die Statistiken erforderlich sind.
RESOURCE-NAME	*resource-name*	Der Name der Profiler-Ressourcendatei, in der die Ereignisdaten für eine spätere Analyse gespeichert werden. Wenn die Erweiterung .nprf (Natural Profiler-Ressourcendatei) oder .nprc (Natural Profiler-Ressource, konsolidiert) nicht angegeben wurde, wird sie automatisch hinzugefügt. Welche Erweiterung verwendet wird, hängt vom Schlüsselwort CONSOLIDATE ab. Standardeinstellung: *INIT-USER_*yyyymmdd_hhiiss* Dabei ist *INIT-USER der Inhalt der entsprechenden Natural-Systemvariablen (Name des Jobs oder Benutzerkennung, unter der die Natural-Session läuft); und *yyyymmdd_hhiiss* ist das Datum und die Uhrzeit der Ressourcengenerierung.
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, der die Ressource zugeordnet ist. Standardwert: Der Name der aktuellen Library.
REPLACE		Gibt an, ob die Ressource ersetzt wird, wenn sie bereits vorhanden ist.
	YES	Ressource ersetzen, falls vorhanden.

Schlüsselwort für INIT	Wert	Beschreibung
	NO	Ressource nicht ersetzen. Wenn die Ressource bereits vorhanden ist, wird eine Nachricht geschrieben und es wird keine Profil-Erstellung durchgeführt.
SAMPLING		Gibt an, ob die CPU-Zeit-Probenahme (Sampling) des Profilers aktiviert ist. Die Probenahme kann die Anzahl der überwachten Ereignisse drastisch reduzieren, wobei fast die gleichen CPU-Zeitergebnisse erzielt werden. Siehe ***Stichprobentechnik (Natural Profiler Sampling)***.
	ON	Sampling aktivieren.
	OFF	Sampling deaktivieren.
INTERVAL	*interval-time*	Das Stichprobenintervall bestimmt, bei welchen CPU-Zeitstempel Ereignisse gesammelt werden. Wenn das Stichprobenintervall größer ist, werden weniger Ereignisse gesammelt. Wenn das Abtastintervall kleiner ist, sind die resultierenden CPU-Zeiten genauer. Gültige Werte für *interval-time*: 1 bis 2147483647 Einheit: Mikrosekunden
CONSOLIDATE		Gibt an, ob die Ereignisdaten konsolidiert (aggregiert) werden, bevor sie in die Ressourcendatei geschrieben werden. Siehe ***Datenkonsolidierung, Codeabdeckung und Datenverarbeitung*** im Abschnitt *Grundlegendes Konzept der Profiler Utility*.
	ON	Die Ereignisdaten werden konsolidiert und in eine NPRC-Ressourcendatei (Natural Profiler Resource Consolidated) geschrieben.
	OFF	Die Ereignisdaten werden nicht konsolidiert in eine NPRF-Ressourcendatei (Natural Profiler Resource File) geschrieben.
TRANSACTION		Diese Option gilt nur, wenn CONSOLIDATE (siehe oben) auf ON gesetzt ist. Gibt an, ob Transaktionsbezeichner (Identifier) zu den konsolidierten Ereignisdaten hinzugefügt werden.
	ON	Transaktionsbezeichner werden den konsolidierten Ereignisdaten hinzugefügt. Transaktionsbezeichner sind eine Voraussetzung für die Bewertung der Transaktionsantwortzeit. Siehe ***Transaktionszusammenfassung***. **Anmerkung:** Die generierte NPRC-Ressourcendatei benötigt mehr Speicherplatz, wenn ON gesetzt ist.

Schlüsselwort für INIT	Wert	Beschreibung
	OFF	Es werden keine Transaktionsbezeichner zu den konsolidierten Ereignisdaten hinzugefügt.
WAIT-FULL	*wait-full-time*	Gibt an, wie lange die Trace-Session wartet, wenn der Profiler-Daten-Pool voll ist. Wenn das Limit erreicht ist und immer noch kein freigegebener Platz verfügbar ist, wird Natural beendet (ABEND SOC1). Der Wert verhindert, dass die Trace-Session endlos auf eine unerwartet ausgefallene Monitor-Session wartet. Gültige Werte für *wait-full-time*: 1 bis 32767 Einheit: Sekunden
WAIT-EMPTY	*wait-empty-time*	Gibt an, wie lange die Profiler-Monitor-Session wartet, wenn der Profiler-Daten-Pool leer ist und die Trace-Session noch aktiv ist. Wenn das Limit erreicht ist, wird eine Nachricht geschrieben und das Profiling gestoppt. Ein Wert von Null (0) bedeutet, dass der Profiler ohne Limit wartet. Ein positiver Wert verhindert, dass die Monitor-Session endlos auf eine möglicherweise fehlerhafte Trace-Session wartet. Gültige Werte für *wait-empty-time*: 0 bis 2147483647 Beim Profiling eines Batch-Natural RPC-Servers oder beim Profiling einer Großrechner-Session von Natural Studio wird der Wert immer auf 0 gesetzt. Siehe ***Batch-Natural RPC-Server profilieren*** und ***Profil-Erstellung einer Großrechner-Session aus Natural Studio***. Einheit: Sekunden
CMPRMIN	*data-set-name*	Gibt den Namen des Eingabe-Dataset für dynamische Parameter für die Profiler-Monitor-Session an. Siehe ***Monitor-Session CMPRMIN***. Standardwert: CMPRMIN (die Monitor-Session verwendet die Parameter der Trace-Session)
CMSYNIN	*data-set-name*	Gibt den Namen des primären Kommandoeingabe-Dataset für die Profiler-Monitor-Session an. Siehe ***Monitor-Session CMSYNIN***. Standardwert: DUMMY (die Monitor-Session verwendet keinen primären Kommandoeingabe-Dataset).

Schlüsselwort für INIT	Wert	Beschreibung
TRACE-EVENT		Gibt an, ob das Profiler-Trace in die Standardausgabe der Profiler-Monitor-Session (MONPRINT-Dataset) geschrieben wird. Siehe *Ereignis-Trace*.
	ON	Das Profiler-Ereignis-Trace schreiben.
	OFF	Das Profiler-Ereignis-Trace nicht schreiben.
TRACE-MONITOR	*trace-level*	Legt die Ebene des internen Tracing der Profiler-Monitor-Session fest. Das interne Trace enthält Informationen wie Profiler-Fehler und wird in die Standardausgabe der Monitor-Session (MONPRINT-Dataset) geschrieben. Siehe *Interner Trace*. Gültige Trace-Ebenen: 0 bis 10
TRACE-CONSOLIDATE		Gibt an, ob das Profiler-Konsolidierungs-Trace in die Standardausgabe geschrieben wird. Die Konsolidierungsverfolgung kann nur geschrieben werden, wenn CONSOLIDATE=ON angegeben ist. Siehe *Konsolidierungs-Trace*.
	ON	Das Profiler-Konsolidierungs-Trace schreiben.
	OFF	Das Profiler-Konsolidierungs-Trace nicht schreiben.

Beispiel für INIT

```
FUNCTION=INIT            /* Initialize Profiling
  RESOURCE=ON            /* Write to resource file
  RESOURCE-NAME='Test'   /* Resource name
  RESOURCE-LIB=PRFDATA   /* Resource library
  REPLACE=YES            /* Replace resource
  SAMPLING=ON            /* Use sampling
  INTERVAL=100           /* Sampling interval
  CONSOLIDATE=OFF        /* Do not consolidate the data
  TRANSACTION=OFF        /* Do not add transaction identifiers
  WAIT-FULL=60           /* Wait sec if pool full
  WAIT-EMPTY=60          /* Wait sec if pool empty
  CMPRMIN=PRFPARMS       /* CMPRMIN for monitor session
  TRACE-EVENT=ON         /* Trace events
  TRACE-MON=3            /* Trace level monitor session
```

Initialisierung der Codeabdeckung (Code Coverage)

Die Funktion `COVERAGE` der Profiler Utility initialisiert die Messung der Natural-Codeabdeckung. Die `COVERAGE`-Funktion muss vor einer der Profiler Utility-Funktionen `START`, `PAUSE` oder `TEXT` oder einem Profiler-Filter ausgeführt werden.

Syntax von `COVERAGE`:

```
FUNCTION=COVERAGE
[RESOURCE={ON|OFF}]
[RESOURCE-NAME=resource-name]
[RESOURCE-LIB=library-name]
[REPLACE={YES|NO}]
[WAIT-FULL={60|wait-full-time}]
[WAIT-EMPTY={60|wait-empty-time}]
[CMPRMIN=data-set-name]
[CMSYNIN=data-set-name]
[TRACE-EVENT={ON|OFF}]
[TRACE-MONITOR={3|trace-level}]
[TRACE-COVERAGE={ON|OFF}]
```

Syntax-Beschreibung:

Schlüsselwort für COVERAGE	Wert	Beschreibung
`RESOURCE`		Gibt an, ob Codeabdeckungsdaten in ein Natural Code Coverage Resource File (NCVF) geschrieben werden sollen.
	`ON`	Codeabdeckungsdaten werden in eine Natural-Code-Coverage-Ressourcendatei geschrieben und können mit NaturalONE oder mit den Datenverarbeitungsfunktionen der Natural Profiler Utility analysiert werden.
	`OFF`	Es werden keine Codeabdeckungsdaten werden in eine Natural-Code-Coverage-Ressourcendatei geschrieben. Diese Option kann verwendet werden, wenn nur das Profiler-Ereignis-Trace oder die Statistiken erforderlich sind.
`RESOURCE-NAME`	*`resource-name`*	Der Name der Natural-Code-Coverage-Ressourcendatei, in der Codeabdeckungsdaten für eine spätere Analyse gespeichert werden. Wenn die Erweiterung `.ncvf` (Natural code coverage resource file) nicht angegeben wurde, wird sie automatisch hinzugefügt. Standardeinstellung: `*INIT-USER_yyyymmdd_hhiiss`

Schlüsselwort für COVERAGE	Wert	Beschreibung
		Dabei ist *INIT-USER der Inhalt der entsprechenden Natural-Systemvariablen ist (Name des Jobs oder Benutzerkennung, unter der die Natural-Session läuft); und *yyyymmdd_hhiiss* ist das Datum und die Uhrzeit der Ressourcengenerierung.
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, der die Ressource zugeordnet ist. Standardwert: Der Name der aktuellen Library.
REPLACE		Gibt an, ob die Ressource ersetzt wird, wenn sie bereits vorhanden ist.
	YES	Ressource ersetzen, falls vorhanden.
	NO	Ressource nicht ersetzen. Wenn die Ressource bereits vorhanden ist, wird eine Nachricht geschrieben und es wird kein Code Coverage durchgeführt.
WAIT-FULL	*wait-full-time*	Gibt an, wie lange die Trace-Session wartet, wenn der Profiler-Daten-Pool voll ist. Wenn das Limit erreicht ist und immer noch kein freigegebener Platz verfügbar ist, wird Natural beendet (ABEND SOC1). Der Wert verhindert, dass die Trace-Session endlos auf eine unerwartet ausgefallene Monitor-Session wartet. Gültige Werte für *wait-full-time*: 1 bis 32767 Einheit: Sekunden
WAIT-EMPTY	*wait-empty-time*	Gibt an, wie lange die Profiler-Monitor-Session wartet, wenn der Profiler-Daten-Pool leer ist und die Trace-Session noch aktiv ist. Wenn das Limit erreicht ist, wird eine Nachricht geschrieben und das Profiling gestoppt. Ein Wert von Null (0) bedeutet, dass der Profiler ohne Limit wartet. Ein positiver Wert verhindert, dass die Monitor-Session endlos auf eine möglicherweise fehlerhafte Trace-Session wartet. Gültige Werte für *wait-empty-time*: 0 bis 2147483647 Bei der Messung der Codeabdeckung eines Batch-Natural RPC-Servers oder einer Großrechner-Session aus Natural Studio wird der Wert immer auf 0 gesetzt. Siehe ***Batch-Natural RPC-Server profilieren*** und ***Profil-Erstellung einer Großrechner-Session aus Natural Studio***. Einheit: Sekunden

Schlüsselwort für COVERAGE	Wert	Beschreibung
CMPRMIN	*data-set-name*	Gibt den Namen des Parameter-Eingabe-Dataset für die Profiler-Monitor-Session an. Siehe ***Monitor-Session CMPRMIN***. Standardwert: CMPRMIN (die Monitor-Session verwendet die Parameter der Trace-Session).
CMSYNIN	*data-set-name*	Gibt den Namen des primären Kommando-Eingabe-Dataset für die Profiler-Monitor-Session an. Siehe ***Monitor-Session CMSYNIN***. Standardwert: DUMMY (die Monitor-Session verwendet keinen primären Kommando-Eingabe-Dataset)
TRACE-EVENT		Gibt an, ob das Profiler-Ereignis-Trace in die Standardausgabe der Profiler-Monitor-Session (MONPRINT-Dataset) geschrieben wird. Siehe ***Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung***.
	ON	Das Profiler-Ereignis-Trace schreiben.
	OFF	Das Profiler-Ereignis-Trace nicht schreiben.
TRACE-MONITOR	*trace-level*	Legt die Ebene des internen Tracing der Profiler-Monitor-Session fest. Das interne Trace enthält Informationen wie Profiler-Fehler und wird in die Standardausgabe der Monitor-Session (MONPRINT-Dataset) geschrieben. Siehe ***Interner Trace***. Gültige Trace-Ebenen: 0 bis 10
TRACE-COVERAGE		Gibt an, ob das Profiler-Codeabdeckungs-Trace in die Standardausgabe geschrieben wird. Siehe ***Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung***.
	ON	Das Profiler-Konsolidierungs-Trace schreiben.
	OFF	Das Profiler-Konsolidierungs-Trace nicht schreiben.

Beispiel für COVERAGE

```
FUNCTION=COVERAGE       /* Initialize code coverage
  RESOURCE=ON           /* Write to resource file
  RESOURCE-NAME='Test'  /* Resource name
  RESOURCE-LIB=COVDATA  /* Resource library
  REPLACE=YES           /* Replace resource
  WAIT-FULL=60          /* Wait sec if pool full
  WAIT-EMPTY=60         /* Wait sec if pool empty
  CMPRMIN=PRFPARMS      /* CMPRMIN for monitor session
  TRACE-EVENT=ON        /* Trace events
  TRACE-MON=3           /* Trace level of monitor session
  TRACE-COVERAGE=ON     /* Trace coverage data
```

Datensammlung starten und anhalten

Der Profiler muss initialisiert werden, bevor die Datensammlung gestartet oder angehalten werden kann. Da die Datensammlung nach der Initialisierung angehalten wird, muss sie in irgendeiner Weise gestartet werden, damit Ereignisdaten aufgezeichnet werden.

Sie können die Datensammlung mittels einer der folgenden Methoden starten und anhalten:

- Profiler Utility-Funktionen benutzen
- Profiler Utility-Programme benutzen
- Anwendungsprogrammierschnittstelle USR8210N benutzen

Profiler Utility-Funktionen benutzen

Die Profiler Utility-Funktionen `START` und `PAUSE` werden verwendet, um die Datensammlung zu starten und anzuhalten. Es gilt die folgende Syntax:

```
FUNCTION=START [COUNT={0|count-number}]
FUNCTION=PAUSE
```

Syntax-Beschreibung:

Schlüsselwort für START	Wert	Beschreibung
`COUNT`	*`count-number`*	Setzt den Ereigniszähler des nächsten überwachten Ereignisses auf den angegebenen Wert. Gültige Werte für *`count-number`*: `0` bis `2147483647` Der Ereigniszähler bleibt unverändert, wenn ein Wert von Null (`0`) angegeben wird.

Profiler Utility-Programme benutzen

Die folgenden Natural-Programme stehen in der System-Library `SYSPRFLR` bereit, um Profiler Utility-Funktionen auszuführen:

Programm	Beschreibung
PRFSTART	Die Datensammlung starten.
PRFPAUSE	Die Datensammlung anhalten.
PRFSTATE	Den Status der Datensammlung abfragen.
PRFFCT	Eine Profiler Utility-Funktionen ausführen: `START`, `PAUSE` oder `STATE`.

Um Profiler Utility-Programme zu benutzen:

- Melden Sie sich bei der Library `SYSPRFLR` an oder kopieren Sie die Programme in die Library `SYSTEM`, in die entsprechende Steplib-Library oder in die erforderliche Library.

  Wenn `PRFFCT` verwendet wird, muss auch die Anwendungsprogrammierschnittstelle `USR8210N` kopiert werden (siehe folgenden Abschnitt).

  Falls `PRFFCT` in einer Client/Server-Umgebung verwendet wird, kopieren Sie `PRFFCT` in die Client-Library und `USR8210N` in die Server-Library.

  Anmerkung: `PRFFCT` erwartet als Eingabewert `START`, `PAUSE` oder `STATE`, um die entsprechende Funktion auszuführen.

Um die Datensammlung zu starten:

- Führen Sie das folgende Programm aus:

```
PRFSTART
```

  Oder:

```
PRFFCT
START
```

Um die Datensammlung anzuhalten:

- Führen Sie das folgende Programm aus:

```
PRFPAUSE
```

  Oder:

```
PRFFCT
PAUSE
```

Um den aktuellen Status der Datensammlung abzufragen:

- Führen Sie das folgende Programm aus:

```
PRFSTATE
```

Oder:

```
PRFFCT
STATE
```

Anwendungsprogrammierschnittstelle USR8210N benutzen

Die Datensammlung kann von der profilierten Natural-Anwendung aus gestartet und angehalten werden, indem die Anwendungsprogrammierschnittstelle (API) `USR8210N` aufgerufen wird. Die API kann auch verwendet werden, um den aktuellen Status des Überwachungsprozesses zu ermitteln. Die API wird in der Library `SYSEXT` ausgeliefert. Weitere Informationen siehe *SYSEXT Utility*.

Um die API zu benutzen:

- Kopieren Sie das Subprogramm `USR8210N` in die Library `SYSTEM`, in die entsprechende Steplib-Library oder in die gewünschte Library.

  Anmerkung: Die API `USR8210N` erwartet als erster Parameter den Wert `START`, `PAUSE` oder `STATE`, um die entsprechende Funktion auszuführen. Die Parameterwerte können in Groß- oder Kleinbuchstaben angegeben werden. Bei der Rückkehr enthält `P-RETURN` den Rückgabecode und `P-MESSAGE` die Erfolgs- oder Fehlermeldung.

Um die Datensammlung zu starten:

- Benutzen Sie die API mit dem `CALLNAT`-Statement:

```
CALLNAT 'USR8210N' 'START' P-RETURN P-MESSAGE /* Start Profiler
```

Um die Datensammlung anzuhalten:

- Benutzen Sie die API mit dem `CALLNAT`-Statement:

```
CALLNAT 'USR8210N' 'PAUSE' P-RETURN P-MESSAGE /* Pause Profiler
```

Um den aktuellen Status der Datensammlung abzufragen:

- Benutzen Sie die API mit dem `CALLNAT`-Statement:

```
CALLNAT 'USR8210N' 'STATE' P-RETURN P-MESSAGE /* Get Profiler state
```

Der Status ist im Feld `P-RETURN` codiert:

P-RETURN	Beschreibung
0	Die Datensammlung mit dem Natural Profiler ist gestartet.
1	Die Datensammlung mit dem Natural Profiler ist angehalten.

Filter zur Begrenzung der gesammelten Daten benutzen

Filter spielen eine wichtige Rolle bei der Reduzierung der Menge der gesammelten Daten. Folgende Filter stehen zur Verfügung:

- Ereignisfilter (Event Filter)
- Programmfilter
- Zählfilter
- Zeitfilter

Anmerkung: Der Profiler muss initialisiert werden, bevor Einstellungen auf einen bestimmten Filter angewendet werden können.

Ereignisfilter (Event Filter)

Der Profiler-Ereignisfilter gibt an, welche Ereignisse gesammelt werden. Darüber hinaus bestimmt es, ob Ereignisdaten aufgezeichnet werden, während Natural-Systemprogramme ausgeführt werden.

Syntax des Ereignisfilters:

```
FILTER=EVENT
[EVENT={[event-type]...|ALL}
[STATEMENT={ON|OFF|COUNT}]
[FNAT={ON|OFF}
```

Syntax-Beschreibung:

Schlüsselwort für EVENT	Wert	Beschreibung
EVENT		Gibt an, welche Ereignisse gesammelt werden.
	event-type	Jedes Ereignis wird mit einem ein- oder zweistelligen Schlüssel codiert. Mehrere Ereignisse werden durch Leerzeichen getrennt. Es werden nur die angegebenen Ereignisse aufgezeichnet. Wenn kein Ereignis angegeben wird, überwacht der Profiler nur die Session- und Pausenereignisse. Mögliche Ereigniseinträge sind: DB, DA, PL, PS, PT, PR, IB, IA, E, CB, CA, U, RS, RI, RO und RW. **Notes:** 1. Informationen zu den Ereigniscodes finden Sie unter *Gesammelte Ereignisse und Daten*. 2. Die folgenden Ereignisse können nicht in der Ereignisliste angegeben werden: ■ SI-Ereignisse (Session-Initialisierung) können vom Profiler nicht im Batch-Modus gesammelt werden. ■ ST-Ereignisse (Session Termination) und MP-Ereignisse (Monitor Pause) werden immer gesammelt. ■ Die Sammlung von NS-Ereignissen (Natural Statement) hängt vom Schlüsselwort STATEMENT ab. 3. Wenn nur ein Zeichen angegeben wird, werden alle Ereignisse, die mit diesem Zeichen beginnen, aufgezeichnet. Zum Beispiel ist EVENT=P äquivalent zu EVENT=PL PR PS PT.

Schlüsselwort für EVENT	Wert	Beschreibung
	ALL	Alle Ereignisse werden aufgezeichnet; das heißt, die Definition ist äquivalent zu `EVENT=DB DA PL PS PT PR IB IA E CB CA U RS RI RO` **Vorsicht:** EVENT=ALL enthält das NS-Ereignis nicht. Die Sammlung von NS-Ereignissen (Natural Statement) hängt vom Schlüsselwort STATEMENT ab.
STATEMENT		Gibt an, ob Natural-Statement-Ereignisse (NS) gesammelt werden.
	ON	Natural-Statement-Ereignisse (NS) werden gesammelt, aber nicht gezählt. Es werden nur Nicht-Statement-Ereignisse gezählt. Natural-Statement-Ereignisse erhalten die gleiche Anzahl wie das vorherige Ereignis.
	OFF	Natural-Statement-Ereignisse (NS) werden nicht gesammelt.
	COUNT	Natural-Statement-Ereignisse (NS) werden gesammelt und gezählt, was bedeutet, dass das Ereigniszählerfeld im Ereignisdatensatz mit jedem Statement-Ereignis erhöht wird. Diese Option kann zu einer schlechteren Performance führen. Siehe *Profiler-Performance im Batch-Modus*.
FNAT		Gibt an, ob Ereignisdaten aufgezeichnet werden, während Natural-Systemprogramme ausgeführt werden.
	ON	Ereignisdaten werden aufgezeichnet, während Natural-Systemprogramme ausgeführt werden.
	OFF	Ereignisdaten werden nicht aufgezeichnet, während Natural-Systemprogramme ausgeführt werden.

Standardfilterwert für die Profil-Erstellung

Standardmäßig (wenn der Ereignisfilter nicht angegeben ist) werden alle Ereignisse außer Natural-Statement-Ereignissen gesammelt und Ereignisdaten von Natural-Systemprogrammen werden nicht aufgezeichnet:

```
FILTER=EVENT            /* Set event filter
  EVENT=ALL             /* All events
  STATEMENT=OFF         /* Do not collect statements
  FNAT=OFF              /* No FNAT
```

Standardfilterwert für Codeabdeckung

Standardmäßig (wenn der Ereignisfilter nicht angegeben ist) werden nur Programmstart- und Natural-Statement-Ereignisse gesammelt. Ereignisdaten von Natural-Systemprogrammen werden nicht aufgezeichnet:

```
FILTER=EVENT            /* Set event filter
  EVENT=PS              /* Program start only
  STATEMENT=ON          /* Collect statements
  FNAT=OFF              /* No FNAT
```

Die Codeabdeckung funktioniert nicht korrekt, wenn Programmstart- oder Natural-Statement-Ereignisse nicht gesammelt werden. Wenn andere Ereignisse überwacht werden, werden sie im Trace angezeigt, aber von der Codeabdeckung ignoriert.

Beispiel für einen Ereignisfilter

```
FILTER=EVENT            /* Set event filter
  EVENT=D PS PR         /* Database and program start/resume events
  STATEMENT=ON          /* Collect statements (no count)
  FNAT=OFF              /* No FNAT
```

Programmfilter

Der Profiler-Programmfilter gibt die Libraries, Programme (Natural-Objekte) und Programmzeilen an, für die Ereignisdaten gesammelt werden. Standardmäßig (wenn der Programmfilter nicht angegeben ist) werden die Daten aller Libraries, Programme und Zeilen gesammelt.

Syntax des Programfilters:

```
FILTER=PROGRAM
[LIBRARY=library-name]
[PROGRAM=program-name]
[LINE-FROM={0|start-number}
[LINE-TO={0|end-number}
```

Syntax-Beschreibung:

Schlüsselwort für PROGRAM	Wert	Beschreibung
LIBRARY	*library-name*	Nur die angegebene Library wird überwacht. Wenn die Spezifikation mit einem Stern (*) endet, werden alle Libraries mit dem entsprechenden Präfix überwacht. Standardeinstellung: alle Libraries
PROGRAM	*program-name*	Nur das angegebene Programm wird überwacht. Wenn die Spezifikation mit einem Stern (*) endet, werden alle Programme mit dem entsprechenden Präfix überwacht. Standardeinstellung: alle Programme

Schlüsselwort für PROGRAM	Wert	Beschreibung
LINE-FROM	*start-number*	Nur Zeilen mit einer Zeilennummer, die größer oder gleich der angegebenen Nummer ist, werden überwacht. Gültige Werte für *start-number*: 0 bis 9999
LINE-TO	*end-number*	Nur Zeilen mit einer Zeilennummer, die kleiner oder gleich der angegebenen Nummer ist, werden überwacht. Wenn die Zahl 0 ist, wird die maximale Zeilennummer verwendet. Gültige Werte für *end-number*: 0 bis 9999

Beispiel für einen Programmfilter:

Das folgende Beispiel überwacht die Zeilen 0500 bis 2000 in allen Natural-Objekten, die in der Library PRFDEMO mit X beginnen.

```
FILTER=PROGRAM             /* Set program filter
  LIBRARY=PRFDEMO          /* Monitored library
  PROGRAM=X*               /* Monitored program
  LINE-FROM=500            /* Monitor from line 0500
  LINE-TO=2000             /* Monitor to line 2000
```

Zählfilter

Der Profiler-Zählfilter (Count Filter) gibt die Ereigniszähler an, für die Daten gesammelt werden. Standardmäßig (wenn der Zählfilter nicht angegeben ist) werden Daten für jede Ereignisanzahl gesammelt.

Wenn der Ereignisfilter STATEMENT=ON gesetzt ist, kann sich der Zählfilter nur auf Nicht-Statement-Ereignisse beziehen, da Statement-Ereignisse keine eindeutige Anzahl erhalten. Die Nicht-Statement-Ereignisse haben die gleiche Anzahl wie bei STATEMENT=OFF und in beiden Fällen kann der gleiche Zählfilter verwendet werden.

Syntax des Zählfilters:

```
FILTER=COUNT
[FROM={0|minimum-count}
[TO={0|maximum-count}
```

Syntax-Beschreibung:

Schlüsselwort für COUNT	Wert	Beschreibung
FROM	*minimum-count*	Es werden nur Ereignisse bei einem Ereigniszähler überwacht, der größer oder gleich der angegebenen Zahl ist. Gültige Werte für *minimum-count*: 0 bis 2147483647
TO	*maximum-count*	Es werden nur Ereignisse bei einem Ereigniszähler überwacht, der kleiner oder gleich der angegebenen Zahl ist. Wenn die Zahl 0 ist, wird der maximale Ereigniszähler verwendet. Gültige Werte für *maximum-count*: 0 bis 2147483647

Beispiel für einen Zählfilter

Eine Profil-Erstellung mit STATEMENT=OFF hat gezeigt, dass zwischen den Ereignissen viel CPU-Zeit mit den Zählern 1200 bis 1400 verbracht wurde. Jetzt soll dieser Bereich, einschließlich der Statements, detaillierter analysiert werden. Mit STATEMENT=COUNT würden auch die Statements gezählt und die Ereignisse würden andere Zählerwerte erhalten. Aber mit STATEMENT=ON werden die Statements nicht gezählt und die Zählerwerte können verwendet werden, um die Datensammlung einzuschränken.

```
FILTER=COUNT              /* Set count filter
  FROM=1200               /* Monitor from event count 1200
  TO=1400                 /* Monitor to event count 1400
```

Zeitfilter

Der Profiler-Zeitfilter gibt die CPU-Zeiten (Einheit: 1/100 Sekunden) an, für die Daten gesammelt werden. Standardmäßig (wenn der Zeitfilter nicht angegeben ist) werden Daten für jede CPU-Zeit gesammelt.

Syntax des Zeitfilters:

```
FILTER=TIME
[FROM={0|minimum-time}
[TO={0|maximum-time}
```

Syntax-Beschreibung:

Schlüsselwort für TIME	Wert	Beschreibung
FROM	*minimum-time*	Es werden nur Ereignisse mit einer CPU-Zeit überwacht, die größer oder gleich der gegebenen Anzahl an 1/100 Sekunden ist. Gültige Werte für *minimum-time*: 0 bis 2147483647
TO	*maximum-time*	Es werden nur Ereignisse mit einer CPU-Zeit überwacht, die kleiner oder gleich der gegebenen Anzahl an 1/100 Sekunden ist. Wenn die Zahl 0 ist, wird die maximale CPU-Zeit verwendet. Gültige Werte für *maximum-time*: 0 bis 2147483647

Beispiel für einen Zählfilter:

In dem folgenden Beispiel werden alle Ereignisse überwacht, die nach einer Sekunde für eine Dauer von zwei Sekunden auftreten:

```
FILTER=TIME             /* Set time filter
  FROM=100              /* Monitor from CPU second 1.00
  TO=300                /* Monitor to CPU second 3.00
```

Stichprobennahme (Sampling) aktivieren

Bei der Sampling-Methode wird ein statistischer Ansatz verwendet, bei dem Daten durch Stichprobennahme gesammelt werden. Das Sampling reduziert die Menge der in die Ressourcendatei geschriebenen Daten erheblich und behält annähernd die gleichen CPU-Zeiten bei wie ohne Sampling.

Die Stichprobennahme steht bei der Profil-Erstellung (Profiling), aber nicht bei der Messung der Codeabdeckung (Code Coverage) zur Verfügung.

Allgemeine Informationen zur Stichprobennahme finden Sie unter ***Stichprobentechnik (Natural Profiler Sampling)*** im Kapitel *Grundlegendes Konzept der Profiler Utility*.

Um das Sampling zu aktivieren:

- Geben Sie die folgenden untergeordneten Schlüsselwörter ein, die der INIT-Funktion der Profiler Utility zugeordnet sind:

```
SAMPLING=ON
INTERVAL=nnn
```

Dabei ist *nnn* das Stichprobenintervall in Mikrosekunden ist.

Anmerkung: Standardmäßig (wenn `SAMPLING` nicht angegeben ist) werden keine Stichproben der Daten genommen. Wenn `SAMPLING=ON` angegeben ist, aber kein `INTERVAL`, beträgt das Standard-Stichprobenintervall 100 Mikrosekunden.

Benutzerdefinierte Ereignisse schreiben

Ein benutzerdefiniertes Ereignis kann aus einem Natural-Programm mithilfe des folgenden Natural-Statement generiert werden:

```
CALL 'CMRDC' 'U' USER-DATA
```

Das erste Zeichen der `USER-DATA` wird als Subtyp des benutzerdefinierten Ereignisses behandelt. Die restlichen Zeichen sind der Text des benutzerdefinierten Ereignisses. Weitere Informationen finden Sie unter *Benutzerdefinierte Ereignisse* in der *SYSRDC Utility*-Dokumentation.

Anmerkungen:

1. Der Natural Profiler im Batch-Modus bietet eine Funktion, mit der Sie benutzerdefinierte Ereignisse aus dem JCL-Eingabestrom generieren können. Diese benutzerdefinierten Ereignisse werden immer unabhängig von den aktuellen Filtereinstellungen aufgezeichnet. Wenn die Profil-Erstellung angehalten wird, aktiviert die Profiler Utility die Profil-Erstellung, bevor sie das benutzerdefinierte Ereignis schreibt, und deaktiviert sie anschließend. Wenn die Probenahme aktiv ist, kann es vorkommen, dass ein benutzerdefiniertes Ereignis herausgefiltert wird.
2. Der Profiler muss initialisiert werden, bevor ein benutzerdefiniertes Ereignis geschrieben werden kann.

Die Funktion `FUNCTION=TEXT` der Profiler Utility wird verwendet, um benutzerdefinierte Ereignisse zu schreiben. Es gilt die folgende Syntax:

```
FUNCTION=TEXT
[TEXT=text]
[TYPE=character]
```

Syntax-Beschreibung:

Schlüsselwort für TEXT	Wert	Beschreibung
TEXT	*text*	Der Text *text* wird dem Profiler-Trace als benutzerdefiniertes Ereignis hinzugefügt. Wenn mehrere TEXT-Schlüsselwörter angegeben sind, werden die entsprechenden Werte verkettet. Die maximale Textgröße beträgt 249 Bytes. Zusätzliche Zeichen werden abgeschnitten. Standardwert: keiner (Leerzeichen)
TYPE	*character*	Das alphanumerische Zeichen *character* gibt den Subtyp des benutzerdefinierten Ereignisses an. Der Subtyp ist Teil der ereignisspezifischen Daten. Der Ereignistyp eines benutzerdefinierten Ereignisses ist immer U gefolgt von einem Leerzeichen. Standardwert: Leerzeichen

Verwenden Sie die folgende Ereignisfiltereinstellung, wenn Sie nur die benutzerdefinierten Ereignisse überwachen möchten, die mit der TEXT-Funktion geschrieben wurden. Zusätzlich zu den TEXT-Funktionseinträgen werden auch die Session- und Pausenereignisse überwacht. Alle anderen Ereignisse, einschließlich benutzerdefinierter Ereignisse, die durch Aufrufe von CMRDC geschrieben werden, werden herausgefiltert.

```
* Monitor only TEXT function entries
FILTER=EVENT/* Set event filter
  EVENT=                /* No events
  STATEMENT=OFF         /* No statements
  FNAT=OFF              /* No FNAT
```

Beispiel für ein benutzerdefiniertes Ereignis

Das folgende Beispiel schreibt ein benutzerdefiniertes Ereignis mit dem Subtyp J und dem Text Start profiling in das Profiler-Ereignis-Trace.

```
FUNCTION=TEXT            /* Write a user-defined event
  TEXT='Start profiling' /* User-defined event text
  TYPE='J'               /* User-defined event subtype
```

Monitor-Session CMPRMIN

Standardmäßig verwendet die Profiler-Monitor-Session die gleichen dynamischen Natural-Parameter wie die Trace-Sitzung, die mit dem CMPRMIN-Eingabe-Dataset angegeben wurde. Ausnahme: Der Profilparameter RDCSIZE (Puffergröße für den Natural Data Collector) ist für die Monitor-Session auf Null (0) gesetzt. Mit dem CMPRMIN-Schlüsselwort der Profiler Utility-Funktion INIT oder COVERAGE kann ein separater dynamischer Parametereingabe-Dataset für die Profiler-Monitor-Session definiert werden. Wenn Sie den separaten Eingabe-Dataset für dynamische Parameter verwenden, beachten Sie Folgendes:

- Geben Sie für die Monitor-Session nur die Natural-Parameter an, die erforderlich sind. Geben Sie keine Parameter an, die für die Anwendungsausführung erforderlich sind (z. B. `RPC`).
- Geben Sie die Parameter `RDCSIZE` und `RDCEXIT` (User Exits für den Natural Data Collector) nur für die Trace-Session an. Die Angabe dieser Parameter für die Monitor-Session führt zu unnötigen Exit-Aufrufen und einer schlechteren Performance.
- Geben Sie den Parameter `PDPSIZE` (Profiler-Daten-Pool-Größe) nur für die Monitor-Session an. Jede Spezifikation im Eingabe-Dataset für dynamische Parameter der Trace-Session wird ignoriert.

Wenn die standardmäßige Einstellung des Natural-Profilparameters `ETID` (Adabas-Benutzerkennung) verwendet wird, kann es vorkommen, dass der folgende Fehler von der Profiler-Monitor-Session empfangen wird:

`NAT3048 Error during Open processing. DB/Subcode` *`nn`*`/8 - ETID=`*`job-name`*`..` (Fehler während der `Open`-Kommandoverarbeitung).

Verwenden Sie in diesem Fall die folgenden Parameter für `CMPRMIN`:

```
ETID=' ',DBCLOSE=ON
```

Alternativ können Sie einen `ETID`-Wert verwenden, der sich vom Jobnamen im separaten dynamischen Parametereingabe-Dataset für die Profiler-Monitor-Session unterscheidet.

Um einen dynamischen Parametereingabe-Dataset für die Profiler-Monitor-Session zu definieren:

- Geben Sie das folgende untergeordnete Schlüsselwort ein, das der Profiler Utility-Funktion `INIT` oder `COVERAGE` zugeordnet ist:

  ```
  CMPRMIN=data-set
  ```

  Dabei ist *`data-set`* der Name des dynamischen Parameter-Eingabe-Dataset für die Profiler-Monitor-Session.

Beispiel für z/OS:

```
//CMSYNIN  DD  *
PROFILER
FUNCTION=INIT             /* Initialize profiling
  CMPRMIN=PRFPARMS        /* Monitor session parameter
  ...
/*
//* Trace Session Parameters
//CMPRMIN  DD   *
RDCSIZE=2,RDCEXIT=NATRDC1,...
/*
//* Monitor Session Parameters
//PRFPARMS DD   *
```

```
ETID=PROFILER,PDPSIZE=10000,...
/*
```

Monitor-Session CMSYNIN

Im Allgemeinen benötigt die Profiler-Monitor-Session keinen primären Kommandoeingabe-Dataset. Mit dem `CMSYNIN`-Schlüsselwort der Profiler Utility-Funktion `INIT` oder `COVERAGE` kann ein separater primärer Kommandoeingabe-Dataset für die Profiler-Monitor-Session definiert werden. Dies ist erforderlich, wenn Ihre Anwendung unter der Kontrolle von Natural Security läuft und Sie keine automatische Anmeldung wünschen, d.h. der Natural-Profilparameter `AUTO` (Automatische Anmeldung) ist auf `OFF` gesetzt. Geben Sie die gewünschten Anmeldeeinstellungen im primären Kommandoeingabe-Dataset für die Profiler-Monitor-Session an, gefolgt vom Profiler-Kommando, das die Profiler-Monitor-Session startet.

Geben Sie für das Profiling Folgendes an:

```
library-name,user-id,password
PROFILER MONITOR
```

Geben Sie für die Codeabdeckung Folgendes an:

```
library-name,user-id,password
PROFILER COVMON
```

Anmerkung: Wenn die obige Kommando-Syntax nicht angewendet wird, tritt der folgende Fehler auf: `Profiler Error-1 => PRBINIT : Profiler INIT function - No response from monitor session after 0.3 seconds.` (Profiler INIT-Funktion - Keine Antwort von der Monitor-Session nach 0,3 Sekunden.)

Um einen primären Kommandoeingabe-Dataset für die Profiler-Monitor-Session zu definieren:

- Geben Sie das folgende untergeordnete Schlüsselwort ein, das der Profiler Utility-Funktion `INIT` oder `COVERAGE` zugeordnet ist:

  ```
  CMSYNIN=data-set
  ```

  Dabei ist *data-set* der Name des primären Kommandoeingabe-Parameter-Dataset für die Profiler-Monitor-Session.

Beispiel für z/OS (mit AUTO=OFF gesetzt):

```
//CMSYNIN  DD  *
library-name,user-id,password
PROFILER
FUNCTION=INIT             /* Initialize profiling
  CMSYNIN=PRFSYNIN        /* Monitor session input data
  ...
/*
//* Monitor Session Input Data
//PRFSYNIN DD   *
library-name,user-id,password
PROFILER MONITOR
/*
```

Batch-Natural RPC-Server profilieren

Die Profil-Erstellung oder die Messung der Codeabdeckung eines Batch-Natural RPC-Servers erfordert, dass das Systemkommando `PROFILER` und die Profiler-Eingabedaten mit dem Natural-Profilparameter `STACK` im RPC-Server-Job eingegeben werden. Die Profiler-Eingabe muss im zweiten Syntaxformat erfolgen (ohne Gleichheitszeichen und Kommas). Siehe auch ***Profiler Utility-Syntax*** in ***Syntax und Schlüsselwörter***.

Anmerkung: Unter z/OS dürfen Sie die Language Environment-Option `POSIX(ON)` nicht verwenden. Wenn Sie SSL für die Kommunikation mit dem EntireX Broker verwenden möchten, müssen Sie Application Transparent Transport Layer Security (AT-TLS) verwenden.

Um die Profil-Erstellung eines Batch Natural RPC-Servers zu starten:

- Geben Sie das folgende Schlüsselwort ein, bevor die Profiler-Funktion `INIT` oder `COVERAGE` ausgeführt wird:

```
RPC
```

Das Schlüsselwort `RPC` zeigt dem Profiler an, dass ein Natural-RPC-Server überwacht wird. Der Profiler überschreibt das Schlüsselwort `WAIT-EMPTY` der Initialisierung mit dem Wert Null (`0`), so dass die Profil-Erstellung immer fortgesetzt wird, wenn der Profiler-Daten-Pool leer ist und die Trace-Session noch aktiv ist.

Es wird empfohlen, einen separaten dynamischen Parametereingabe-Dataset (Monitor-Session `CMPRMIN`) für die Profiler-Monitor-Session zu verwenden, wenn Sie einen Batch Natural RPC-Server profilieren. Geben Sie den `RPC`-Parameter in diesem Dataset nicht an. Siehe ***Monitor-Session CMPRMIN***.

Die Ereignisdaten des Batch Natural RPC-Servers werden in eine Profiler-Ressourcendatei geschrieben. Der Name und die Library der Ressourcendatei können mit Profiler-Schlüsselwörtern angegeben werden (siehe *Profilierung initialisieren*). Wir empfehlen Ihnen, den Natural RPC-Server zu stoppen, bevor der Profiler die Ressourcendatei weiterverarbeitet.

Wenn Sie ein Profil eines Natural-RPC-Servers erstellen, können Sie die Datensammlung mit den für die Profiler Utility bereitgestellten Programmen starten und anhalten.

Um die Profiler-Datensammlung in einer Client/Server-Umgebung zu starten und anzuhalten:

1 Kopieren Sie das Programm `PRFFCT` aus der System-Library `SYSPRFLR` in die Client-Library und die Anwendungsprogrammierschnittstelle `USR8210N` aus der System-Library `SYSEXT` in die Server-Library.

2 Führen Sie das Programm `PRFFCT` in der Client-Library aus.

`PRFFCT` erwartet als Eingabe den Wert `START` oder `PAUSE`, um die entsprechende Funktion auszuführen. Wenn Sie den Wert `STATE` eingeben, wird der aktuelle Status der Datensammlung angezeigt.

Weitere Informationen siehe *Datensammlung starten und anhalten*.

Beispiel für Natural RPC Batch-Profilierung

Das folgende Beispiel für z/OS zeigt die Profiler-Eingabedaten für einen Batch Natural RPC-Server, der mit dem Natural-Profilparameter `STACK` angegeben wurde:

```
STACK=(
PROFILER RPC:
TRACE:3:
FUNCTION:INIT:
  TRACE-EVENT:OFF:
  TRACE-MON:3:
  CMPRMIN:CMPRMINX:
  RESOURCE:ON:
  RESOURCE-NAME:RPCTEST:
  RESOURCE-LIB:PRFDATA:
  REPLACE:YES:
FILTER:EVENT:
  EVENT:ALL:
  STATEMENT:ON:
  FNAT:OFF:
END-PROFILER;
LOGON PRFDEMO
)
```

Profil-Erstellung einer Großrechner-Session aus Natural Studio

Sie können die Profiler Utility im Batch-Modus benutzen, um ein Profil einer Mainframe-Anwendung zu erstellen oder die Codeabdeckung für eine Großrechner-Anwendung auszuführen, die remote von Natural Studio auf einem Natural Development Server ausgeführt wird.

Dazu müssen Sie das Systemkommando `PROFILER` und die `PROFILER`-Eingabedaten mit dem dynamischen Natural-Profilparameter `STACK` angeben, wenn Sie Ihre Remote-Großrechner-Umgebung mappen. Die Profiler-Eingabe muss im zweiten Syntaxformat eingegeben werden (ohne Gleichheitszeichen und Kommas). Siehe auch ***Profiler Utility-Syntax*** in ***Syntax und Schlüsselwörter***.

Sie können das Profiler-Schlüsselwort `INCLUDE` verwenden, um die Profiler-Eingabe aus einem Natural-Text-Objekt zu lesen und so die mit dem `STACK`-Parameter eingegebene Datenmenge zu reduzieren. Ein Beispiel finden Sie im Textobjekt `XNDV`, das in der System-Library `SYSPRFLR` ausgeliefert wird. In diesem Beispiel wird die Profil-Erstellung initialisiert und sofort die Überwachung gestartet.

Der Profiler überschreibt automatisch das Schlüsselwort `WAIT-EMPTY` der Initialisierung mit dem Wert Null (`0`), so dass die Profil-Erstellung immer fortgesetzt wird, wenn der Profiler-Daten-Pool leer ist und die Trace-Session noch aktiv ist.

Die Ereignisdaten der Natural Studio-Großrechner-Session werden in eine Profiler-Ressourcendatei geschrieben. Der Name und die Library der Ressourcendatei können mit Profiler-Schlüsselwörtern angegeben werden (siehe ***Profilierung initialisieren*** oder ***Initialisierung der Codeabdeckung (Code Coverage)***). Wir empfehlen Ihnen, die Großrechner-Sitzung zu trennen, bevor Sie die Ressourcendatei auswerten.

Wenn Sie ein Profil einer Natural Studio-Großrechner-Sitzung erstellen, können Sie die Datensammlung mit den für die Profiler Utility mitgelieferten Programmen **starten und anhalten** (siehe entsprechenden Abschnitt).

Um die Profiler-Datenerfassung in einer Natural Studio-Großrechner-Sitzung zu starten oder anzuhalten:

1. Melden Sie sich bei der Library `SYSPRFLR` an.
2. Führen Sie das Programm `PRFSTART` aus, um die Sammlung zu starten.
3. Führen Sie das Programm `PRFPAUSE` aus, um die Sammlung anzuhalten.

Regeln und Einschränkungen

Die folgenden Regeln und Einschränkungen gelten für die Profil-Erstellung einer Großrechner-Session aus Natural Studio:

- Die Profil-Erstellung mit der Profiler Utility im Batch-Modus funktioniert nicht, wenn Programme remote von NaturalONE ausgeführt werden. Verwenden Sie den NaturalONE Profiler, wenn Sie Großrechner-Programme profilieren möchten, die in einer NaturalONE-Umgebung ausgeführt werden.
- Die Profil-Erstellung einer Natural Studio-Großrechner-Session mit der Profiler Utility im Batch-Modus funktioniert nicht, wenn der Natural Development Server einen CICS-Adapter verwendet.
- Wenn die Profil-Erstellung mit dem `STACK`-Parameter gestartet wird, verwenden Sie die Profiler-Einstellung `TRACE=0`.
- Das Monitor-Tracing (`TRACE-MONITOR`) der Profiler-Session wird in die Natural Development Server-Ausgabe geschrieben.

Beispiel für STACK für Mainframe Großrechner-Profiling

Das folgende Beispiel enthält eine `STACK`-Parameterangabe für eine Profiler-Dateneingabe, die dynamisch beim Mappen auf eine Remote-Großrechner-Umgebung von Natural Studio gesetzt wird:

```
STACK=(PROFILER RESOURCE-LIB:PRFDATA:INCLUDE:XNDV)
```

Die Ressource wird in die Library `PRFDATA` geschrieben.

Ereignisdaten konsolidieren

Mit der Profiler Utility-Funktion `CONSOLIDATE` können Sie Ereignisdaten konsolidieren.

Beachten Sie, dass die **Natural Profiler Rich GUI-Schnittstelle** ebenfalls eine Funktion zur Konsolidierung von Profiler-Ereignisdaten bietet.

Allgemeine Informationen zur Datenkonsolidierung finden Sie unter ***Datenkonsolidierung*** im Kapitel *Grundlegendes Konzept der Profiler Utility*.

Syntax von `CONSOLIDATE`:

```
FUNCTION=CONSOLIDATE
[RESOURCE={ON|OFF}]
[RESOURCE-NAME=resource-name]
[RESOURCE-LIB=library-name]
[REPLACE={YES|NO}]
[TRANSACTION={ON|OFF}]
[IO-TIME={ON|OFF}]
[EXPORT={ON|OFF}]
[FORMAT={TEXT|COMMA|SEMICOLON}]
[TRACE-EVENT={ON|OFF}]
[TRACE-CONSOLIDATE={ON|OFF}]
```

Syntax-Beschreibung:

Schlüsselwort für CONSOLIDATE	Wert	Beschreibung
RESOURCE		Gibt an, ob die konsolidierten Ereignisdaten in eine konsolidierte Natural Profiler-Ressourcendatei (NPRC) geschrieben werden.
	ON	Die konsolidierten Ereignisdaten werden in eine NPRC-Ressourcendatei geschrieben.
	OFF	Die konsolidierten Ereignisdaten werden nicht in eine NPRC-Ressourcendatei geschrieben. Diese Einstellung ist nützlich, wenn Sie das Ereignis-Trace oder die Statistikdaten drucken oder die Daten exportieren möchten und die konsolidierte NPRC-Ressourcendatei nicht benötigen.
RESOURCE-NAME	*resource-name*	Der Name der Natural Profiler-Ressourcendatei (NPRF), die Sie konsolidieren möchten. Die Dateierweiterung `.nprf` wird automatisch hinzugefügt. Standardwert: Der Name der zuletzt erstellten NPRF-Ressourcendatei in der Library Bei `RESOURCE=ON` werden die konsolidierten Daten in eine NPRC-Ressourcendatei mit demselben Ressourcennamen geschrieben.
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, die die NPRF-Ressourcendatei enthält, die Sie konsolidieren möchten. Standardwert: Der Name der aktuellen Library. Diese Library wird auch als Ziel-Library für die konsolidierte NPRC-Ressourcendatei verwendet.
REPLACE		Gibt an, ob eine vorhandene NPRC-Ressourcendatei ersetzt wird.

Schlüsselwort für CONSOLIDATE	Wert	Beschreibung
	YES	Ersetzen einer vorhandene NPRC-Ressourcendatei mit demselben Namen.
	NO	Kein Ersetzen einer vorhandene NPRC-Ressourcendatei mit demselben Namen. Eine Nachricht wird zurückgegeben, wenn bereits eine Ressourcendatei mit demselben Namen vorhanden ist. In diesem Fall wird keine Konsolidierung durchgeführt.
TRANSACTION		Gibt an, ob Transaktionskennungen zu den konsolidierten Ereignisdaten hinzugefügt werden.
	ON	Transaktionskennungen werden zu den konsolidierten Ereignisdaten hinzugefügt. Weitere Informationen siehe *Transaktionszusammenfassung*. **Anmerkung:** Die generierte NPRC-Ressourcendatei benötigt mehr Speicherplatz, wenn ON eingestellt ist.
	OFF	Transaktionskennungen werden den konsolidierten Ereignisdaten nicht hinzugefügt.
IO-TIME		Gibt an, ob die E/A-Zeit (IB-Ereignis) und die Natural-RPC-Client-Zeit (RW-Ereignis) in den konsolidierten Daten enthalten sind.
	ON	E/A- und Natural-RPC-Client-Zeit sind in den konsolidierten Daten enthalten.
	OFF	E/A- und Natural-RPC-Client-Zeit sind nicht in den konsolidierten Daten enthalten. Diese Einstellung ist nützlich, wenn die Ereignisdaten aus der Profiler Utility im Online-Modus stammen und Sie planen, die Performance in NaturalONE oder mit der Profiler Utility-READ-Funktion zu analysieren. Für eine Performance-Analyse mit der Natural Profiler Rich GUI-Schnittstelle ist diese Einstellung nicht erforderlich, da diese eine ähnliche Funktion bietet.
EXPORT		Gibt an, ob die konsolidierten Ereignisdaten in die Arbeitsdatei (Work File) 7 geschrieben werden.
	ON	In die Arbeitsdatei 7 schreiben.
	OFF	Nicht in die Arbeitsdatei 7 schreiben.
FORMAT		Gibt das Format an, in dem die exportierten Daten in Arbeitsdatei 7 geschrieben werden.
	TEXT	Daten im Freitextformat schreiben.
	COMMA	Daten im CSV-Format mit einem Komma (,) als Trennzeichen schreiben.

Schlüsselwort für CONSOLIDATE	Wert	Beschreibung
	SEMICOLON	Daten im CSV-Format mit einem Semikolon (;) als Trennzeichen schreiben.
TRACE-EVENT		Gibt an, ob das Profiler-Ereignis-Trace in die Standardausgabe geschrieben wird. Siehe *Ereignis-Trace*.
	ON	Profiler-Ereignis-Trace schreiben.
	OFF	Profiler-Ereignis-Trace nicht schreiben.
TRACE-CONSOLIDATE		Gibt an, ob der Profiler-Konsolidierungs-Trace in die Standardausgabe geschrieben wird. Siehe *Konsolidierungs-Trace*.
	ON	Profiler-Konsolidierungs-Trace schreiben.
	OFF	Profiler-Konsolidierungs-Trace nicht schreiben.

Beispiel für eine Konsolidierung

Im folgenden Beispiel wird die Profiler-Ressource `Test.nprf` in der Library `PRFDATA` konsolidiert und die konsolidierten Daten werden in die Profiler-Ressource Test.nprc geschrieben. E/A- und Natural-RPC-Client-Zeit sind in den konsolidierten Daten enthalten.

Darüber hinaus werden die konsolidierten Daten im CSV-Format (semikolongetrennte Werte) in Arbeitsdatei 7 geschrieben.

Die Ereignis- und Konsolidierungs-Traces sind ausgeschaltet.

```
FUNCTION=CONSOLIDATE    /* Consolidate Profiler data
  RESOURCE=ON           /* Write to resource
  RESOURCE-NAME='Test'  /* Resource name
  RESOURCE-LIB=PRFDATA  /* Resource library
  REPLACE=YES           /* Replace resource
  TRANSACTION=OFF       /* Do not add transaction identifiers
  IO-TIME=ON            /* Include I/O and RPC client times
  EXPORT=ON             /* Write to Work File 7
  FORMAT=S              /* CSV format with semicolon separator
  TRACE-EVENT=OFF       /* No event trace
  TRACE-CONSOLIDATE=OFF /* No consolidation trace
```

Ereignisdaten auswerten

Wenn das Profil einer Natural-Anwendung erstellt wird, schreibt die Natural Profiler Utility die Ereignisdaten in eine NPRF-Ressourcendatei. Konsolidierte Daten werden in einer NPRC-Ressourcendatei und Abdeckungsdaten in einer NCVF-Ressourcendatei gespeichert. Die `READ`-Funktion der Profiler Utility liest und wertet die Profiler-Ressourcendaten aus und schreibt die Ergebnisse in die Standardausgabe oder in eine Natural-Arbeitsdatei. Die durchgeführten Auswertungen hängen, wie in der folgenden Tabelle beschrieben, vom Typ der gelesenen Ressourcendatei ab:

Typ der Ressourcendatei	Auswertung	Beschreibung
NPRF	Ereignis-Trace	Chronologische Liste der Profiler-Ereignisdaten.
	Programm-Trace	Programmablauf der profilierten Anwendung.
	Statistik	Statistik der Profil-Erstellung und der NPRF-Ressourcendatei.
NPRC	Konsolidierungs-Trace	Liste der konsolidierten Daten mit Trefferzahlen und zusammengefasster verstrichener (elapsed) Zeit und CPU-Zeit.
	Programmzusammenfassung	Tabelle der ausgeführten Natural-Objekte. Die Tabelle zeigt, welche Ereignisse während der Objektausführung aufgetreten sind und welche CPU-Zeit für die Ausführung des Objekts aufgewendet wurde.
	Zeilenzusammenfassung	Tabelle der ausgeführten Natural-Quellcode-Zeilen. Die Tabelle zeigt, wie viele Ereignisse während der Zeilenausführung aufgetreten sind und wie viel CPU- und verstrichene (elapsed) Zeit bei der Ausführung der Zeile aufgewendet wurde.
	Transaktionszusammenfassung	Tabelle der ausgeführten Transaktionen. Die Tabelle zeigt, wie viele Ereignisse während der Transaktionsausführung aufgetreten sind und die Antwortzeit (verstrichene (elapsed) Zeit) der Transaktionen.
	Statistik	Statistikdaten des Profiling, der Konsolidierung und der NPRC-Ressourcendatei.
NCVF	Statement-Abdeckung	Liste der in den Quellcode-Zeilen abgedeckten Statements. Die Liste zeigt den Prozentsatz der Statement-Abdeckung für jede Statement-Zeile in dem Quellcode der aufgerufenen Programme.

Typ der Ressourcendatei	Auswertung	Beschreibung
	Programmabdeckung	Tabelle der Codeabdeckungsergebnisse ausgeführter Natural-Objekte. Die Programmabdeckungstabelle listet alle Natural-Objekte auf, die während des Abdeckungslaufs ausgeführt wurden. Für jedes Objekt zeigt es den Prozentsatz der Abdeckung, die Anzahl der abgedeckten und nicht abgedeckten („missed“) Statements und die Gesamtzahl der Statements.
	Statistik	Statistikdaten für Profil-Erstellung, Abdeckung und die NCVF-Ressourcendatei.

Dieser Abschnitt behandelt die folgenden Themen:

- Lesefunktion (READ) der Profiler Utility
- Beispiel für READ
- Ereignis-Trace
- Konsolidierungs-Trace
- Programm-Trace
- Programmzusammenfassung
- Zeilenzusammenfassung
- Transaktionszusammenfassung
- Programmabdeckung
- Statement-Abdeckung
- Microsoft Excel-Vorlage zur Visualisierung der Abdeckungsergebnisse benutzen
- Statistiken

Lesefunktion (READ) der Profiler Utility

Die `READ`-Funktion der Profiler Utility liest und wertet die Ressourcendaten aus.

Syntax von `READ`:

```
FUNCTION=READ
[RESOURCE-NAME=resource-name]
[RESOURCE-LIB=library-name]
[RESOURCE-TYPE={NPRF|NPRC|NCVF}]
[EVENT={ON|OFF}]
[PROGRAM={ON|OFF}]
[LINE={ON|OFF}]
[TRANSACTION={ON|OFF}]
[STATISTICS={ON|OFF}]
```

```
[PRINT={ON|OFF}]
[EXPORT={ON|OFF}]
[FORMAT={TEXT|COMMA|SEMICOLON}]
```

Syntax-Beschreibung:

Schlüsselwort für READ	Wert	Beschreibung
RESOURCE-NAME	*resource-name*	Der Name der NPRF-, NPRC- oder NCVF-Ressourcendatei, die Sie lesen möchten. Wenn keine Dateierweiterung angegeben ist, wird die mit dem Schlüsselwort RESOURCE-TYPE angegebene Erweiterung automatisch hinzugefügt. Standardwert: Der Name der zuletzt erstellten NPRF-, NPRC- oder NCVF-Ressourcendatei in der Library, abhängig von der RESOURCE-TYPE-Angabe.
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, die die NPRF-, NPRC- oder NCVF-Ressource enthält, die Sie lesen möchten. Standardwert: Der Name der aktuellen Library.
RESOURCE-TYPE		Gibt den Standard-Ressourcentyp (Dateierweiterung) an, der verwendet werden soll, wenn keine Erweiterung mit RESOURCE-NAME angegeben ist.
	NPRF	Der Standard-Ressourcentyp ist NPRF mit der Erweiterung .nprf.
	NPRC	Der Standard-Ressourcentyp ist NPRC mit der Erweiterung .nprc.
	NCVF	Der Standard-Ressourcentyp ist NCVF mit der Erweiterung .ncvf.
EVENT		Gibt an, ob der Natural Profiler Ereignisse auswertet. Siehe auch ***Ereignis-Trace***, ***Konsolidierungs-Trace*** und ***Statement-Abdeckung***.
	ON	NPRF: Das Natural Profiler-Ereignis-Trace schreiben. NPRC: Das Natural Profiler-Konsolidierungs-Trace schreiben. NCVF: Das Ergebnis des Statement-Abdeckung-Trace schreiben.
	OFF	Ereignisse nicht auswerten.
PROGRAM		Gibt an, ob der Natural Profiler Programme auswertet. Siehe auch ***Programm-Trace***, ***Programmzusammenfassung*** und ***Programmabdeckung***.
	ON	NPRF: Das Natural Profiler-Programm-Trace schreiben. NPRC: Die Natural Profiler-Programmzusammenfassung schreiben. NCVF: Die Programmabdeckungstabelle schreiben.
	OFF	Programme nicht auswerten.

Schlüsselwort für READ	Wert	Beschreibung
LINE		Diese Option ist nur für NPRC-Ressourcen verfügbar. Gibt an, ob der Natural Profiler ausgeführte Quellcodezeilen auswertet. Siehe auch *Zeilenzusammenfassung*.
	ON	Die Natural Profiler-Zeilenzusammenfassung schreiben.
	OFF	Keine ausgeführten Quellcodezeilen auswerten.
TRANSACTION		Diese Option ist nur für NPRC-Ressourcen verfügbar. Gibt an, ob der Natural-Profiler Transaktionen auswertet. Weitere Informationen siehe *Transaktionszusammenfassung*.
	ON	Transaktionen auswerten, um eine Transaktionszusammenfassung zu generieren und transaktionsbezogene Werte in der **Programmzusammenfassung** und der **Zeilenzusammenfassung** anzuzeigen.
	OFF	Keine Transaktionen auswerten.
STATISTICS		Gibt an, ob der Natural Profiler Statistiken schreibt. Siehe auch *Profiler-Statistikdaten*. **Anmerkung:** Statistikdaten für die Natural-Codeabdeckung werden unter Linux und Windows nicht erfasst.
	ON	Statistiken schreiben.
	OFF	Keine Statistiken schreiben.
PRINT		Gibt an, ob das Ergebnis in die Standardausgabe geschrieben wird.
	ON	In die Standardausgabe schreiben.
	OFF	Nicht in die Standardausgabe schreiben.
EXPORT		Gibt an, ob die ausgewerteten Daten in die Arbeitsdatei 7 geschrieben werden.
	ON	In die Arbeitsdatei 7 schreiben.
	OFF	Nicht in die Arbeitsdatei 7 schreiben.
FORMAT		Gibt das Format an, in dem die exportierten Daten in Arbeitsdatei 7 geschrieben werden.
	TEXT	Die Daten im Freitextformat schreiben.
	COMMA	Die Daten im CSV-Format mit einem Komma (,) als Trennzeichen schreiben.
	SEMICOLON	Die Daten im CSV-Format mit einem Semikolon (;) als Trennzeichen schreiben.

Beispiel für READ

Das folgende Beispiel liest die Natural Profiler-Ressource `Test.nprf` in der Library `PRFDATA` und schreibt das Ereignis-Trace, das Programm-Trace und die Profiler-Statistiken in die Standardausgabe und in die Arbeitsdatei 7 im Textformat.

```
FUNCTION=READ            /* Read Profiler Data
  RESOURCE-NAME='Test'   /* Resource name
  RESOURCE-LIB=PRFDATA   /* Resource library
  RESOURCE-TYPE=NPRF     /* Use resource type NPRF
  EVENT=ON               /* Print event trace
  PROGRAM=ON             /* Print program trace
  STATISTICS=ON          /* Print statistics
  PRINT=ON               /* Write to standard output
  EXPORT=ON              /* Write to Work File 7
  FORMAT=TEXT            /* Export in text format
```

Ereignis-Trace

Wenn `EVENT=ON` für eine NPRF-Ressourcendatei angegeben wird, wird das Profiler-Ereignis-Trace generiert.

Das Ereignis-Trace zeigt die Daten jedes Natural-Ereignisses, das während der Ausführung der Anwendung aufgetreten ist. Das Trace kann referenziert werden, wenn detaillierte Informationen zu einem Ereignis erforderlich sind. Wenn beispielsweise während der Anwendungsausführung ein Natural-Fehler aufgetreten ist, zeigt das Ereignis-Trace die entsprechende Fehlernummer und Meldung an.

Wenn das Ereignis-Trace in die Standardausgabe (`PRINT=ON`) geschrieben oder im Textformat exportiert wird (`EXPORT=ON`, `FORMAT=TEXT`), ähnelt sie dem Ereignis-Trace, das von der Profiler-Monitor-Session geschrieben wurde, während die Anwendung profiliert wurde (siehe *Ereignis-Trace*). Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, enthalten sie alle vom Profiler bereitgestellten Datenfelder (siehe *Gesammelte Daten*).

Beispiel für ein Ereignis-Trace

Das folgende Beispiel zeigt einen Auszug aus einem Ereignis-Trace:

```
Natural Profiler Event Trace
----------------------------
     Count Time             CPU-Time (ms) Ev Lev Library  Program  Line CC-Lib   CC-Name  Statement  Local-Data
         0 10:20:58.219911         63.318   MP 003 SYSPRFD  PRBINIT  8350                  Call       Monitor pause requested
       102 10:20:58.277586         76.106   PL 000                   0000                             Execute SYSEDMD/MENU
       103 10:20:58.277591         76.139   PS 001 SYSEDMD  MENU     0000                  PgmStart   00010/02430 Type: P
       103 10:20:58.277594         76.151   NS 001 SYSEDMD  MENU     0250                  Compute    Assign/Compute/Move
       103 10:20:58.277596         76.155   NS 001 SYSEDMD  MENU     0270                  Fetch      Fetch
       104 10:20:58.277598         76.169   DB 001 SYSEDMD  MENU     0270                  Fetch      00010/02430 S1
...
```

Erläuterungen:

- Die Spalte **Count** zeigt die Nummer des Ereignisses an. Monitor Pause (`MP`)-Ereignisse und Natural Statement (`NS`)-Ereignisse werden nicht gezählt und erhalten die Nummer des vorherigen Ereignisses.
- Die Spalten **Time** und **CPU-Time** zeigen die Ereigniszeit bzw. den CPU-Zeitstempel der Ereignisausführung an.
- Das Ereignis mit der Nummer `104` ist ein Database Before (`DB`)-Ereignis, das durch ein Adabas S1-Kommando verursacht wird, das für die Datei `00010/02430` ausgegeben wurde und durch ein `FETCH`-Statement in der Zeile `0270` des Natural-Objekts `MENU` ausgelöst wurde.

Weitere Erläuterungen zu den Trace-Spalten und Ereignistypen finden Sie in den Abschnitten *Event-Trace* und *Gesammelte Ereignisse und Daten*.

Konsolidierungs-Trace

Wenn `EVENT=ON` für eine NPRC-Ressourcendatei angegeben wird, wird das Natural Profiler-Konsolidierungs-Trace generiert. Das Konsolidierungs-Trace wird auch generiert, wenn `CONSOLIDATE=ON` und `TRACE-CONSOLIDATE=ON` für die INIT-Funktion der Profiler Utility gesetzt sind oder wenn `TRACE-CONSOLIDATE=ON` für die Profiler Utility-Funktion `CONSOLIDATE` gesetzt ist.

Das Konsolidierungs-Trace zeigt allgemeine Ereignisdaten, zusammengefasste Werte der verstrichenen (elapsed) Zeit und CPU-Zeit sowie die Trefferanzahl des konsolidierten Datensatzes. Wenn zwei Trace-Einträge die gleichen allgemeinen Ereignisdaten zeigen, haben sie unterschiedliche ereignisspezifische Daten, die nicht im Konsolidierungs-Trage angezeigt werden.

Die konsolidierten Datensätze dienen als Grundlage für weitere Auswertungen wie die NaturalONE-Hot Spots oder die Datenauswertung und Programmanalyse der Profiler Rich GUI-Schnittstelle. Das Konsolidierungs-Trace kann verwendet werden, um die konsolidierten Daten zu validieren.

Wenn das Konsolidierungs-Trace in die Standardausgabe (`PRINT=ON`) geschrieben wird, ähnelt es dem Konsolidierungs-Trace, das von der Profiler-Datenkonsolidierung geschrieben wurde (siehe *Ereignisdaten konsolidieren*). Wenn die Daten exportiert werden, enthalten sie alle vom Profiler bereitgestellten konsolidierten Datenfelder.

Beispiel für ein Konsolidierungs-Trace:

Das folgende Beispiel zeigt einen Auszug aus einem Konsolidierungs-Trace:

```
Natural Profiler Consolidation Trace
------------------------------------
    Count Transact Ev User     Lev Library  Program  Line CC-Lib   CC-Name  Statement   Hit-Count  Elapsed(ms)      CPU(ms)
        1          DA PRF082D  000                   0000                                      1       75.692        0.870
        2          DA PRF082D  000                   0000                                      1        0.002        0.004
        3          DA PRF082D  000                   0000                                      1        0.006        0.025
        4          NS PRF082D  006 SYSLIBS  A82CLS   0010 SYSAOSSU C-COPYRT Reset            43        0.043        0.118
        5          NS PRF082D  006 SYSTEM   NOMSTCS  4360                   End               1        0.000        0.003
        6          PL PRF082D  006 SYSTEM   NOMSTCS  0970                   Callnat           1        0.008        0.058
        7          PL PRF082D  006 SYSTEM   NOMSTCS  1020                   Perform           1        0.004        0.017
...
```

Erläuterungen:

- Die Spalte **Count** zeigt die Nummer des konsolidierten Datensatzes.

- Die Spalte **Transact** zeigt die Transaktionskennung.

  Die Transaktionskennung beginnt mit `1` und wird mit jedem `IA`- (nach Terminal-E/A) oder `RI`-Ereignis (eingehende RPC-Nachricht) erhöht. Transaktionskennungen sind nur verfügbar, wenn die Daten mit der Option `TRANSACTION=ON` konsolidiert werden.
- Der konsolidierte Datensatz `4` zeigt, dass die `RESET`-Anweisung in der Zeile `0010` des Copycode `C-COPYRT` (im Natural-Objekt `A82CLS` enthalten) 43 Mal ausgeführt wurde und eine verstrichene (elapsed) Gesamtzeit von `0,043` Millisekunden (ms) und eine Gesamt-CPU-Zeit von `0,118` ms verbrauchte.
- Die Anwendung lief auf einer z/OS-Maschine mit zIIP (IBM System z Integrated Information Processor). Unter dieser Bedingung kann die CPU-Zeit höher sein als die verstrichene (elapsed) Zeit.

Weitere Erläuterungen zu den Trace-Spalten und Ereignistypen finden Sie in den Abschnitten *Event-Trace* und *Gesammelte Ereignisse und Daten*.

Programm-Trace

Wenn `PROGRAM=ON` für eine NPRF-Ressourcendatei angegeben wird, wird das Profiler-Programm-Trace erstellt. Das Programm-Trace zeigt den Programmfluss der profilierten Anwendung. Im Allgemeinen zeigt der Programmablauf ausschließlich Programm- und Session-Ereignisse an (eine Liste möglicher Ereignistypen finden Sie unter *Gesammelte Ereignisse und Daten*).

Wenn der Programmablauf in die Standardausgabe (`PRINT=ON`) geschrieben oder im Textformat exportiert wird (`EXPORT=ON`, `FORMAT=TEXT`), werden die Programmnamen entsprechend der Programmebene eingerückt (siehe Beispiel unten), um einen schnellen Überblick über die Anwendungsaufrufstruktur zu erhalten.

Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, werden die Programmnamen nicht eingerückt. Zusätzlich zur Ausgabe im Textformat enthalten die exportierten Daten den CPU-Zeitstempel und die zusammengefasste Adabas-Zeit.

Beispiel für einen Programm-Trace

Das folgende Beispiel zeigt einen Auszug eines Programm-Trace und die Summen der Anwendungsausführung:

```
Natural Profiler Program Trace
------------------------------
Time            Ev Library  CC-Name  Line Lev Program  Events
10:20:58.309812 PL                   0000 000
10:20:58.309817 PS SYSEDMD           0000 001 .OPTTEST  D=4 N=2
10:20:58.357694 PL SYSEDMD           5620 001 .OPTTEST
10:20:58.357704 PS SYSEDMD           0000 002 ..CALLMON3 N=3
10:20:58.385263 PL SYSEDMD           0980 002 ..CALLMON3
10:20:58.385274 PS SYSEDMD           0000 003 ...OP3DISC  D=3 N=4
10:20:58.412207 PL SYSEDMD           1670 003 ...OP3DISC
10:20:58.412221 PS SYSEDMD           0000 004 ....OPTINFO  N=57
10:20:58.443203 PL SYSEDMD           5830 004 ....OPTINFO
10:20:58.443210 PS SYSEDMD           0000 005 ....:OPTPARM1 D=3 N=19
10:20:58.449549 PL SYSEDMD           1960 005 ....:OPTPARM1
10:20:58.449555 PS SYSEDMD           0000 006 ....:.OPTPARM2 D=3 N=10
10:20:58.458286 PL SYSEDMD           0560 006 ....:.OPTPARM2
10:20:58.458300 PS SYSEDMD           0000 007 ....:..OPTPARM3 N=16
10:20:58.458390 PL SYSEDMD           1530 007 ....:..OPTPARM3
10:20:58.458408 PS SYSLIBS           0000 008 ....:...NAT41004 D=10 C=6 N=7345
10:20:58.471017 PT SYSLIBS           5235 008 ....:...NAT41004
10:20:58.471017 PR SYSEDMD           1530 007 ....:..OPTPARM3 N=2898
10:20:58.473293 PL SYSEDMD           1530 007 ....:..OPTPARM3
10:20:58.473297 PS SYSLIBS           0000 008 ....:...NAT41004 D=5 C=6 N=1416
10:20:58.475581 PT SYSLIBS           5235 008 ....:...NAT41004
10:20:58.475581 PR SYSEDMD           1530 007 ....:..OPTPARM3 N=466
10:20:58.475957 PT SYSEDMD           2190 007 ....:..OPTPARM3
10:20:58.475957 PR SYSEDMD           0560 006 ....:.OPTPARM2 N=283
10:20:58.476187 PT SYSEDMD           0860 006 ....:.OPTPARM2
10:20:58.476187 PR SYSEDMD           1960 005 ....:OPTPARM1 N=42
10:20:58.476222 PT SYSEDMD           7510 005 ....:OPTPARM1
10:20:58.476222 PR SYSEDMD           5830 004 ....OPTINFO  D=3 N=10
10:20:58.497926 PL SYSEDMD           6080 004 ....OPTINFO
10:20:58.521954 PR SYSEDMD           1670 003 ...OP3DISC  N=241
10:21:11.205102 PR SYSEDMD           0980 002 ..CALLMON3 D=7 N=6070
10:21:41.704996 PR SYSEDMD           5620 001 .OPTTEST  D=8 I=3 N=26
10:21:41.731229 PT SYSEDMD           7370 001 .OPTTEST
10:21:41.731229 PR                   0000 000          D=14 I=1
10:21:42.248348 ST                   0000 000

Totals
------
Ev Event                    Count
S  Session ................ 1
P  Program ................ 5297
D  Database Call .......... 2140
I  Terminal I/O ........... 12
C  External Program Call .. 6510
E  Runtime Error .......... 43
N  Natural Statement ...... 857384
R  RPC Request............. 0
U  User-Defined Event ..... 0
M  Monitor Pause .......... 2
```

Erläuterungen:

- Für jedes aufgeführte Ereignis wird der Zeitpunkt des Ereignisses, die aktive Library, das Programm (Natural-Objekt), der Copycode, die Zeilennummer und die Programmebene angezeigt.
- Dem Programmnamen folgt die Anzahl der Ereignisse, die von einem Programmereignis bis zum nächsten Programmereignis aufgetreten sind.
- Ereignisse, die zu einer Ereignisgruppe gehören, werden unter Verwendung der maximalen Anzahl der entsprechenden Ereignistypen zu einer Anzahl zusammengefasst. Beispiel: Ein Database Before-(`DB`)- und ein Database After-(`DA`)-Ereignis werden zu einem Datenbankereignis (`D=1`) zusammengefasst.
- Im obigen Beispiel wurde das Natural-Objekt `OPTTEST` auf Ebene `1` gestartet. Dieses Programm ruft das Subprogramm `CALLMON3` auf, das weitere Subprogramme aufruft. Die höchste Stufe 8 wird erreicht, wenn das Subprogramm `NAT41004` ausgeführt wird. Während der ersten Ausführung führt dieses Subprogramm 10 Datenbankaufrufe (`D=10`), 6 externe Programmaufrufe (`C=6`) und 7345 Natural-Statements (N=7345) durch.
- Der Abschnitt `Totals` (Summen) am Ende des Programm-Trace zeigt die maximale Anzahl jeder Ereignisgruppe. Zum Beispiel: Die Summe von 2140 Datenbankaufrufe entspricht 2140 Database Before-(`DB`)- und 2140 Database After-(DA)-Ereignissen.
- Die Summen der Ereignisgruppen **Session** (`S`) und **Program** (`P`) sind nur unter `Totals` (Summen) aufgeführt; sie sind nicht neben dem Programmnamen aufgeführt.

Weitere Erläuterungen zu den Trace-Spalten finden Sie im Abschnitt *Ereignis-Trace*.

Erläuterungen zu Ereignistypen und zugehörigen Ereignisgruppen finden Sie im *Ereignisse*.

Programmzusammenfassung

Wenn `PROGRAM=ON` für eine NPRC-Ressourcendatei angegeben ist, wird die Profiler-Programmzusammenfassung generiert.

Die Programmzusammenfassung zeigt für jedes Natural-Objekt, wie viele Natural-Ereignisse aufgetreten sind, die gesamte CPU-Zeit (in Millisekunden) und den Prozentsatz der CPU-Zeit, die das Natural-Objekt bezogen auf die gesamte CPU-Zeit aufgewendet hat.

Monitor-Pause-Ereignisse und Ereignisse auf Ebene 0 werden nicht für die Programmzusammenfassung berücksichtigt. Ereignisse, die zu einer Ereignisgruppe gehören, werden zu einer Zählung zusammengefasst: siehe *Ereignisse*.

Programmstarts und Ladeanforderungen werden separat aufgeführt.

Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, wird die Anzahl jedes Ereignistyps aufgelistet. Darüber hinaus werden die verstrichene (elapsed) Zeit und die Adabas-Zeiten (absolute und prozentuale Werte) angezeigt. Die exportierten Zeitwerte werden in Mikrosekunden angezeigt.

Beispiel für eine Programmzusammenfassung

Das folgende Beispiel zeigt den Auszug einer Programmzusammenfassung:

```
Natural Profiler Program Summary
--------------------------------
Library  Program      Start      Load  Database       I/O  External     Error Statement      User CPU-Time (ms)  CPU %
SYSEDMD  ADA-CL          41         0        40         0        41         0       621         0         3.785   0.14
SYSEDMD  ADA-RC          45         0        44         0        45         0       545         0         4.704   0.17
SYSEDMD  AOS-CL         115        97        15         0         0         0      2507         0        42.890   1.63
SYSEDMD  AOS-OP         169       154        22         0         0         0      6975         0        70.286   2.68
SYSEDMD  BYTE             1         0         0         0         0         0        11         0         0.034   0.00
SYSEDMD  CALLMON3         1         5        23         0         0         0      7089         0        20.001   0.76
SYSEDMD  CALLNOM          6         6        19         0         0         0        18         0         1.342   0.05
SYSEDMD  CALLNOPM         2         2         4         0         0         0        16         0         0.395   0.01
SYSEDMD  CALLNOPN         1         1         4         0         0         0         8         0         0.244   0.00
SYSEDMD  CALLNOPS         3         4        23         0         0         1        31         0         1.841   0.07
SYSEDMD  DISNOP           1         7         6         0         0         0       515         0         2.260   0.08
SYSEDMD  DISNO4I          1        47         3         0         1         0      8075         0        25.516   0.97
SYSEDMD  DISNO4IS        57         0         0         0       624         0     36877         0       105.650   4.03
SYSEDMD  DISNRS           1         0         0         0        44         0       511         0         3.343   0.12
SYSEDMD  DISNSP           1        18        15         0         0         0      1850         0         6.074   0.23
SYSEDMD  DISNTMZ          1         4        11         0         0         0       324         0         2.309   0.08
SYSEDMD  MENU             1         1         3         0         0         0         2         0         0.235   0.00
SYSEDMD  MONACSH          1         6         6         0         0         0      1217         0         3.470   0.13
SYSEDMD  MONADA           1      3176        71         0         0         0    272180         0       680.214  25.98
SYSEDMD  MONAREP          1         9        28         0         0         0      1964         0         6.378   0.24
...
Total                  5294      5293      2122         7      6510        43    857384         0      2617.326 100.00
```

Erläuterungen:

- Das Natural-Objekt `MONADA` verbrauchte die meiste CPU-Zeit: 680,214 ms, was 25,98 Prozent der gesamten CPU-Zeit entspricht.
- `MONADA` wurde einmal gestartet, es lud `3176` andere Natural-Objekte, führte `71` Datenbankaufrufe und `272180` Natural-Statements durch. Es gab keine Eingabe/Ausgabe, keinen externen Aufruf und keinen Fehler im Programm.
- Am Ende der Programmzusammenfassung werden die Gesamtzahlen der Profil-Erstellung aufgelistet.

Zeilenzusammenfassung

Wenn `LINE=ON` für eine NPRC-Ressourcendatei angegeben wird, wird die Profiler-Zeilenzusammenfassung generiert.

Die Zeilenzusammenfassung zeigt für jede Quellcodezeile in einem Natural-Objekt die Anzahl der aufgetretenen Natural-Ereignisse (Hit-Count/Trefferzahl), die CPU- und die verstrichene (elapsed) Zeit (in Millisekunden und Prozent), die für die Zeile aufgewendet wurde. Der Prozentsatz der Zeiten wird im Verhältnis zu den Gesamtzeiten der Anwendung berechnet.

Die Zeilenzusammenfassung zählt keine Monitor-Pause-Ereignisse und Ereignisse auf Ebene 0.

Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, wird die Anzahl jedes Ereignistyps aufgelistet. Zusätzlich werden die Adabas-Zeiten (absolute und prozentuale Werte) angezeigt. Die exportierten Zeitwerte werden in Mikrosekunden angezeigt.

Beispiel für eine Zeilenzusammenfassung

Das folgende Beispiel zeigt den Auszug einer Zeilenzusammenfassung:

```
Natural Profiler Line Summary
-----------------------------
Library  Program  Line CC-Lib   CC-Name  Hit-Count      CPU (ms)  CPU %  Elapsed (ms)  Ela %
PRFTEST  XINT     0000                          1         0.016   0.46         0.003   0.01
PRFTEST  XINT     0140                          1         0.005   0.14         0.001   0.00
PRFTEST  XINT     0150                          1         0.006   0.17         0.002   0.01
PRFTEST  XINT     0160                          1         0.004   0.11         0.001   0.00
PRFTEST  XINT     0170                         23         0.128   3.75         0.029   0.18
PRFTEST  XINT     0180                         10         0.049   1.43         0.012   0.07
PRFTEST  XINT     0190                         10         0.054   1.58         0.010   0.06
...
Total                                         371         3.408 100.00        15.992 100.00
```

Erläuterungen:

- Die Zeile `0170` im Natural-Objekt `XINT` verbrauchte `0,128` ms der CPU-Zeit und `0,029` ms der verstrichenen (elapsed) Zeit. Dies entspricht `3,75` Prozent der gesamten CPU-Zeit und `0,18` Prozent der gesamten verstrichenen (elapsed) Zeit. `23` Ereignisse (Hit-Count/Trefferzahl) wurden in der Zeile ausgeführt.
- Am Ende der Zeilenzusammenfassung werden die Gesamtzahlen der Profil-Erstellung aufgelistet.

Transaktionszusammenfassung

Eine Transaktion ist der Code, der zwischen zwei aufeinanderfolgenden Terminal-Ein-/Ausgabe-Vorgängen ausgeführt wird. Die verstrichene (elapsed) Zeit, die für die Ausführung einer Transaktion aufgewendet wurde, wird als Antwortzeit (Response Time) bezeichnet.

Die vom Profiler generierte Transaktionszusammenfassung zeigt, wie viele Natural-Ereignisse bei jeder Transaktion aufgetreten sind, die Antwortzeit (in Millisekunden) und den Prozentsatz der Antwortzeit, die die Transaktion bezogen auf die gesamte Antwortzeit aufgewendet hat.

Erforderliche Voraussetzungen für Transaktionsbewertungen

Die Profiler-Daten müssen mit der `CONSOLIDATE`-Funktion auf `TRANSACTION=ON` konsolidiert werden, damit die konsolidierten Datensätze in der NPRC-Ressourcendatei Transaktionsbezeichner enthalten, die für die Auswertung der Transaktionsantwortzeit erforderlich sind.

`TRANSACTION=ON` wird für die READ-Funktion angegeben.

Für Transaktionen ausgewertete Ereignisdaten

Im Allgemeinen bezieht sich ein Terminal-Ein-/Ausgabe-Ereignis auf eine Transaktion. Die Anzahl der Terminal-Ein-/Ausgabe-Ereignisse ist nicht in der Standardausgabe aufgeführt (für `PRINT=ON`).

Daten, die im CSV-Format (kommagetrennte Werte) exportiert werden, enthalten die Anzahl jedes Ereignistyps, einschließlich Terminal-Eingabe/Ausgabe. Darüber hinaus enthält die Datenliste die CPU-Zeit und die Adabas-Zeiten (absolute und Prozentwerte). Exportierte Zeitwerte werden in Mikrosekunden angezeigt.

Die Transaktionszusammenfassung berücksichtigt keine Before Terminal E/A (`IB`)-Ereignisse, RPC Wait for Client-Ereignisse (`RW`), Monitor Pause (`MP`)-Ereignisse und Ereignisse auf Ebene 0. Ereignisse, die zu einer Ereignisgruppe gehören, werden zu einer Zählung zusammengefasst.

Erläuterungen zu Ereignistypen und zugehörigen Ereignisgruppen finden Sie im Abschnitt *Ereignisse*.

Geänderte Zeilen- und Programmzusammenfassung

Wenn `TRANSACTION=ON` für die `READ`-Funktion einer NPRC-Ressourcendatei angegeben wird, berücksichtigen die Programmzusammenfassung und die Zeilenzusammenfassung auch den Transaktionsbezeichner (Identifier) und aggregieren die Programm- und Positionswerte für jede Transaktion separat. In der **Transaktionszeilenzusammenfassung** gibt die Spalte `ID` die Programmzeile an, in der die Transaktion gestartet wurde.

Beispiel für eine Transaktionszusammenfassung

Das folgende Beispiel zeigt eine Transaktionszusammenfassung:

```
Natural Profiler Transaction Summary
------------------------------------
Transact TA-Lib   TA-Prog  TA-CC TA-Line   Program  Database  External     Error Statement      User  Elapsed (ms)  Time%
       1                                          1         0         1         0         0         0         0.829   0.50
       2 SYSPRFLR PRFMENM           0020         9        13         8         0         0         0        17.618  10.65
       3 SYSEDM   MOPTTEST          0020         4         0         0         0         0         0         6.167   3.72
       4 SYSEDM   OPTWLS80          0470         0         0         0         0         0         0         1.108   0.66
       5 SYSEDM   OPTWLS80          0470         2         0         0         0         0         0         1.133   0.68
       6 SYSEDM   OPTTEST           1750         1         0         0         0         0         0         1.180   0.71
       7 SYSEDM   MOPTTEST          0020        27        33        30         0         0         0        94.236  56.96
       8 SYSEDM   OPTTEST           5590         1         0         0         0         0         0         1.185   0.71
       9 SYSEDM   MOPTTEST          0020        37        33        56         0         0         0        37.311  22.55
      10 SYSEDM   OPTTEST           5760         1         0         0         0         0         0         1.096   0.66
      11 SYSEDM   MOPTTEST          0020         6         0         6         0         0         0         2.164   1.30
      12 SYSPRFLR PRFMENM           0020         2         0         3         0         0         0         1.394   0.84
Total                                           86        79       103         0         0         0       165.421 100.00
```

Erläuterungen am Beispiel von Transaktion `7`:

- Transaktion `7` verbrauchte die längste (verstrichene) Antwortzeit: 94,236 Millisekunden (ms), was 56,96 Prozent der gesamten Antwortzeit entspricht.
- Transaktion `7` startete eine E/A-Vorgang in der Library `SYSEDM` (`TA-Lib`-Spalte), Programm `MOPTTEST` (`TA-Prog`), Programmzeile `0020` (`TA-Line`).
- Transaktion `7` hat auf `27` Programme zugegriffen und `33` Datenbankaufrufe und `30` externe Programmaufrufe durchgeführt.
- Im obigen Beispiel sind alle Zähler für Statement-Ereignisse `0`, da Statement-Ereignisse während diesem Profiler-Lauf nicht gesammelt wurden.
- Am Ende der Transaktionszusammenfassung werden die Gesamtzahlen (`Total`) der profilierten Transaktionen aufgelistet.

Programmabdeckung

Die Programmabdeckungstabelle wird generiert, wenn `PROGRAM=ON` für eine NCVF-Ressourcendatei angegeben ist.

Die Programmabdeckungstabelle zeigt die Codeabdeckungsergebnisse für jedes aufgerufene Natural-Objekt. Wenn die Tabelle im Textformat angegeben wird, werden nur die GP-Abdeckungsergebnisse (einschließlich Copycodes) angezeigt. Im CSV-Format (kommagetrennte Werte) zeigt die Tabelle Zeilen, die Copycode-Werte enthalten, zusätzliche Spalten mit Quellcodezählern (Copycodes nicht enthalten) und Informationen zu `INCLUDE`-Statements.

Im Textformat enthält die Tabelle die Anzahl der Abdeckungen für jede aufgerufene Library und für die gesamte Anwendung.

Die Tabelle enthält die folgenden Spalten:

Spalte	Beschreibung	
Evaluation	Der Auswertungstyp. Mögliche Typen sind:	
	Program	Für Programmabdeckungsdaten.
	Event (Ereignis)	Für Statement-Abdeckungsdaten.
	Statistics	Für Profiler-Statistikdaten.
Object Count	Die Anzahl der katalogisierten Objekte (GPs), die in der Tabelle aufgeführt sind.	
Object Type	Der Typ des Natural-Objekts, z.B. Programm und Subprogramm.	
Library	Die Natural-Library, die das Objekt enthält.	
Object	Der Name des Natural-Objekts.	
Copycode ID	Der eindeutige Bezeichner der Copycode-Instanz im katalogisierten Objekt (GP). Das Programm erhält die Copycode-ID `0`.	
Copycode Library	Die Library, aus der der Copycode eingefügt wird.	
Copycode Name	Der Name des Copycode.	
GP Coverage%	Der Prozentsatz der Objektabdeckung, wobei `INCLUDE`-Statements aufgelöst werden.	
GP Covered	Die Anzahl der abgedeckten (ausgeführten) Statements, wobei I`NCLUDE`-Statements aufgelöst werden.	
GP Missed	Die Anzahl der „missed“ (nicht ausgeführten) Statements im Objekt, wobei `INCLUDE`-Statements aufgelöst werden.	
GP Total	Die Gesamtzahl aller ausführbaren Statements im Objekt, wobei `INCLUDE`-Statements aufgelöst werden.	
Src Coverage%	Der Prozentsatz der Objektabdeckung, wobei `INCLUDE`-Statements nicht aufgelöst werden.	
Src Covered	Die Anzahl der abgedeckten (ausgeführten) Statements, wobei `INCLUDE`-Statements nicht aufgelöst werden.	
Src Missed	Die Anzahl der „missed“ (nicht ausgeführten) Statements im Objekt, wobei `INCLUDE`-Statements nicht aufgelöst werden.	
Src Total	Die Gesamtzahl aller ausführbaren Statements im Objekt, wobei `INCLUDE`-Statements nicht aufgelöst werden.	
First Statement	Die ID des ersten Statements des Objekts oder Copycodes.	
INCLUDE CC-ID	Nur bei Copycode. Die Copycode-ID des Objekts oder des Copycode, der den Copycode enthält.	
INCLUDE Object	Nur bei Copycode. Der Name des Objekts oder des Copycode, der den Copycode enthält.	

Spalte	Beschreibung
INCLUDE Line	Nur bei Copycode. Die Zeilennummer des `INCLUDE`-Statement, das den Copycode enthält.

Beispiel für die Programmabdeckung

Das folgende Beispiel zeigt das Ergebnis der Programmabdeckung im Textformat

```
Program Coverage
----------------
Library  Object   Ty Coverage%  Covered    Missed    Total
COVDEMO  TESTCOVN N      84.0%       37         7       44
COVDEMO  TESTCOVP P      69.2%        9         4       13
COVDEMO  -------- --      80.7%       46        11       57
Totals   -------- --      80.7%       46        11       57
```

Erläuterungen:

- Die Anwendung greift auf zwei Objekte zu: Das `TESTCOVN`-Subprogramm (`N`) und das `TESTCOVP`-Programm (`P`).
- In `TESTCOVN` gibt es `44` ausführbare Statements (Objektcode-Anweisungen), von denen `37` abgedeckt (ausgeführt) und `7` nicht ausgeführt wurden („missed"), was eine Gesamtabdeckung von `84,0%` ergibt.
- Die aufsummierten Werte der beiden Objekte, auf die in der Library `COVDEMO` zugegriffen wird, zeigen eine Abdeckung von `80,7%`.
- Die Gesamtabdeckung beträgt ebenfalls `80,7%`, weil die Objekte nur auf eine Library zugreifen.

Statement-Abdeckung

Die Statement-Abdeckung wird generiert, wenn `EVENT=ON` für eine NCVF-Ressourcendatei angegeben wird.

Für die Statement-Abdeckung liest die Profiler Utility den Quellcode der überwachten Objekte. Zuerst sucht sie den Quellcode in der Library, die die Natural Profiler-Ressource enthält. Wenn der Quellcode in dieser Library nicht gefunden wird, wird er in der Library gesucht, die mit den Profiler-Daten angegeben ist. Wenn der Quellcode auch dort nicht gefunden wird oder wenn der Quellcode nicht mit den gesammelten Daten übereinstimmt, werden im Statement-Abdeckungsreport Quellcodezeilen nicht gedruckt. Die Profiler Utility löst `INCLUDE`-Statements auf und fügt den Quellcode des entsprechenden Copycodes in das inkludierende Programm ein. Wenn die `INCLUDE`-Struktur nicht aufgelöst werden kann, werden die Copycodes separat gedruckt.

Wenn ein Quellcode nach der Ausführung der Abdeckung geändert wurde, kann die Profiler Utility nicht mehr die vollständigen Informationen bereitstellen. Um dies zu verhindern, kopieren Sie die Ressourcendatei zusammen mit den zugehörigen Programmquellen in eine andere Library. Wenn die Profiler Utility die Ressource liest, werden die Quellcodes aus der Ressource-Library

genommen und eine Änderung der ursprünglichen Quellcodes hat keinen Einfluss auf die Statement-Abdeckung.

Die Statement-Abdeckung zeigt den Prozentsatz der Statements, die für jede Quellcodezeile der aufgerufenen Programme abgedeckt werden. Wenn das Ergebnis im Textformat geschrieben ist, werden für jedes in der Statistik aufgeführte Objekt die Objektabdeckungswerte vor den Statement-Abdeckungsdaten angezeigt. Wenn das Ergebnis im CSV-Format (kommagetrennte Werte) geschrieben wird, werden zusätzliche Informationen zur Abdeckung der Statement-Abdeckung bereitgestellt.

Die Tabelle enthält die folgenden Spalten:

Spalte	Beschreibung	
Evaluation	Der Auswertungstyp. Mögliche Typen sind:	
	Program	Für Programmabdeckungsdaten.
	Event (Ereignis)	Für Statement-Abdeckungsdaten.
	Statistics	Für Profiler-Statistikdaten.
Object Count	Die Anzahl der in der Tabelle aufgeführten Objekte (GPs).	
Library	Die Natural-Library, die die Objekte enthält.	
Object	Der Name des Natural-Objekts.	
Copycode ID	Der eindeutige Bezeichner der Copycode-Instanz im zugehörigen katalogisierten Objekt. Das Programm erhält die Copycode-ID `0`.	
Copycode Library	Die Library, die den Copycode enthält (wenn Copycode aktiv ist).	
Copycode Name	Der Name des Copycodes (wenn Copycode aktiv ist).	
Line	Die Zeilennummer im Natural Source-Objekt, z. B. `0120`.	
Source	Die Natural-Quellcodezeile, die eine Statement-Definition enthält, z. B. `MOVE #A TO #B`.	
Coverage%	Der Prozentsatz der Statement-Abdeckung der Zeile.	
Covered	Die Anzahl der in der Zeile abgedeckten (ausgeführten) Statements.	
Missed	Die Anzahl der „missed“ (nicht ausgeführten) Statements in der Zeile.	
Total	Die Gesamtzahl aller ausführbaren Statements (Objektcode-Anweisungen) in der Zeile.	
Item Coverage	Gibt an, welche Statement-Elemente (Objektcode-Anweisungen) in der Zeile abgedeckt oder nicht abgedeckt wurden („missed“). Jedes Statement wird entweder durch `1` oder `0` dargestellt, wobei `1` ein abgedecktes Statement und `0` ein nicht abgedecktes („missed“) Statement anzeigt. Zum Beispiel: Ein Wert von `x100` zeigt an, dass nur das erste von drei Statements in der Zeile abgedeckt ist.	
Mark	Zeigt den Abdeckungsstatus der Zeile an. Die Spalte `Mark` kann verwendet werden, um die Abdeckungsergebnisse in Tools wie Microsoft Excel zu visualisieren. Mögliche `Mark`-Werte sind unter ***Microsoft Excel-Vorlage zur Visualisierung der Abdeckungsergebnisse benutzen*** aufgeführt.	

Beispiel für Statement-Abdeckung:

Im folgenden Beispiel wird angenommen, dass die Entwicklungsabteilung eine neue Version des `TESTCOVN`-Subprogramms an die Qualitätssicherungsabteilung geliefert hat. Nach dem Ausführen der Testprogramme zeigt die Statement-Abdeckung des Subprogramms das folgende Ergebnis (Textformat):

```
Statement Coverage
------------------
M Cov% CC-Lib   CC-Name  Line Source
*                        0010 * Test function Coverage
*                        0020 * Subprogram TESTCOVN
+                        0030 DEFINE DATA
+                        0040 PARAMETER
+                        0050 1 FUNC     (I2)  /* Function
+                        0060 1 RET-CODE (I4)  /* Return code
+                        0070 END-DEFINE
*                        0080 *
*                        0090 /* Return 0 by default
C 100%                   0100 RESET RET-CODE
*                        0110 *
C 100%                   0120 DECIDE ON FIRST VALUE OF FUNC
P  50%                   0130   VALUE 0
M   0%                   0140     PRINT 'Test function 0'
P  66%                   0150   VALUE 1
C 100%                   0160     PRINT 'Test function 1'
C 100%                   0170   VALUE 2
C 100%                   0180     PRINT 'Test function 2'
C 100%                   0190   VALUE 3
C 100%                   0200     PRINT 'Test function 3'
C 100%                   0210   VALUE 4
C 100%                   0220     PRINT 'Test function 4'
C 100%                   0230   VALUE 5
C 100%                   0240     PRINT 'Test function 5'
C 100%                   0250   VALUE 6
C 100%                   0260     PRINT 'Test function 6'
C 100%                   0270   VALUE 7
C 100%                   0280     PRINT 'Test function 7'
C 100%                   0290   VALUE 8
C 100%                   0300     PRINT 'Test function 8'
P  33%                   0310   VALUE 9
M   0%                   0320     PRINT 'New test function 9'
C 100%                   0330   NONE VALUE
M   0%                   0340     RET-CODE := 1 /* Unsupported function
+                        0350 END-DECIDE
*                        0360 *
C 100%                   0370 END
```

Erläuterungen:

- Die Spalte `Mark` (`M`) zeigt an, ob eine Zeile abgedeckt (`C` = Covered), nicht abgedeckt (`M` = Missed) oder teilweise abgedeckt (`P` = Partially) ist.
- Keine Testfälle decken die Funktionen `Test function 0` und `New test function 9` ab (bezeichnet mit `M` und `0%` Abdeckung). Der Fall `NONE VALUE` wird ebenfalls nicht abgedeckt.
- Alle anderen Testfälle sind abgedeckt (bezeichnet mit `C` und `100%` Abdeckung).
- Ein Natural `VALUE`-Statement entspricht mehreren Objektcode-Statements. Die Abdeckung von `50%` und `60%` für `VALUE 0`- und `VALUE 1`-Statements zeigt an, dass nur ein Teil dieser Objektcode-Statements abgedeckt ist.

  Dies liegt daran, dass eines der generierten Objektcode-Statements aus technischen Gründen zum vorangehenden Statement und die anderen zum aktuellen `VALUE`-Statement gehören.

Infolge dieser Abdeckungsanalyse müssen die Testfälle so angepasst werden, dass auch die `Test function 0` und die `Test function 9` (und möglicherweise der Fehlerfall mit einem nicht unterstützten Funktionscode) abgedeckt sind.

Microsoft Excel-Vorlage zur Visualisierung der Abdeckungsergebnisse benutzen

Voraussetzungen: Microsoft Excel und Natural für Windows oder Natural für Linux.

Wenn Sie das Abdeckungsergebnis mit Microsoft Excel analysieren möchten, können Sie die Microsoft Excel-Vorlage verwenden, die mit Natural für Windows und Natural für Linux geliefert wird. Führen Sie dazu die folgenden Schritte aus:

1. Führen Sie die Profiler `READ`-Funktion aus und schreiben Sie die Ausgabedaten im CSV-Format (kommagetrennte Werte) in Arbeitsdatei 7 (`Work File 7`). Zum Beispiel:

```
FUNCTION=READ            /* Read Profiler Data
  RESOURCE-NAME='Test'   /* Resource name
  RESOURCE-LIB=PRFDATA   /* Resource library
  RESOURCE-TYPE=NCVF     /* Use resource type NCVF
  EVENT=ON               /* Print statement coverage
  PROGRAM=ON             /* Print program coverage
  STATISTICS=ON          /* Print statistics
  PRINT=ON               /* Write to standard output
  EXPORT=ON              /* Write to Work File 7
  FORMAT=COMMA           /* Export in CSV format
```

2. Falls Ihr Microsoft Excel Semikolons als Trennzeichen benötigt, geben Sie Folgendes an:

```
FORMAT=SEMICOLON      /* Export in CSV format
```

3. Exportieren Sie die Daten von Arbeitsdatei 7 mit einem beliebigen Tool (z. B. FTP) als CSV-formatierte Datei in eine Windows-Umgebung.
4. Öffnen Sie die CSV-Datei mit Microsoft Excel.
5. Ordnen Sie die Daten neu an, so dass sich jeder Auswertungstyp (Programm, Ereignis, Statistik) auf einem eigenen Arbeitsblatt in der Microsoft Excel-Datei befindet.
6. Öffnen Sie die gelieferte Vorlage `TESTCOV.XLSX` mit Microsoft Excel. Die Vorlage ist im Unterverzeichnis `RES` (Ressourcen) der Natural-System-Library `SYSPRFLR` enthalten.
7. Kopieren Sie für jedes Arbeitsblatt das Format aus der Vorlage in Ihr Microsoft Excel:
   - Klicken Sie auf die obere linke Ecke der Tabelle in der Vorlage, um alle Daten in der Tabelle zu markieren.
   - Klicken Sie auf die Microsoft Excel-Funktion **Copy format**.
   - Klicken Sie auf die obere linke Ecke der Tabelle in Ihrem Arbeitsblatt, um das Format zu kopieren.

   Jetzt sind alle Einträge wie in der Vorlage formatiert. Die Quellcodezeilen sind wie folgt gefärbt und markiert:

Farbe	Markierung	Beschreibung
Grün	`C`	Alle Statements in der Zeile sind abgedeckt („Covered“).
Gelb	`P`	Die Statements in der Zeile sind teilweise abgedeckt („Partially“).
Rosa	`M`	Alle Statements in der Zeile werden nicht abgedeckt („Missed“).
Grau	`*`	Ein Kommentar oder eine leere Zeile.
Rot	`E`	Fehler aufgetreten. Zum Beispiel, wenn die Abdeckungsanalyse eine Zeilennummer gesammelt hat, aber die entsprechende Quellcodezeile nicht gefunden wird.
Keine (weiß)	`+`	Alle anderen Zeilen, z.B. Fortsetzungszeilen eines Statement.

Beispiel für ein Microsoft Excel-Arbeitsblatt

Das folgende Beispiel zeigt einen Arbeitsblattauszug der Codeabdeckung für das Programm `TESTCOVP` mit inkludiertem Copycode `TESTCOVC` ohne die Spalten, die den Objektnamen und die Library enthalten:

Copycode ID	Copycode Library	Copycode Name	Line	Source	Coverage%	Covered	Missed	Total	Item Coverage	Mark
0			10	* Test Coverage		0	0	0		*
0			20	* Program TESTCOVP		0	0	0		*
0			30	DEFINE DATA LOCAL		0	0	0		+
0			40	1 FUNC (I2) /* function		0	0	0		+
0			50	1 RET-CODE (I4) /* Return code		0	0	0		+
0			60	END-DEFINE		0	0	0		+
0			70	FOR FUNC = 1 TO 8	100	4	0	4	x1111	C
0			80	/* Test the subprogram functions		0	0	0		*
0			90	CALLNAT 'TESTCOVN' FUNC RET-CODE	100	1	0	1	x1	C
0			100	INCLUDE TESTCOVC 'RET-CODE' 'FUNC'	100	1	0	1	x1	C
1	COVDEMO	TESTCOVC	10	* Test Coverage		0	0	0		*
1	COVDEMO	TESTCOVC	20	* Copycode TESTCOVC		0	0	0		*
1	COVDEMO	TESTCOVC	30	IF &1& > 0	100	1	0	1	x1	C
1	COVDEMO	TESTCOVC	40	IF &1& = 1	0	0	1	1	x0	M
1	COVDEMO	TESTCOVC	50	PRINT 'Unsupported function' &2&	0	0	1	1	x0	M
1	COVDEMO	TESTCOVC	60	ELSE	0	0	1	1	x0	M
1	COVDEMO	TESTCOVC	70	PRINT 'Return code:' &1&	0	0	1	1	x0	M
1	COVDEMO	TESTCOVC	80	END-IF		0	0	0		+
1	COVDEMO	TESTCOVC	90	END-IF		0	0	0		+
0			110	END-FOR	100	1	0	1	x1	C
0			120	*		0	0	0		*
0			130	END	100	1	0	1	x1	C

Erläuterungen:

- Die Quellcodezeilen des Copycode `TESTCOVC` sind in den Quellcode des Programms `TESTCOVP` direkt nach dem entsprechenden `INCLUDE`-Statement eingefügt.
- Die (rosa) Zeilen `40` bis `70` des Copycodes enthalten nicht abgedeckte („missed“) Statements, was bedeutet, dass sie im Testlauf nicht ausgeführt wurden.
- Alle anderen Zeilen (grün), die ausführbare Statements enthalten, sind abgedeckt.
- Ein `FOR`-Statement entspricht vier Objektcode-Statements. Alle vier Statements werden, wie in `x1111` in der Spalte **Item Coverage** angezeigt, behandelt.

Statistiken

Wenn `STATISTICS=ON` angegeben wird, werden die Profiler-Statistiken aufgelistet.

Wenn die Daten im CSV-Format (kommagetrennte Werte) exportiert werden, werden die Eigenschaften und Werte der Profiler-Statistiken als separate Spalten zum Ereignis- oder Konsolidierungs-Trace hinzugefügt. Wenn Abdeckungsdaten im CSV-Format exportiert werden, werden die Statistikwerte in zusätzlichen Zeilen hinzugefügt, die durch den Wert `Statistics` in der Spalte `Evaluation` gekennzeichnet sind.

Beispiel für Statistik:

Das folgende Beispiel zeigt einen Auszug aus den Statistikdaten einer NPRC-Ressourcendatei:

```
************************************************************************
* 11:02:39          ***** NATURAL PROFILER UTILITY *****         2015-08-05
* User SAGTEST1                - Statistics -                       RESDATA
*
* General Info
* Machine class ..................... MAINFRAME
* Environment ....................... Batch
...
* Profiler Resource File
* Resource name ..................... EDM-MONITOR.nprc
* Resource type ..................... Natural Profiler Resource Consolidated
* Resource allocation date ........... 2015-07-27 10:36:19.6
* Resource size (bytes) .............. 565160
...
*
* Monitor Session
* Monitor start time ................. 2015-07-27 10:20:57.2
* Monitor end time ................... 2015-07-27 10:21:42.8
* Monitor elapsed time (sec) ......... 45.519604
*
* Trace Session
* First library ..................... SYSEDMD
* First program ..................... MENU
* Highest level ..................... 10
* Trace start time .................. 10:20:58.219911
* Trace end time .................... 10:21:42.248348
* Trace elapsed time (sec) ........... 44.028437
...
* Data Processing
* Number of events .................. 895936
...
* Data Consolidation
* Consolidation ..................... ON
* Consolidation records .............. 21624
* Consolidation elapsed time (sec) ... 15.643516
* Consolidation factor ............... 41.4
* Consolidation records/block ........ 191.3
* Bytes/consolidation record ......... 25.8
*
...
************************************************************************
```

Erläuterungen:

- Die Ressource `EDM-MONITOR.nprc` wurde am 27.07.2015 um 10:36:19 Uhr zugewiesen und hat eine Größe von 565160 Bytes.
- Die profilierte Anwendung lief am selben Tag um 10:20:58 Uhr für 44,0 Sekunden und startete mit dem Programm `MENU` in der Library `SYSEDMD`.

- Die profilierte Anwendung generierte insgesamt 895936 Natural-Ereignisse. Die Datenkonsolidierung dauerte 15,6 Sekunden und reduzierte die Anzahl der Datensätze auf 21624, was einem Konsolidierungsfaktor von 41,4 entspricht.

Alle bereitgestellten Statistikinformationen werden im Abschnitt *Profiler-Statistikdaten* erläutert.

Profiler-Ressourcendateien verwalten

Im Allgemeinen werden Profiler-Ressourcen mit der Natural Utility SYSMAIN, NaturalONE oder Natural Studio als NPRF-, NPRC- oder NCVF-Dateien aufgeführt. Diese Tools bieten auch Funktionen zum Kopieren, Umbenennen und Löschen von Ressourcendateien.

Die Natural Profiler Rich GUI-Schnittstelle listet auch die Profiler-Ressourcen auf und bietet eine Funktion zum Löschen einer Profiler-Ressourcendatei.

Darüber hinaus können Sie Profiler Utility-Funktionen verwenden, um Profiler-Ressourcendateien aufzulisten und zu löschen.

Dieser Abschnitt behandelt die folgenden Themen:

- Profiler-Ressourcendateien auflisten
- Löschen einer Ressourcendatei

Profiler-Ressourcendateien auflisten

Die Funktion `LIST` der Profiler Utility listet die Profiler-Ressourcendateien einer bestimmten Natural-Library sowie das Datum und die Uhrzeit auf, zu der die Ressourcendateien zugeordnet wurden.

Syntax von `LIST`:

```
FUNCTION=LIST
[RESOURCE-LIB=library-name]
[RESOURCE-TYPE={NPRF|NPRC|NCVF}]
[PRINT={ON|OFF}]
[EXPORT={ON|OFF}]
[FORMAT={TEXT|COMMA|SEMICOLON}]
```

Syntax-Beschreibung:

Schlüsselwort für LIST	Wert	Beschreibung
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, die die Profiler-Ressourcendateien enthält, die Sie auflisten möchten. Standardwert: Der Name der aktuellen Library.
RESOURCE-TYPE		Gibt den Typ der aufzulistenden Ressourcendateien an: NPRF, NPRC oder NCVF. Standardeinstellung: Alle Typen werden aufgelistet, wenn hier kein Wert angegeben ist.
	NPRF	Listet nur NPRF-Ressourcendateien (Natural Profiler Resource File) auf.
	NPRC	Listet nur NPRC-Ressourcendateien (Natural Profiler Resource Consolidated) auf.
	NCVF	Listet nur NCVF-Ressourcendateien (Natural Code Coverage File) auf.
PRINT		Gibt an, ob das Ergebnis in die Standardausgabe geschrieben wird.
	ON	In die Standardausgabe schreiben.
	OFF	Nicht in die Standardausgabe schreiben.
EXPORT		Gibt an, ob das Ergebnis in Natural-Arbeitsdatei (Work File) 7 geschrieben wird.
	ON	In Natural-Arbeitsdatei 7 schreiben.
	OFF	Nicht in Natural-Arbeitsdatei 7 schreiben.
FORMAT		Gibt das Format an, in dem die exportierten Daten in Arbeitsdatei 7 geschrieben werden.
	TEXT	Daten im Freitextformat schreiben.
	COMMA	Daten im CSV-Format schreiben mit einem Komma (,), das als Trennzeichen verwendet wird.
	SEMICOLON	Daten im CSV-Format schreiben mit einem Semikolon (;), das als Trennzeichen verwendet wird.

Beispiel für LIST:

Das folgende Beispiel listet die NPRF-Profiler-Ressourcendateien der Library PRFDATA auf. Die Liste wird in die Standardausgabe und in die Arbeitsdatei 7 im Textformat geschrieben.

```
FUNCTION=LIST           /* List Profiler resource files
  RESOURCE-LIB=PRFDATA  /* Resource library
  RESOURCE-TYPE=NPRF    /* List NPRF resource files
  PRINT=ON              /* Write to standard output
  EXPORT=ON             /* Write to Work File 7
  FORMAT=TEXT           /* Export in text format
```

Ausgabe:

```
Natural Profiler Resources
--------------------------
Library: PRFDATA
Resource type: nprf

Count Date       Time     Name
    1 2015-06-15 14:32:18 Hello1.nprf
    2 2015-06-26 18:39:57 QDTest1.nprf
    3 2015-06-24 22:00:35 QETest1.nprf
    4 2015-06-30 14:32:42 Studio.nprf
    5 2015-07-02 15:02:32 Test.nprf

Number of nprf resources in library PRFDATA: 5
```

Löschen einer Ressourcendatei

Wenn Sie eine große Ressourcendatei löschen oder ersetzen, kann es vorkommen, dass Sie die folgende Fehlermeldung erhalten:

```
Error - NAT3047 Maximum value for Adabas parameter NISNHQ was exceeded.
```

(Fehler - NAT3047 Maximalwert für den Adabas-Parameter NISNHQ wurde überschritten.)

In diesem Fall haben Sie drei Möglichkeiten:

1. Bitten Sie Ihren Datenbankadministrator, den Wert des Adabas-Parameters `NISNHQ` zu erhöhen.
2. Benutzen Sie die **Delete**-Funktion der Profiler Rich GUI-Schnittstelle. Sie verwendet die gleiche Technik wie die `DELETE`-Funktion der Profiler Utility.
3. Benutzen Sie die `DELETE`-Funktion der Profiler Utility, um ein „unsauberes“ Löschen der Ressource durchzuführen. Diese Funktion löscht die Ressource nicht in einem einzigen großen Schritt, sondern in Blöcken (mit einem Ende der Transaktion nach jedem Block). Wenn die `DELETE`-Funktion aus irgendeinem Grund fehlschlägt, müssen Sie sie wiederholen, um inkonsistente Daten zu beseitigen.

   Syntax von `DELETE`:

```
FUNCTION=DELETE
[RESOURCE-NAME=resource-name]
[RESOURCE-TYPE={NPRF|NPRC|NCVF|NPRK|NONE}]
[RESOURCE-LIB=library-name]
```

   Syntax-Beschreibung:

Schlüsselwort für DELETE	Wert	Beschreibung
RESOURCE-NAME	*resource-name*	Der Name der Profiler-Ressourcendatei, die Sie löschen möchten. Mögliche Erweiterungen sind .nprf (Natural Profiler resource file), .nprc (Natural Profiler resource consolidated), .ncvf (Natural code coverage file) oder .nprk (Natural Profiler resource key). Wenn keine Erweiterung angegeben wurde, wird die mit dem Schlüsselwort RESOURCE-TYPE angegebene Erweiterung automatisch hinzugefügt. Standardwert: keiner Wenn RESOURCE-TYPE=NPRK angegeben ist und kein RESOURCE-NAME, wird der Name der Profiler-Ressourcendatei, die den NaturalONE Profiler-Schlüssel enthält, als Ressourcenname verwendet.
RESOURCE-TYPE		Der Standard-Ressourcentyp (Erweiterung), der gelöscht werden soll, wenn keine Erweiterung mit RESOURCE-NAME angegeben ist. Standardwert: NPRF
	NPRF	Der Standard-Ressourcentyp ist NPRF mit der Erweiterung .nprf.
	NPRC	Der Standard-Ressourcentyp ist NPRC mit der Erweiterung .nprc.
	NCVF	Der Standard-Ressourcentyp ist NCVF mit der Erweiterung .ncvf.
	NPRK	Der Standard-Ressourcentyp ist NPRK mit der Erweiterung .nprk.
	NONE	Die Ressource mit dem Kurznamen *resource-name* wird gelöscht.
RESOURCE-LIB	*library-name*	Der Name der Natural-Library, die die Ressource enthält, die Sie löschen möchten. Standardwert: Der Name der aktuellen Library

Beispiel für DELETE

```
FUNCTION=DELETE          /* Delete a Profiler resource file
  RESOURCE-NAME='Test'   /* Resource name
  RESOURCE-TYPE=NPRF     /* Resource type
  RESOURCE-LIB=PRFDATA   /* Resource library
```

Profiler-Eingaben aus Natural-Textobjekten einfügen

Der Profiler kann Eingabedaten von einem Natural-Textobjekt lesen. Die Syntax der Daten im Natural-Textobjekt ist die gleiche wie für den primären Kommandoeingabedatensatz CMSYNIN (siehe *Syntax und Schlüsselwörter*).

Um Profiler-Eingabedaten aus einem Natural-Textobjekt zu inkludieren:

- Geben Sie die folgenden Profiler-Schlüsselwörter ein:

```
INCLUDE-LIB=library-name
INCLUDE=object-name
```

  Die Schlüsselwortsyntax wird in *Profiler Utility-Schlüsselwörter* erklärt.

Die Daten im Natural-Textobjekt werden den Profiler-Eingabedaten in der Zeile nach dem `INCLUDE`-Schlüsselwort hinzugefügt. Die Profiler-Eingabedaten können mehrere `INCLUDE`-Schlüsselwörter enthalten, und die zugehörigen Natural-Textobjekte können auch `INCLUDE`-Schlüsselwörter enthalten. Wenn ein Natural-Textobjekt ein `END-PROFILER`-Schlüsselwort enthält, wird die Profiler Utility beendet und alle verbleibenden Daten in dem (den) Natural-Textobjekt(en) werden ignoriert.

In der Natural-System-Library `SYSPRFLR` werden Textobjekte ausgeliefert, deren Namen mit `X` beginnen, und die als Profiler-Eingabe verwendet werden können. Die einzelnen Profiler-Funktionen, die sie ausführen, werden in den Quellcodes dieser Objekte beschrieben.

Wir empfehlen, keine Objekte in der System-Library `SYSPRFLR` zu ändern, da sie überschrieben oder entfernt werden können, wenn eine neue Natural-Version installiert wird. Kopieren Sie die erforderlichen Objekte in eine User Library, bevor Sie sie bearbeiten.

Beispiele für INCLUDE

Das folgende Beispiel fügt den Inhalt des Natural-Textobjekts `MYPROF` aus der Library `MYLIB` zu den Profiler-Eingabedaten hinzu:

```
INCLUDE-LIB=MYLIB
INCLUDE=MYPROF
```

Im folgenden Beispiel wird der Inhalt des Natural-Textobjekts `XINIT` aus der Library `SYSPRFLR` zu den Profiler-Eingabedaten hinzugefügt. Das Objekt initialisiert und startet die Profil-Erstellung ohne Konsolidierung und ohne Statement-Ereignissammlung. Darüber hinaus wird die Profiler Utility beendet, so dass nach dem `INCLUDE`-Schlüsselwort keine weitere Profiler-Eingabe erwartet wird.

```
INCLUDE=XINIT
```

Ereignisverfolgung (Event-Trace)

Der Natural Profiler sammelt detaillierte Informationen über jedes Natural-Ereignis, das auftritt, während eine Natural-Anwendung ausgeführt wird. Diese Daten können in der Ereignisverfolgung (Event-Trace) eingesehen werden.

Die für die Natural-Code-Abdeckung geschriebenen Traces werden im Abschnitt *Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung* beschrieben.

Die Profiler Utility bietet die folgenden Optionen zum Schreiben einer Profiler-Ereignisverfolgung:

- Schreiben des Trace in die Standardausgabe der Profiler-Monitor-Session (`MONPRINT`-Dataset), während die Anwendung profiliert wird.
- Schreiben des Trace in die Standardausgabe, während die NPRF-Daten konsolidiert werden. In diesem Fall zeigt die Ereignisverfolgung die Delta-Werte der verstrichenen (elapsed) Zeit und der CPU-Zeit anstelle von ereignisspezifischen Daten.
- Schreiben des Trace beim Lesen einer Profiler NPRF-Ressourcendatei mit der Profiler Utility-Funktion `READ`.

Anmerkung: Die Ereignisverfolgung (Event-Trace) kann auch in NaturalONE aufgelistet werden.

Um die die Ereignisverfolgung zu aktivieren:

- Geben Sie das folgende untergeordnete Schlüsselwort der `INIT`-Funktion der Profiler Utility ein:

  ```
  TRACE-EVENT=ON
  ```

  Geben Sie das folgende untergeordnete Schlüsselwort der `CONSOLIDATE`-Funktion der Profiler Utility ein:

  ```
  TRACE-EVENT=ON
  ```

  Geben Sie das folgende untergeordnete Schlüsselwort der `READ`-Funktion der Profiler Utility ein:

```
EVENT=ON
```

Die Profiler-Ereignisverfolgung enthält die folgenden Spalten:

Spalte	Beschreibung
Count	Ereigniszähler.
Time	Zeitpunkt des Ereignisses. Einheit: *Stunde*:*Minute*:*Sekunde*.*Mikrosekunden*
CPU-Time	Session-CPU-Zeit. Einheit: Mikrosekunden
Ev	Ereignistyp. Siehe ***Gesammelte Ereignisse und Daten***.
Lev	Programmebene.
Library	Programm-Library.
Program	Programmname (Natural-Objekt) .
Line	Zeilennummer des ausgeführten Programm-Statement.
CC-Lib	Copycode-Library (wenn Copycode aktiv ist).
CC-Name	Copycode-Name (wenn Copycode aktiv ist).
Statement	Natural-Statement, das zurzeit ausgeführt wird. Aus technischen Gründen gibt es keine Eins-zu-Eins-Beziehung zwischen einem Natural-Quellcode-Statement und dem entsprechenden Objektcode im katalogisierten Objekt. Daher können sich die in der Profiler-Ereignisverfolgung aufgeführten Anweisungen von den Anweisungen im Quellcode unterscheiden.
Local-Data	Ereignisspezifische Daten wie die Adabas-Datenbankkennung (DBID) und die Dateinummer (FNR). Diese Daten werden nur für die Funktionen INIT und READ der Profiler Utility angezeigt.
Elapsed (ms)	Verstrichene Zeit für die Verarbeitung des Ereignisses. Einheit: Millisekunden Diese Daten werden nur für die `CONSOLIDATE`-Funktion der Profiler Utility angezeigt.
CPU-Delta	CPU-Zeit, die für die Verarbeitung des Ereignisses aufgewendet wurde. Einheit: Millisekunden

Beispiel für eine Ereignisverfolgung (Event Trace)

Im folgenden Beispiel druckt die Funktion `READ` der Profiler Utility das Event Trace aus:

```
FUNCTION=READ                /* Read event data
  EVENT=ON                   /* Write event trace
```

Das Event Trace wird in die Standardausgabe geschrieben:

```
Count Time             CPU-Time (ms) Ev Lev Library  Program  Line CC-Lib   CC-Name  Statement  Local-Data
    0 17:38:17.200951         42.324 MP 003 SYSPRFLR PRBINIT  8370                 Call       Monitor pause requested
    0 17:38:17.204508         43.471 MP 003 SYSPRFLR PRBSTART 1760                   Call       Start of block filter
   11 17:38:17.218379          48.874 DB 000                    0000                              00010/00032 S1
   12 17:38:17.218941         48.897 DA 000                   0000                              00010/00032 S1   Rsp: 0
   13 17:38:17.218944          48.910 PL 000                    0000                               Execute PRFDEMO/XPROF
   14 17:38:17.218945          48.916 PS 001 PRFDEMO  XPROF    0000                     PgmStart   00010/00032 Type: P
   15 17:38:17.218956          48.979 IB 001 PRFDEMO  XPROF    0300                     Input      Out: 133 In: 0
   16 17:38:17.219235          49.046 IA 001 PRFDEMO  XPROF    0300                     Input      Out: 133 In: 80
   17 17:38:17.219258          49.182 DB 001 PRFDEMO  XPROF    0370                     Callnat    00010/00032 S1
   18 17:38:17.220426         49.211 DA 001 PRFDEMO  XPROF    0370                   Callnat    00010/00032 S1   Rsp: 0
   19 17:38:17.220427          49.216 DB 001 PRFDEMO  XPROF    0370                     Callnat    00010/00032 S1
   ...
```

Ablaufverfolgung (Tracing) beim Messen der Natural-Codeabdeckung

Beim Messen der **Natural-Codeabdeckung (Code Coverage)** stehen zwei Trace-Optionen zur Verfügung:

- Die Ereignisverfolgung (**Event Trace**) listet alle überwachten Ereignisse auf. Dies sind standardmäßig die Ereignisse Program Start (`PS`) und Natural Statement (`NS`). Wenn der Profiler das GP liest, schreibt er außerdem einen Ereignisverfolgungseintrag für jedes im GP gefundene Statement. Diese Einträge werden durch den Pseudo-Ereignistyp `GP` angezeigt.
- Die Codeabdeckungsverfolgung (Coverage Trace) listet alle Datensätze auf, die in die Natural-Abdeckungsressourcendatei (Natural Coverage Resource File) geschrieben werden.

Wenn die Abdeckungsressource mit der Profiler Utility-Funktion `READ` gelesen wird, ist die folgende Trace-Option verfügbar:

Die interne Datenverfolgung listet alle Datensätze auf, die aus der Natural-Abdeckungsressourcendatei gelesen werden.

Um die Ablaufverfolgung für die Codeabdeckung zu aktivieren:

1 Aktivieren Sie die Ereignisverfolgung, indem Sie das folgende untergeordnete Schlüsselwort der `COVERAGE`-Funktion der Profiler Utility angeben:

```
TRACE-EVENT=ON
```

2 Aktivieren Sie die Ablaufverfolgung für die Codeabdeckung, indem Sie das folgende untergeordnete Schlüsselwort der `READ`-Funktion der Profiler Utility angeben:

```
TRACE-COVERAGE=ON
```

3 Aktivieren Sie die interne Ablaufverfolgung, indem Sie das folgende untergeordnete Schlüsselwort der `READ`-Funktion der Profiler Utility angeben:

```
TRACE=9
```

Die folgende Tabelle beschreibt die in den Ablaufverfolgungen (Traces) aufgeführten Eigenschaften und gibt mit (X) an, für welchen Trace-Typ die Daten bereitgestellt werden.

Event Trace = Ereignisverfolgung, Coverage Trace = Codeablaufverfolgung, Internal Trace = Interne Ablaufverfolgung:

Eigenschaft	Event Trace	Coverage Trace	Internal Trace	Beschreibung
Count	X	X	X	Der Ereigniszähler.
Ev	X	X	X	Der Ereignistyp. Siehe *Gesammelte Ereignisse und Daten*. Der Pseudo-Ereignistyp `GP` zeigt ein im GP gelesenes Statement an.
Library	X	X	X	Der Name der Natural-Library, die das Programm/Objekt enthält.
Program/Object	X	X	X	Der Name des Natural-Programms/Objekts.
Ty	-	X	X	Der Objekttyp, z.B. `P`, für Programm.
CC-Lib	X	X	X	Der Name der Natural-Library, die den Copycode enthält (wenn Copycode aktiv ist).
CC-Name	X	X	X	Der Name des Copycodes.
Line	X	X	X	Die Nummer der Quellcodezeile.
Statement	X	X	-	Das ausgeführte Natural-Statement. Aus technischen Gründen gibt es keine Eins-zu-Eins-Beziehung zwischen einem Natural-Quellcode-Statement und dem entsprechenden Objektcode im katalogisierten Objekt. Daher können sich die in der Profiler-Ereignisverfolgung aufgeführten Statements von den Statements in der Quelle unterscheiden. Für die Codeabdeckung wird der Statement-Objektcode nicht in der Ressourcendatei gespeichert. Daher kann er nur während der Datensammlung aufgelistet werden.

Eigenschaft	Event Trace	Coverage Trace	Internal Trace	Beschreibung
GP-Offset	X	-	-	Der Offset im GP. Er identifiziert eindeutig das Statement-Element zum Zeitpunkt der Ausführung.
Size	X	-	-	Die Größe des Statement im GP.
CC-ID	-	X	X	Die Copycode-ID. Sie identifiziert die Copycode-Instanz im GP eindeutig. Das Programm erhält die Copycode-ID `0`.
Par-CC	-	X	X	Nur für Copycode. Die übergeordnete Copycode-ID, die die Copycode-ID des Objekts/Copycodes ist, das/der den aktuellen Copycode enthält.
INCL	-	X	-	Nur für Copycode. Die Zeilennummer des `INCLUDE`-Statement, das den Copycode enthält.
FirstS	-	X	X	Die Kennung (ID) des ersten Statement des Objekts oder Copycodes.
Stmts	-	X	X	Die Gesamtzahl der ausführbaren Statements im Objekt, wobei alle `INCLUDE`-Statements aufgelöst werden.
Item	X	X	X	Die Elementkennung (Item ID) des Statement. Sie identifiziert eindeutig das Statement in der Ressourcendatei.
Cover	X	X	X	Das Abdeckungskennzeichen (Flag) des Statement: (`0` oder `1`). Wenn das GP gelesen wird, werden alle Flags mit `0` initialisiert. Immer wenn ein Statement ausgeführt wird, wird das Flag auf `1` gesetzt.
Same	X	-	-	Wenn das GP gelesen wird, zeigt es an, wie viele nachfolgende Statements den gleichen Objektcode und die gleiche Zeilennummer haben. Diese Statements werden verworfen. Immer wenn ein verworfenes Statement überwacht wird, wird das Statement, das sich vor dem verworfenen Statement befindet, als abgedeckt markiert. Die Spalte zeigt den GP-Offset des markierten Statement.

Beispiel für einen Coverage Trace

Im folgenden Beispiel schreibt die Funktion `COVERAGE` der Profiler Utility die Abdeckungsverfolgung:

```
FUNCTION=COVERAGE           /* Initialize code coverage
  TRACE-COVERAGE=ON         /* Write coverage trace
```

Das **Coverage Trace** wird in die Standardausgabe der Profiler-Monitor-Session (`MONPRINT`-Dataset) geschrieben:

```
                      Natural Coverage Trace                                ↩

                      ----------------------                                ↩

Count Ev Library  Program  Ty CC-Lib   CC-Name  Line Statement  CC-ID Par-CC INCL ↩
 FirstS  Stmts    Item Cover
    1 PI COVDEMO  TESTCOVN N                                      0      0 0000 ↩
      1     44         1
    2 NS COVDEMO  TESTCOVN                     0130 Compute        0            ↩
                     4 0
    3 NS COVDEMO  TESTCOVN                     0140 Print          0            ↩
                     5 0
    4 NS COVDEMO  TESTCOVN                     0150 Goto           0            ↩
                     6 0
    5 NS COVDEMO  TESTCOVN                     0310 If             0            ↩
                    39 0
    6 NS COVDEMO  TESTCOVN                     0310 Compute        0            ↩
                    40 0
    7 NS COVDEMO  TESTCOVN                     0320 Print          0            ↩
                    41 0
    8 NS COVDEMO  TESTCOVN                     0340 Compute        0            ↩
                    43 0
    9 NS COVDEMO  TESTCOVN                     0100 Reset          0            ↩
                     1 1
   10 NS COVDEMO  TESTCOVN                     0120 Reset          0            ↩
                     2 1
   11 NS COVDEMO  TESTCOVN                     0130 If             0            ↩
                     3 1
   12 NS COVDEMO  TESTCOVN                     0150 If             0            ↩
                     7 1
...
```

Interner Trace

Das Profiler-interne Trace schreibt Profiler-Meldungen wie z.B. Fehler oder Warnungen.

Das interner Trace kann für Folgendes aktiviert werden:

- Die Profiler-Monitor-Sessions (**Datensammlung/Data Collection**). Die Daten werden in die Standardausgabe der Monitor-Session geschrieben.
- Die Profiler-Datenverarbeitungsfunktionen. Die Daten werden in die Standardausgabe geschrieben.

- Die Profiler Rich GUI-Schnittstelle bietet ein aufklappbares Feld für die interne Trace-Aktivierung. Die Daten werden in die Standardausgabe geschrieben.

Um die interne Ablaufverfolgung für die Profiler-Trace-Session oder die Datenverarbeitungsfunktionen zu aktivieren:

- Geben Sie das folgende Profiler-Schlüsselwort ein:

```
TRACE=n
```

Dabei ist *n* die Trace-Ebene (siehe ***Trace-Ebenen***).



Anmerkungen:

1. Standardmäßig (wenn `TRACE` nicht angegeben ist) wird Trace-Ebene 2 (Warnungen) verwendet.
2. Die Ablaufverfolgung wird aktiviert, sobald das Schlüsselwort `TRACE` angegeben ist. Es wird daher empfohlen, das `TRACE`-Schlüsselwort möglichst bald anzugeben.
3. Wenn Sie die Profiler Utility mehrmals in dem Job ausführen, müssen Sie bei jeder Ausführung das Schlüsselwort `TRACE` angeben.

Um die interne Ablaufverfolgung für die Profiler-Monitor-Session zu aktivieren:

- Geben Sie das folgende untergeordnete Schlüsselwort der `INIT`- oder der `COVERAGE`-Funktion der Profiler Utility ein:

```
TRACE-MONITOR=n
```

Dabei ist *n* die Trace-Ebene (siehe ***Trace-Ebenen***).



Anmerkung: Standardmäßig (wenn `TRACE-MONITOR` nicht angegeben ist) wird Trace-Ebene 3 (Statistik) verwendet.

Trace-Ebenen

Die Trace-Ebene, die von den Profiler Trace- und Monitor-Sessions und von den Profiler-Datenverarbeitungsfunktionen verwendet werden, sind in der folgenden Tabelle aufgeführt. Im Allgemeinen enthält eine höhere Trace-Ebene auch die Informationen der unteren Trace-Ebenen. Wenn Sie beispielsweise Trace-Ebene 3 (Statistik) auswählen, werden auch Fehlermeldungen und Warnungen protokolliert.

Wir empfehlen, mindestens Trace-Ebene 2 (Warnungen) zu verwenden, damit Fehlermeldungen und Warnungen protokolliert werden. Für die Profiler-Monitor-Session ist Trace-Ebene 3 (Statistik)

eine gute Wahl. Dann werden die Statistiken des Profiler-Laufs gedruckt (siehe Profiler-Statistiken). Höhere Trace-Ebenen für die Monitor-Session können extrem ausführlich sein und die Ausgabe kann mit dem Ereignis-Trace (falls aktiviert) vermischt werden.

Trace-Ebene	Name	Beschreibung
0	No trace	Die interne Ablaufverfolgung des Profilers ist deaktiviert.
1	Error	Fehlermeldungen werden protokolliert.
2	Warning	Warnungen werden protokolliert.
3	Statistics	Trace-Session: Die Werte, die für die Profiler Utility -Funktion `INIT` oder `COVERAGE` verwendet werden, werden gedruckt. Monitor-Session: Die ***Profiler-Statistikdaten*** werden gedruckt. Datenkonsolidierung: Die ***Profiler-Statistikdaten*** werden einschließlich der Konsolidierungsstatistiken gedruckt.
4	Function	Nachrichten für verwendete **Profiler Utility-Schlüsselwörter** (`FUNKTION`, `FILTER` usw.) werden protokolliert.
5	Block	Die Statistiken für jeden Datenblock, der in die Profiler-Ressourcendatei geschrieben wurde, werden gedruckt.
6	Details	Detaillierte Informationen werden protokolliert.
7		Nicht benutzt.
8		Nicht benutzt.
9	Data	Abdeckungsressourcendaten beim Lesen einer NCVF-Abdeckungsressourcendatei verfolgen.
10	Internal	Interne Verwendung.

Beispiel

Im folgenden Beispiel wird das interne Trace des Profilers für die Trace-Session auf Ebene `4` (Funktion) gesetzt:

```
* Set Profiler internal trace
TRACE=4                     /* Trace level
```

Ausgabe der Profiler-Trace-Session für Trace-Ebene 4:

```
PRBMAIN : Profiler trace level: 4
PRBMAIN : Profiler On-Error: Terminate
********************************************************************************
* 15:45:14          ***** NATURAL PROFILER UTILITY *****         2014-12-17
* User SAGPRFD1               - Function INIT -                     PRBINIT
*
* Keyword              Wert
* -------------------- ------------------------------------------------------
* Resource             ON
* Resource-Lib         SAGPROF
* Resource-Name        Test01.nprf
* Replace              Y
* Wait-Full            60
* Wait-Empty           60
* Sampling             OFF
* Consolidate          OFF
* Trace-Monitor        5
* Trace-Event          OFF
* Trace-Consolidate    OFF
* -------------------- ------------------------------------------------------
********************************************************************************
PRBINIT : Profiler INIT function - Start monitor session.
PRBINIT : Profiler INIT function - Monitor session started. Time: 1.0 sec.
PRBINIT : Set trace session Id ...: 0000000000000001
PRBINIT : Set monitor session Id .: 0000000100000000
PRBINIT : Trace session successfully initialized.
PRBFEVEN: Event filter: SI ST DB DA PL PS PT PR IB IA E CB CA U RS RI RO NS
PRBSTART: Profiling started.
PRBMAIN : Profiler - End of input.
********************************************************************************
...
Output of the application
...
```

Profiler-Statistikdaten

Zusätzlich zu den Ereignisdaten sammelt der Profiler statistische Daten, die in die Profiler-Ressourcendatei geschrieben werden.

Die Profiler Utility bietet die folgenden Optionen zum Schreiben und Anzeigen von Profiler-Statistikdaten:

- Statistikdaten in die Standardausgabe der Profiler-Monitor-Session (`MONPRINT`-Dataset) schreiben, während die Anwendung profiliert wird oder die Messung der Codeabdeckung durchgeführt wird.
- Statistikdaten in die Standardausgabe schreiben, während die Daten konsolidiert werden.

- Statistikdaten schreiben, während eine Profiler-Ressourcendatei mit der Profiler Utility-`READ`-Funktion gelesen wird.
- Eine Ressource über die Natural Profiler Rich GUI-Schnittstelle auswählen, um die entsprechenden Eigenschaften und Statistiken anzuzeigen.

Um Profiler-Statistikdaten zu schreiben, führen Sie einen der folgenden Schritte aus:

- Geben Sie das folgende untergeordnete Schlüsselwort der `INIT`- oder der `COVERAGE`-Funktion der Profiler Utility ein:

```
TRACE-MONITOR=3
```

  oder eine höhere Trace-Ebene (siehe ***Trace-Ebenen***). Trace-Ebene 3 ist auch die Standardebene für die Profiler-Monitor-Session.

- Geben Sie das folgende Schlüsselwort ein, bevor Sie die `CONSOLIDATE`-Funktion der Profiler Utility starten:

```
TRACE=3
```

  oder eine höhere Trace-Ebene (siehe ***Trace-Ebenen***).

- Geben Sie das folgende untergeordnete Schlüsselwort der `READ`-Funktion der Profiler Utility ein:

```
STATISTICS=ON
```

Die Profiler-Statistikdaten werden in Kategorien angezeigt, wobei die Eigenschaften eines ähnlichen Typs zusammengefasst werden. Folgende Kategorien stehen zur Verfügung:

- Allgemeine Informationen
- Profiler-Ressourcendatei
- Monitor-Session
- Trace-Session
- Datenverarbeitung
- Ereignistyp-Statistiken
- Monitor-Pause-Statistiken
- Datenkonsolidierung
- Abdeckung (Coverage)
- Transaktionen

Anmerkung: Die im folgenden Abschnitt aufgeführten Eigenschaften sind die Eigenschaften, die der Profiler in allen Umgebungen bereithält. Die Profiler-Statistiken enthalten nur die Eigenschaften, die für den aktuellen Lauf relevant sind. Daher werden nicht alle im Abschnitt aufgeführten Eigenschaften in jedem Fall angezeigt.

Allgemeine Informationen

Informationen zur Umgebung und Natural Profiler anzeigen.

Eigenschaft	Beschreibung
Machine class	Der Name der Maschinenklasse, auf der die Natural-Anwendung läuft.
Environment	Die Umgebung, in der die Natural-Anwendung läuft, z.B. NaturalONE, Batch oder RPC.
Codepage	Die Codepage, die während der Überwachung der Natural-Anwendung verwendet wurde.
User	Die Kennung (ID) des Benutzers, der die Anwendung ausführt (Wert der Systemvariablen `*USER`). Bei einem Batch-Job kann es der Name des Jobs sein.
Profiler version	Die interne Version des Profilers. NaturalONE-Umgebung: Die Version des Profilers auf dem Server.
Profiler revision	Die interne Revision des Profilers.
Profiler revision date	Datum und Uhrzeit der Erstellung der Profiler-Revision.
Profiler client version	NaturalONE-Umgebung: Die Version des Profiler-Clients.
Profiler trace library	NaturalONE-Umgebung: Der Name der Natural-Library, die das interne Profiler-Trace und das Profiler-Ereignis-Trace enthält.
Profiler trace level	Die Ebene des internen Trace des Profilers.
Profiler trace member	NaturalONE-Umgebung: Der Name des Natural-Textobjekts, das das interne Profiler-Trace enthält.
Profiler event trace	Zeigt an, ob die Profiler-Ereignisverfolgung aktiviert wurde (`ON/OFF`).
Profiler event trace member	NaturalONE-Umgebung: Der Name des Natural-Textobjekts, das die Profiler-Ereignisverfolgung enthält.
Utility trace level	NaturalONE-Umgebung: Die Ebene des Natural Utility Trace.

Profiler-Ressourcendatei

Informationen zu Profiler-Ressourcendateien anzeigen.

Eigenschaft	Einheit	Beschreibung
Resource name		Der Name der Natural Profiler-Ressourcendatei.
Resource type		Der Typ der Natural Profiler-Ressourcendatei: Natural Profiler Resource File (NPRF), Natural Profiler Resource Consolidated (NPRC) oder Natural Code Coverage File (NCVF).
Resource short name		Großrechner: Der Kurzname der Natural Profiler-Ressourcendatei.
Resource library		Der Name der Natural-Library, die die Natural Profiler-Ressourcendatei enthält.
Resource DBID		Die Datenbankkennung (DBID) der Natural-Library, die die Natural Profiler-Ressourcendatei enthält.

Eigenschaft	Einheit	Beschreibung
Resource FNR		Die Dateinummer (FNR) der Natural-Library, die die Natural Profiler-Ressourcendatei enthält.
Resource allocation date	*yyyy-mm-dd hh:ii:ss.t*	Datum und Uhrzeit, wann die Natural Profiler-Ressourcendatei angelegt wurde.
Resource size	Bytes	Die Größe der Natural Profiler-Ressourcendatei. Umfasst die Ressourcen-Headers, die Ereignisdaten und die Statistikdaten. Die Ressourcengröße wird unabhängig davon berechnet, ob die Ressource angelegt wird oder nicht.
Resource block size	Bytes	Die maximale Größe eines Ressourcenblocks. Ein Ressourcenblock besteht aus einem Ressourcenblock-Header und einem Datenblock.
Resource version		Die Version des Natural Profiler-Ressourcen-Layouts.

Monitor-Session

Statistiken der Profiler-Monitor-Session anzeigen.

Eigenschaft	Einheit	Beschreibung
Monitor start time	*yyyy-mm-dd hh:ii:ss.t*	Datum und Uhrzeit des Starts der Monitor-Session.
Monitor end time	*yyyy-mm-dd hh:ii:ss.t*	Datum und Uhrzeit des Endes der Monitor-Session.
Monitor elapsed time	sec	Die gesamte verstrichene Zeit, die von der Monitor-Session verbraucht wurde.

Trace-Session

Statistikdaten der Profiler-Trace-Session anzeigen. Die Profiler-Trace-Session beinhaltet auch die Anwendungsausführung.

Eigenschaft	Einheit	Beschreibung
First library		Die erste Library, die überwacht wurde. Die Libraries `SYSTEM`, `SYSLIB*` und `SYSPRF*` werden ignoriert.
First program		Das erste Programm, das überwacht wurde.
Highest level		Höchste Ebene der überwachten Natural-Objekte.
Trace start time	*hh:ii:ss.microsec*	Die Startzeit der Ablaufverfolgung. Bei NaturalONE ist dies die Zeit des `SI`-Ereignisses (Session Initialization). Im Batch-Modus ist die Sitzung bereits initialisiert, wenn die Überwachung gestartet wird. Daher ist die Startzeit die Zeit des ersten Ereignisses (normalerweise ein Monitor-Pause-Ereignis).
Trace end time	*hh:ii:ss.microsec*	Die Endezeit der Ablaufverfolgung. Dies ist im Allgemeinen die Zeit des `ST`-Ereignisses (Session Termination).

Eigenschaft	Einheit	Beschreibung
Trace elapsed time	sec	Die verstrichene Zeit, die von der Trace-Session von der Startzeit bis zur Endezeit verbraucht wird.
Application CPU time	ms	Die gesamte CPU-Zeit, die von der Anwendung verbraucht wird.
Monitor CPU time	ms	Die gesamte CPU-Zeit, die der Natural Data Collector verbraucht. Diese Zeit wird bei einem Natural-Linux- oder Windows-Server nicht erfasst.
Total CPU time	ms	Die gesamte CPU-Zeit, die von der Trace-Session verbraucht wird. Es ist die Summe der CPU-Zeit der Anwendung und der CPU-Zeit des Monitors.
Sampling interval	microsec	Die Stichprobenintervallzeit (CPU-Zeit in Mikrosekunden). Der Wert Null (0) bedeutet, dass keine Stichprobennahme aktiv war.
Data pool empty		Die Anzahl der Profiler-Leseanforderungen, die den Profiler-Daten-Pool leer (und eine Session aktiv) vorfanden.
Data pool empty after full		Die Anzahl der Profiler-Leseanforderungen, die den Profiler-Daten-Pool leer vorfanden, obwohl er zuvor voll war. Wenn dieser Zähler größer als 0 ist, ist der Profiler-Daten-Pool zu klein, was zu einer schlechten Performance führt.
Data pool overflow		Die Anzahl der Profiler-Daten-Pool-Überläufe (mit verlorenen Daten). Daten-Pool-Überläufe sollten nicht mehr auftreten. Diesen Wert gibt es nur noch aus Gründen der Abwärtskompatibilität mit früheren Versionen von Natural.
No session active		Die Anzahl der Leseanforderungen, bei denen der Profiler-Daten-Pool leer und keine Trace-Session aktiv war. Dies kann nur bei Profiler-Leseanforderungen geschehen, die vor der Session-Initialisierung oder nach Beendigung der Session eingereicht wurden.

Datenverarbeitung

Statistiken über die Datenverarbeitung, Komprimierung und Übertragung anzeigen.

Eigenschaft	Einheit	Beschreibung
Number of events		Die Gesamtzahl der Ereignisse.
Highest event number		Die höchste Ereigniszahl, wie sie vom Natural Data Collector angegeben wurde. Beachten Sie, dass der Natural Data Collector nur Nicht-Statement-Ereignisse zählt, wenn er von NaturalONE aufgerufen wird. Im Batch-Modus hängt es vom Statement-Filter ab, ob Statement-Ereignisse gezählt werden oder nicht.
Number of data blocks		Die Anzahl der Ereignisdatenblöcke, die an NaturalONE gesendet oder in die Ressource geschrieben werden.

Eigenschaft	Einheit	Beschreibung
Utility buffer size	Bytes	Die Größe des Utility-Puffers, der für die Datenübertragung vom Server zu NaturalONE verwendet wird. Im Allgemeinen enthält der Puffer die Header-Informationen und funktionsspezifischen Daten.
Data block size	Bytes	Die maximale Menge an Ereignisdaten, die in einem Aufruf vom Server zu NaturalONE übertragen werden kann. Die gleiche Datenblockgröße wird zum Speichern der Ereignisdaten in der Ressourcendatei verwendet.
RDC data length	Bytes	Die Gesamtgröße der vom Natural Data Collector erhaltenen Daten.
Uncompressed data length	Bytes	Die Gesamtgröße der Profiler-Daten im unkomprimierten Format.
Compressed data length	Bytes	Die Gesamtgröße der komprimierten Daten, die an NaturalONE gesendet oder in die Profiler-Ressourcendatei geschrieben werden.
Identical bytes trimmed left		Die Anzahl der identischen Bytes, die bei der Vorwärtsdatenkomprimierung links abgeschnitten wurden.
Blanks trimmed right		Die Anzahl der Leerzeichen, die bei der rückwärts gerichteten Datenkomprimierung rechts abgeschnitten wurden.
Compression header length	Bytes	Die Gesamtgröße der Komprimierungs-Header, die mit jedem komprimierten Ereignisdatensatz gespeichert werden.
Compression rate	Prozent	Der Prozentsatz der Datenreduktion durch die Komprimierung. Je höher die Komprimierungsrate, desto weniger Daten müssen übertragen oder gespeichert werden. Die Formel der Komprimierungsrate wird unten beschrieben.
Events/block		Die durchschnittliche Anzahl der Ereignisse, die in einem Ereignisdatenblock enthalten sind.
Bytes/event		Die durchschnittliche Länge eines komprimierten Ereignisdatensatzes in Bytes. Diese Eigenschaft ist nicht für konsolidierte oder Abdeckungsdaten verfügbar.

Die Komprimierungsrate wird mittels folgender Formel berechnet:

$$\text{CompressionRate} := 100 \times \frac{\text{BytesTrimmedLeft} + \text{BytesTrimmedRight} - \text{CompressionHeaderLength}}{\text{UncompressedDataLength}}$$

Ereignistyp-Statistiken

Statistiken der Ereignistypen anzeigen.

Eigenschaft	Beschreibung
Unknown event	Die Anzahl der unbekannten Ereignisse.
Session initialization	Die Anzahl der Session-Initialisierungsereignisse.
Session termination	Die Anzahl der Session-Beendigungsereignisse.
Program load	Die Anzahl der Programmlade-Ereignisse.
Program start	Die Anzahl der Programmstart-Ereignisse.
Program termination	Die Anzahl der Programmbeendigungsereignisse.
Program resume	Die Anzahl der Programmwiederaufnahme-Ereignisse.
Program information	Die Anzahl der Programm-Informationsereignisse.
Before database call	Die Anzahl der Vor-Datenbankaufruf-Ereignisse.
After database call	Die Anzahl der Nach-Datenbankaufruf-Ereignisse.
Before terminal I/O	Die Anzahl der Vor-Terminal-E/A-Ereignisse.
After terminal I/O	Die Anzahl der Nach-Terminal-E/A-Ereignisse.
Before external program call	Die Anzahl der Vor-Externem-Programmaufruf-Ereignisse.
After external program call	Die Anzahl der Nach-Externem-Programmaufruf-Ereignisse.
Runtime error	Die Anzahl der Laufzeitfehler-Ereignisse.
Natural statement	Die Anzahl der Natural Statement-Ereignisse. Aus technischen Gründen können mehrere Natural-Statements zu einem Statement-Ereignis zusammengeführt werden, und umgekehrt kann ein Natural-Statement mehrere Statement-Ereignisse abdecken.
Outbound RPC message	Die Anzahl der ausgehenden RPC-Nachrichtenereignisse.
Inbound RPC message	Die Anzahl der eingehenden RPC-Nachrichtenereignisse.
Start RPC request execution	Die Anzahl der RPC-Ausführungsanforderungsstart-Ereignisse.
RPC Wait for Client	Die Anzahl der RPC-Wartet-auf-Client-Ereignisse.
User trace call	Die Anzahl der benutzerdefinierten Ereignisse.
Monitor pause	Die Anzahl der Monitor-Pause-Ereignisse.
Monitor filter	Die Anzahl der Monitor-Filter-Ereignisse. Filter-Ereignisse werden nicht aufgezeichnet.

Monitor-Pause-Statistiken

Statistiken der Monitor-Pausentypen-Ereignissen anzeigen.

Eigenschaft	Beschreibung
Pause - unknown type	Die Anzahl der Monitor-Pause-Ereignisse mit unbekanntem Pausentyp.
Pause - requested	Die Anzahl der angeforderten Monitor-Pause-Ereignisse.
Pause - start of block filter	Die Anzahl der Monitor-Pause-Ereignisse, die durch den Start eines Blockfilters verursacht werden (Library-, Programm-, Zeilen-, FNAT-, Ereignisanzahl- oder Zeit-Filter).
Pause - data pool full	Die Anzahl der Monitor-Pause-Ereignisse, die durch einen vollen Daten-Pool verursacht werden.
Pause - data pool overflow	Die Anzahl der Monitor-Pause-Ereignisse, die durch einen Daten-Pool-Überlauf verursacht werden.

Datenkonsolidierung

Statistiken der Datenkonsolidierung anzeigen.

Eigenschaft	Einheit	Beschreibung
Consolidation		Gibt an, ob die Profiler-Daten konsolidiert sind (`ON/OFF`). Bei der Konsolidierung werden ähnliche Ereignisse in einem Konsolidierungsdatensatz zusammengefasst.
Consolidation records		Die Gesamtzahl der Konsolidierungsdatensätze. Im Allgemeinen umfasst ein Konsolidierungsdatensatz mehrere Ereignisse.
Consolidation elapsed time	Sekunden	Die verstrichene Zeit in Sekunden, die für die Datenkonsolidierung mit der Profiler Utility-Funktion `CONSOLIDATE` erforderlich ist. Dieser Wert wird nicht angegeben, wenn die Konsolidierung während der Datensammlung durchgeführt wird (Profiler Utility-Funktion `INIT`). In diesem Fall ist die für die Konsolidierung erforderliche Zeit in der **Monitor elapsed time** enthalten.
Consolidation factor		Die durchschnittliche Anzahl der Ereignisse, die zu einem Konsolidierungsdatensatz zusammengefasst sind. Je höher der Konsolidierungsfaktor, desto besser die Konsolidierung. `ConsolidationFactor := NumberOfEvents / ↩ ConsolidationRecords`
Consolidation records/block		Die durchschnittliche Anzahl der Konsolidierungsdatensätze, die in einem Datenblock enthalten sind.
Bytes/consolidation record		Die durchschnittliche Länge in Bytes eines komprimierten Konsolidierungsdatensatzes.

Eigenschaft	Einheit	Beschreibung
Consolidate I/O time		Gibt an, ob die E/A- und Natural RPC-Client-Zeit in den konsolidierten Daten enthalten sind.

Abdeckung (Coverage)

Statistiken zur Natural-Codeabdeckung anzeigen.

Anmerkung: Statistikdaten zur Natural-Codeabdeckung werden nur auf dem Großrechner gesammelt.

Eigenschaft	Beschreibung
Coverage	Zeigt an, ob die Natural-Codeabdeckung durchgeführt wird(`ON/OFF`).
Missed statements recorded	Zeigt an, ob nicht abgedeckte („missed“) Statements aufgezeichnet werden (`ON/OFF`).
Coverage records	Die Gesamtzahl der Abdeckungsdatensätze. Dies sind Programminformationen und Natural-Statement-Aufzeichnungen.
Program information records	Die Anzahl der in die Ressourcendatei geschriebenen Programminformationsdatensätze. Jeder Programminformationsdatensatz enthält programm- und copycode-bezogene Informationen.
Coverage records/block	Die durchschnittliche Anzahl der Abdeckungsdatensätze, die in einem Datenblock enthalten sind.
Bytes/coverage record	Die durchschnittliche Länge in Bytes eines komprimierten Abdeckungsdatensatzes.
Programs covered	Die Anzahl der abgedeckten Programme.
Programs NOC-ed	Die Anzahl der abgedeckten Programme, die mit dem Natural Optimizer Compiler kompiliert wurden.
Statement coverage	Der Prozentsatz der Statements aller aufgerufenen Programme, die von der Anwendung abgedeckt wurden.
Statements covered	Die Anzahl der abgedeckten (ausgeführten) Statements.
Statements total	Die Gesamtzahl der ausführbaren Statements aller Programme, auf die zugegriffen wird.

Transaktionen

Statistikdaten der ausgeführten Transaktionen anzeigen.

Anmerkung: Transaktionsstatistikdaten sind nur verfügbar, wenn Profiler-Daten mit der auf `TRANSACTION=ON` gesetzten `CONSOLIDATE`-Funktion konsolidiert wurden.

Eigenschaft	Einheit	Beschreibung
Transaction		Gibt an, ob Transaktionsidentifikatoren zu den konsolidierten Datensätzen hinzugefügt werden (`ON`/`OFF`). Transaktionsidentifikatoren sind erforderlich, um die Antwortzeit für Transaktionen auszuwerten.
Number of transactions		Die Gesamtzahl der verarbeiteten Transaktionen. Eine Transaktion ist der Code, der zwischen zwei aufeinanderfolgenden Terminal-E/A-Vorgängen ausgeführt wird.
Total response time	Sekunden	Die gesamte verstrichene (elapsed) Zeit, die für die Verarbeitung aller Transaktionen aufgewendet wurde.
Average response time	Sekunden	Die durchschnittlich aufgewendete verstrichene (elapsed) Zeit pro Transaktion.
Max response time	Sekunden	Die maximale (höchste) verstrichene (elapsed) Zeit, die für die Verarbeitung von Transaktionen aufgewendet wird.
ID max response time		Der Transaktionsidentifikator der Transaktion mit der höchsten verstrichenen (elapsed) Zeit. Zusätzlich zum Transaktionsidentifikator gibt die Statistik den Programmnamen, den Copycode-Namen (falls vorhanden) und die Programmzeile an, von der aus der Terminal-E/A-Vorgang ausgegeben wurde.

Weitere Informationen zur Auswertung von Transaktionen finden Sie unter *Transaktionszusammenfassung*.

54 Natural Profiler Rich GUI-Schnittstelle

Die **Tools and Utilities** in NaturalONE bieten eine Rich GUI-Schnittstelle für den Profiler, das die Auswertung der Profiler-Daten in einer grafischen, interaktiven Browser-Oberfläche ermöglicht. In diesem Kapitel erhalten Sie einen kurzen Überblick über dieses Tool. Weitere Informationen siehe *NaturalONE* > *Using NaturalONE* > *Rich GUI Interface of the Natural Profiler*-Dokumentation.

Über die Rich GUI-Schnittstelle für den Natural Profiler können Sie die Profiler-Ressourcendateien lesen und analysieren. Wie eine Profiler-Ressourcendatei erstellt wird, erfahren Sie in den Kapiteln *Profiler Utility im Online-Modus benutzen* und *Profiler Utility im Batch-Modus benutzen*.

Um die Natural Profiler Rich GUI-Schnittstelle zu starten:

1 Starten Sie NaturalONE.

2 Wählen Sie in Ihrem Arbeitsbereich ein beliebiges Natural-Projekt, eine Library oder ein Objekt aus, wo Sie das Natural Profiler Rich GUI-Schnittstelle starten möchten.

3 Rufen Sie das Kontextmenü auf und wählen Sie **NaturalONE** > **Tools and Utilities**, um die `SYSUTIL` Utility zu starten.

4 Markieren Sie **Natural Profiler** in der Baumstruktur und wählen Sie **Execute**.

Die Natural Profiler Rich GUI-Schnittstelle wird gestartet.

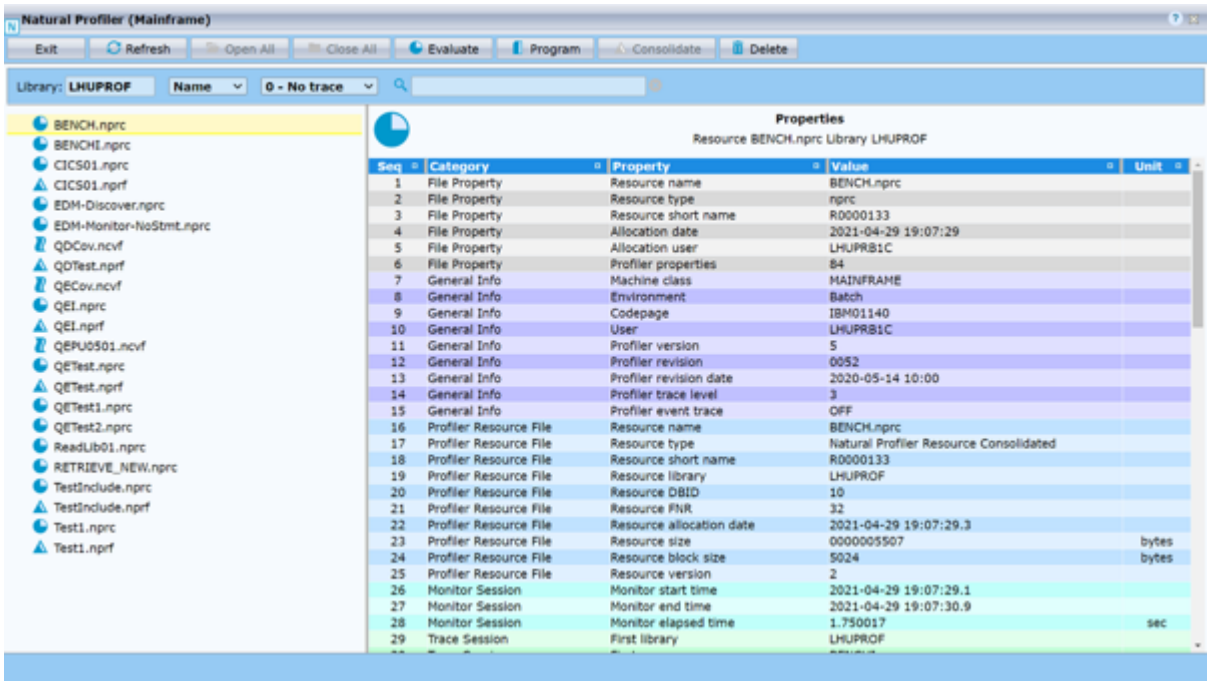
Die Natural Profiler Rich GUI-Schnittstelle bietet die folgenden Funktionen:

- Listet die Profiler-Ressourcen einer bestimmten Library auf.
- Zeigt die Eigenschaften und Statistiken der Profil-Erstellung für eine ausgewählte Profiler-Ressource an.
- Bietet die Auswertung der Profiler-Daten in interaktiven Kreisgrafiken. Diese zeigen die Verteilung der Profiler-KPIs für ausgewählte Kriterien, z.B. Verteilung der CPU-Zeit für Programme.

- Bietet die Analyse überwachter Programme und des Quellcodes eines ausgewählten Programms. Für einen ausgewählten KPI (z.B. CPU-Zeit) wird die Verteilung der KPI über die Programme und die Programmzeilen angezeigt.
- Ermöglicht die Konsolidierung der Profiler-Daten. Eine ausgewählte NPRF-Ressource wird gelesen und die Daten werden konsolidiert (aggregiert). Die resultierenden Daten werden unter Verwendung des gleichen Namens wie bei der entsprechenden NPRF-Ressource in eine NPRF-Ressource geschrieben.
- Sie können eine Profiler-Ressource-Datei löschen.
- Bietet eine Hilfe-Funktion, die eine detaillierte Beschreibung der aktuellen Seite liefert.

Die folgenden Abbildungen zeigen die Hauptseiten der Natural Profiler Rich GUI-Schnittstelle.

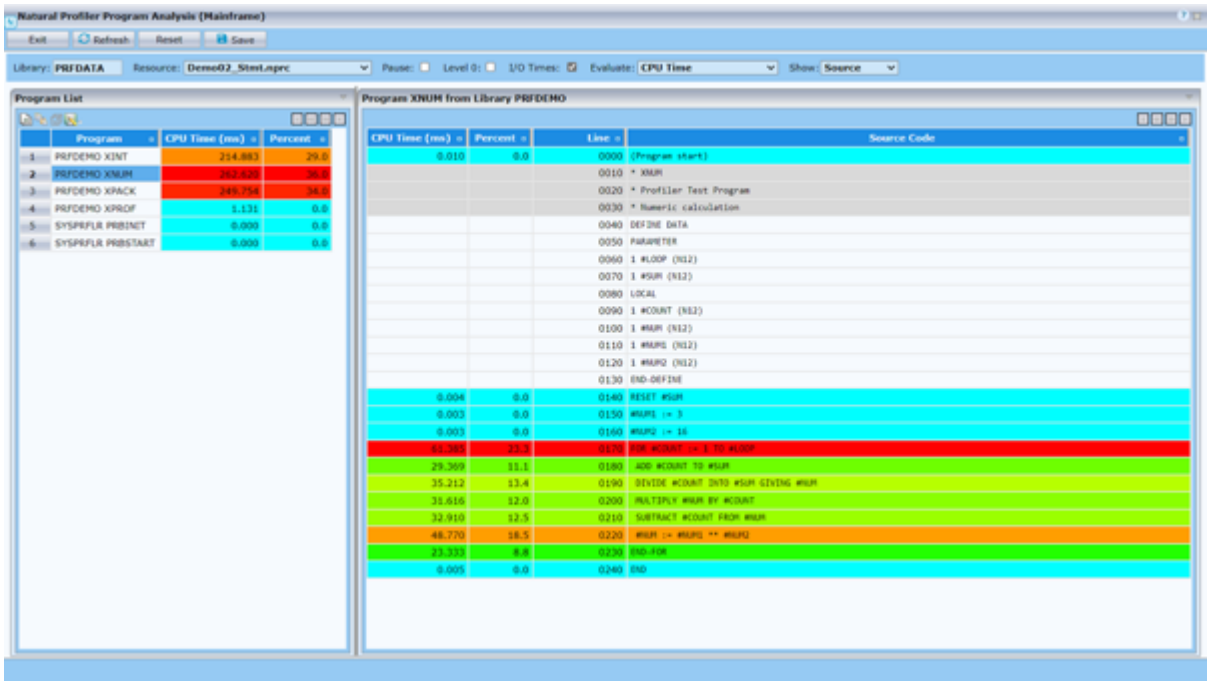
Die **Natural Profiler**-Seite erscheint als erste Seite nach dem Start. Sie enthält eine Liste der Ressourcen einer ausgewählten Library sowie die Eigenschaften und Statistikdaten einer ausgewählten Ressource.



Auf der Seite **Natural Profiler Data Evaluation** werden interaktive Kreisgrafiken für die Auswertung der Profiler-Daten verwendet:



Auf der Seite **Natural Profiler Program Analysis** erfolgt die Analyse des Quellcodes der profilierten Objekte:



XI Recording Utility

55 Recording Utility

- Verwendungszweck 522
- Aufgezeichnete Daten und Funktionen 523
- Aufzeichnen einer Session 523
- Wiedergabe einer Aufzeichnung 524
- Manipulieren einer Aufnahme 527

Mit diesem Dienstprogramm können Sie während einer Natural-Session eingegebene Kommandos und Eingabedaten aufzeichnen und die aufgezeichnete Sitzung wieder ausführen.

Verwandte Themen:

- *Bildschirme aufzeichnen* im Kapitel *Terminalkommandos nach Funktionsgruppen* in der *Terminalkommandos*-Dokumentation.
- *Recordings - Aufzeichnungen* im Abschnitt *Natural-Scratch-Pad-Datei* in der *Operations*-Dokumentation
- Profilparameter *RFILE - Datei für Aufzeichnung per Recording-Funktion* in der *Parameter-Referenz*-Dokumentation

Verwendungszweck

Die Recording Utility kann für folgende Zwecke benutzt werden:

- **Demonstration**
  Anstatt mehrere Kommandos einzugeben, wie zum Beispiel bei der Eingabe von Daten von Hand, können Sie eine aufgezeichnete Abfolge von Tastaturaktionen wiedergeben, um einen Standardvorgang zu demonstrieren.
- **Anwendungsentwicklung**
  Wenn Sie dieselben Änderungen bei mehreren Objekten anwenden, z. B., Programme oder Masken (Maps), können Sie mit einer Aufzeichnung den Arbeitsaufwand reduzieren und gleichzeitig sicherstellen, dass die Änderungen bei allen betroffenen Objekten tatsächlich gleich sind.
- **Testen**
  Sie können einen Standard-Testvorgang ausführen, indem Sie einfach eine Aufzeichnung wiedergeben.
- **Qualitätssicherung**
  Bevor und nachdem Sie Änderungen an einer Anwendung vorgenommen haben, können Sie eine Aufzeichnung abspielen und die Ergebnisse der beiden Durchläufe vergleichen, um sicherzustellen, dass bestimmte Dinge nicht von den Änderungen betroffen waren.
- **Benutzerschulung**
  Sie können die Wiedergabe von Aufzeichnungen in Schulungsprogramme für Benutzer einbinden, um ihnen bestimmte Abläufe zu zeigen. Außerdem können Sie die Tastatureingaben der Benutzer in einer Sitzung aufzeichnen und sie dann über Fehler oder Möglichkeiten zur effizienteren Ausführung von Aktionen informieren. Die Aufzeichnung von Benutzeraktionen kann Ihnen auch helfen, Fehler in der Benutzeroberfläche einer Anwendung zu erkennen.

Aufgezeichnete Daten und Funktionen

Die Recording Utility zeichnete Folgendes auf:

- Alle Eingabedaten und Kommandos (einschließlich Terminalkommandos), die im Bildschirm eingegeben werden.
- Alle Funktionstasten (PF-Tasten), die gedrückt werden.
- Die aktuelle Position des Cursors, so wie sie in der Systemvariablen `*CURSOR` enthalten ist (siehe *Systemvariablen*-Dokumentation).

Aufzeichnen einer Session

Im Folgenden wird beschrieben, wie Sie eine Aufzeichnung aktivieren und deaktivieren.

- Angabe von Libraries
- Aktivierung einer Aufzeichnung
- Deaktivierung einer Aufzeichnung

Angabe von Libraries

Um eine Library anzugeben, in der alle anschließenden Aufzeichnungen gespeichert werden sollen:

- Geben Sie folgenden Terminalkommando ein:

```
%B=library-name
```

Wenn Sie den Aufzeichnungsvorgang aktivieren, ohne eine Angabe für *library-name* gemacht zu haben, dann ist der Name der Library, in der die Aufzeichnung gespeichert wird, der gleich dem Wert der Systemvariablen `*INIT-USER` (siehe *Systemvariablen*-Dokumentation) zum Zeitpunkt der Aktivierung des Aufzeichnungsvorgangs.

Wenn Sie sich während des Aufzeichnens einer Session bei einer anderen Library anmelden, bleibt die Library, in der die Speicherung der Aufzeichnung stattfindet, dieselbe (d.h., entweder die mit `%B=` angegebene oder die in `*INIT-USER` enthaltene Library). Das bedeutet, dass Sie mit nur einer Aufzeichnung Tastaturaktionen über mehrere Anwendungen hinweg aufzeichnen können.

Aktivierung einer Aufzeichnung

Um eine Aufzeichnung zu aktivieren:

- Geben Sie folgendes Terminalkommando ein:

  ```
  %Bname
  ```

  Alle anschließend erfolgenden Tastaturaktionen werden aufgezeichnet.

name bezeichnet den Namen, unter dem die aufgezeichneten Daten in Source-Form als Natural-Objekts des Typs Recording gespeichert werden. Sie können diese Source wie jede andere Source behandeln (z.B. speichern, kopieren). Sie dürfen sie jedoch nicht bearbeiten: Aufzeichnungen enthalten Binärdaten, die durch einen Editor zerstört werden.

name kann nur einmal angegeben werden. Wenn in der für die Aufzeichnung benannten Libray bereits ein Record-Objekt mit dem gleichen Namen existiert, gibt Natural die Meldung `Error in recording activation` (Fehler bei der Aufzeichnungsaktivierung) zurück.

Vorsicht: Jede Situation, die zu einem Backout-Transaktion oder einem Rollback führt (z. B. eine Zeitüberschreitung bei Nichtaktivität), während eine Aufzeichnung läuft, wird einen Teil der Aufzeichnung löschen, wodurch die gesamte Aufzeichnung unbrauchbar wird.

Auf ein in die Aufzeichnung eingefügtes Terminalkommando `%Aname` sollte das Terminalkommando `%B` folgen, siehe Abschnitt *Aufzeichnen mit %A*.

Deaktivierung einer Aufzeichnung

Um eine Aufzeichnung zu deaktivieren:

- Geben Sie folgendes Terminalkommando ein:

  ```
  %B
  ```

  Die Aufzeichnung ist beendet.

Wiedergabe einer Aufzeichnung

Wird eine Aufzeichnung wiedergegeben, dann wird, zum Beispiel, die Abfolge von Kommandos und Funktionen faktisch erneut ausgeführt.

Die Aufzeichnung ist unabhängig vom Terminaltyp, d.h., eine Session, die auf einem Terminal aufgenommen wurde, kann auf einem Terminal eines anderen Typs wiedergegeben werden. Es ist auch möglich, eine Aufzeichnung im Batch-Betrieb wiederzugeben. Eine aufgezeichnete Online-Session kann sich bei der Wiedergabe in Batch natürlich anders verhalten.

Folgende Themen werden behandelt:

- Einzelschritt- und Hintergrund-Modus
- Aktivieren einer Wiedergabe
- Unterbrechen einer Wiedergabe

Einzelschritt- und Hintergrund-Modus

Eine Aufzeichnung kann in zwei Modi wiedergegeben werden: Einzelschritt-Modus und Hintergrund-Modus.

Im Hintergrund-Modus wird die gesamte Aufzeichnung unsichtbar wiedergegeben, d.h., alle Tastaturaktionen der Aufzeichnung werden ausgeführt, ohne dass während der Ausführung der Aufzeichnung irgendetwas auf Ihrem Terminal-Bildschirm angezeigt wird. Eine Aufzeichnung, die im Hintergrund-Modus wiedergegeben wird, können Sie nicht unterbrechen, es sei denn, die Aufzeichnung enthält das Terminalkommando `%R`. Erklärung siehe Abschnitt ***Manipulieren einer Aufnahme***.

Im Einzelschritt-Modus wird eine Aufzeichnung schrittweise wiedergegeben und alle Tastaturaktionen werden im Bildschirm angezeigt. Sie können eine beliebige Funktionstaste benutzen, um schrittweise voranzugehen. Im Einzelschritt-Modus haben Sie auch die Möglichkeit, die Aufzeichnung zu unterbrechen, indem Sie CLEAR drücken. Erklärung siehe Abschnitt ***Unterbrechen einer Wiedergabe***.

Standardmäßig erfolgt die Wiedergabe einer Aufzeichnung im Hintergrund-Modus.

Um den Wiedergabemodus einzustellen:

1 Um den Einzelschritt-Modus zu aktivieren, geben Sie das Terminalkommando `G` wie folgt ein:

```
%GON
```

2 Um den Einzelschritt-Modus zu deaktivieren und den Hintergrund-Modus zu aktivieren, geben Sie das Terminalkommando `G` wie folgt ein:

```
%GOFF
```

3 Um zwischen Einzelschritt-Modus und Hintergrund-Modus hin- und herzuschalten, geben Sie folgendes Terminalkommando ein:

```
%G
```

Aktivieren einer Wiedergabe

Um eine Aufzeichnung wiederzugeben:

- Geben Sie folgendes Terminalkommando ein:

  ```
  %Aname
  ```

  Die unter dem angegebenen Namen gespeicherte Aufzeichnung wird wieder ausgeführt.

Aufzeichnen mit %A*name*

Wenn Sie das Kommando %A*name* absetzen, während eine Session aufgezeichnet wird, dann wird die mit %A*name* angegebene Aufzeichnung nicht ausgeführt, sondern das Kommando %A*name* wird in die Objekt-Source eingefügt, die gerade aufgenommen wird. Auf diese Weise können Sie eine Aufzeichnung aus einer anderen Aufzeichnung heraus ausführen und eine Reihe von Aufzeichnungen miteinander verketten. In einander verschachtelte Aufzeichnung sind jedoch nicht möglich. Die Ausführung der Aufzeichnung, die das Kommando %A*name* enthält, stoppt nach diesem Kommando und wird nicht wieder fortgesetzt, wenn die Ausführung von *name* endet. Das hat zur Folge, dass die nach dem Kommando %A*name* aufgezeichneten Daten niemals wiedergegeben werden. Um dies zu vermeiden, sollten Sie, unmittelbar nachdem Sie %A*name* in eine Aufzeichnung eingegeben haben, das Kommando %B eingeben.

Unterbrechen einer Wiedergabe

Um eine Wiedergabe zu unterbrechen:

- Drücken Sie CLEAR.

Wenn Sie eine Aufzeichnung unterbrochen haben, haben Sie folgende Möglichkeiten:

- Sie können Ihre Session ab dem Punkt, an dem Sie die Aufzeichnung gestoppt haben, normal fortsetzen.
- Sie können zusätzliche Tastaturaktionen in die Aufzeichnung einfügen:

  Nachdem Sie CLEAR gedrückt haben, geben Sie das Kommando %B ein, und alle Aktionen, die Sie durchführen, werden so lange in die Source der Aufzeichnung eingefügt, bis Sie wieder %B eingeben. Danach wird die Ausführung der Aufzeichnung fortgesetzt.
- Sie können den nächsten Schritt in der Aufzeichnung verändern:

  Nachdem Sie CLEAR gedrückt haben, geben Sie das Kommando %R ein. Danach geben Sie die Eingabedaten für den nächsten Schritt ein. Die neu eingegebenen Eingabedaten überschreiben die Eingabedaten für diesen Schritt in der aufgezeichneten Source. Wenn Sie ENTER drücken, wird der Schritt mit den neuen Eingabedaten ausgeführt. Anschließend wird die Ausführung der Aufzeichnung fortgesetzt.

- Sie können eine Hilfe-Routine ausführen:

  Nachdem Sie CLEAR gedrückt haben, geben Sie das Kommando %J und direkt danach den Namen der gewünschten Hilfe-Routine ein. Die Hilfe-Routine wird aufgerufen und die Ausführung der Aufzeichnung wird fortgesetzt, wenn die Ausführung der Hilfe-Routine beendet ist.

Manipulieren einer Aufnahme

Sie können, indem Sie das Terminalkommando %R aufzeichnen, einen Einzelschritt in einer Aufzeichnung manipulieren, wenn diese wiedergegeben wird. Dies gilt im Einzelschritt-Modus und im Hintergrund-Modus. Im Hintergrund-Modus ist %R die einzige Möglichkeit, um mit einer Aufzeichnung, die gerade wiedergegeben wird, zu interagieren. Ein solches Eingreifen kann zum Beispiel erforderlich sein, um eine Eingabemöglichkeit für sensible Daten wie z.B. Passwörter zu schaffen, die zum Zeitpunkt der Aufzeichnung nicht bekannt sind.

Wenn das Terminalkommando %R (letzten Bildschirm erneut anzeigen) aufgezeichnet worden ist, dann ist der nachfolgende Bildschirm für eine Eingabe durch den Benutzer geöffnet, wenn die Aufzeichnung wiedergegeben wird, d.h., Die Eingabedaten für diesen Bildschirm werden nicht aus der Aufzeichnung genommen, sondern von dem, was der Benutzer eingibt. Anschließend wird die Ausführung der Aufzeichnung fortgesetzt.

XII SYSAPI Utility

56 SYSAPI Utility

- Voraussetzungen 532
- SYSAPI Utility aufrufen und beenden 532
- Reservierte Schlüsselwörter 533
- SYSAPI Utility benutzen 533

Mit dem Dienstprogramm („Utility“) SYSAPI können Sie Anwendungsprogrammierschnittstellen (Application Programming Interfaces/APIs), die von Natural Add-on-Produkten (z.B. Entire Output Management) zur Verfügung gestellt werden, finden und testen.

Die API eines Natural Add-on-Produkts ist ein Natural-Subprogramm (katalogisiertes Objekt), das dazu benutzt wird, auf Daten zuzugreifen, eventuell Daten zu ändern oder Dienste auszuführen, die für ein Natural Add-on-Produkt oder eine Unterkomponente spezifisch sind.

Die API eines Natural Add-on-Produkts wird in der Natural Library und/oder in der Systemdatei geliefert, die für Objekte vorgesehen sind, welche für ein bestimmtes Natural Add-on-Produkt spezifisch sind. Anleitungen zur Verwendung der APIs eines Natural Add-on-Produkts sind in der Dokumentation des betreffenden Add-on-Produkts enthalten.

Die SYSAPI Utility bietet zu jeder API eines Natural Add-on-Produkts eine Funktionsbeschreibung, ein Beispiel-Programm und API-spezifische Schlüsselwörter.

Verwandtes Thema:

SYSEXT Utility

Voraussetzungen

- Das betreffende Natural Add-on-Produkt muss in Ihrer Umgebung installiert sein.
- Die Version des installierten Natural Add-on-Produkts muss die Funktionalität der SYSAPI Utility unterstützen.

SYSAPI Utility aufrufen und beenden

Um die SYSAPI Utility aufzurufen:

- Geben Sie folgendes Systemkommando ein:

  ```
  SYSAPI
  ```

  Das Hauptmenü der SYSAPI Utility wird angezeigt. Es enthält eine Liste mit einem oder mehreren Natural Add-on-Produkten und zu jedem von ihnen eine oder mehrere zugehörige Gruppen von APIs.

  Jede Gruppe repräsentiert ein besonderes API-Merkmal, das für das Natural Add-on-Produkt verfügbar ist, und enthält Beispiel-Programme, die sich auf dieses Merkmal beziehen.

  Sie können eine Gruppe auswählen, um alle zugehörigen APIs in einem Untermenü anzuzeigen, siehe *SYSAPI Utility benutzen*.

Anmerkung: In der Kommandozeile **Command** können Sie ein beliebiges Natural-Systemkommando eingeben.

Um die SYSAPI Utility zu beenden:

- Drücken Sie PF3 oder PF12.

Reservierte Schlüsselwörter

Reservierte Schlüsselwörter beziehen sich auf Meta-Informationen zu APIs, z.B. der Natural Add-on-Produkt-Version, in der eine API hinzugefügt worden ist. Reservierte Schlüsselwörter beginnen immer mit einem Plus-Zeichen (+):

Reserviertes Schlüsselwort	Beschreibung
+NEW-*PROD-version*	Eine API, die zu einem bestimmten Produkt in einer bestimmten Version hinzugefügt worden ist.
+MOD-*PROD-version*	Eine API, die zu einem bestimmten Produkt gehört und in einer bestimmten Version geändert worden ist.

SYSAPI Utility benutzen

Nachdem Sie die SYSAPI Utility aufgerufen haben, können Sie eine bestimmte Gruppe von APIs auswählen, indem Sie ein beliebiges Zeichen in das entsprechende Eingabefeld eingeben oder indem Sie das Feld mit dem Cursor markieren. Das daraufhin angezeigte SYSAPI-Untermenü enthält eine alphabetisch sortierte Liste aller APIs der gewählten Gruppe.

Dieser Abschnitt beschreibt, wie Sie das Untermenü der SYSAPI Utility benutzen, z.B. um Auswahl- oder Suchkriterien einzugeben.

Folgende Themen werden behandelt:

- Bestandteile des SYSAPI-Untermenüs
- PF-Tasten

- Zeilenkommandos

Bestandteile des SYSAPI-Untermenüs

Das SYSAPI-Untermenü enthält eine Liste mit alphabetisch sortierten APIs. Über den Spalten, die mit **Interface** und **Description** beschriftet sind, befinden sich Auswahlfelder, in denen Sie Auswahlkriterien eingeben können.

Die Eingabefelder für Schlüsselwörter können sowohl zur Schlüsselwortauswahl als auch zur Schlüsselwortsuche benutzt werden.

Die folgende Tabelle enthält eine ausführliche Beschreibung aller Bestandteile des SYSAPI-Untermenüs und Hinweise zu ihrer Benutzung:

Bestandteil	Erklärung	Verwendung
Cmd	Eingabefeld für ein Zeilenkommando, das Sie zu einer API ausführen können. Siehe *Zeilenkommandos*.	Geben Sie ein Zeilenkommando ein, z.B.: `K` Alle bei der angegebenen API relevanten Schlüsselwörter auflisten.
Interface	Der Name des API-Subprogramms.	Geben Sie einen Stern (*) oder ein mit einem Stern eingegrenztes Präfix ein
Description	Kurzbeschreibung des Verwendungszwecks der API.	Geben Sie eine Zeichenkette ein, z.B.: `printer` Alle APIs mit einer Beschreibung auflisten, die die Zeichenkette `printer` oder `Printer` enthält.
Keywords	Alle Schlüsselwörter auflisten oder ein Schlüsselwort als Auswahlkriterium eingeben. Maximale Eingabe: zwei Schlüsselwörter.	Siehe *Schlüsselwort-Suche* oder *Schlüsselwort-Auswahl*.
And/Or	Geben Sie die logische Bedingung an, mit der mehrere Schlüsselwörter verknüpft werden sollen: `A` (**And**) oder `O` (**Or**). `A` ist die Standardeinstellung.	Geben Sie `A` oder `O` ein. Siehe auch *Schlüsselwort-Auswahl*.
Command	Kommandozeile zur Eingabe von Natural-Systemkommandos.	Geben Sie ein Natural-Systemkommando ein.

Anmerkung: Wenn ein Natural-Add-on-Produkt für eine zusätzliche Sprache lokalisiert worden ist, können Sie die Sprache in Ihren Profilparameter-Einstellungen angeben, siehe *ULANG - Benutzersprache* in der *Parameter-Referenz*-Dokumentation. Als Ergebnis enthält die für dieses Produkt zurückgegebene Liste APIs für die Sprache, die Sie angegeben haben.

Schlüsselwort-Suche

- Geben Sie im Feld **Keyword** einen Stern (*) oder ein mit einem Stern (*) eingegrenztes Präfix ein, z.B.:

```
L*
```

Als Ergebnis der Suche erscheint ein Menü, in dem Schlüsselwörter angezeigt werden, die mit L beginnen (Beispiel):

```
        Search for Keywords

Mark Keyword
---- L*__________________
  _  LIST
  _  LOG
  _  LOGGING
  _  LOGICAL
```

Um ein bestimmtes Schlüsselwort als Auswahlkriterium auszuwählen, können Sie in dem entsprechenden Eingabefeld in der Spalte **Mark** ein beliebiges Zeichen eingeben. Als Ergebnis wird eine Liste aller APIs, die dieses Auswahlkriterium erfüllen, angezeigt.

Anmerkung: Die Benutzung der Schlüsselwort-Suche in beiden Eingabefeldern resultiert in zwei Menüs, die nacheinander verarbeitet und angezeigt werden.

Schlüsselwort-Auswahl

1 Geben Sie ein vollständiges Schlüsselwort ein. Sie können Schlüsselwörter auch in beiden Eingabefeldern eingeben, z.B.:

```
USER
```

und

```
LOG
```

2 Wählen Sie die logische Bedingung, mit der mehrere Schlüsselwörter verknüpft werden sollen:

A (**And**) oder O (**Or**).

Je nach angegebener logischer Verknüpfung erhalten Sie entweder eine Liste aller APIs mit dem Schlüsselwort USER **und** dem Schlüsselwort LOG (And) oder eine Liste aller APIs mit dem Schlüsselwort USER **oder** dem Schlüsselwort LOG (Or).

PF-Tasten

Sie können die folgenden, mit Funktionen belegten PF-Tasten benutzen:

PF-Taste	Name	Funktion
PF1	**Help**	Anzeige der kontextabhängigen Hilfe. Zu jedem Eingabefeld ist ein spezifischer Hilfetext vorhanden. In anderem Kontext, zum Beispiel in der Kommandozeile, wird ein allgemeiner Hilfetext angezeigt.
PF2	**Reset**	Die Inhalte aller Auswahlfelder werden gelöscht und die Liste der APIs wird zurückgesetzt.
PF3	**Exit**	Beenden der SYSAPI Utility oder Verlassen des aktuellen Menüs oder Fensters.
PF6	**--**	Zum Anfang der Liste blättern.
PF7	**-**	Eine Seite nach oben blättern.
PF8	**+**	Eine Seite nach unten blättern.
PF9	**++**	Zum Ende der Liste blättern.
PF12	**Canc**	SYSAPI Utility oder das aktuelle Menü oder Fenster verlassen.

Zeilenkommandos

Zeilenkommandos werden verwendet, um Objektoperationen durchzuführen. Sie können ein Zeilenkommando in der Spalte **Cmd** neben der gewünschten API eingeben. Für eine Liste der gültigen Zeilenkommandos können Sie ein Fragezeichen (`?`) eingeben oder PF1 drücken.

Folgende Zeilenkommandos stehen zur Verfügung:

Zeilenkommando	Funktion
`K`	Schlüsselwörter auflisten, die für die angegebene API relevant sind.
`T`	Falls Text-Objekt verfügbar sind: Beschreibung der entsprechenden API anzeigen. Andernfalls benutzen Sie das Zeilenkommando `L`, um das Beispiel-Programm wegen einer Beschreibung anzuzeigen.
`L`	Beispiel-Programm anzeigen (`LIST`).
`E`	Beispiel-Programm bearbeiten (`EDIT`).
`X`	Beispiel-Programm ausführen (`EXECUTE`).
`.`	Zum SYSAPI-Hauptmenü zurückkehren.

XIII SYSCP Utility

57 SYSCP Utility

- SYSCP Utility aufrufen und beenden 540
- Codepages verwalten - Menü: Code Page Maintenance 542
- Alle Codepages anzeigen - Funktion: All Code Pages 557
- Geladene Codepages - Funktion: Loaded Code Pages 563
- Unicode-Eigenschaften - Funktion: Unicode Properties 564
- ICU Information 565

Mit dem Dienstprogramm („Utility“) SYSCP können Sie sich Informationen zu Codepages und ICU-Datenbibliotheken (*ICU = International Components for Unicode*) anzeigen lassen, die in der aktuellen Natural-z/OS-Umgebung zur Verfügung stehen. Darüber hinaus können Sie das Dienstprogramm benutzen, um die Codepage-Zuweisung bei Programmierungsobjekten, Datendefinitionsmodulen (DDMs), Profilen und Fehlermeldungen zu ändern. Sie können mit diesem Dienstprogramm auch Codepages hinzufügen, entfernen oder konvertieren.

Das Dienstprogramm SYSCP leistet außerdem Hilfestellung zur Vermeidung von Problemen, die auftreten können, wenn eine Codepage in Natural nicht definiert oder nicht freigeschaltet ist oder wenn Objekte in ein nicht korrektes Codepage- oder Unicode-Format konvertiert werden.

Ausführliche Informationen dazu, wie Unicode, Codepages und Unicode-spezifische Elemente von Natural unterstützt werden, finden Sie in den Beschreibungen und Präsentationen, die Sie sich mit dem Dienstprogramm **SYSEXV** anzeigen lassen können, sowie in den Dokumenten, die unter ***Verwandte Themen*** aufgeführt sind.

Die Benutzung der Funktionen des Dienstprogramms SYSCP kann in Natural Security benutzerspezifisch eingeschränkt werden. Ausführliche Informationen siehe *SYSCP-Dienstprogrammprofile* in der *Natural Security*-Dokumentation.

Verwandte Themen:

- Informationen zur Benutzung von Lokalisierungsdaten auf der Basis von ICS finden Sie in der *Unicode- und Codepage-Unterstützung*-Dokumentation.
- Unicode: Unicode Consortium Website auf ***http://www.unicode.org/***
- ICU: International Components for Unicode Website auf ***http://site.icu-project.org/***
- ICU: Converter Explorer Documentation auf ***http://demo.icu-project.org/icu-bin/convexp***
- Systemkommando `CPINFO`: dient ebenfalls zum Anzeigen relevanter Natural Codepage-Einstellungen.

SYSCP Utility aufrufen und beenden

Der folgende Abschnitt beschreibt, wie Sie die SYSCP Utility aufrufen, wie Sie eine Funktion in der Utility ausführen können und wie Sie die SYSCP Utility beenden.

Um die SYSCP Utility aufzurufen:

- Setzen Sie folgendes Systemkommando ab:

  ```
  SYSCP
  ```

  Das Hauptmenü (**Menu**) der Natural SYSCP Utility erscheint (Beispiel):

```
10:04:36               ***** NATURAL SYSCP UTILITY *****            2021-08-09
User XYZ                             - Menu -              ICU Version 65.1
                                                         Unicode Version 12.1



                          Function

                        _  Code Page Maintenance
                        _  All Code Pages
                        _  Loaded Code Pages
                        _  Unicode Properties
                        _  ICU Information
                        _  Help
                        _  Exit









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

Im oberen rechten Bildschirmbereich werden die aktuelle ICU-Version und die Unicode-Version angezeigt.

Die einzelnen Menüfunktionen werden separat im Anschluss an diesen Abschnitt beschrieben.

Um eine SYSCP-Funktion auszuführen:

- Stellen Sie im SYSCP-Hauptmenü **Menu** den Cursor in das Eingabefeld vor der gewünschten Funktion und drücken Sie ENTER.

  Oder:

  Geben Sie im SYSCP-Hauptmenü **Menu** im Eingabefeld vor der gewünschten Funktion ein beliebiges Zeichen ein und drücken Sie ENTER.

Anmerkung: In der Kommandozeile **Command ==>** in jedem beliebigen Bildschirm der SYSCP Utility können Sie ein beliebiges Natural-Systemkommando eingeben. Durch das Absetzen eines Systemkommandos wird die SYSCP Utility beendet.

Um die SYSCP Utility zu beenden:

- Drücken Sie PF3 oder PF12.

Oder:

Wählen Sie im SYSCP **Menu** die Funktion **Exit**.

Codepages verwalten - Menü: Code Page Maintenance

Das Menü **Code Page Maintenance** bietet Pflegefunktionen, die Sie auf Programmierobjekte (N), DDMs (D), Profile (P), Benutzerfehlermeldungen (E) und Systemfehlermeldungen (S) anwenden können. Bei einem angegebenen Objekt können Sie Codepage-Informationen auflisten (L), zuweisen (A) oder entfernen (R). Darüber hinaus können Sie eine Codepage konvertieren (C) oder ihre Konvertierung für zugewiesene (K) oder nicht zugewiesene (C) Objekte prüfen.

Alle Codepage-Pflegefunktionen beziehen sich auf den Standard-IANA-Namen (siehe auch **Cmd** in **All Code Pages**). Das heißt, Sie können keinen anderen Codepage-Namen als den IANA-Namen benutzen, wenn Sie eine Codepage-Pflegefunktion ausführen.

Die Ergebnisse einer Codepage-Pflegefunktion werden in einem Report-Bildschirm ausgegeben. Beschreibung siehe *Ergebnisreport der Funktion*.

Wenn Sie die Funktion **Code Page Maintenance** aufrufen, erscheint folgender Bildschirm (Beispiel):

```
10:06:16               ***** NATURAL SYSCP UTILITY *****             2021-08-09
User XYZ                   - Code Page Maintenance -

        Code Function                                  Code Object

          L  List Code Page Information                  N  Programming Objects
          C  Check Conversion of Unassigned Objects      D  DDMs
          A  Assign Code Page Information                P  Profiles
          K  Check Conversion of Assigned Objects        E  User Error Messages
          T  Convert to Different Code Page              S  System Error Messages
          R  Remove Code Page Information
          U  Convert to Unshaped Form
          ?  Help
          .  Exit

 Function _                                    Object .. N

 Library  XYZ___     DBID ..... _____     FNR ..... _____
                       Password ..          Cipher ..


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

Die im Menü **Code Page Maintenance** enthaltenen Felder und Funktionen und die Optionen und Merkmale, die die Menü-Funktionen bieten, werden in den folgenden Abschnitten beschrieben:

- Feldbeschreibungen: Code Page Maintenance
- Codepage-Informationen von Profilen auflisten - Funktion: List Code Page Information of Profiles
- Konvertierung nicht zugewiesener Objekte prüfen - Funktion: Check Conversion of Unassigned Objects
- Codepage-Informationen zu Objekten zuweisen - Funktion: Assign Code Page Information to Objects
- Konvertierung zugewiesener Objekte prüfen - Funktion: Check Conversion of Assigned Objects
- In andere Codepage konvertieren - Funktion: Convert to Different Code Page
- Codepage-Informationen von einer Source entfernen - Funktion: Remove Code Page Information from Sources
- Konvertierung in die Grundform - Funktion: Convert to Unshaped Form
- Namen als Auswahlkriterium angeben
- Liste zur Objektauswahl
- Ergebnisreport der Funktion

Feldbeschreibungen: Code Page Maintenance

Die im Menü **Code Page Maintenance** enthaltenen Felder werden in der folgenden Tabelle beschrieben:

Feld	Erklärung
Function	Funktionsschlüssel zum Aufrufen der auszuführenden Funktion.
Object	Schlüssel zur Angabe des Objekttyps, auf den die Funktion angewendet werden soll. `N` Programmierobjekte `D` DDMs `P` Profile `E` Benutzerfehlermeldungen `S` Systemfehlermeldungen
Library	Name der Natural-Library, die die Objekte enthält, an denen eine Codepage-Pflegefunktion ausgeführt werden soll. Standardmäßig ist das Feld mit dem Namen der aktuellen Library vorbelegt, der aus der Systemvariablen `*LIBRARY-ID` entnommen wird. Siehe *Systemvariablen*-Dokumentation. Nicht verwendet bei DDMs, Profilen und Systemfehlermeldungen.
DBID	Datenbankkennung (DBID) der Natural-Systemdatei, in der die angegebene Library gespeichert ist. Wird kein Wert (oder `0`) angeben, dann wird die entsprechende Systemdatei benutzt.
FNR	Dateinummer (FNR) der Natural-Systemdatei, in der die angegebene Library gespeichert ist. Wird kein Wert (oder `0`) angeben, dann wird die entsprechende Systemdatei benutzt.

Feld	Erklärung
Password	Wenn die angegebene Systemdatei passwortgeschützt ist, müssen Sie das korrekte, aus 8 Zeichen bestehenden Adabas-Passwort eingeben.
Cipher	Wenn die angegebene Systemdatei verschlüsselt ist, müssen Sie den korrekten, aus 8 Zeichen bestehenden Adabas-Chiffrierschlüssel eingeben.

Codepage-Informationen von Profilen auflisten - Funktion: List Code Page Information of Profiles

Sie können sich die Codepage-Informationen zu Objekten eines bestimmten Typs anzeigen lassen, indem Sie ein L im Feld **Function** und den **Code zur Angabe des Objekts** im Feld **Object** eingeben. Der folgende Beispielbildschirm zeigt das Ergebnis der Eingabe des Codes P. Es wird eine Liste aller Profile und, falls zugewiesen, der jeweiligen Codepage angezeigt:

```
10:10:24                ***** NATURAL SYSCP UTILITY *****              2021-08-09
User XYZ           - List Code Page Information of Profiles -
                                                        Listed Library SAG
Cmd  Name      Code Page                            Type
---  *________ *____________________________________ *__
__   USER1                                           P
__   USER2                                           P
__   USER3     IBM01140                              P
__   USER4     IBM01140                              P
__   USER5                                           P
__   USER6                                           P
__   USER7                                           P
__   USER8     IBM01140                              P
__   USER9                                           P
__   USER10                                          P
__   USER11                                          P
__   USER12                                          P
__   USER13                                          P
__   USER14                                          P


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

Die in einem **List Code Page Information**-Bildschirm enthaltenen Felder und Spalten werden in der folgenden Tabelle beschrieben. Zu beachten ist, dass das Vorhandensein einiger Felder und Spalten vom gewählten Objekttyp abhängig ist.

Feld / Spalte	Erklärung	Objekt
Listed Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N Programmierobjekte E Benutzerfehlermeldungen P Profile
Cmd	Eingabefeld zum Eingeben des folgenden Zeilenkommandos, das für ein markiertes Programmierobjekt ausgeführt werden soll: LD Directory-Informationen anzeigen. Dieses Zeilenkommando entspricht dem Systemkommando LIST DIRECTORY *object-name*. Beschreibung siehe *Directory-Informationen anzeigen* in der *Systemkommando*-Dokumentation.	Alle Objekte.
Name	Name des Objekts.	Alle Objekte außer E (Benutzerfehlermeldungen) und S (Systemfehlermeldungen).
Code Page	Codepage-Information (IANA-Name) zu dem Objekt. Bei einem Objekt, dem keine Codepage zugewiesen worden ist, ist diese Spalte leer.	Alle Objekte.
Type	Typ des angezeigten Objekts.	N Programmierobjekte P Profile
Error Text	Gekürzter Text der Meldung.	Objekttypen E (Benutzerfehlermeldungen) und S (Systemfehlermeldungen).
Numbr	Nummer der Meldung.	Objekttypen E (Benutzerfehlermeldungen) und S (Systemfehlermeldungen).
LC	Das Sprachschlüsselzeichen des Objekts. Siehe *LANGUAGE.	Objekttypen E (Benutzerfehlermeldungen) und S (Systemfehlermeldungen).
S/L	Angabe, ob es sich bei der Meldung um eine Kurz- oder Langmeldung handelt.	Objekttypen E (Benutzerfehlermeldungen) und S (Systemfehlermeldungen).

Objekte filtern

Durch Angabe von Auswahlkriterien können Sie die Liste der im Bildschirm **List Code Page Information** angezeigten Objekte verkürzen.

Um Auswahlkriterien anzugeben:

1 Bei jeder Spalte, über der ein Eingabefeld vorhanden ist, können Sie Auswahlkriterien angeben. Überschreiben Sie den standardmäßig vorhandenen Stern (*) durch einen der Eingabewerte, die im Abschnitt *Namen als Auswahlkriterium angeben* aufgeführt sind.

2 Überschreiben Sie im Eingabefeld unter der Spaltenüberschrift **Type** den standardmäßig vorhandenen Stern (*) durch einen oder mehrere (maximal 11) der folgenden Typschlüssel ohne Trennzeichen:

Code	Objekttyp	Code	Objekttyp
P	Programm	A	Parameterdatenbereich (PDA)
N	Subprogramm	G	Globaler Datenbereich (GDA)
S	Subroutine	L	Lokaler Datenbereich (LDA)
M	Map (Maske)	C	Copycode
H	Helproutine	T	Text
7	Function	4	Class
3	Dialog		
5	Processor		
*	Alle Typen		

Konvertierung nicht zugewiesener Objekte prüfen - Funktion: Check Conversion of Unassigned Objects

Mit dieser Funktion können Sie prüfen, ob ein nicht zugewiesenes Objekt in eine Codepage konvertiert werden kann.

Ein nicht zugewiesenes Objekt ist ein Objekt ohne Codepage-Information, das ursprünglich unter einer Natural-Version gespeichert wurde, in der keine Codepage-Informationen unterstützt wurden. Da keine Codepage-Information geliefert wird, müssen Sie entscheiden, welche Codepage für das zur Konvertierung zu prüfende Objekt angegeben werden soll. Dies ist abhängig von dem Zeichensatz, der in dem Objekt benutzt wird.

Wenn Sie zum Beispiel die Funktion **Check Conversion of Unassigned Objects** auf **Profiles** anwenden, wird folgender Bildschirm angezeigt:

```
10:12:28              ***** NATURAL SYSCP UTILITY *****             2021-08-09
User XYZ         - Check Conversion of Unassigned Profiles -


   Check if profiles that have no code page information can be
   converted from a given code page to a target code page.

   Use selection list .. Y

   Source code page .... IBM01140_____________________________________________

   Target code page .... IBM01140_____________________________________________

   Profile name ........ *________     DBID ...... 10___     FNR ..... 32___
                                       Password ..           Cipher ..
   Profile types ....... *__




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

Die im Bildschirm **Check Conversion of Unassigned Objects** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	Alle Objekte.
Source code page	Name der Codepage, die verwendet werden soll, um zu prüfen, ob die angegebenen Objekte (den bisher noch keine Codepages zugewiesen worden sind) von dieser Codepage in die im Feld **Target code page** eingegebene Codepage konvertiert werden kann. Im Falle einer erfolgreichen Konvertierungsprüfung kann die im Feld **Target code page** angegebene Codepage in der Funktion **Assign Code Page Information to Objects** verwendet werden. Der eingegebene Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	Alle Objekte.
Target code page	Name der Codepage, die für die Konvertierung der angegebenen, nicht zugewiesenen Objekte geprüft werden soll. Der eingegebene Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	Alle Objekte.

Feld	Erklärung	Objekt
Object name/Object name range	Name eines einzelnen Objekts oder ein Bereich von Namen, die verarbeitet werden sollen. Gültige Eingabewerte siehe *Namen als Auswahlkriterium angeben*.	Alle Objekte.
Language code	Sprachcodezeichen für das Objekt. Siehe `*LANGUAGE` in der *Systemvariablen*-Dokumentation.	E, S
Short/long error msgs	Typ der zu verarbeitenden Fehlermeldung: S Verarbeitung für Kurzmeldungen L Verarbeitung für Langmeldungen */A Verarbeitung für beliebige Fehlermeldungen	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Profile types	Typ der zu verarbeitenden Fehlermeldung: S Nur Verarbeitung für Kurzmeldungen L Nur Verarbeitung für Langmeldungen */A Verarbeitung aller Fehlermeldungen	P

Codepage-Informationen zu Objekten zuweisen - Funktion: Assign Code Page Information to Objects

Mit dieser Funktion können Sie Codepage-Informationen zu einem nicht zugewiesen Objekt zuweisen. Der Code dieses Objekts wird dabei *nicht* in die angegebene Codepage konvertiert.

Außerdem können Sie mit dieser Funktion die Codepage-Informationen zu Objekten ändern, zu denen bereits eine Codepage zugewiesen ist. In diesem Fall ändert sich lediglich der Codepage-Name (IANA-Name) des Objekts. Der Quellcode des Objekts wird dabei *nicht* konvertiert.

Die im Bildschirm **Assign Code Page Information to Sources** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	Alle Objekte.
Forced assignment	Gibt an, ob Objekte, die bereits Codepage-Informationen haben, oder Objekte ohne Codepage-Informationen verarbeitet werden sollen. Mögliche Werte:	Alle Objekte.

Feld	Erklärung	Objekt
	Y Yes/Ja. Die zwangsweise Zuordnung wird aktiviert: Die Codepage-Informationen ändern sich in die angegebene Codepage bei Objekten, die bereits Codepage-Informationen haben. N No/Nein. Die zwangsweise Zuordnung wird deaktiviert (Standardeinstellung): Die angegebene Codepage wird nur zu Objekten zugewiesen, die keine Codepage-Informationen haben.	
Code page	Name der Codepage, die den angegeben Objekten zugewiesen werden soll. Der eingegebene Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	Alle Objekte.
Object name/Object name range	Siehe **Object name/Object name range** in *Konvertierung nicht zugewiesener Objekte prüfen*.	Alle Objekte.
Language code	Siehe **Language code** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Short/long error msgs	Siehe **Short/long error msgs** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Profile types	Siehe **Profile types** in *Konvertierung nicht zugewiesener Objekte prüfen*.	P

Konvertierung zugewiesener Objekte prüfen - Funktion: Check Conversion of Assigned Objects

Mit dieser Funktion können Sie prüfen, ob ein zugewiesenes Objekt von seiner aktuellen Codepage (laut Eintrag in der Objekt-Directory-Information) in eine andere Codepage konvertiert werden kann.

Ein „zugewiesenes Objekt“ ist ein Objekt, das Codepage-Informationen hat.

Die im Bildschirm **Check Conversion of Assigned** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	Alle Objekte.
Current code page	Die Codepage (oder ein Bereich von Codepages), die als Objektauswahlkritierium verwendet werden soll. Gültige Eingabewerte siehe *Namen als Auswahlkriterium angeben*. Die Standardeinstellung ist ein Stern (*) als Platzhalterzeichen für alle Codepages.	Alle Objekte.
New code page	Name der Codepage, die für die Konvertierung der angegebenen zugewiesenen Objekte geprüft werden soll. Der eingegebene Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	Alle Objekte.
Object name/Object name range	Siehe **Object name/Object name range** in *Konvertierung nicht zugewiesener Objekte prüfen*.	Alle Objekte.
Language code	Siehe **Language code** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Short/long error msgs	Siehe **Short/long error msgs** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Profile types	Siehe **Profile types** in *Konvertierung nicht zugewiesener Objekte prüfen*.	P

In andere Codepage konvertieren - Funktion: Convert to Different Code Page

Mit dieser Funktion können Sie ein zugewiesenes Objekt von seiner aktuellen Codepage (laut Eintrag in der Objekt-Directory-Information) in eine andere Codepage konvertieren. Sie können damit kein nicht zugewiesenes Objekt konvertieren.

Die im Bildschirm **Convert to Different Code Page** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	Alle Objekte.
Current code page	Die Codepage (oder ein Bereich von Codepages), die als Objektauswahlkriterium verwendet werden soll. Gültige Eingabewerte siehe *Namen als Auswahlkriterium angeben*. Die Standardeinstellung ist ein Stern (*) als Platzhalterzeichen für alle Codepages.	Alle Objekte.
New code page	Name der Codepage, in die die Konvertierung der angegebenen Objekte erfolgen soll. Der eingegebene Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	Alle Objekte.
Object name/Object name range	Siehe **Object name/Object name range** in *Konvertierung nicht zugewiesener Objekte prüfen*.	Alle Objekte.
Language code	Siehe **Language code** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Short/long error msgs	Siehe **Short/long error msgs** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Profile types	Siehe **Profile types** in *Konvertierung nicht zugewiesener Objekte prüfen*.	P

Codepage-Informationen von einer Source entfernen - Funktion: Remove Code Page Information from Sources

Mit dieser Funktion können Sie die Codepage-Informationen (laut Eintrag in der Objekt-Directory-Information) von einem zugewiesenen Objekt entfernen.

Vorsicht: Dabei ist zu beachten, dass beim Entfernen der Codepage-Information keine Quellcode-Konvertierung erfolgt.

Die im Bildschirm **Remove Code Page Information from Sources** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	Alle Objekte.
Current code page	Die Codepage (oder ein Bereich von Codepages), die als Objektauswahlkriterium verwendet werden soll. Gültige Eingabewerte siehe *Namen als Auswahlkriterium angeben*. Die Standardeinstellung ist ein Stern (*) als Platzhalterzeichen für alle Codepages.	Alle Objekte.
Object name/Object name range	Siehe **Object name/Object name range** in *Konvertierung nicht zugewiesener Objekte prüfen*.	Alle Objekte.
Language code	Siehe **Language code** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Short/long error msgs	Siehe **Short/long error msgs** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	Alle Objekte.
Profile types	Siehe **Profile types** in *Konvertierung nicht zugewiesener Objekte prüfen*.	P

Konvertierung in die Grundform - Funktion: Convert to Unshaped Form

Mit dieser Funktion können Sie in ihrer Darstellungsform („shaped“) vorliegende arabische Zeichen (Codepage IBM420) durch die entsprechenden Zeichen in der Grundform („unshaped“) ersetzen. Dies ist in den folgenden Objekten möglich: Programmierungsobjekte (N), DDMs (D), Benutzerfehlermeldungen (E) und Systemfehelermeldungen (S).

Die im Bildschirm **Convert Natural Sources to Unshaped Form** enthaltenen Felder werden in der folgenden Tabelle beschrieben.

Feld	Erklärung	Objekt
Use selection list	Gibt an, ob selektive oder automatisierte Verarbeitung für die angegebenen Objekte benutzt wird. Siehe *Liste zur Objektauswahl*.	N, D,E, S
Current code page	Eine Codepage (oder ein Bereich von Codepages), die als Objektauswahlkriterium verwendet werden soll. Gültige Eingabewerte siehe *Namen als Auswahlkriterium angeben*. Der Standard-Name ist der IANA-Name, so wie er von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben wird.	N, D, E, S

Feld	Erklärung	Objekt
Object name	Siehe **Object name/Object name range** in *Konvertierung nicht zugewiesener Objekte prüfen*.	N, D, E, S
Language code	Siehe **Language code** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Short/long error msgs	Siehe **Short/long error msgs** in *Konvertierung nicht zugewiesener Objekte prüfen*.	E, S
Library	Siehe **Library** in *Codepages verwalten - Menü: Code Page*.	N, E
DBID	Siehe **DBID** in *Codepages verwalten - Menü: Code Page*.	N, D, E, S
FNR	Siehe **FNR** in *Codepages verwalten - Menü: Code Page*.	N, D, E, S
Password	Siehe **Password** in *Codepages verwalten - Menü: Code Page*.	N, D, E, S
Cipher	Siehe **Cipher** in *Codepages verwalten - Menü: Code Page*.	N, D, E, S

Namen als Auswahlkriterium angeben

Als Auswahlkriterium können Sie einen Namen oder einen Bereich von Namen angeben.

In der folgenden Liste der Optionen steht *value* für eine beliebige Kombination aus einem oder mehreren Zeichen.

	Eingabe	Ausgewählte Elemente
	value	Alle Elemente mit Namen gleich *value*.
	*	Alle Elemente.
	?	Alle Elemente mit einem beliebigen einzelnen Zeichen für jedes eingegebene Fragezeichen (?).
Führende Zeichen	*value**	Alle Elemente mit Namen, die mit *value* beginnen. Beispiel: AB* Ausgewählt: AB, AB1, ABC, ABEZ Nicht ausgewählt: AA1, ACB
Platzhalterzeichen	*value*?	Ein Platzhalterzeichen (Wildcard). Alle Elemente mit Namen, die mit *value* beginnen und mit einem beliebigen einzelnen Zeichen für jedes eingegebene Fragezeichen (?) enden. Beispiel: ABC? Ausgewählt: ABCA, ABCZ Nicht ausgewählt: AXC, ABCAA
	value? *value*?	Alle Elemente, die mit *value* in Kombination mit Stern (*) und Fragezeichen (?) in beliebiger Reihenfolge übereinstimmen. Beispiel: A?C*Z Ausgewählt: ABCZ, AXCBBBZ, ANCZ Nicht ausgewählt: ACBZ, ABDEZ, AXCBBBZA
	*value**value*?	
	value*?*value	

	Eingabe	Ausgewählte Elemente
Startwert	*value*>	Alle Elemente mit Namen größer/gleich *value*. Beispiel: AB> Ausgewählt: AB, AB1, BBB, ZZZZZZZ Nicht ausgewählt: AA1, AAB
Endwert	*value*<	Ein Endwert: Alle Elemente mit Namen kleiner/gleich *value*. Beispiel: AX< Ausgewählt: AB, AWW, AX Nicht ausgewählt: AXA, AY

Liste zur Objektauswahl

Um festzustellen, ob selektive oder automatisierte Verarbeitung bei einer Pflegefunktion benutzt wird, können Sie die Option **Use selection list** setzen. Wenn selektive Verarbeitung benutzt wird, dann wird in jedem Auswahlbildschirm vor dem Ausführen der Funktion eine Auswahlliste der angegebenen Objekte angezeigt.

Die Option **Use selection list** gilt nicht bei der Funktion **List Code Page Information of Sources**.

Mögliche Einstellungen der Option **Use selection list**:

Y	Yes/Ja. Selektive Verarbeitung ist aktiviert (Standardeinstellung): Es erscheint eine Auswahlliste mit allen Objekten, die die angegebenen Auswahlkriterien erfüllen. In dieser Liste können Sie die zu verarbeitenden Objekte auswählen.
N	No/Nein. Selektive Verarbeitung ist aktiviert. Die Funktion wird unmittelbar bei allen Objekten ausgeführt, die die angegebenen Auswahlkriterien erfüllen.

Beispiel einer Liste zur Objektauswahl:

```
10:15:29               ***** NATURAL SYSCP UTILITY *****             2021-08-09
User SAG            - Check Conversion of Assigned Sources -
      Target code page IBM01140
Cmd  Name      Code Page                          Message
---  --------  ---------------------------------- ------------------------------
__   LDA1      IBM01147
__   LDA2      IBM01147
__   LDA3      IBM01147
__   LDA4      IBM01147
__   MAP1      IBM01147
__   MAP2      IBM01147
__   MAP3      IBM01147
__   PGM2      IBM01147
__   PGM3      IBM01147
__   PROG1     IBM01147
__   PROG3     IBM01147
__   PROG4     IBM01147



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

Die in einem Bildschirm zur Objektauswahl enthaltenen Felder und Spalten werden in der folgenden Tabelle beschrieben:

Feld / Spalte	Erklärung	
Target code page	Codepage, die verwendet werden soll, um eine Source-Objekt-Zuweisung oder Konvertierung zu prüfen oder durchzuführen.	
Cmd	Eingabefeld für eines der folgenden Zeilenkommandos, das für ein gewähltes Objekt ausgeführt werden soll.	

	`EX`
	
	oder
	`X`
		Führt die Pflegefunktion aus.
		
		Sie können PF5 drücken, wenn Sie das Zeilenkommando an alle Objekte in einem Schritt absetzen wollen.

	`LD`	Zeigt die Directory-Informationen des Objekts an. Dieses Zeilenkommando entspricht dem Kommando `LIST DIRECTORY` *`object-name`*. Siehe *Directory-Informationen anzeigen* in der *Systemkommandos*-Dokumentation.
Name	Name des Objekts, das die angegebenen Auswahlkriterien erfüllt.	
Code Page	Aktuelle Codepage-Informationen des Objekts.	

Feld / Spalte	Erklärung
Message	Diese Spalte enthält nur dann einen Text, wenn Sie die Ausführung der Pflegefunktion beendet haben. Dann enthält die Spalte eine Meldung, die den Verarbeitungsstand des Objekts anzeigt. Siehe auch *Ergebnisreport der Funktion*.

Ergebnisreport der Funktion

Nachdem die Ausführung einer Pflegefunktion beendet worden ist, werden die Ergebnisse der Verarbeitung in einem Report-Bildschirm angezeigt. Dieser sieht ähnlich aus wie der im Abschnitt *Liste zur Objektauswahl* abgebildete **Beispiel**-Bildschirm zur Objektauswahl.

Die in einem Ergebnisreport-Bildschirm enthaltenen Felder und Spalten werden in der folgenden Tabelle beschrieben:

Feld / Spalte	Erklärung
Target code page	Die Codepage, die verwendet wird, um eine Zuordnung oder Konvertierung für ein Source-Objekt zu prüfen oder durchzuführen.
Cmd	Keine Eingabe möglich.
Name	Der Name des Objekts, das die angegebenen Auswahlkriterien erfüllt.
Code Page	Die aktuellen Codepage-Informationen des Objekts.
Message	Diese Spalte enthält Meldungen, die den Verarbeitungsstand der zur Verarbeitung ausgewählten Objekte angeben. Diese Meldungen zeigen die erfolgreiche Ausführung einer Funktion oder mögliche Gründe für einen Fehler an. Mögliche Meldungen: ■ **Assignment possible** Zuweisung möglich: Dem Objekt kann die angegebenen Codepage zugewiesen werden. ■ **Conversion error, at least one code point not translated** Fehler bei der Konvertierung, mindestens ein Codepoint wurde nicht übersetzt: Das Objekt kann der angegebenen Codepage nicht zugewiesen oder nicht in diese konvertiert werden. ■ **Code page assigned** Codepage zugewiesen: Das Objekt wurde der angegebenen Codepage zugewiesen. ■ **Conversion possible** Konvertierung möglich: Das Objekt kann in die angegebene Codepage konvertiert werden. ■ **Code page converted** Codepage konvertiert: Das Objekt wurde aus seiner aktuellen Codepage in eine andere Codepage konvertiert.

Feld / Spalte	Erklärung
	■ **Not converted** Nicht konvertiert: Das Objekt wurde nicht in die angegebene Codepage konvertiert, weil es schon in dieser Codepage kodiert ist.

Alle Codepages anzeigen - Funktion: All Code Pages

Mit der **Hauptmenü**-Funktion **All Code Pages** können Sie alle in Ihrer Natural-Umgebung verfügbaren Codepages laden und sich anzeigen lassen (Beispiel):

```
10:16:43               ***** NATURAL SYSCP UTILITY *****             2021-08-09
User SAG                       - All Code Pages -

Cmd Stat Name                                                              Units
--- ---- ------------------------------------------------------------------ -----
 _   D   UTF-8                                                              1 - 3
 _   D   UTF-16                                                             2 - 2
 _   D   UTF-16BE                                                           2 - 2
 _   D   UTF-16LE                                                           2 - 2
 _   D   UTF-32                                                             4 - 4
 _   D   UTF-32BE                                                           4 - 4
 _   D   UTF-32LE                                                           4 - 4
 _   D   UTF16_PlatformEndian                                               2 - 2
 _   D   UTF16_OppositeEndian                                               2 - 2
 _   D   UTF32_PlatformEndian                                               4 - 4
 _   D   UTF32_OppositeEndian                                               4 - 4
 _   D   UTF-7                                                              1 - 4
 _   D   IMAP-mailbox-name                                                  1 - 4
 _   D   SCSU                                                               1 - 3


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

Sie können folgende PF-Tasten benutzen:

- PF8 (oder ENTER): Eine Seite in der Liste nach unten blättern.
- PF7: Eine Seite in der Liste nach oben blättern.
- PF5: Liste nach Codepage-Namen in aufsteigender Reihenfolge sortieren. Je nachdem wie umfangreich die Liste ist, kann es erforderlich sein, dass Sie den Sortierpuffer vergrößern. Dazu können Sie den Profilparameter `SORT` benutzen. Beschreibung siehe *SORT - Steuerung des Sortierprogramms* in der *Parameter-Referenz*-Dokumentation.

Die im Bildschirm **All Code Pages** enthaltenen Spalten werden in der folgenden Tabelle beschrieben:

Spalte	Erklärung
Cmd	Eingabefeld für eines der folgenden Zeilenkommandos, die für die ausgewählte Codepage ausgeführt werden sollen:
N	Alle für die Codepage verwendeten Namen anzeigen: Der IANA-Name (*IANA = Internet Assigned Numbers Authority*) ist der eindeutige Standardname der Codepage. Der IANA-Name wird von Natural als Standard-Codepage-Name (siehe Profilparameter `CP` in der *Parameter Reference*-Dokumentation) bei Konvertierungen nach und aus Unicode benutzt. Der IANA-Name wird von der Systemvariablen `*CODEPAGE` (siehe *Systemvariablen*-Dokumentation) zurückgegeben. CCSID (*Coded Character Set Identifier*) bezeichnet den Zeichensatz gemäß IBM-Identifizierung. Alias-Namen: ein oder mehrere alternative Namen für die Codepage.
C	Anzeige aller Codepoints der ausgewählten Codepage. Siehe ***Codepoint-Liste***.
T	Aufruf eines Fensters zum Testen der Codepoint-Konvertierung nach oder aus Unicode. Siehe ***Test-Konvertierung - Funktion: Test Conversion***.
Stat	Alle während einer Natural-Session benutzten Codepages müssen im NATCONFG-Modul vordefiniert und freigeschaltet sein. Diese Spalte zeigt den NATCONFG-Status der Codepage:
E	Die Codepage ist im NATCONFG-Modul definiert und freigeschaltet.
D	Die Codepage ist im NATCONFG-Modul definiert, aber ausgeschaltet.
N	Die Codepage ist im NATCONFG-Modul nicht definiert.
	Informationen zum NATCONFG-Modul siehe *Natural Configuration Tables* in der *Operations*-Dokumentation.
Name	Der interne ICU-Name.
Units	Die den Codepoints zugewiesenen Code-Einheiten (minimale und maximale Anzahl an Bytes).

Folgende Themen werden behandelt:

- Codepoint-Liste
- Test-Konvertierung - Funktion: Test Conversion

Codepoint-Liste

Diese Funktion zeigt die Liste aller Codepoints der ausgewählten Codepage (Beispiel):

```
13:38:33               ***** NATURAL SYSCP UTILITY *****             2007-08-06
+--------------------------- Code Points of UTF-8 -----------------------------+
! CP: 00000000  U: 0000        NULL                                            !
! CP: 00000001  U: 0001  ?     START OF HEADING                                !
! CP: 00000002  U: 0002  ?     START OF TEXT                                   !
! CP: 00000003  U: 0003  ?     END OF TEXT                                     !
! CP: 00000004  U: 0004  ?     END OF TRANSMISSION                             !
! CP: 00000005  U: 0005  ?     ENQUIRY                                         !
! CP: 00000006  U: 0006  ?     ACKNOWLEDGE                                     !
! CP: 00000007  U: 0007  ?     BELL                                            !
! CP: 00000008  U: 0008  ?     BACKSPACE                                       !
! CP: 00000009  U: 0009  ?     CHARACTER TABULATION                            !
! CP: 0000000A  U: 000A  ?     LINE FEED (LF)                                  !
! CP: 0000000B  U: 000B  ?     LINE TABULATION                                 !
! CP: 0000000C  U: 000C  ?     FORM FEED (FF)                                  !
! CP: 0000000D  U: 000D  ?     CARRIAGE RETURN (CR)                            !
! CP: 0000000E  U: 000E  ?     SHIFT OUT                                       !
! CP: 0000000F  U: 000F  ?     SHIFT IN                                        !
+-------------------------------------------------------------------------------+
 _   D   ibm-912_P100-1995                                              1 - 1
 _   D   ibm-913_P100-2000                                              1 - 1
 _   N   ISCII,version=0                                                1 - 4
 _   N   ISCII,version=1                                                1 - 4


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

Die Liste enthält folgende Informationen:

- Die Byte-Folge der Codepoints (`CP`) der Codepage.
- Die Byte-Folge der entsprechenden Unicode-Codepoints (`U`).
- Das Unicode-Zeichen. Falls das Zeichen von der aktuellen Terminal-Emulation nicht interpretiert werden kann, wird stattdessen das Ersetzungszeichen (gemäß Definition in der Codepage, hier: ?) angezeigt.
- Der normative Name des Unicode-Zeichens.

In der folgenden Tabelle werden die zu jedem Codepoint zur Verfügung gestellten PF-Tasten erklärt:

PF-Taste	Funktion
PF4	Nicht anwendbar bei einer Codepage mit einer 1-Byte-Einheit als Maximum. Öffnet das Fenster **Leading Bytes of Code Point** (siehe Abschnitt **Führende Bytes angeben**), in dem Sie den Byte-Bereich eingeben können, den Sie sich ansehen möchten. Drücken Sie PF3 oder ENTER, um die von Ihnen eingegebenen führenden Bytes zu bestätigen und das Fenster zu schließen. Drücken Sie PF12, um Ihre aktuelle Eingabe abzubrechen und das Fenster zu schließen.
PF5	Aufruf des Bildschirms **Unicode Properties** (siehe dort) zu dem Listenelement, auf dem der Cursor platziert ist.
PF6	Zurücksetzen des ersten (nicht führenden) Bytes des Byte-Bereichs auf den hexadezimalen Wert 0x00.
PF7	Im ausgewählten Byte-Bereich um eine Seite nach oben blättern (siehe auch ***Führende Bytes angeben***). In einer UTF-16- oder UTF-32-Codepage können Sie durch alle Byte-Bereiche blättern.
PF8 (oder ENTER)	Im ausgewählten Byte-Bereich um eine Seite nach unten blättern (siehe auch ***Führende Bytes angeben***). In einer UTF-16- oder UTF-32-Codepage können Sie durch alle Byte-Bereiche blättern.
PF10	Bewegt sich zur Position ganz links im Bildschirm.
PF11	Bewegt sich zur Position ganz rechts im Bildschirm.

Führende Bytes angeben - Fenster: Leading Byte of Code Point

Diese Funktion ist nicht anwendbar bei einer Codepage mit einer 1-Byte-Einheit als Maximum.

Sie können das Fenster **Leading Byte of Code Point** benutzen, um sich den Byte-Bereich (hexadezimale Werte 0x00 bis 0xFF) eines bestimmten führenden (*leading*) Bytes bei einem Codepoint anzusehen.

Im folgenden Beispiel einer UTF-8-Codepage sind die hexadezimalen Werte 0x22 und 0x32 als die führenden Bytes angegeben worden:

```
+--------- Leading Bytes of Code Point ---------+
!                                                !
!    Maximum number of bytes .. 3                !
!                                                !
!                                                !
!    Enter leading bytes ...... 00 22 32 00      !
!                                                !
!                                                !
!                                                !
+------------------------------------------------+
```

Nach Drücken von PF3 (oder ENTER) zeigt die Codepoint-Liste dann die Bytes von hexadezimal 0x00223200 bis 0x002232FF.

Anmerkung: Bei byte-vertauschten Codepages, z.B. UTF-16LE oder UTF-32LE, werden die Bytes in umgekehrter Byte-Reihenfolge gelesen.

Test-Konvertierung - Funktion: Test Conversion

Mit dieser Funktion können Sie die Codepoint-Konvertierung von einer ausgewählten Codepage in die Standard-Codepage (Wert von `*CODEPAGE`) testen, die mit dem Profilparameter `CP` definiert wird:

- von einer alphanumerischen Zeichenkette nach Unicode-Codepoints und umgekehrt oder
- von hexadezimalen Werten nach Unicode-Codepoints und umgekehrt.

Das folgende Beispiel zeigt das Konvertierungsfenster einer Codepage (hier: `ibm-1140_P100-1997`), das folgende Informationen enthält:

- die den Codepoints zugewiesene Anzahl der Byte-Einheiten (minimale und maximale Anzahl an Bytes),
- eine alphanumerische Zeichenkette und ihre äquivalenten hexadezimalen Werte und
- die entsprechenden Unicode-Codepoints.

```
+------------------- Test Conversion of ibm-1140_P100-1997 -------------------+
!                                                                             !
! Code page units .. 1 - 1   (minimum, maximum of bytes)                      !
!                                                                             !
!                                                                             !
! Code Page Characters                                                        !
! Alphanumeric .. ABC                                                         !
! Hexadecimal ... C1 C2 C3 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 !
!                 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 !
!                                                                             !
! Unicode                                                                     !
! Code points ... 0041 0042 0043 0020 0020 0020 0020 0020 0020 0020           !
!                 0020 0020 0020 0020 0020 0020 0020 0020 0020 0020           !
!                 0020 0020 0020 0020 0020 0020 0020 0020 0020 0020           !
!                 0020 0020 0020 0020 0020 0020 0020 0020 0020 0020           !
!                                                                             !
!                                                                             !
!                                                                             !
+-----------------------------------------------------------------------------+
```

Um ein Zeichen oder einen Codepoint zu konvertieren:

1 Aktivieren Sie das Feld, in dem Sie die zu konvertierende Literal-Zeichenkette oder Code Unit-Sequenz eingeben wollen:

Drücken Sie PF6, um im Feld **Alphanumeric** (Standard-Eingabefeld) eine Literal-Zeichenkette einzugeben.

Oder:

Drücken Sie PF7, um im Feld **Hexadecimal** hexadezimale Werte einzugeben.

Oder:

Drücken Sie PF8, um im Feld **Unicode** Unicode-Codepoints einzugeben.

2 Drücken Sie ENTER.

Der in einem der Felder eingegebene Wert wird in äquivalente Codepoints oder eine äquivalente Literal-Zeichenkette konvertiert.

Geladene Codepages - Funktion: Loaded Code Pages

Mit dieser Funktion können Sie sich die in Ihrer aktuellen Natural-Umgebung bereits geladenen Codepages anzeigen lassen.

Beispiel:

```
09:52:33               ***** NATURAL SYSCP UTILITY *****             2020-07-22
User XYZ                     - Loaded Code Pages -

 Name                                                                 Member
 -------------------------------------------------------------------- --------
 ibm-1140_P100-1997                                                   I66C0067
 ibm-1125_P100-1997                                                   I66C0060
 ibm-939_P120-1999                                                    I66C0203
 ibm-930_P120-1999                                                    I66C0199
 ibm-1147_P100-1997                                                   I66C0074









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

Sie können folgende PF-Tasten benutzen:

- PF8 (oder ENTER): Eine Seite in der Liste nach unten blättern.
- PF7: Eine Seite in der Liste nach oben blättern.
- PF5: Liste nach Codepage-Namen in aufsteigender Reihenfolge sortieren. Je nachdem wie umfangreich die Liste ist, kann es erforderlich sein, dass Sie den Sortierpuffer vergrößern. Dazu können Sie den Profilparameter `SORT` benutzen. Beschreibung siehe *SORT - Steuerung des Sortierprogramms* in der *Parameter-Referenz*-Dokumentation.

Die Spalten im Bildschirm **Loaded Code Pages** werden in der folgenden Tabelle beschrieben:

Spalte	Erklärung
Name	Der interne ICU-Name der geladenen Codepage.
Member	Der Name des Datenelements, das das Lademodul aus dem STEPLIB Dataset ist.

Unicode-Eigenschaften - Funktion: Unicode Properties

Mit dieser Funktion können Sie sich anzeigen lassen, ob die Unicode-Eigenschaft (Property) eines Unicode-Zeichens für ein in der Standard-Codepage (Wert der Systemvariablen `*CODEPAGE`) enthaltenes Zeichen wahr (`yes`) oder falsch (`no`) ist, wie im folgenden Bildschirm am Beispiel des Buchstabens `A` in einer `IBM01140`-Codepage gezeigt wird.

```
14:43:19               ***** NATURAL SYSCP UTILITY *****             2008-09-23
User SAG                     - Unicode Properties -

Default code page ... IBM01140

Alpha character ..... A     C1       hexadecimal    Substitution .. ? 3F
Unicode code point .. 0041

Unicode char. name .. LATIN CAPITAL LETTER A

Alphabetic .......... yes           Control ........ no
Alphanumeric ........ yes           Space .......... no
Lower case .......... no            Whitespace ..... no
Upper case .......... yes           Blank .......... no
Digit ............... no            Punctuation .... no
Hexadecimal ......... yes           Combining ...... no
Graphic ............. yes           Surrogate ...... no
Printable ........... yes           Right to left .. no


Command ===>


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

Im Feld **Alpha character** können Sie das Zeichen eingeben, dessen Eigenschaften Sie sich ansehen möchten. Drücken Sie PF5, wenn Sie einen Unicode-Codepoint eingeben wollen.

Eine Erklärung der im Bildschirm angezeigten Unicode-Zeicheneigenschaften können Sie der Dokumentation *Unicode Character Database* des Unicode Consortium auf der Website ***http://www.unicode.org/Public/4.1.0/ucd/UCD.html*** entnehmen.

ICU Information

Mit dieser Funktion können Sie sich Informationen zur benutzten ICU anzeigen lassen, z.B.:

```
18:27:23               ***** NATURAL SYSCP UTILITY *****            2022-03-16
User SAG                      - ICU Information -

Product ................ International Components for Unicode for Software AG
Product code ........... ICS
Product version ........ 312
Cumulated fix ..........

Architecture level ..... 9
Revision ............... 155

ICU version ............ 65.1
Unicode version ........ 12.1
Data library ...........

*CODEPAGE .............. IBM01140
*LOCALE ................ en_US




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit                                                 Canc
AÔ                                                              14,027      ↩
```

Product	Produktname: ICU for Adabas & Natural (ICS)
Product code	ICS
Product version	ICS-Produktversion
Cumulated fix	Cumulated Fix (falls vorhanden und angewendet) für die aktuelle ICS-Produktversion
Architecture level	IBM Architecture Level, bei Benutzung auf z/OS. Standardwert ist Null (`0`), was bedeutet, dass kein Architecture Level benutzt wird.
Revision	ICS-Revisionsnummer
ICU version	Durch ICS unterstützte ICU-Version
Unicode version	Durch ICS unterstützte Unicode-Version
*CODEPAGE	Zurzeit benutzte Codepage. Standardmäßig ist dies `IBM01140`. Siehe auch Systemvariable `*CODEPAGE`.

*LOCALE	Zurzeit benutzte Locale. Standardmäßig ist dies `en_US`. Siehe auch Systemvariable `*LOCALE`.



Anmerkung: Ähnliche Informationen werden durch das Systemkommando `CPINFO` angezeigt.

XIV SYSEDT Utility

58 SYSEDT Utility

- Library-Profil für Natural Security definieren 570
- SYSEDT Utility aufrufen und eine Funktion ausführen 570
- General Information - Allgemeine Informationen 572
- Generation Parameters - Generierungs-Parameter 574
- Users - Benutzer 575
- Logical Files - Logische Dateien 576
- Recovery Files - Wiederherstellungsdateien 577
- Administration Facilities - Verwaltungsfunktionen 577
- Hilfe zu Direktkommandos und Menüfunktionen 578

Das Dienstprogramm SYSEDT (im Folgenden „SYSEDT Utility“) bietet Verwaltungsfunktionen für den Editor Buffer Pool, der als Datencontainer für den Software AG Editor dient. Die SYSEDT Utility kann für Folgendes verwendet werden:

- Anzeigen von Parametern und Laufzeitinformationen des Editor Buffer Pools.
- Ändern von Parametern zur Steuerung und Initialisierung des Editor Buffer Pools und seiner Arbeitsdatei.
- Löschen von logischen Arbeitsdateien und Wiederherstellungsdateien.

Verwandte Dokumente:

- *Operating the Software AG Editor* in der *Operations*-Dokumentation
- *EDBP - Software AG Editor Buffer Pool-Definitionen* und *EDPSIZE - Größe des Software AG Editor-Hilfs-Buffer-Pool* in der *Parameter-Referenz-Dokumentation*

Library-Profil für Natural Security definieren

Wenn bei Ihnen Natural Security verwendet wird, müssen Sie ein Library-Sicherheitsprofil (Library Security Profile) für das Dienstprogramm SYSEDT erstellen.

Weitere Informationen siehe *Library Maintenance* in der *Natural Security*-Dokumentation.

SYSEDT Utility aufrufen und eine Funktion ausführen

Dieser Abschnitt beschreibt, wie Sie die SYSEDT Utility aufrufen und eine SYSEDT-Funktion ausführen.

Um die SYSEDT Utility aufzurufen:

- Geben Sie das folgende Natural-System-Kommando ein:

  ```
  SYSEDT
  ```

  Es erscheint das SYSEDT-Hauptmenü (**Main Menu**).

  Beispiel:

```
15:47:50             ***** NATURAL SYSEDT UTILITY *****             2009-01-29
User MMO                        - Main Menu -                    TID 1     33



                    Code   Function

                      G    General Information
                      P    Generation Parameters
                      U    Users
                      F    Logical Files
                      R    Recovery Files
                      A    Administration Facilities
                      .    Exit

              Code ..




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Cont  Help  Menu  Exit        Admin Files Users Recov       GInfo Parms Canc
```

Im Hauptmenü verfügbare Funktionen werden in anschließenden Kapiteln beschrieben.

Um eine SYSEDT-Funktion auszuführen:

- Geben Sie im Feld **Code** des SYSEDT-Hauptmenüs den aus einem einzelnen Zeichen bestehenden Code ein, der der auszuführenden Funktion entspricht, und drücken Sie Enter. Zum Beispiel:

  ```
  G
  ```

  Die Funktion **General Information** (Allgemeine Informationen) wird ausgeführt.

  Oder:

  Drücken Sie im SYSEDT-Hauptmenü die PF-Taste, die der Funktion entspricht, die Sie ausführen möchten. Zum Beispiel:

  PF10 (GInfo)

  Die Funktion **General Information** (Allgemeine Informationen) wird ausgeführt.

  Oder:

  Geben Sie das Systemkommando `SYSEDT` gefolgt von einem beliebigen Funktionscode aus dem SYSEDT-Hauptmenü ein. Zum Beispiel:

```
SYSEDT G
```

Die Funktion **General Information** (Allgemeine Informationen) wird ausgeführt.

Eine Liste aller verfügbaren SYSERR-Direktbefehle finden Sie in der Hilfefunktion, die unter *Hilfe zu Direktkommandos und Menüfunktionen* beschrieben ist.

General Information - Allgemeine Informationen

Diese Funktion ruft den Bildschirm General Information auf, der einen Überblick über den aktuellen Status des Editor Buffer Pools gibt. Die auf dem Bildschirm vorhandenen Felder werden in der folgenden Tabelle beschrieben. Ausführliche Informationen zu diesen Feldern finden Sie unter *Editor Work File* und *Editor Buffer Pool* in der *Operations*-Dokumentation.

Feld	Beschreibung
Usage Statistics	**Nutzungsstatistik** Die aktuell verfügbare Gesamtzahl, die aktuell genutzte Anzahl und der aktuell genutzte Prozentsatz der verfügbaren Anzahl der nachfolgenden Elemente.
Buffer Pool Blocks	**Buffer Pool-Blöcke** Die Anzahl der Blöcke im Editor Buffer Pool. Siehe auch *Obtaining Free Blocks*.*
Work File Records	**Arbeitsdatei-Datensätze** Die Anzahl der Datensätze in der Arbeitsdatei des Editors. Siehe auch *Editor Work File*.*
Control	**Steuerung** Die Anzahl der Steuerungsdatensätze, die immer eins ist. Siehe auch *Control Record*.*
Work	**Work-Sätze** Die Anzahl der Work-Datensätze. Siehe auch *Work Record* .*
Recovery	**Wiederherstellung** Die Anzahl der Wiederherstellungssätze. Siehe auch *Recovery Records*.*
Logical Files	**Logische Dateien** Die Anzahl der logischen Dateien.
Requests	**Anforderungen** Die Gesamtzahl der Lese- und Schreibanforderungen, die Anzahl der Lese- und Schreibanforderungen für Buffer Pool-Blöcke (Spalte **Pool**) und die Anzahl der Lese- und Schreibanforderungen für Work- oder Recovery-Dateien (Spalte **File**). Die Spalte **Copy** zeigt die Leseanforderungen, die (im Gegensatz zu den

Feld	Beschreibung
	gesperrten Leseanforderungen) zur Löschung des entsprechenden Puffer Pool-Blocks führen.
Read Work	**Anzahl der Leseanforderungen** Die Anzahl der Leseanforderungen für logische Dateisätze. Ein logischer Dateisatz kann sich im Buffer Pool (Spalte **Pool**) oder in der Arbeitsdatei (Spalte **File**) befinden. Er kann durch eine gesperrte oder eine Kopieranforderung gelesen werden: gesperrt bedeutet, dass der Datensatz für eine bestimmte Zeit im Buffer Pool gehalten wird; kopieren bedeutet, dass er aus dem Buffer Pool gelöscht wird, nachdem er gelesen wurde.
Write Work	**Anzahl der Schreibanforderungen** Die Anzahl der Schreibanforderungen für logische Dateisätze. Ein Datensatz kann entweder in den Buffer Pool (**Pool**) geschrieben oder in die Arbeitsdatei (**File**) verschoben werden, wenn keine freien Blöcke verfügbar sind.
Read Recovery	**Wiederherstellungssätze lesen** Die Anzahl der Leseanforderungen für Wiederherstellungssätze in der Arbeitsdatei des Editors.
Write Recovery	**Wiederherstellungssätze schreiben** Die Anzahl der Schreibanforderungen für Wiederherstellungsdatensätze in der Arbeitsdatei des Editors.
Timeout Values in Seconds	**Zeitüberschreitungswerte** Elemente mit Timeout-Werten, die in Sekunden angegeben werden. Diese Timeout-Werte können nach Drücken von PF5 (Update/Ändern) geändert und nach erneutem Drücken von PF5 (Save/Speichern) dynamisch gesetzt werden. Die geänderten Werte werden bei einem Neustart des Buffer Pool nicht beibehalten. Stattdessen werden die Werte aus dem Workfile-Steuerungsdatensatz verwendet. Siehe auch *Obtaining Free Blocks*.*
Logical Files	**Löschung logischer Dateien** Die Zeit, nach der eine logische Datei gelöscht wird, wenn während dieser Zeit nicht auf sie zugegriffen wurde.
Files Delete Check	**Löschung aller Dateien** Die Zeit, nach der alle logischen Dateien periodisch überprüft werden, ob sie gelöscht werden können.
Changed Blocks	**Geänderte Blöcke** Die Zeit, nach der Blöcke, die geändert wurden, durch Schreiben in die Arbeitsdatei freigegeben werden können.
Unchanged Blocks	**Unveränderte Blöcke** Die Zeit, nach der Blöcke, die nicht verändert wurden, durch Schreiben in die Arbeitsdatei freigegeben werden können.

Feld	Beschreibung
Locked Blocks	**Gesperrte Blöcke** Die Zeit, nach der Blöcke, die mit gesperrt gelesen wurden, durch Schreiben in die Arbeitsdatei freigegeben werden können.

* Beschreibung siehe *Operations*-Dokumentation

Generation Parameters - Generierungs-Parameter

Die Funktion **Generation Parameters** ruft einen Bildschirm auf, der die aktuellen Parametereinstellungen des Editor-Buffer-Pools anzeigt. Sie können die vorhandenen Parameterwerte ändern.

Parameter	Beschreibung
DDNAME	Der Name der Editor-Arbeitsdatei für die JCL-Definition.
DSNAME	Der Name des Dataset (Datei) der Arbeitsdatei. Dieser Parameter wird in einer Com-plete-Umgebung nicht angezeigt.
RECNUM	Die Gesamtzahl der Sätze der Arbeitsdatei.
LRECL	Die Länge der Arbeitsdatei-Datensätze.
PWORK	Der Prozentsatz der Arbeitsdatei-Datensätze, die als Arbeitsdatei-Datensätze verwendet werden.
RWORK	Der Prozentsatz der für reguläre logische Dateien verwendeten Arbeitsdatensätze.
MAXLF	Die maximale Anzahl der logischen Dateien im Editor Buffer-Pool.
FTOUT	Der Timeout-Wert (in Sekunden), nach dem eine logische Datei, auf die nicht zugegriffen wurde, gelöscht werden soll.
DTOUT	Die Zeitspanne (in Sekunden), in der logische Dateien auf Löschung geprüft werden sollen.
CTOUT	Der Timeout-Wert (in Sekunden) für geänderte Buffer Pool-Blöcke.
UTOUT	Der Timeout-Wert (in Sekunden) für unveränderte Buffer Pool-Blöcke.
LTOUT	Der Timeout-Wert (in Sekunden) für gesperrte Buffer Pool-Blöcke.
ITOUT	Der Timeout-Wert (in Sekunden) für die Buffer Pool-Initialisierung.

Bei den oben genannten Parametern handelt es sich um Schlüsselwort-Subparameter des Profilparameters EDBP (oder des entsprechenden Makros NTEDBP), der zur Definition der anfänglichen Editor Buffer Pool-Einstellungen verwendet wird.

Ausführliche Informationen zu EDBP und allen verfügbaren Schlüsselwort-Subparametern finden Sie unter *EDBP - Software AG Editor Buffer Pool-Definitionen* in der *Parameter-Referenz*-Dokumentation.

Die Spalte **Start** auf dem Bildschirm **Generation Parameters** bezieht sich auf den Neustart des Buffer Pools (siehe auch *Initializing the Editor Buffer Pool* in der *Operations*-Dokumentation).

Die folgenden Startwerte können für die Parameter angezeigt werden:

Wert	Beschreibung
L	Der Wert für den entsprechenden Parameter wird entweder aus dem Editor-Parametermodul oder aus der Arbeitsdatei-Definition übernommen.
C	Eine Änderung des entsprechenden Parameterwertes erzwingt einen Kaltstart des Buffer Pool. Wiederherstellungssätze gehen verloren.
W	Eine Änderung des entsprechenden Parameterwerts führt zu einem Warmstart des Buffer Pool. Wiederherstellungssätze werden beibehalten.

Um Parameterwerte zu ändern:

1 Drücken Sie im Bildschirm **Generation Parameters** PF5 (Update/Ändern).

Die schreibgeschützten Felder neben den Parametern werden zu Eingabefeldern.

2 Ersetzen Sie den/die gewünschten Parameterwert(e) und drücken Sie PF5 (Save/Speichern), um die neuen Werte im Steuerungsdatensatz der Editor-Arbeitsdatei zu speichern. Sie werden aktiviert, wenn der Editor Buffer Pool erneut gestartet wird.

Wenn Sie den Wert von PWORK ändern, werden Sie in einem Fenster aufgefordert, den Kaltstart der Editor-Arbeitsdatei zu bestätigen. Geben Sie YES ein, um den neuen Parameterwert zu bestätigen und einen Buffer Pool-Kaltstart für die Initialisierung der Arbeitsdatei durchzuführen. (Jede andere Eingabe oder keine Eingabe setzt den Parameter auf seinen vorherigen Wert zurück).

Users - Benutzer

Die Funktion **Users** ruft einen Bildschirm auf, der die folgenden Informationen enthält:

Spalte	Beschreibung
User ID	Die Natural-Benutzerkennung.
Logical Files	Die Anzahl der logischen Dateien, die pro Benutzer definiert sind.
Pool Blocks	Die Anzahl der Buffer Pool-Blöcke pro Benutzer.
Work Records	Die Anzahl der Arbeitsdatensätze pro Benutzer.
Recovery Files	Die Anzahl der Wiederherstellungsdateien pro Benutzer.
Recovery Records	Die Anzahl der Wiederherstellungsdatensätze pro Benutzer.

Bei jedem aufgeführten Benutzer können Sie eines der folgenden Zeilenkommandos ausführen. Geben Sie das Zeilenkommando in der Spalte **C** neben dem gewünschten Benutzer ein.

Zeilenkommando	Beschreibung
? oder *	Öffnet ein Fenster mit einer Liste aller gültigen Zeilenkommandos und den entsprechenden PF-Tasten, falls vorhanden.
.	Bewirkt das Verlassen des aktuellen Bildschirms und Rückkehr zum vorherigen Bildschirm.
/ oder P	Setzt die Zeile an den oberen Rand des Bildschirms.
F	Wählt die logischen Dateien dieses Benutzers aus.
R	Wählt die Wiederherstellungsdateien dieses Benutzers aus.
D	Löscht alle logischen Dateien und/oder Wiederherstellungsdateien für diesen Benutzer.

Logical Files - Logische Dateien

Die Funktion **Logical Files** ruft einen Bildschirm auf, der die folgenden Informationen enthält:

Spalte	Beschreibung
File No.	Die Nummer der logischen Datei.
User ID	Die Natural-Benutzerkennung.
Type	Der Typ der logischen Datei.
Pool Blks	Die Anzahl der Buffer Pool-Blöcke, die derzeit pro logischer Datei verwendet werden.
File Recs	Die Anzahl der aktuell pro logischer Datei zugewiesenen Arbeitsdateisätze.
Last Access	Das Datum und die Uhrzeit der letzten Lese- oder Schreibanforderung pro logischer Datei.

Bei jeder aufgeführten logischen Datei können Sie eines der folgenden Zeilenkommandos ausführen. Geben Sie das Zeilenkommando in der Spalte **C** neben dem gewünschten Benutzer ein.

Zeilenkommando	Beschreibung
? oder *	Öffnet ein Fenster mit einer Liste aller gültigen Zeilenkommandos und den entsprechenden PF-Tasten, falls vorhanden.
.	Bewirkt das Verlassen des aktuellen Bildschirms und Rückkehr zum vorherigen Bildschirm.
/ oder P	Positioniert die Zeile an den oberen Rand des Bildschirms.
S	Wählt die logischen Dateien dieses Benutzers aus.
D	Löscht die logische Datei.

Recovery Files - Wiederherstellungsdateien

Die Funktion **Recovery Files** ruft einen Bildschirm auf, der die folgenden Informationen enthält:

Column	Beschreibung
User ID	Die Natural-Benutzerkennung.
Member	Der Name des Library Members.
Library	Der Name der Library.
Type	Der Library-Typ.
Recs	Die Anzahl der Wiederherstellungsdatensätze pro Wiederherstellungsdatei.
Creation Date	Das Datum der Erstellung der Wiederherstellungsdatei.
Creation Time	Die Uhrzeit der Erstellung der Wiederherstellungsdatei.

Bei jeder aufgeführten Wiederherstellungsdatei können Sie eines der folgenden Zeilenkommandos ausführen. Geben Sie das Zeilenkommando in der Spalte **C** neben dem gewünschten Benutzer ein.

Zeilenkommando	Beschreibung
? oder *	Öffnet ein Fenster mit einer Liste aller gültigen Zeilenkommandos und den entsprechenden PF-Tasten, falls vorhanden.
.	Bewirkt das Verlassen des aktuellen Bildschirms und Rückkehr zum vorherigen Bildschirm.
/ oder P	Positioniert die Zeile an den oberen Rand des Bildschirms.
S	Wählt die Wiederherstellungsdateien dieses Benutzers aus.
D	Löscht die Wiederherstellungsdatei.

Administration Facilities - Verwaltungsfunktionen

Die Funktion **Administration Facilities** ruft einen Bildschirm auf, den Sie benutzen können, um den Editor Buffer Pool oder die Utility SYSEDT zu beenden oder den Inhalt des Editor Buffer Pool zur Problemdiagnose im Dump-Format (hexadezimal) anzuzeigen.

Um den Bildschirm Administration Facilities zu benutzen:

- Beenden Sie den Editor Buffer Pool, indem Sie den Funktionscode T eingeben.

Es erscheint ein Fenster, in dem Sie zur Bestätigung aufgefordert werden.

Wenn Sie `YES` eingeben, öffnet sich ein zusätzliches Fenster, in dem Sie gefragt werden, ob der Editor Buffer Pool sofort neu gestartet werden soll oder nicht.

- Wenn Sie erneut `YES` eingeben, wird der Buffer Pool sofort neu gestartet, was Ihnen die Möglichkeit gibt, geänderte Generierungsparameter sofort zu aktivieren.
- Wenn Sie `NO` eingeben, verlassen Sie die Utility SYSEDT und können Aktionen außerhalb Ihrer TP-Umgebung durchführen, z.B. die Größe Ihrer Editor-Arbeitsdatei ändern.

Oder:

Zeigen Sie den Speicherauszug (Dump) des Editor Buffer Pool an, indem Sie den Funktionscode `D` eingeben.

Die Offsets und Adressen des Editor Buffer Pool werden auf dem Bildschirm angezeigt.

Drücken Sie bei Bedarf `PF10` (RelO), um den Offset `00000000` an den oberen Rand des Bildschirms zu setzen. `PF1` (Help) ruft eine Liste aller verfügbaren PF-Tasten auf.

Oder:

Rückkehr zum SYSEDT-Hauptmenü durch Eingabe eines Punktes (`.`).

Hilfe zu Direktkommandos und Menüfunktionen

Sie können Online-Hilfe-Informationen zu allen verfügbaren SYSEDT-Direktkommandos oder zu der aktuell verwendeten Funktion erhalten.

Um alle SYSEDT-Direktkommandos anzuzeigen:

- Geben Sie in der Kommandozeile eines beliebigen SYSEDT-Bildschirms eines der folgenden Zeichen ein:

```
?
```

oder

```
*
```

Alle in der SYSEDT Utility verfügbaren Direktkommandos werden in alphabetischer Reihenfolge aufgelistet.

Um die Hilfe zu einer bestimmten SYSEDT-Funktion anzuzeigen:

- Geben Sie in der Kommandozeile eines SYSEDT-Bildschirms, der durch eine SYSEDT-Funktion aufgerufen wurde, Folgendes ein:

  ```
  H
  ```

  Oder:

  Drücken Sie PF1 (Help).

  Es werden Hilfe-Informationen zu den auf dem aktuellen SYSEDT-Bildschirm enthaltenen Feldern und Spalten angezeigt.

XV SYSERR Utility

Wenn Sie eine Natural-Anwendung entwickeln, können Sie Fehler- und/oder Informationsmeldungen von Ihrem Natural-Code trennen und sie separat verwalten. Dies erleichtert es, Meldungen zu standardisieren, vordefinierte Meldungsbereiche für verschiedene Meldungstypen zu haben, Meldungen in andere Sprachen zu übersetzen oder an eine Meldung einen Langtext anzuhängen, der sie näher erläutert.

Mit dem Dienstprogramm („Utility“) SYSERR können Sie anwendungsspezifische Meldungen schreiben. Darüber hinaus können Sie mit der SYSERR Utility die Texte der vorhandenen Natural-Systemmeldungen kundenspezifisch anpassen (nicht empfohlen).

Anmerkungen:

1. Die Benutzung der Funktionen der SYSERR Utility kann in Natural Security benutzerspezifisch eingeschränkt werden. Ausführliche Informationen siehe *SYSERR Utility Profiles* in der *Natural Security*-Dokumentation.
2. Die Anwendungsprogrammierschnittstelle (API) USR0020N in der Natural-System-Library SYSEXT kann benutzt werden, um Meldungen von der Systemdatei FNAT oder FUSER zu lesen. Dadurch ist es z.B. möglich, Langmeldungen (als Teil Ihres benutzerdefinierten Hilfesystems) in einer Anwendung anzeigen zu können, ohne dazu die Natural-System-Library SYSERR benutzen zu müssen. Weitere Informationen siehe in der Beschreibung der Utility **SYSEXT**.

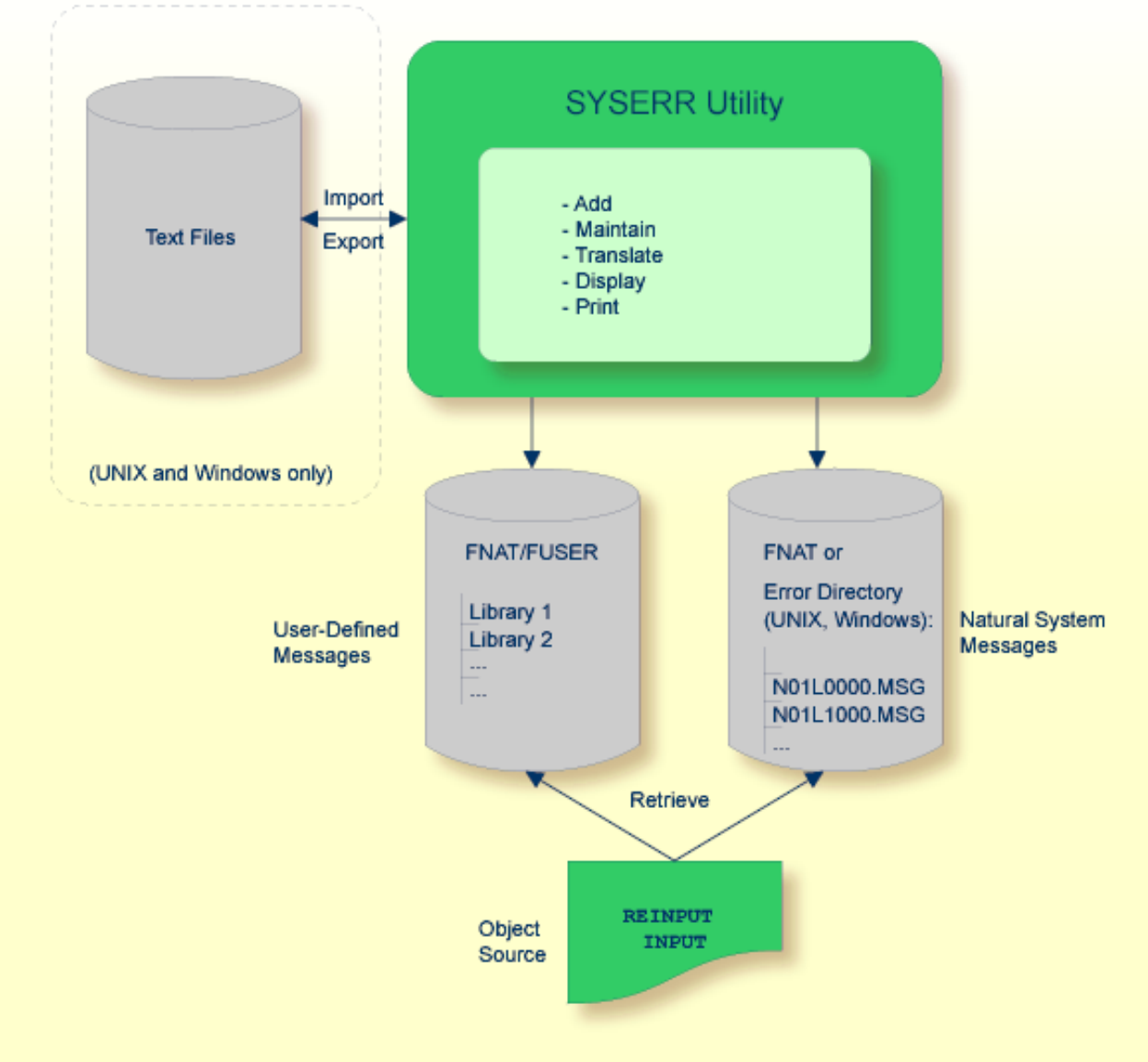
Allgemeine Informationen zu Meldungen
SYSERR aufrufen
Funktionen
Parameter
Direktkommandos
Umwandlung in Großbuchstaben - ERRUPPER
Zeichen ersetzen - ERRCHAR
Meldungen in verschiedenen Libraries verwalten

59 Allgemeine Informationen zu Meldungen

- Meldungstypen 585
- Meldungssprachen 586
- Meldungen und Codepage-Unterstützung 586
- Ausgabe von Meldungen 586
- Abrufen von Natural-System-Kurzmeldungen 587
- Abrufen von benutzerdefinierten Kurzmeldungen 588
- Informationen zur Meldung erhalten 588

Dieses Kapitel beschreibt, welche Meldungstypen es in Natural gibt, welche Meldungssprachen mit der Utility SYSERR verwaltet werden können und wie Meldungen in Ihrer Natural-Systemumgebung abgesetzt und abgerufen werden.

Die folgende Abbildung veranschaulicht die Funktionen und Merkmale der SYSERR Utility und zeigt, wie Meldungen innerhalb von Natural verarbeitet werden:



Meldungstypen

Es gibt zwei Typen von Meldungen: Natural-Systemmeldungen und benutzerdefinierte Meldungen.

Natural-Systemmeldungen werden vom Natural-Nukleus und von Natural Utilities ausgegeben. Die von uns ausgelieferten Natural-Systemmeldungen werden als Meldungsdateien in der Natural-Systemdatei FNAT gespeichert. Sie beginnen stets mit `NAT`. Darauf folgt eine vierstellige Nummer, zum Beispiel: NAT0230.

Benutzerdefinierte Meldungen werden von Anwendungen ausgegeben, die ein Benutzer geschrieben hat. Benutzerdefinierte Meldungen werden als Meldungsdateien in Libraries (einschließlich SYS-Libraries) in der Natural-Systemdatei FUSER oder FNAT gespeichert.

Eine Meldung kann in verschiedene Sprachen übersetzt werden. Jede Sprache wird in einer getrennten Meldungsdatei gespeichert. Pro Library und Meldungsdatei können 9999 Meldungen gespeichert werden.

Es gibt vier Typen von Meldungstexten:

- Natural-System-Kurzmeldung
- Natural-System-Langmeldung
- Benutzerdefinierte Kurzmeldung
- Benutzerdefinierte Langmeldung

Eine Kurzmeldung ist eine einzeilige Meldung, die in der Meldungszeile angezeigt wird, wenn die entsprechende Fehlersituation auftritt.

Eine Langmeldung ist eine ausführliche Erläuterung der entsprechenden Kurzmeldung und enthält Anleitungen zur Lösung von Problemen.

Vorsicht: Änderungen an Natural-Systemmeldungen können zur Folge haben, dass nach der Freigabe einer neuen Natural-Version falsche Meldungen ausgegeben werden können oder die vorgenommenen Änderungen verloren gehen können.

Meldungssprachen

Meldungen können in bis zu 60 Sprachen erstellt werden, siehe auch Beschreibung der Systemvariablen `*LANGUAGE` in der *Systemvariablen*-Dokumentation.

Es gelten folgende Regeln und Einschränkungen:

- Natural-System-Kurzmeldungen müssen zuerst in Englisch eingegeben werden. Sie können erst danach in eine andere Sprache übersetzt werden.
- Natural-System-Langmeldungen können in Englisch eingegeben, aber nicht in andere Sprachen übersetzt werden.
- Benutzerdefinierte Kurzmeldungen können in beliebiger Sprache eingeben und danach in eine andere Sprache übersetzt werden.
- Benutzerdefinierte Langmeldungen können in beliebiger Sprache eingeben werden, jedoch nur wenn die entsprechende Kurzmeldung bereits in dieser Sprache existiert.

Meldungen und Codepage-Unterstützung

Bei jeder Kurz- und Langmeldung kann eine Codepage gespeichert werden.

Mit dem Session-Parameter `SRETAIN=OFF` wird die Codepage der aktuellen Session (s. Systemvariable `*CODEPAGE`) einer Fehlermeldung zugewiesen, wenn diese gespeichert wird.

Wenn mit Session-Parameter `SRETAIN=OFF` eine Fehlermeldung mit SYSERR gelesen wird, wird diese in die Codepage der aktuellen Session konvertiert. Weitere Informationen siehe *Codepage-Unterstützung bei Editoren, Systemkommandos und Utilities* im Abschnitt *Entwicklungsumgebung* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.

Ausgabe von Meldungen

Dieser Abschnitt enthält Informationen zu den Natural-Statements `INPUT` und `REINPUT`, die benutzt werden, um in einem Natural-Programm eine Natural-System-Kurzmeldung oder eine benutzerdefinierte Kurzmeldung auszugeben.

Um eine Natural-System-Kurzmeldung in einem Programm auszugeben:

- Geben Sie eines der folgenden Natural-Statements an:

```
INPUT WITH TEXT *-nnnn ' '
```

oder

```
REINPUT WITH TEXT *-nnnn
```

dabei ist *nnnn* die Nummer der angeforderten Meldung (führende Nullen können weggelassen werden.

Um eine benutzerdefinierte Kurzmeldung in einem Programm auszugeben:

- Geben Sie eines der folgenden Natural-Statements an:

```
INPUT WITH TEXT *nnnn ' '
```

oder

```
REINPUT WITH TEXT *nnnn
```

dabei ist *nnnn* die Nummer der angeforderten Meldung (führende Nullen können weggelassen werden.

Dynamisches Ersetzen von Meldungstext

Ein Meldungstext kann variable Teile enthalten. Diese werden durch die Notation `:n:` gekennzeichnet. Dabei steht *n* für Ausprägungen 1 bis 7. Diese variablen Teile werden zur Laufzeit durch einen Wert ersetzt.

Ausführliche Informationen siehe *operand3* im Abschnitt *INPUT-Syntax 1 - Dynamisch generierter Eingabeschirm* und *operand3* im Abschnitt *REINPUT* in der *Statements*-Dokumentation.

Abrufen von Natural-System-Kurzmeldungen

Wenn ein Programm eine Natural-System-Kurzmeldung referenziert, sucht Natural nach der angeforderten Meldungsnummer in der Natural-Systemdatei FNAT in folgender Reihenfolge:

1. Unter dem aktuellen Sprachcode gemäß Festlegung durch die Systemvariable `*LANGUAGE`,
2. Unter dem Sprachcode 1 (Englisch).

Wird keines von beiden gefunden, referenziert ein Programm eine Meldung, die nicht existiert, und Sie erhalten nur die Meldungsnummer mit dem Präfix `NAT`, zum Beispiel: NAT0230.

Abrufen von benutzerdefinierten Kurzmeldungen

Wenn ein Programm eine benutzerdefinierte Kurzmeldung referenziert, sucht Natural in folgender Reihenfolge:

- zuerst nach der angeforderten Meldungsnummer *nnnn* unter dem aktuellen Sprachcode gemäß Festlegung durch die Systemvariable `*LANGUAGE` (siehe *Systemvariablen*-Dokumentation),
- Wenn diese Meldung nicht existiert, sucht Natural nach der angeforderten Meldungsnummer *nnnn* unter dem Sprachcode 1 (Englisch).
- Und wenn diese Meldung auch nicht existiert, sucht Natural nach der Meldungsnummer *n*`000` (wobei *n* die erste Stelle der angeforderten Meldungsnummer ist) unter dem Sprachcode 1 (Englisch).

Diese drei Suchschritte werden zuerst in der aktuellen Library durchgeführt. Wurde dort nichts gefunden, werden weitere Libraries auf die gleiche Weise durchsucht, bis eine entsprechende Meldung gefunden wird.

Die Suchreihenfolge in Libraries ist Folgende:

1. Die aktuelle Library gemäß Festlegung durch die Systemvariable `*LIBRARY-ID`,
2. die Steplibs (wenn Natural Security installiert ist, in der Reihenfolge, die im Natural Security-Profil der aktuellen Library angegeben ist),
3. die Standard-Steplib gemäß Festlegung durch die Systemvariable `*STEPLIB`,
4. die Library SYSTEM in der Systemdatei FUSER (*),
5. die Library SYSTEM in der Systemdatei FNAT (*).

(*) Wenn der Name der aktuellen Library mit SYS beginnt, dann wird SYSTEM FNAT vor SYSTEM FUSER durchsucht.

Informationen zur Meldung erhalten

Wenn Sie eine Kurzmeldung erhalten, können Sie zusätzliche Informationen zur Problemsituation erhalten.

- Mit dem Systemkommando `HELP` können Sie Natural-System-Langmeldungen oder benutzerdefinierte Langmeldungen anzeigen.
- Mit dem Systemkommando `LASTMSG` erhalten Sie - jeweils zu der zuletzt aufgetretenen Fehlersituation - die ausgegebene Fehlermeldung sowie alle vorherigen (nicht ausgegebenen) Fehlermeldungen, die zu diesem Fehler geführt haben.

Beschreibung der beiden Systemkommandos siehe *Systemkommandos*-Dokumentation.

60 SYSERR aufrufen

- SYSERR ausführen 592

In diesem Kapitel wird beschrieben, wie Sie die SYSERR Utility ausführen.

SYSERR ausführen

Um die SYSERR Utility auszuführen:

- Geben Sie in der Kommandozeile des **Natural Main Menu** das folgende Systemkommando ein:

```
SYSERR
```

Das Menü der SYSERR Utility wird angezeigt:

```
18:23:46              ***** NATURAL SYSERR UTILITY *****              2022-03-04
                                    - Menu -

                 Code  Function
                 ----  ----------------------------------------
                  AD   Add new messages
                  DE   Delete messages
                  DI   Display messages
                  MO   Modify messages
                  PR   Print messages
                  SC   Scan in messages
                  SE   Select messages from a list
                  TR   Translate messages into another language
                  ?    Help
                  .    Exit
                 ----  ----------------------------------------
           Code .. __   Message type .... US
                        Library ......... XHTRI___
                        Message number .. 1___ - 9999
                        Language codes .. 1________
Please enter code.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit                                                 Canc  ↩
 ↩
```

Im Menü der SYSERR Utility können Sie alle SYSERR-Funktionen ausführen, die zum Anlegen und Pflegen von Meldungen zur Verfügung stehen. Die einzelnen Funktionen werden im Kapitel *Funktionen* erklärt. Die zutreffenden Parameter werden allgemein im Kapitel *Parameter* erläutert. Einschränkungen, die für die Benutzung dieser Parameter gelten, werden bei der jeweils betroffenen Funktion im Kapitel *Funktionen* beschrieben (s.u.).

Die SYSERR Utility bietet ein umfangreiches Online-Hilfesystem. Um Hilfe zu einem Feld zu erhalten, können Sie entweder in dem betreffenden Feld ein Fragezeichen (?) eingeben und ENTER drücken oder den Cursor in das Feld stellen und PF1 drücken.

61 Funktionen

- Meldungen anlegen 596
- Meldungen löschen 600
- Meldungen anzeigen 601
- Meldungen ändern 602
- Meldungen drucken 605
- Meldungen durchsuchen 607
- Meldungen aus Liste auswählen 610
- Meldungen in andere Sprachen übersetzen 613

Um eine Funktion der SYSERR Utility aufzurufen:

- Geben Sie den Code, der der gewünschten Funktion entspricht, sowie einen oder mehrere Parameter in die Eingabefelder des Menüs der SYSERR Utility ein und drücken Sie ENTER.

Dieses Kapitel beschreibt die im Menü zur Verfügung stehenden Funktionen und die Parameter, die zu jeder Funktion angegeben werden können. Allgemeine Informationen zum Gebrauch von Parametern finden Sie im Kapitel *Parameter*.

Meldungen anlegen

Um neue Meldungen anzulegen:

1 Standardmäßig unterscheidet die SYSERR Utility Groß- und Kleinschreibung. Falls Sie eine Umsetzung von Klein- in Großschreibung wünschen, können Sie folgendes Terminalkommando eingeben:

```
%U
```

Alle Kleinbuchstaben, die Sie beim Anlegen eines Meldungstextes eintippen, werden dann in Großbuchstaben umgesetzt. Dies gilt für die gesamte Dauer der aktuellen Natural-Session. Ebenso wird Meldungstext in Großbuchstaben umgesetzt, wenn der Profilparameter `TS` auf `ON` gesetzt ist (siehe *Parameter-Referenz*-Dokumentation).

Ausführliche Informationen zum Terminalkommando `%U` finden Sie in der *Terminalkommandos*-Dokumentation.

2 Rufen Sie das Menü der SYSERR Utility auf und geben Sie die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** `AD` (Add new messages/neue Meldungen anlegen)
Message type	**Meldungstyp:** `NS` Natural-System-Kurzmeldungen `NL` Natural-System-Langmeldungen `US` Benutzerdefinierte Kurzmeldungen `UL` Benutzerdefinierte Langmeldungen Eine Langmeldung kann erst dann angelegt werden, wenn die entsprechende Kurzmeldung existiert, weil die Langmeldung als Erklärung der Kurzmeldung gedacht ist.

Feld	Eingabewert
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des anzulegenden Meldungsbereichs. Falls Sie nur eine Meldung anlegen möchten, geben Sie entweder die Nummer der neuen Meldung im linke Feld **Message number** ein und löschen Sie den Inhalt des rechten Feldes oder geben Sie die Nummer in beide Felder ein.
Language codes	**Sprachecode:** Der Code der Sprache, in der die Meldung angelegt werden soll. Wenn der Meldungstyp `NS` oder `NL` ist, muss der Sprachcode `1` für Englisch sein. Bei anderen Meldungstypen wird der erste in das Feld eingegebene Sprachcode verwendet, alle anderen werden nicht beachtet.

3 Drücken Sie ENTER.

Der Bildschirm **Add Short Message** wird angezeigt (Beispiel):

```
18:35:50              ***** NATURAL SYSERR UTILITY *****             2022-03-04
Lower Case ON               - Add Short Message -

Number        Short Message (German)
------------  ------------------------------------------------------------------
XHTRI1004     Kurzmeldungsbeispiel Nummer 1004
              ....+....1....+....2....+....3....+....4....+....5....+..

Sample ......









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Mod               Exit                    -     +     Long              Canc  ↩
 ↩
```

Das Feld **Number** zeigt die Meldungsnummer (`1004` im obigen Beispiel). Sie hat die Library-Kennung als Präfix (`SYSERR` im obigen Beispiel).

4 Geben Sie in der Eingabezeile bei der Meldungsnummer einen Kurzmeldungstext ein und drücken Sie ENTER.

Es erscheint die Meldung: `Message has been added.`

Oder:

Wenn die mit `Sample` beschriftete Zeile, wie im obigen Beispiel gezeigt, einen Beispiel-Meldungstext enthält, kopieren Sie diesen Text in die Eingabezeile, indem Sie `.C` eingeben und dann ENTER drücken. Wenn der Beispiel-Meldungstext die Zeichenkette `0000` enthält, wird diese Zeichenkette durch die neue Meldungsnummer ersetzt, siehe folgenden Beispiel-Bildschirm:

```
18:35:50              ***** NATURAL SYSERR UTILITY *****             2022-03-04
Lower Case ON               - Add Short Message -

Number        Short Message (German)
------------  ------------------------------------------------------------------
SYSERR1004    Kurzmeldungsbeispiel Nummer 1004
              ....+....1....+....2....+....3....+....4....+....5....+..

Sample ...... Kurzmeldungsbeispiel Nummer 0000









Message has been added.

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

Eine Anleitung zum Erstellen einer Beispiel-Meldung finden Sie beim Kommando `SAMPLE` im Abschnitt *Direktkommandos*.

5 Drücken Sie PF9, um einen entsprechenden Langmeldungstext anzulegen.

Der Bildschirm **Add Long** *`message-number`* erscheint (Beispiel):

```
18:42:41        - Add Long XHTRI1004 (German) - Lower Case ON        2022-03-04
 1 Tx. Kurzmeldungsbeispiel Nummer 1004
 2     .
 3     .
 4 Ex. .
 5     .
 6     .
 7     .
 8     .
 9     .
10     .
11     .
12     .
13     .
14     .
15     .
16     .
17     .
18 Ac. .
19     .
20     .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Add               Exit                              Short Copy        Canc  ↩
 ↩
```

6 Geben Sie in den drei Eingabebereichen Text ein: **Tx.** (Text), **Ex.** (Explanation/Erklärung) und **Ac.** (Action/Maßnahme).

7 Drücken Sie ENTER, um die Langmeldung zu speichern.

8 Drücken Sie PF9, um zur Anzeige der Kurzmeldung zurückzukehren oder um die nächste Kurzmeldung in aufsteigender Reihenfolge anzulegen, wenn Sie eine Bereich von Meldungsnummern ausgewählt haben.

9 Drücken Sie PF3 oder PF12, um zum Menü der SYSERR Utility zurückzukehren.

Oder:

Drücken Sie PF8 oder PF7, um die nächste Kurzmeldung in aufsteigender oder absteigender Reihenfolge anzulegen, wenn Sie eine Bereich von Meldungsnummern ausgewählt haben.

Meldungen löschen

Um Meldungen zu löschen:

- Geben Sie in den Feldern des Menüs der SYSERR Utility die folgenden Werte ein:

  Rufen Sie das Menü der SYSERR Utility auf und geben Sie die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** `DE` (Delete messages/Meldungen löschen)
Message type	**Meldungstyp:** `NS` Natural-System-Kurzmeldungen `NL` Natural-System-Langmeldungen `US` Benutzerdefinierte Kurzmeldungen `UL` Benutzerdefinierte Langmeldungen Eine Langmeldung kann gelöscht werden, ohne zuvor die entsprechende Kurzmeldung zu löschen, jedoch nicht umgekehrt. Wenn Sie versuchen, eine Kurzmeldung zu löschen, zu der eine Langmeldung existiert, werden Sie aufgefordert, die Löschung beider Meldungen zu bestätigen.
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des zu löschenden Meldungsbereichs.
Language codes	**Sprachcodes:** Code(s) der Sprache(n), in der bzw. denen die Meldungen gelöscht werden sollen. Sie können einen Stern (*) eingeben, um anzuzeigen, dass die angegebenen Meldungen in allen vorhandenen Sprachen gelöscht werden sollen.

Meldungen anzeigen

Um Meldungen anzuzeigen:

1 Geben Sie in den Feldern des Menüs der SYSERR Utility die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** `DI` (Display messages/Meldungen anzeigen)
Message type	**Meldungstyp:** `NS` Natural-System-Kurzmeldungen `NL` Natural-System-Langmeldungen `US` Benutzerdefinierte Kurzmeldungen `UL` Benutzerdefinierte Langmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des anzuzeigenden Meldungsbereichs.
Language codes	**Sprachcodes:** Der Code der Sprache, in der die Meldungen angezeigt werden sollen. Es wird nur ein Sprachcode akzeptiert. Wenn mehr als ein Sprachcode angegeben wird, wird nur der erste Code verwendet, alle anderen werden nicht beachtet.

2 Drücken Sie ENTER.

Bei Kurzmeldungen erscheint der Bildschirm **Display Short Messages** (Beispiel):

```
18:44:27              ***** NATURAL SYSERR UTILITY *****           2022-03-04
                          - Display Short Messages -

Number         Short Message (German)
------------  ------------------------------------------------------------
NAT0001      Fehlende/undefinierte Syntax; Variablenname/Schluesselwort undef. ↩

NAT0002       Keine Datei zu angegebenem Name bzw. Nummer vorhanden.
NAT0003       Ungueltige Zeichenkette fuer Dateiname oder Dateinummer.
NAT0004       DEFINE DATA muss erstes Statement sein, wenn vorhanden.
NAT0005       Schliessende Klammer fehlt in arithmetischem/logischem Ausdruck.
NAT0006      ESCAPE-Statement benutzt, aber keine Verarbeitungsschleife aktiv.
NAT0007      Ungueltige THRU oder TO Klausel in READ LOGICAL oder HISTOGRAM. ↩









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

Drücken Sie PF8, um eine Seite weiter zu blättern.

Bei Langmeldungen wird der Bildschirm **Display Long Message** angezeigt. Um die Meldungen nacheinander anzuzeigen, können Sie PF8 (+) drücken. Um zurückzublättern, können Sie PF7 (-) drücken.

Der Bildschirm **Display Long Message** ähnelt dem Bildschirm **Modify Long** *message-number*, der im Abschnitt *Meldungen ändern* abgebildet ist.

Meldungen ändern

Um Meldungen zu ändern:

1 Standardmäßig unterscheidet die SYSERR Utility Groß- und Kleinschreibung. Falls Sie eine Umsetzung von Klein- in Großschreibung wünschen, können Sie folgendes Terminalkommando eingeben:

```
%U
```

Alle Kleinbuchstaben, die Sie beim Anlegen eines Meldungstextes eintippen, werden dann in Großbuchstaben umgesetzt. Dies gilt für die gesamte Dauer der aktuellen Natural-Session.

Ebenso wird Meldungstext in Großbuchstaben umgesetzt, wenn der Profilparameter `TS` auf `ON` gesetzt ist (siehe *Parameter-Referenz*-Dokumentation).

Ausführliche Informationen zum Terminalkommando `%U` finden Sie in der *Terminalkommandos*-Dokumentation.

2 Geben Sie in den Feldern des Menüs der SYSERR Utility die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** `MO` (Modify messages/Meldungen ändern)
Message type	**Meldungstyp:** `NS` Natural-System-Kurzmeldungen `NL` Natural-System-Langmeldungen `US` Benutzerdefinierte Kurzmeldungen `UL` Benutzerdefinierte Langmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des zu ändernden Meldungsbereichs.
Language codes	**Sprachcodes:** Der Code der Sprache, in der die Meldungen geändert werden sollen. Es wird nur ein Sprachcode akzeptiert. Wenn mehr als ein Sprachcode angegeben wird, wird nur der erste Code verwendet, alle anderen werden nicht beachtet.

3 Drücken Sie ENTER.

Bei Kurzmeldungen erscheint der Bildschirm **Modify Short Message** (Beispiel):

```
18:56:38              ***** NATURAL SYSERR UTILITY *****             2022-03-04
Lower Case ON             - Modify Short Message -


Number        Short Message (German)
------------  ------------------------------------------------------------------
XHTRI1004     Kurzmeldungsbeispiel Nummer 1004
              ....+....1....+....2....+....3....+....4....+....5....+..


 1 Tx. Eingabe fehlt.
 2     .
 3     .
 4 Ex. Eingabewert fehlt im Feld XYZ.
 5     Geben Sie einen alphanumerischen Wert ein.                            ↩

 6     .
 7     .
 8     .
18 Ac. Geben Sie einen Wert im Feld XYZ ein.
19     .
20     .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Mod               Exit                    -     +           Copy        Canc  ↩
 ↩
```

Zu Referenzzwecken wird die Langmeldung in der unteren Hälfte des Bildschirms angezeigt.

Beim Ändern von Langmeldungen wird der Bildschirm **Modify Long** *message-number* angezeigt:

```
19:03:13      - Modify Long XHTRI1004 (German) - Lower Case ON       2022-03-04
 1 Tx. Eingabe fehlt.
 2     .
 3     .
 4 Ex. Eingabewert fehlt im Feld XYZ.
 5     Geben Sie einen alphanumerischen Wert ein.
 6     .
 7     .
 8     .
 9     .
10     .
11     .
12     .
13     .
14     .
15     .
16     .
17     .
18 Ac. Geben Sie einen Wert im Feld XYZ ein.
19     .
20     .

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Mod               Exit                    -     +           Copy        Canc  ↩
 ↩
```

4 Drücken Sie ENTER, um Änderungen zu speichern.

5 Drücken Sie PF8 oder PF7, um die nächste Meldung in aufsteigender oder absteigender Reihenfolge zu ändern, wenn Sie einen Nummernbereich ausgewählt haben.

Meldungen drucken

Um Meldungen zu drucken:

1 Geben Sie in den Feldern des Menüs der SYSERR Utility die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** PR (Print messages/Meldungen drucken)
Message type	**Meldungstyp:** NS Natural-System-Kurzmeldungen NL Natural-System-Langmeldungen US Benutzerdefinierte Kurzmeldungen

Feld	Eingabewert
	`UL` Benutzerdefinierte Langmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des zu druckenden Meldungsbereichs.
Language codes	**Sprachcodes:** Der Code der Sprache, in der die Meldungen gedruckt werden sollen. Es wird nur ein Sprachcode akzeptiert. Wenn mehr als ein Sprachcode angegeben wird, wird nur der erste Code verwendet, alle anderen werden nicht beachtet.

2 Drücken Sie ENTER.

Das Fenster **Print Natural System Messages** wird angezeigt (Beispiel):

```
+------Print Natural System Messages-------+
!                                           !
!   Library .......... XHTRI                !
!   Language code .... 2                    !
!                                           !
!   Long texts, too .. N                    !
!   Message number ... 1___ -  25           !
!   Lines per page ... 60_                  !
!   Left margin ...... 10                   !
!   Top margin ....... 0_                   !
!   Bottom margin .... 0_                   !
!   Printer .......... PRT1____             !
!                                           !
+-------------------------------------------+
```

3 Geben Sie in den Feldern des Fensters die Druckoptionen und den logischen Druckernamen an.

Ausführliche Informationen zu logischen Druckern siehe Statement `DEFINE PRINTER` in der *Natural Statements*-Dokumentation.

4 Drücken Sie ENTER, um die ausgewählten Meldungen an einen Drucker auszugeben.

Um alle Natural-Systemmeldungen zu drucken:

- Geben Sie in den Feldern des SYSERR-Menüs die folgenden Werte ein:

Code	PR
Message type	NS oder NL
Message number	1 - 9999
Language code	1 (Englisch) oder 2 (Deutsch)

Die Eingabe einer Library-Kennung ist nicht erforderlich und wird nicht beachtet.

Meldungen durchsuchen

Mit der Scan-Funktionen können Sie Meldungen nach einer bestimmten Zeichenkette durchsuchen. Nur Kurzmeldungen können durchsucht werden.

Um Meldungen zu durchsuchen:

1 Geben Sie in den Feldern des Menüs der SYSERR Utility die folgenden Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** SC (Scan messages/Meldungen durchsuchen)
Message type	**Meldungstyp:** NS Natural-System-Kurzmeldungen US Benutzerdefinierte Kurzmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des zu durchsuchenden Meldungsbereichs.
Language codes	**Sprachcodes:** Geben Sie maximal neun Sprachcodes aus den Bereichen 1 - 9, A - Z und a - y ein oder geben Sie einen Stern (*) für alle Sprachen ein.

2 Drücken Sie ENTER.

Ein Eingabefenster wird angezeigt (Beispiel):

```
+--------------------------------------+
|   Scan value(s)       Or/And/Not     |
|   ----------------    ----------     |
|   _________________       OR_        |
|   _________________                  |
|   _________________                  |
|   _________________                  |
|                                      |
|   Absolute ..... X                   |
|   Immediate .... _                   |
|                                      |
+--------------------------------------+
```

In den Feldern können Sie die Suchkriterien eingeben, die für die Suche verwendet werden sollen:

Scan value(s)	**Suchwert(e):** In die vier leeren Felder können Sie bis zu vier Zeichenketten eingeben, nach denen gesucht werden soll. Die Suchfunktion findet die angegebenen Terme sowohl in Groß- als auch in Kleinbuchstaben.
Or/And/Not	**Verknüpfung:** Sie können eine Boolesche Suchanfrage durchführen, indem Sie einen der folgenden Operatoren eingeben:

`OR`	**ODER-Verknüpfung** Sucht nach einer oder mehreren Zeichenketten, die in den **Scan value(s)**-Eingabefeldern eingeben sind. Dies ist die Standardeinstellung.
`AND`	**UND-Verknüpfung** Sucht nach allen Zeichenketten, die in den **Scan value(s)**-Eingabefeldern eingeben sind.
`NOT`	**Negation** Sucht nach keiner der Zeichenketten, die in den **Scan value(s)**-Eingabefeldern eingeben sind.

Wenn Sie nur eines der **Scan value(s)**-Eingabefelder ausfüllen, wird der Operator nicht beachtet.

Absolute	**Absolut:** Wenn Sie dieses Feld markieren, wird die Zeichenkette auch dann gefunden, wenn sie Teil eines Wortes ist. Wenn Sie zum Beispiel nach dem Wert `meter` suchen, findet die Suchfunktion auch Wörter wie Parameter oder Millimeter.

	Wenn Sie die Feldmarkierung entfernen, wird die Suche auf passende, ganze Wörter eingeschränkt.
Immediate	**Unmittelbar:** Wenn Sie dieses Feld markieren, werden Meldungen einzeln, und zwar nacheinander, angezeigt. Andernfalls wird eine Liste mit Meldungen angezeigt, wenn die Suche abgeschlossen ist. Wenn Sie im Feld **Language codes** mehr als eine Sprache oder einen Stern (*) angeben, muss das Feld **Immediate** markiert sein.

3 Geben Sie Suchkriterien an, zum Beispiel:

```
+--------------------------------------+
!   Scan value(s)       Or/And/Not     !
!   ----------------    ----------     !
!   BUFFER_____________       AND      !
!   POOL_______________                !
!   ___________________                !
!   ___________________                !
!                                      !
!   Absolute ..... X                   !
!   Immediate .... _                   !
!                                      !
+--------------------------------------+
```

Im obigen Beispiel einer UND-Verknüpfung findet die Suchfunktion alle Kurzmeldungen, die sowohl das Wort `buffer` als auch das Wort `pool` enthalten.

4 Drücken Sie ENTER.

Alle Meldungen, auf die die angegebenen Suchkriterien zutreffen, werden im Bildschirm aufgelistet, zum Beispiel:

```
19:13:57              ***** NATURAL SYSERR UTILITY *****             2022-03-04
                           - Scan in Short Messages -

Number         Short Message (German)
-------------- ------------------------------------------------------------
NAT0777        Buffer pool voll.









End of scan reached.


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

Das Wort, in dem die gesuchte Zeichenkette gefunden wird, wird hervorgehoben dargestellt.

Wenn Sie in diesem Bildschirm PF10 drücken, werden die für den aktuellen Suchvorgang verwendeten Suchkriterien angezeigt.

Meldungen aus Liste auswählen

Mit dieser Funktion können Sie sich einen Bereich von Meldungen anzeigen lassen und aus der Auflistung einzelne Meldungen zur Weiterverarbeitung auswählen. Es können nur Kurzmeldungen angezeigt werden.

Um Meldungen auszuwählen:

1 Geben Sie in den Feldern im SYSERR-Menü folgende Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** SE (Select messages/Meldungen auswählen)
Message type	**Meldungstyp:** NS Natural-System-Kurzmeldungen US Benutzerdefinierte Kurzmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des für die Auswahl anzuzeigenden Meldungsbereichs.
Language codes	**Sprachcode:** Der Code der Sprache, in der die Meldungen angezeigt werden sollen. Wenn mehr als ein Code angegeben wird, werden nur die Kurzmeldungstexte der ersten Sprache angezeigt. Sie können einen Stern (*) eingeben, um sich die zu jeder Meldung vorhandenen Sprachen anzeigen zu lassen.

2 Drücken Sie ENTER.

Der Bildschirm **Select Messages** wird angezeigt, zum Beispiel:

```
19:16:36              ***** NATURAL SYSERR UTILITY *****              2022-03-04
                                - Select Messages -
                                                                    Languages
Se Number       Short Message (German)                              short     long
-- ------------ --------------------------------------------------- --------- ----
__ NAT1004      Mit FIND UNIQUE keinen oder mehr als 1 Datensatz 2          1
__ NAT1005      Mehr Datensaetze gefunden als in WITH LIMIT ange 2          1
__ NAT1006      Der Indexwert ist '0' oder groesser als das Maxi 2          1
__ NAT1007      Ganzzahl-Wert ausserhalb des gueltigen Wertebere 2          1
__ NAT1008      Variabler Index in 'RETAIN-List-Record-Buffer' v 2          1
__ NAT1009      Programm nach :1: Datenbank-Aufrufen abgebrochen 2          1
__ NAT1010      STORE/UPDATE/DELETE wegen UPDATE OFF-Kommando ab 2          1
__ NAT1011      Betaetigter Funktionstaste ist keine Funktion zu 2          1
__ NAT1012      Aenderungen auf der Systemdatei sind Ihnen nicht 2          1
__ NAT1013      Das eingegebene Terminalkommando existiert nicht 2          1
__ NAT1016      Programm wurde durch den Benutzer abgebrochen.   2          1
__ NAT1017      Update nicht erlaubt fuer reine Lese-Datenbank ( 2          1
__ NAT1018      Ungueltiger Datenbanktyp fuer Systemdatei.       2          1
__ NAT1019      Magnetkartenleser muss mit SET KEY sensitiv gema 2          1
__ NAT1024      < 1 > Total object size (:1: bytes) exceeds :2:             1
__ NAT1028      Eine rekursive Datenbankschleife darf nicht gest 2          1

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

3 In der Spalte **Se** können Sie eines der folgenden Zeilenkommandos eingeben:

Kommando	Funktion
DE	Löschung der Meldung.
DI	Wechsel vom Bildschirm **Select Messages** zum Bildschirm **Display Short Messages** (siehe Abschnitt *Meldungen anzeigen*. Darüber hinaus wird die mit diesem Zeilenkommando ausgewählte Meldung a den Anfang der Liste gestellt und die Anzahl der angezeigten Meldungen wird wie beim Zeilenkommando `.X` reduziert.
LA	Anzeige des Codes der Sprachen, für die Übersetzungen existieren.
MO	Änderung der Meldung.
PR	Ausgabe der Meldung auf einem Drucker.
SH	Anzeige der Kurzmeldung. Dieses Kommando ist nur dann verfügbar, wenn ein Stern (*) in das Feld **Language codes** im SYSERR-Menü eingegeben worden ist.
TR	Übersetzung der Meldung in eine andere Sprache.

Kommando	Funktion
.X	Definition eines kürzeren Meldungsbereichs zwecks Reduzierung der Anzahl der angezeigten Meldungen. Dabei wird die mit dem Kommando ausgewählte Meldung an den Anfang der Liste gestellt und alle Meldungen, die oberhalb dieser Meldung aufgelistet waren, werden aus der Anzeige entfernt. Der Meldungsbereich im SYSERR-Menü wird entsprechend zurückgesetzt und beginnt jetzt mit der hier im Bildschirm **Select Messages** ausgewählten Meldung.
.Y	Definition eines kürzeren Meldungsbereichs zwecks Reduzierung der Anzahl der angezeigten Meldungen: Alle Meldungen, die unterhalb der mit dem Kommando ausgewählten Meldung gelistet waren, werden aus der Anzeige entfernt. Der Meldungsbereich im SYSERR-Menü wird entsprechend zurückgesetzt und endet jetzt mit der hier im Bildschirm **Select Messages** ausgewählten Meldung.

4 Drücken Sie ENTER, um fortzufahren.

Meldungen in andere Sprachen übersetzen

Mit dieser Funktion können Sie Kurzmeldungen von einer Sprache in eine oder mehrere andere Sprachen übersetzen. Wie Sie Langmeldungen in andere Sprachen übersetzen können, wird im Abschnitt *Meldungen anlegen* beschrieben.

Um eine Kurzmeldung zu übersetzen:

1 Geben Sie in den Feldern des SYSERR-Menüs folgende Werte ein:

Feld	Eingabewert
Code	**Funktionscode:** TR (Translate messages/Meldungen übersetzen)
Message type	**Meldungstyp:** NS Natural-System-Kurzmeldungen US Benutzerdefinierte Kurzmeldungen
Library	**Library-Kennung:** Name (ID) einer existierenden Natural-Library.
Message number	**Meldungsnummer:** Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des für die Auswahl anzuzeigenden Meldungsbereichs.

Feld	Eingabewert
Language codes	**Sprachcodes:** Geben Sie maximal neun Sprachcodes ein. Die Sprachcodes sind alphanumerische Ein-Zeichen-Codes aus den Bereichen 1 - 9, A - Z und a - y.

2 Drücken Sie ENTER.

Der Bildschirm **Translate Short Message** wird angezeigt, zum Beispiel:

```
19:33:54             ***** NATURAL SYSERR UTILITY *****             2022-03-04
Lower Case ON             - Translate Short Message -

Number ...... XHTRI1004
Languages ... 12.....................................................

------------ ....+....1....+....2....+....3....+....4....+....5....+..
English      Short message English    (1)_____________________________________
German       Kurzmeldung Deutsch      (2)_____________________________________
Spanish      Mensaje breve Espanol    (4)_____________________________________
Italian      Messaggio breve Italiano (5)_____________________________________
             ________________________________________________________________
             ________________________________________________________________
             ________________________________________________________________
             ________________________________________________________________
             ________________________________________________________________
------------ ....+....1....+....2....+....3....+....4....+....5....+..

 1
 4      *** Long message not found ***
18

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

Im Feld **Languages** werden die Codes der Sprachen angezeigt, in denen die Meldung bereits existiert (im obigen Beispiel: 1, 2, 4 und 5).

Im Bereich unterhalb der Felder **Number** und **Languages** werden alle Sprachen aufgelistet, für die ein Sprachcode zuvor im Feld **Language codes** des SYSERR-Menüs eingegeben worden ist (im obigen Beispiel: 1, 2, 3, 4, 5). Übersetzungen in Englisch (1), Spanisch (4) und Italienisch (5) existieren bereits, während neue Übersetzungen in Deutsch (2) und Französisch (3) eingegeben werden können.

Zu Referenzzwecken werden (falls vorhanden) im unteren Bereich des Bildschirms drei Zeilen der Langmeldung angezeigt, die der Sprache entsprechen, die an erster Stelle im Bereich Sprachen/Kurzmeldungen (im obigen Beispiel: Englisch) aufgelistet ist. Die Zeilen 1, 4 und 18 werden standardmäßig angezeigt. Sie können eine beliebige andere Zeile der Langmeldung

anzeigen, indem Sie eine der drei Nummern (**1**, **4** oder **18**) mit einer anderen Zeilennummer überschreiben und ENTER drücken.

3 Geben Sie die Übersetzung in der Eingabezeile neben der angegebenen neuen Sprache ein.

4 Drücken Sie ENTER.

Um Übersetzungen von Kurzmeldungen zu ändern:

1 Drücken Sie im Bildschirm **Translate Short Message** die Taste PF10.

Das Fenster **Options** erscheint, zum Beispiel:

```
+------------------------ Options -------------------------+
!                                                          !
!  Modification of all fields allowed ..... N             !
!                                                          !
!  Currently recognized language codes ..... 123456789    !
!                                                          !
+----------------------------------------------------------+
```

2 Ersetzen Sie im oberen Feld den Wert `N` (Standardwert) durch `Y` und geben Sie im unteren Feld den oder die erforderlichen Sprachcodes ein. Sie können für die Übersetzung bis zu neun neue Sprachen angeben.

Um eine Übersetzung in eine leere Eingabezeile zu kopieren:

1 Geben Sie im Bildschirm **Translate Short Message** das Kommando `.C` an den ersten zwei Stellen einer leeren Zeile ein.

2 Platzieren Sie den Cursor auf eine beliebige Stelle einer Kurzmeldung, die schon in einer anderen Sprache existiert. (Sie können nur Text kopieren, der im Anzeigen-Modus erscheint.)

3 Drücken Sie ENTER.

62 Parameter

- Meldungstyp 618
- Library 618
- Meldungsnummer 618
- Sprachcodes 619

In diesem Kapitel werden die Parameter beschrieben, die zu einer im SYSERR-Menü vorhandenen Funktion angegeben werden können. Einschränkungen, die für die Verwendung von Parametern bei einer bestimmten Funktion gelten, sind im Kapitel *Funktionen* beschrieben.

Meldungstyp

Gibt den Typ der zu verarbeitenden Meldung an. Die vorhandenen Meldungstypen sind in der folgenden Tabelle aufgeführt:

Typ	Erklärung
NS	Natural-System-Kurzmeldungen
NL	Natural-System-Langmeldungen
US	Benutzerdefinierte Kurzmeldungen
UL	Benutzerdefinierte Langmeldungen

Library

Gibt die Library an, für die Meldungen angelegt oder gepflegt werden sollen. Die Angabe einer Library ist nicht erforderlich, wenn auf Natural-Systemmeldungen zugegriffen wird (Meldungstypen NS und NL). Bei diesen Meldungstypen werden Angaben im Feld **Library** ignoriert.

Meldungsnummer

Gibt die erste und die letzte Nummer eines Meldungsbereichs an. Die höchste Meldungsnummer für eine Library und Sprache ist 9999. Die Meldungsnummer ist nicht zulässig. Um nur eine Meldungsnummer anzugeben, können Sie entweder die Nummer der Meldung im linken **Message number**-Feld eingeben und den Wert im rechten Feld löschen oder die Nummer in beiden Feldern eingeben.

Sprachcodes

Gibt maximal 9 bis 60 verfügbare Sprachcodes an. Die Sprachcodes sind alphanumerische Ein-Zeichen-Codes aus den Bereichen 1 - 9, A - Z und a - y. Um verfügbare Sprachcodes anzuzeigen oder auszuwählen, können Sie ein Fragezeichen (?) an der ersten Stelle im Feld **Language codes** eingeben und ENTER drücken. Weitere Informationen siehe Systemvariable *LANGUAGE in der *Systemvariablen*-Dokumentation.

63 Direktkommandos

Im SYSERR-Menü können Sie folgende Kommandos ausführen, indem Sie sie in die Kommandozeile eingeben:

Kommando	Erklärung
LAYOUT	Angabe von gültigen Meldungsbereichen zur Kategorisierung von Meldungen. Überlappungen von Bereichen sind möglich. Eine neue Meldung kann nur angelegt werden, wenn ihre Nummer in dem im Layout angegebenen Bereich liegt.
NEXT	Suche nach der nächsten freien Meldungsnummer innerhalb des angegebenen Meldungsnummernbereichs. Frei bedeutet, dass die Meldungsnummer verfügbar und noch nicht in einer Sprache einer Meldungsdatei zugewiesen worden ist.
NEXTTAB	Wie NEXT, liefert jedoch eine Liste mit Meldungsnummern, aus der Sie eine Nummer auswählen können.
RESTART	Reinitialisierung von SYSERR (und seiner Standardwerte). Die Utility wird dabei nicht verlassen.
SAMPLE	Aufruf des Fensters **Edit SAMPLE message**. Dort können Sie Mustermeldungen anlegen und ändern, die Sie als Vorlage zum Anlegen von neuen Kurzmeldungen verwenden können. **Um eine Mustermeldung anzulegen:** ■ Geben Sie im Editierbereich des Fensters **Edit SAMPLE message** den erforderlichen Meldungstext ein oder ändern Sie den vorhandenen Text. Wenn Sie die Zeichenkette 0000 (in Kombination mit Text oder ohne), wird die Zeichenkette 0000 durch die Nummer der neuen Meldung ersetzt, wenn die Meldung kopiert wird. Siehe auch **Schritt 4** unter *Meldungen anlegen* im Kapitel *Funktionen*. ■ Geben Sie im Feld **Read or Write sample** ein W ein, um Ihre Eingaben zu speichern. ■ Geben Sie im Feld **Library** den Namen (ID) der Library ein, für die die Mustermeldung verwendet werden soll. Wenn Sie das Feld leer lassen, gilt die Mustermeldung für Natural-Systemmeldungen. ■ Drücken Sie PF3, um das Fenster **Edit SAMPLE message** zu verlassen. Die können eine Mustermeldung pro Sprache und Library definieren.

Kommando	Erklärung
SECURITY	Aufruf eines Security-Fensters, in dem Sie ein Passwort und einen Chiffrierschlüssel für den Zugriff auf Adabas- und VSAM-Dateien, die durch Security geschützt sind.
SHIFT	Wenn SHIFT aktiviert ist, wird der Text einer Kurzmeldung automatisch an den linken Rand verschoben, wenn das Anlegen einer neuen Meldung oder das Ändern einer Meldung bestätigt wird.
TRACE	Es wird die Anzahl der Datenbankzugriffe gezählt. Wenn die angegebene Anzahl an Meldungen erreicht ist, wird ein Fenster angezeigt. Die Standardanzahl ist 900. Wenn der Wert auf 0 gesetzt ist, wird die Nachverfolgungsfunktion ausgeschaltet. Die Kommandos TRACE ON und TRACE OFF können direkt in der Kommandozeile eingegeben werden. TRACE ON setzt den Zugriffszähler auf 900. TRACE OFF setzt den Zugriffszähler auf 0.
USEREXIT	Aufruf des Programms USEREXIT in der Natural-System-Library SYSERR.

64 Umwandlung in Großbuchstaben - ERRUPPER

Natural-Systemmeldungen werden in Kleinbuchstaben geliefert. Falls Ihr Terminal nicht in der Lage ist, Kleinbuchstaben korrekt anzuzeigen, können Sie die Meldungen von Kleinbuchstaben in Großbuchstaben konvertieren, indem Sie das Programm ERRUPPER in der Natural-System-Library SYSERR ausführen.

Nachdem Sie die Meldungen in Großbuchstaben konvertiert haben, können Sie sie jedoch nicht zurück in Kleinbuchstaben konvertieren. Um die Meldungen in Kleinbuchstaben wiederherzustellen, haben Sie zwei Möglichkeiten:

- Sie können die Meldungen mit dem Programm ERRLODUS oder mit dem Object Handler erneut laden.
- Sie können die Meldungen in Kleinbuchstaben in einen freien Sprachcode entladen, indem Sie vor der Umwandlung das Programm ERRULDUS oder den Object Handler benutzen, so dass immer eine Sicherheitskopie existiert.

Ausführliche Informationen siehe ***Meldungen entladen - ERRULDUS*** und ***Meldungen laden - ERRLODUS*** und die *Object Handler*-Dokumentation.

65 Zeichen ersetzen - ERRCHAR

Falls Ihr Terminal nicht in der Lage ist, bestimmte Buchstaben korrekt anzuzeigen, können Sie nach diesen Zeichen suchen und sie durch neue Zeichen Ihrer Wahl ersetzen, indem Sie das Programm ERRCHAR in der Natural-System-Library SYSERR ausführen.

Sie können damit jedoch nur Zeichen in Natural-System-Kurzmeldungen ersetzen. Bei der Benutzung von ERRCHAR suchen Sie nach einem bestimmten Zeichen und ersetzen den Hexadezimalcode, der dieses Zeichen darstellt, durch einen anderen Hexadezimalcode.

Nach dem Ausführen des Programms ERRCHAR wird das Menü **ERRCHAR** angezeigt. Es bietet folgende Funktionen:

- Suche nach einem gegebenen Zeichen
- Suchen und Ersetzen von Zeichen
- Anzeige einer Meldung im Hexadezimalformat
- EBCDIC-Zeichentabelle für Ihr Terminal
- Übersetzen unter Verwendung des Zeichensatzes ERRCSET

Im Menü **ERRCHAR** sind folgende Eingabefelder vorhanden:

Feld	Erklärung
Message Number	Der Bereich der Meldungen, die in den Such- oder Such-/Ersetzungsvorgang einbezogen werden sollen.
Language Code	Der Sprachcode der Natural-System-Kurzmeldungen, die in den Such- oder Such-/Ersetzungsvorgang einbezogen werden sollen.
Scan Value	Der Hexadezimalwert, nach dem gesucht werden soll.
Replace Value	Der Hexadezimalwert, der alle gefundenen Suchwerte ersetzen soll. Benutzen Sie die Funktion **EBCDIC character table for your terminal**, um zu bestimmen, welche Zeichen Ihr Terminal darstellen kann.

66 Meldungen in verschiedenen Libraries verwalten

- Meldungen entladen - ERRULDUS 628
- Meldungen laden - ERRLODUS 630

Sie können Meldungen zwischen verschiedenen Libraries transferieren und Meldungen in verschiedenen Libraries umbenennen, finden, auflisten oder löschen. Je nach betroffener Natural-Systemumgebung oder -umgebungen können Sie die Meldungen entladen und in Arbeitsdateien (Workfiles) laden oder sie direkt von einer Library in eine andere kopieren oder verschieben.

Um Meldungen umzubenennen, zu finden, aufzulisten oder zu löschen:

- Für Libraries, die sich in der aktuellen Natural-Umgebung auf einer z/OS-Plattform befinden, können Sie die SYSMAIN-Utility benutzen (siehe entsprechende Dokumentation).

  Oder:

  Für Libraries, die in verschiedenen Natural-Umgebungen auf verschiedenen z/OS-, UNIX oder Windows-Plattformen enthalten sind, können Sie den Object Handler benutzen (siehe entsprechende Dokumentation).

Um Meldungen von einer oder in eine Arbeitsdatei zu entladen oder zu laden:

- Für Libraries, die in einer lokalen z/OS-Umgebung enthalten sind, können Sie die Programme ERRULDUS und ERRLODUS benutzen (Beschreibung siehe folgende Abschnitte).

  Oder:

  Für Libraries, die in verschiedenen Natural-Umgebungen auf verschiedenen z/OS-, UNIX- oder Windows-Plattformen enthalten sind, können Sie den Object Handler benutzen (siehe entsprechende Dokumentation).

Meldungen entladen - ERRULDUS

Das Programm ERRULDUS dient dazu, alle von Natural unterstützten Meldungstypen zu entladen, d.h., benutzerdefinierte Kurz- und Langmeldungen, die mit der SYSERR Utility erstellt wurden, und Natural-Systemmeldungen. Die Meldungen werden aus der Systemdatei FNAT oder FUSER gelesen und in die Arbeitsdatei (Workfile) 2 geschrieben.

Um ERRULDUS aufzurufen:

- Geben Sie in der Kommandozeile des SYSERR-Menüs `ERRULDUS` ein.

  Das Menü **ERRULDUS** wird angezeigt (Beispiel):

```
16:11:13              ***** NATURAL SYSERR UTILITY *****               08-09-18
                   - ERRULDUS (Unload Texts to Work File 2) -

                     Code  Function
                     ----  ---------------------------------
                      US   User supplied short error texts
                      UL   User supplied long  error texts
                      U    User supplied short AND long texts
                      NS   NATURAL short error texts
                      NL   NATURAL long  error texts
                      H    NATURAL help texts
                      .    Exit
                     ----  ---------------------------------
   Code ............ __
   Source Library .. SYSERR__     Source Language Code .. 01
   Target Library .. SYSERR__     Target Language Code .. 01
   Error Number .... 1___ - 9999
   Replace ......... N
   Report .......... ON_

Please enter valid code.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Exec  Help        Exit                                                  Canc
```

Das **ERRULDUS**-Menü enthält folgende Eingabefelder:

Feld	Erklärung
Code	Der zu entladende Meldungstyp. Gültige Werte: `US` Benutzerdefinierte Kurzmeldungen `UL` Benutzerdefinierte Langmeldungen `U` Benutzerdefinierte Kurz- und Langmeldungen `NS` Natural-System-Kurzmeldungen `NL` Natural-System-Langmeldungen `H` Natural-Hilfetexte `.` Verarbeitung beenden
Source Library	Name der Library, aus der Meldungen entladen werden sollen. Der Name kann unter Verwendung eines Sterns (*) verkürzt werden. Wenn Sie nur einen Stern (*) eingeben, werden Meldungen aus allen Libraries entladen. Bei den Meldungstypen `NS` und `NL` wird die Angabe im Feld **Source Library** nicht beachtet.
Source Language Code	Sprachcode der Meldungen, die entladen werden sollen.
Target Library	Name der Library, in die Meldungen geladen werden sollen. Bei den Meldungstypen `NS` und `NL` wird die Angabe im Feld **Target Library** nicht beachtet.
Target Language Code	Code der Sprache, in die Meldungen geladen werden sollen.

Feld	Erklärung
Error Number	Zwei Nummern mit bis zu vier Stellen. Diese entsprechen der ersten und der letzten Nummer des zu entladenden Meldungsbereichs. Wird keine erste Nummer angegeben, wird `1` als erste Nummer angenommen.
Replace	Mögliche Werte: `Y` (Yes): Die Ziel-Library soll überschrieben werden. `N` (No): Die Ziel-Library soll nicht überschrieben werden (Standardeinstellung).
Report	Mögliche Werte: `ON`: Nach Beendigung des Herunterladens wird eine Liste aller Meldungen, die heruntergeladen wurden, angezeigt. `OFF`: Die Liste wird nicht angezeigt.

ERRULDUS liest Eingaben solange, bis im Feld **Code** ein Punkt (.) festgestellt wird (= Verarbeitung beenden). Nach dem Herunterladen wird eine Statistik-Liste angezeigt.

Meldungen laden - ERRLODUS

Das Programm ERRLODUS dient dazu, Meldungen zu laden, die zuvor mit dem Programm ERRULDUS entladen worden sind.

Die Meldungen werden von der Arbeitsdatei (Workfile) 2 gelesen und in die Systemdatei FNAT oder FUSER geschrieben. Wenn beim Entladen die Option **Replace** mit `Y` markiert wurde, überschreibt ERRLODUS existierende Meldungen in der Systemdatei.

Wenn Sie ERRLODUS als `ON` angeben, wird eine Liste aller Meldungen, die geladen (hinzugefügt oder ersetzt) wurden, angezeigt, wenn der Ladevorgang abgeschlossen ist.

Anmerkung: Unter Natural Security ist, unabhängig von der Einstellung der **Utilities**-Option im Security-Profil der Library, die Online-Benutzung des Programms ERRLODUS nur für Benutzer des Typs Administrator gestattet. Siehe auch Abschnitt Utilities unter *Library Maintenance* in der *Natural Security*-Dokumentation.

XVI SYSEXT Utility

67 SYSEXT Utility

- Einleitung zu SYSEXT 634
- SYSEXT Utility aufrufen und beenden 635
- SYSEXT Utility benutzen 638
- Interface-Versionen 643
- Reservierte Schlüsselwörter 644
- Eine Natural API verwenden 644

Mit dem Dienstprogramm („Utility“) SYSEXT können Sie Natural-Anwendungsprogrammierschnittstellen (Application Programming Interfaces/APIs), die in der aktuellen System-Library SYSEXT enthalten sind, finden und testen.

Eine Natural API ist ein Natural-Subprogramm (katalogisiertes Objekt), das dazu benutzt wird, auf Daten zuzugreifen, eventuell Daten zu ändern oder Dienste auszuführen, die für Natural-Statements nicht zugänglich sind. Natural APIs beziehen sich auf Natural, eine Unterkomponente oder ein Unterprodukt.

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.

Verwandte Themen:

- *Application Programming Interfaces - Natural Security*-Dokumentation
- *Application Programming Interfaces - Natural SAF Security*-Dokumentation
- APIs, die von Natural Add-on-Produkten zur Verfügung gestellt werden, siehe Dienstprogramm („Utility“) **SYSAPI**.

Einleitung zu SYSEXT

Die SYSEXT Utility bietet zu jeder Natural API eine Funktionsbeschreibung, ein Beispiel-Programm und API-spezifische Schlüsselwörter.

Hauptmerkmale der SYSEXT Utility sind:

- **API-Objekte**
  - API-Subprogramm
  - Beispiel-Programm zum Testen
  - Text-Objekt mit Funktionsbeschreibung und Interface-Versionen
- **Funktionen**
  - APIS auflisten mit Beschreibung, Kategorie und Produktcode
  - Mehrfachauswahl zur API-Abfrage
  - Suche nach Schlüsselwort und Kategorie
  - Umschalten zwischen Anzeigemodi
  - Objekt-Operationen

Bei Natural APIs verfügbare Objekte

Im folgenden Abschnitt werden die Natural-Objekttypen aufgeführt, die zu jeder Natural API typischerweise vorhanden sind. Zusätzliche Objekte, die zu einer bestimmten API verfügbar sein können, werden an dieser Stelle nicht behandelt.

Alle Objekte, die einen Bezug zur API haben, sind in der Library SYSEXT in der Systemdatei FNAT enthalten.

In der folgenden Tabelle steht *nnnn* für die vierstellige Nummer, durch die die API sowie das entsprechende Beispiel-Programm und Text-Objekt und (falls vorhanden) zugehörige Copycodes identifizierbar sind.

Objektname	Erklärung
USR*nnnn*N	**API-Subprogramm** Ein Subprogramm (katalogisiertes Objekt), das die vorgesehene API-Funktion ausführt.
USR*nnnn*P	**Beispiel-Programm** Ein Programm (Source-Objekt), das benutzt werden kann, um die Wirkung der API zu testen. Das Beispiel-Programm ruft das entsprechende Subprogramm USR*nnnn*N auf.
USR*nnnn*T	**Text-Objekt** Ein Text-Objekt, das eine Kurzbeschreibung und eine ausführliche Beschreibung, Informationen zur Verwendung, Schlüsselwörter und **Interface-Versionen** auflistet: . Zum Anzeigen eines Text-Objekts können Sie das Zeilenkommando `T` benutzen, siehe ***Zeilenkommandos***.
USR*nnnnX*	**Copycode** Bei einigen APIs stehen Copycodes zur Verfügung, die Funktionen bieten, die in Beziehung zur API stehen. Die Copycodes haben den Namen `USR`*nnnnX*, wobei *X* ein Identifizierungszeichen ist (z.B. "Z", "Y" usw.).

SYSEXT Utility aufrufen und beenden

Die SYSEXT Utility können Sie mit folgendem Systemkommando und den im Syntax-Diagramm gezeigten Parametern aufrufen:

```
SYSEXT [ ALL       ] [ FIRST  ] [ DESCENDING ]
       [ CURRENT   ] [ SECOND ] [ ASCENDING  ]
```

Die Parameter, die Sie mit dem Systemkommando angeben können, werden in der folgenden Tabelle beschrieben:

Parameter	Beschreibung
ALL	Auflistung aller APIs (Standardeinstellung).
CURRENT	Auflistung der aktuellen APIs, d.h. alle singulären, d.h. nur einmal vorhanden, APIs und die aktuelle Version von APIs mit Interface-Versionen (mit Schlüsselwort +CURRENT-VERSION). Siehe *Interface-Versionen*
FIRST	Anzeige des ersten Menüs mit Produktcode-Einzelheiten (Standardeinstellung). Beispiel siehe **unten**.
SECOND	Anzeige des zweiten Menüs mit Kategorie-Einzelheiten. Beispiel siehe **unten**.
ASCENDING	Anzeige der APIs in aufsteigender Reihenfolge (Standardeinstellung).
DESCENDING	Anzeige der APIs in absteigender Reihenfolge.

Um die SYSEXT Utility aufzurufen:

- Geben Sie folgendes Systemkommando ein:

```
SYSEXT
```

Es erscheint ein Menü mit einer Liste aller vorhandenen Natural APIs gemäß den Angaben im ersten Menü (Beispiel):

```
07:43:19            ***** NATURAL SYSEXT UTILITY *****           2010-09-15
User SAG                            - Menu -                  Library SYSEXT

Cmd Interface Description                                                Prod
--- USR*____  ____________________________________________________________ *__
 _  USR0010N  Get SYSPROF information                                     NAT
 _  USR0011N  Get information on logical file                            NAT
 _  USR0020N  Read any error message from FNAT or FUSER                  NAT
 _  USR0040N  Get type of last error                                     NAT
 _  USR0050N  Get SYSPROD information                                    NAT
 _  USR0060N  Copy LFILE definition from FNAT to FUSER                   NAT
 _  USR0070P  Define editor default profile 'SYSTEM'                     NAT
 _  USR0080N  Get or set type and name of editor contents                NAT
 _  USR0100N  Maintain record length of VSAM datasets                    NVS
 _  USR0120N  Read Natural short error message                           NAT
 _  USR0210N  Save, catalog or stow Natural object                       NAT
 _  USR0220N  Read Natural long error message                            NAT
 _  USR0320N  Read user short error message from FNAT or FUSER           NAT

 Category .. _____________________________  Keyword .. ___________________

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

Sie können die SYSEXT Utility mit beliebigen Parametern (s.o.) aufrufen, z.B. um das zweite Menü direkt beim Start der Utility anzuzeigen:

```
SYSEXT SECOND
```

Folgendes Menü wird angezeigt (Beispiel):

```
07:43:19              ***** NATURAL SYSEXT UTILITY *****            2010-09-15
User SAG                            - Menu -                   Library SYSEXT

Cmd Interface Description                          Category
--- USR*____  ____________________________________ *_____________________________
 _  USR0010N  Get SYSPROF information              SYSTEM COMMANDS
 _  USR0011N  Get information on logical file      SYSTEM FILES
 _  USR0020N  Read any error message from FNAT o   ERROR MESSAGES
 _  USR0040N  Get type of last error               ERROR HANDLING
 _  USR0050N  Get SYSPROD information              SYSTEM COMMANDS
 _  USR0060N  Copy LFILE definition from FNAT to   SYSTEM FILES
 _  USR0070P  Define editor default profile 'SYS   EDITOR
 _  USR0080N  Get or set type and name of editor   EDITOR
 _  USR0100N  Maintain record length of VSAM dat   VSAM
 _  USR0120N  Read Natural short error message     ERROR MESSAGES
 _  USR0210N  Save, catalog or stow Natural obje   NATURAL OBJECTS
 _  USR0220N  Read Natural long error message      ERROR MESSAGES
 _  USR0320N  Read user short error message from   ERROR MESSAGES

 Category .. ____________________________  Keyword .. ____________________

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Reset Exit  Desc  Curr  --    -     +     ++    <           Canc   ↩
```

Alternativ können Sie die SYSEXT Utility ohne Parameter aufrufen und dann PF-Tasten benutzen, um das angezeigte Menü wunschgemäß anzupassen. Siehe ***PF-Tasten***.

Um die SYSEXT Utility zu beenden:

- Drücken Sie im Menü der SYSEXT Utility PF3 oder PF12.

  Oder:

  Geben Sie in der Kommandozeile einen Punkt (.) oder `EXIT` ein.

SYSEXT Utility benutzen

Folgende Themen werden behandelt:

- Bestandteile des SYSEXT Utility-Menüs
- PF-Tasten
- Zeilenkommandos

- Utility-Kommandos

Bestandteile des SYSEXT Utility-Menüs

Das SYSEXT Utility-Menü enthält eine Liste der APIs. Zu jeder API wird Folgendes angezeigt: Name der API (**Interface**), eine Beschreibung (**Description**) und, je nach gewähltem Menü, eine Klassifizierung nach Produktcode (**Prod**) oder Kategorie (**Category**).

Mit Ausnahme der Spalte **Cmd** befindet sich über jeder Spalte ein Auswahlfeld, in das Sie Auswahlkriterien eingeben können.

Das Eingabefeld **Category** im unteren Bereich des Menüs können Sie außerdem zur Suche nach verfügbaren Kategorien benutzen.

Das Eingabefeld **Keywords** können Sie als Suchfeld benutzen, um verfügbare Schlüsselwörter oder APIs abzurufen.

Darüber hinaus können Sie Auswahlkriterien bei mehreren Auswahlfeldern eingeben. Zum Beispiel können Sie in einem einzelnen Schritt APIs auswählen, die mit `USR4` beginnen und als weitere Auswahlkriterien das Schlüsselwort `PF-KEY` und die Kategorie `NATURAL ENVIRONMENT` aufweisen.

Die folgende Tabelle enthält eine ausführliche Beschreibung aller Bestandteile des SYSEXT Utility-Menüs:

Bestandteil	Erklärung	Verwendung
Cmd	Eingabefeld für ein Zeilenkommando, mit dem Sie ein Text-Objekt oder ein Beispiel-Programm zur Ausführung markieren können. Siehe *Zeilenkommandos*.	Geben Sie ein Zeilenkommando ein, z.B.: `K` Alle bei der angegebenen API relevanten Schlüsselwörter auflisten.
Interface	Der Name des API-Subprogramms. APIs mit Interface-Versionen werden hervorgehoben angezeigt.	Geben Sie einen Stern (*) oder ein mit einem Stern eingegrenztes Präfix ein, z.B.: `USR4*` Alle APIs mit Präfix `USR4` auflisten.
Description	Kurzbeschreibung des Verwendungszwecks der API.	Geben Sie eine Zeichenkette ein, z.B.: `default` Alle APIs mit einer Beschreibung auflisten, die die Zeichenkette `default` oder `Default` enthält. `*` Alle APIs mit einer Beschreibung auflisten, die die Zeichenkette * enthält.

Bestandteil	Erklärung	Verwendung
Prod	Der Produktcode von Natural (NAT) oder eines Natural Add-on-Produkts, das von der API betroffen ist. Verfügbare Produktcodes: NAT = Natural, NCI = Natural for CICS, NDB = Natural for Db2, NVS = Natural for VSAM, NDV = Natural Development Server, PRD = Predict, RPC = Natural RPC (Remote Procedure Call). Mit Ausnahme von NAT werden alle Produktcodes hervorgehoben angezeigt. Die Anzeige erfolgt nur im ersten Menü.	Geben Sie einen Namen oder ein mit einem Stern (*) eingegrenztes Präfix ein, z.B.: N* Alle APIs mit einem Produktcode auflisten, der mit N beginnt.
Category (Auswahlfeld)	Die Kategorie, durch die die API gemäß ihrem Funktionsbereich und Verwendungszweck klassifiziert wird. Eine API kann nur zu einer Kategorie gehören. Die Anzeige erfolgt nur im zweiten Menü, und zwar über der Liste der APIs.	Geben Sie einen Namen oder ein mit einem Stern (*) eingegrenztes Präfix ein, z.B.: NATURAL OBJECTS Alle APIs mit der Kategorie NATURAL OBJECTS auflisten.
Category (Suchfeld)	Alle Kategorien auflisten. Die Anzeige erfolgt über der Liste der APIs.	Siehe *Kategorie-Suche*.
Keyword	Alle Schlüsselwörter auflisten oder ein Schlüsselwort als Auswahlkriterium eingeben.	Siehe *Schlüsselwort-Suche* oder *Schlüsselwort-Auswahl*.
Command	Kommandozeile zur Eingabe von Utility-Kommandos.	Geben Sie ein Utility-Kommando ein. Siehe *Utility-Kommandos*.

Kategorie-Suche

- Geben Sie einen Stern (*) oder ein Präfix im Suchfeld **Category** ein. Wahlweise können Sie einen Stern (*) zur Eingrenzung angeben, z.B.:

```
N*
```

Als Ergebnis der Suche erscheint ein Menü (Beispiel):

```
        Search for Categories

 Mark Category
 ---- N*_____________________________
   _  NATURAL ENVIRONMENT
   _  NATURAL OBJECTS
```

Um eine bestimmte Kategorie als Auswahlkriterium auszuwählen, können Sie in dem entsprechenden Eingabefeld in der Spalte **Mark** ein beliebiges Zeichen eingeben. Als Ergebnis wird eine Liste aller APIs, die dieses Auswahlkriterium erfüllen, angezeigt.

Schlüsselwort-Suche

- Geben Sie im Feld **Keyword** einen Stern (*) oder ein mit einem Stern (*) eingegrenztes Präfix ein, z.B.:

```
PF*
```

Als Ergebnis der Suche erscheint ein Menü mit einem separaten Fenster, in dem Schlüsselwörter angezeigt werden, die mit `PF` beginnen (Beispiel):

```
        Search for Keywords

Mark Keyword
---- PF*_________________
  _  PF-KEY
  _  PF-KEY LINE
```

Um ein bestimmtes Schlüsselwort als Auswahlkriterium auszuwählen, können Sie in dem entsprechenden Eingabefeld in der Spalte **Mark** ein beliebiges Zeichen eingeben. Als Ergebnis wird eine Liste aller APIs, die dieses Auswahlkriterium erfüllen, angezeigt.

Schlüsselwort-Auswahl

- Geben Sie ein vollständiges Schlüsselwort ein, z.B.:

```
PF-KEY
```

Als Ergebnis der Suche erscheint eine Liste aller APIs mit Schlüsselwort `PF-KEY`.

Anmerkung: Ein Stern (*), der nicht am Ende eines Kriteriums angegeben wird, wird nicht als Platzhalter, sondern „buchstäblich“ interpretiert. Beispiel: Die Eingabe von `*LANGUAGE` ergibt eine Liste mit APIs mit dem Schlüsselwort `*LANGUAGE`.

PF-Tasten

Sie können die folgenden, mit Funktionen belegten PF-Tasten benutzen:

PF-Taste	Name	Funktion
PF1	**Help**	Anzeige der kontextabhängigen Hilfe. Zu jedem Eingabefeld ist ein spezifischer Hilfetext vorhanden. In anderem Kontext, zum Beispiel in der Kommandozeile, wird ein allgemeiner Hilfetext angezeigt.
PF2	**Reset**	Die Inhalte aller Auswahlfelder werden gelöscht und die Liste der APIs wird zurückgesetzt.
PF3	**Exit**	Beenden der SYSEXT Utility oder Verlassen des aktuellen Menüs oder Fensters.
PF4	**Asc/Desc**	Umschalten zwischen aufsteigender (**Asc**) und absteigender (**Desc**) Reihenfolge der aufgelisteten APIs.
PF5	**All/Curr**	Umschalten zwischen Menüs, die alle APIs (**All**) anzeigen, und Menüs, die nur aktuelle APIs (**Curr**) anzeigen. Siehe *Interface-Versionen*.
PF6	**--**	Zum Anfang der Liste blättern.
PF7	**-**	Eine Seite nach oben blättern.
PF8	**+**	Eine Seite nach unten blättern.
PF9	**++**	Zum Ende der Liste blättern.
PF10	**<**	Wechsel zum ersten Menü.
PF11	**>**	Wechsel zum zweiten Menü.
PF12	**Canc**	SYSEXT Utility oder das aktuelle Menü oder Fenster verlassen.

Zeilenkommandos

Zeilenkommandos werden verwendet, um Objektoperationen durchzuführen. Sie können ein Zeilenkommando in der Spalte **Cmd** neben der gewünschten API eingeben. Für eine Liste der gültigen Zeilenkommandos können Sie ein Fragezeichen (`?`) eingeben oder PF1 drücken.

Folgende Zeilenkommandos stehen zur Verfügung:

Zeilenkommando	Funktion
`K`	Schlüsselwörter auflisten, die für die angegebene API relevant sind.
`T`	Text-Objekt USR*nnnn*T mit der Beschreibung der entsprechenden API anzeigen. Die Beschreibung umfasst Zweck, Funktion und Aufrufkonventionen der API sowie relevante Schlüsselwörter und die Kategorie der API.
`L`	Beispiel-Programm USR*nnnn*P anzeigen (`LIST`).
`E`	Beispiel-Programm USR*nnnn*P bearbeiten (`EDIT`). **Anmerkung:** Dieses Editierkommando steht nicht zur Verfügung, wenn Programm-Editor, Datenbereich-Editor (Data Area Editor) und Masken-Editor (Map Editor) in Ihrer Umgebung

Zeilenkommando	Funktion
	abgeschaltet (deaktiviert) sind. Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung* in der *Editoren*-Dokumentation.
R	Beispiel-Programm USR*nnnn*P kompilieren und ausführen (RUN).
X	Beispiel-Programm USR*nnnn*P ausführen (EXECUTE).
.	SYSEXT Utility verlassen.

Utility-Kommandos

In der Kommandozeile können Sie die folgenden Utility-Kommandos eingeben:

- **EXIT**
  SYSEXT Utility verlassen.
- **REFRESH**
  Aktualisieren der API-Informationen unter Verwendung von Daten aus den Objekten, die in der aktuellen Library SYSEXT enthalten sind. Dieses Kommando wird nur dann benötigt, wenn eine API-Beschreibung oder ein Schlüsselwort geändert oder ein Text-Objekt hinzugefügt oder entfernt worden ist.

  Nach erfolgter Aktualisierung erscheint eine Meldung zur Bestätigung, dass die Text-Module EXT-XML1 und EXT-XML2 in der Library SYSEXT generiert worden sind.

  Anmerkung: Sie dürfen die Source-Objekte EXT-XML1 und EXT-XML2 *nicht* ändern. Diese werden zum Konfigurieren der SYSEXT Utility benötigt und sind nur für unseren internen Gebrauch bestimmt.

Interface-Versionen

Interface-Versionen können als eine Sammlung von APIs gesehen werden mit (fast) der gleichen Funktionalität, aber mit unterschiedlich erweiterten Parameter Spezifikationen. Sie decken somit den Entwicklungszyklus ab, der explizit wegen der Kompatibilität (zwischen späteren und früheren Versionen) aufbewahrt wird.

Wenn Interface-Versionen zu einer API existieren, werden diese im entsprechenden Text-Objekt USR*nnnn*T angezeigt. Interface-Versionen sind in einer Liste gemäß der Version angeordnet, zu der sie gehören. Das ganz rechts angeordnete Listenelement gehört zu der aktuellen Version. Der Status wird durch das Schlüsselwort +CURRENT-VERSION ausgedrückt. Alle übrigen Listenelemente gehören zu früheren Versionen und sind durch das Schlüsselwort +PREVIOUS-VERSION gekennzeichnet. APIs ohne Interface-Versionen werden als singulär, d.h. nur einmal vorhanden, bezeichnet.

APIs mit Interface-Versionen werden im Menü hervorgehoben dargestellt.

Die Liste aller aktuellen APIs (siehe PF5) besteht aus allen nur einmal vorhandenen APIs und der aktuellen Version von APIs mit Interface-Versionen (mit Schlüsselwort `+CURRENT-VERSION`).

Reservierte Schlüsselwörter

Reservierte Schlüsselwörter beziehen sich auf Meta-Informationen zu APIs, z.B. der Natural-Version, in der eine API hinzugefügt worden ist. Reservierte Schlüsselwörter beginnen immer mit einem Plus-Zeichen (+):

Reserviertes Schlüsselwort	Beschreibung
+CURRENT-VERSION	Die aktuelle Version einer API mit Interface-Versionen (siehe *Interface-Versionen*).
+PREVIOUS-VERSION	Eine frühere Version einer API mit Interface-Versionen (siehe *Interface-Versionen*).
+NEW-*PROD-version*	Eine API, die zu einem bestimmten Produkt in einer bestimmten Version hinzugefügt worden ist. Beispiel: `+NEW-NAT-9.2.4` bezieht sich auf eine API, die zum Produkt Natural in Version `9.2.4` hinzugefügt worden ist.
+MOD-*PROD-version*	Eine API, die zu einem bestimmten Produkt gehört und in einer bestimmten Version geändert worden ist. Beispiel: +MOD-NAT-9.2.3 bezieht sich auf eine API, die zum Produkt Natural gehört und in Version 9.2.3 geändert worden ist.

Eine Natural API verwenden

Wenn Sie eine der Natural APIs, die in der System-Library SYSEXT enthalten sind, verwenden möchten, können Sie einen der folgenden Schritte ausführen:

- Definieren Sie die System-Library SYSEXT in der Systemdatei FNAT als eine Steplib Library für die Benutzer-Library, die die Natural-Objekte enthält, welche diese API verwenden. Dadurch sind bei einem Upgrade Ihrer Natural-Version keine API-spezifischen Maßnahmen erforderlich.
- Kopieren Sie die benötigte API in die System-Library SYSEXT in der Systemdatei FNAT. Sie müssen so bei einem Natural-Upgrade nur eine einzige Library auf APIs überprüfen.
- Kopieren Sie die benötigte API in die System-Library SYSTEM in der Systemdatei FNAT (nicht zu empfehlen).
- Kopieren Sie die benötigte API in die Benutzer-Library (oder in eine ihrer Steplibs) in der Systemdatei FUSER, die die Natural-Objekte enthält, welche diese API verwenden (nicht zu empfehlen).

Sie können eine API nur in der Natural-Version verwenden, in der sie ausgeliefert wird. Es wird dringend geraten, die APIs nur in der Systemdatei FNAT zu speichern. Dadurch wird sichergestellt, dass immer die richtige Version ausgeführt wird.

Um eine Natural API zu verwenden:

1 Benutzen Sie im rufenden Programm das Statement `DEFINE DATA`, um die Parameter anzugeben, die im Text-Objekt `USR`*`nnnn`*`T` der betreffenden API aufgelistet sind. In dem Beispiel-Programm `USR`*`nnnn`*`P` werden die Parameter in einem `DEFINE DATA LOCAL`-Statement definiert. Alternativ können Sie die Parameter außerhalb des rufenden Programms in einer separaten LDA (Local Data Area) oder PDA (Parameter Data Area) angeben, indem Sie ein `DEFINE DATA LOCAL USING`-Statement benutzen, das diesen Datenbereich referenziert.

2 Geben Sie folgendes Statement ein:

```
CALLNAT 'USRnnnnN' parameters
```

Weitere Informationen siehe *CALLNAT*-Statement in der *Statements*-Dokumentation.

Anmerkung: Eine nicht standardmäßige Verwendung ist immer im jeweiligen Text-Objekt `USR`*`nnnn`*`T` dokumentiert.

Wenn Sie einen in der System-Library SYSEXT enthaltenen Copycode verwenden möchten, können Sie folgenden Schritt ausführen:

Kopieren Sie den benötigten Copycode in die Benutzer-Library in der Systemdatei FUSER, die die Natural-Objekte enthält, welche diese API verwenden

Um einen Copycode zu verwenden:

1 Benutzen Sie im rufenden Programm das Statement `INCLUDE`, um die Parameter anzugeben, die im Text-Objekt `USR`*`nnnn`*`T` der betreffenden API oder im Copycode selbst aufgelistet sind. In dem Beispiel-Programm `USR`*`nnnn`*`P` werden die Parameter in einem `DEFINE DATA LOCAL`-Statement definiert. Alternativ können Sie die Parameter außerhalb des rufenden Programms in einer separaten LDA (Local Data Area) oder PDA (Parameter Data Area) angeben, indem Sie ein `DEFINE DATA LOCAL USING`-Statement benutzen, das diesen Datenbereich referenziert.

2 Bei einigen Copycodes sind zusätzliche Datendefinitionen erforderlich. Diese sind im Text-Objekt `USR`*`nnnn`*`T` der API und im Copycode selbst beschrieben.

3 Geben Sie folgendes Statement ein:

```
INCLUDE USRnnnnX 'parameter'...
```

Weitere Informationen siehe *INCLUDE*-Statement in der *Statements*-Dokumentation.

Anmerkung: Eine nicht standardmäßige Verwendung ist immer im jeweiligen Text-Objekt USR*nnnn*T dokumentiert.

XVII SYSEXV Utility

68 SYSEXV Utility

- SYSEXV starten und beenden 650
- Untermenüs und Beispielprogramme aufrufen und beenden 652
- Suchbegriff eingeben 652
- SYSEXV-PF-Tasten 653
- SYSEXV-Zeilenkommandos 653
- Kategorie auswählen 654
- Version auswählen 655
- Member auswählen 657

Das Dienstprogramm („Utility“) SYSEXV stellt Beispiel-Programme zur Verfügung, die die Verwendung von Natural-Funktionalität und Merkmalen veranschaulichen, die mit der aktuellen oder einer vorangegangenen Version von Natural eingeführt wurden.

In diesem Kapitel wird die zeichenbasierte Benutzeroberfläche von SYSEXV beschrieben. Informationen zur Rich-GUI-Oberfläche von SYSEXV finden Sie in *NaturalONE* > *Using NaturalONE* > *Using Natural Tools and Utilities* > *Rich GUI Interface of SYSEXV*.

Alle Beispiel-Programme sind als Quellcode-Objekte verfügbar. Eine ausführliche Beschreibung der Funktionalität erhalten Sie, wenn Sie die Programme ausführen.

SYSEXV starten und beenden

Um SYSEXV aufzurufen:

- Geben Sie das folgende Systemkommando ein:

  ```
  SYSEXV
  ```

  Das SYSEXV-Hauptmenü wird angezeigt:

```
16:08:04            ***** NATURAL SYSEXV UTILITY *****             2026-08-12
User LHU          - Show Features of Natural Versions -       Library SYSEXV


The SYSEXV utility provides example programs that demonstrate the use of     ↩

Natural features introduced in the current or previous versions of Natural.  ↩




 _  Category selection                                                       ↩

 _  Version selection                                                        ↩

 _  Member selection                                                         ↩

 _  Help                                                                     ↩

 _  Exit                                                                     ↩









Search: ____________________________                                         ↩


Command ===>                                                                ↩

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

Die Beispielprogramme können nach Kategorie oder Version gruppiert oder alphabetisch nach Member-Namen aufgelistet werden.

Um SYSEXV zu beenden:

- Markieren Sie im SYSEXV-Hauptmenü den Menüeintrag **Exit**.

  Oder:

  Drücken Sie PF3 oder geben Sie `Exit` oder einen Punkt (`.`) in die Kommandozeile ein.

  Oder:

  In einem beliebigen Bildschirm: Drücken Sie PF12 oder geben Sie `Canc` (Abbruch) in die Kommandozeile ein.

Untermenüs und Beispielprogramme aufrufen und beenden

Um ein Untermenü oder ein Beispielprogramm aufzurufen:

- Markieren Sie die entsprechende Zeile mit einem beliebigen Zeichen. Verwenden Sie jedoch kein Zeichen mit einer Sonderbedeutung, wie sie im Abschnitt *SYSEXV-Zeilenkommandos* aufgeführt sind.

  Wenn Sie mehrere Zeilen markieren, werden diese nacheinander verarbeitet.

  Wenn keine Zeile markiert ist und sich der Cursor auf einer Eingabezeile befindet, wird der entsprechende Eintrag verarbeitet.

Um ein Untermenü zu verlassen:

- Drücken Sie PF3 oder geben Sie `Exit` oder einen Punkt (`.`) in die Kommandozeile ein.

  Oder:

  Geben Sie einen Punkt (`.`) als Zeilenkommando ein.

  Der vorherige Bildschirm wird angezeigt. Wenn auf diesem Bildschirm mehrere Einträge markiert waren, wird der nächste Eintrag verarbeitet.

Um ein Beispielprogramm zu beenden:

- Drücken Sie PF3.

  Sie verlassen den aktuellen Bildschirm und kehren zum vorherigen Bildschirm zurück. Wenn auf diesem Bildschirm mehrere Einträge markiert waren, wird der nächste Eintrag verarbeitet.

Suchbegriff eingeben

Sie können bei jeder Auswahl (Hauptmenü, Kategorie, Version, Member-Auswahl) im Feld **Search** einen Suchbegriff angeben, sodass nur die Beispielprogramme angezeigt werden, deren Member-Name, Version, Kategorie oder Beschreibung diesen Begriff enthält.

SYSEXV-PF-Tasten

Die folgenden PF-Tasten stehen zur Verfügung:

PF-Taste	Name	Funktion
PF1	`Help`	SYSEXV-Online-Hilfe anzeigen.
PF2	`Menu`	Alle Auswahlfeldinhalte löschen und zum Hauptmenü wechseln.
PF3	`Exit`	Aktuelles Menü verlassen oder Beispielprogramm beenden.
PF4	`Categ`	Kategoerieauswahl anzeigen.
PF5	`Vers`	Versionsauswahl anzeigen.
PF6	`--`	Zum Anfang der Liste springen.
PF7	`-`	Eine Seite nach obern blättern.
PF8	`+`	Eine Seite nach unten blättern.
PF9	`++`	Zum Ende der Liste springen.
PF10	`Membr`	Member-Auswahl anzeigen.
PF12	`Canc`	SYSEXV-Utility-Benutzung abbrechen.

Alternativ können Sie den Namen einer Taste in der Kommandozeile eingeben und Enter drücken, damit die entsprechende Funktion ausgeführt wird.

SYSEXV-Zeilenkommandos

Die folgenden Zeilenkommandos stehen zur Verfügung:

Zeilenkommando	Funktion
Beliebig (außer den folgenden)	Untermenü aufrufen oder ein Beispielprogramm aufrufen.
`I`	Zu dem markierten Beispiel werden die folgenden Informationen angezeigt: ■ Name des Beispiel-Members, ■ Version, mit der die entsprechende Funktionalität eingeführt wurde ■ kurze Beschreibung der Funktionalität, ■ Kategorien, zu denen das Beispiel-Member gehört.
`?`	SYSEXV-Online-Hilfe-Bildschirme anzeigen.
`.`	Aktuelles Menü verlassen.

Kategorie auswählen

Jedes Beispiel ist einer oder mehreren Kategorien zugeordnet. Standardmäßig werden auf dem Bildschirm **Category Selection** alle Kategorien angezeigt.

```
16:09:14             ***** NATURAL SYSEXV UTILITY *****             2026-08-12
User LHU                   - Category Selection -              Library SYSEXV

 _  Array
 _  Database
 _  Date and Time
 _  Debug and Test
 _  Edit Mask
 _  Error
 _  Function and Subprogram
 _  Input and Output
 _  Internet and XML
 _  Server
 _  Session Parameter
 _  Statement Enhancement
 _  String Handling
 _  System Function
 _  System Variable

Search: ____________________________

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

Markieren Sie eine Kategorie, um die zu dieser Kategorie gehörenden Beispielprogramme aufzulisten, beispielsweise die Kategorie **Statement Enhancement** (Statement-Verbesserung):

```
16:10:30             ***** NATURAL SYSEXV UTILITY *****              2026-08-12
User LHU            - Category: Statement Enhancement -         Library SYSEXV

 _  Arithmetic expressions in FOR statement                          V82FOR
 _  Arrays for replacements(n:m) in EXAMINE statement               V82EXAM2
 _  Comparators in READ and HISTOGRAM statement                      V4-COMP
 _  Direction and position clause in EXAMINE statement              V82EXAM1
 _  Enhanced COMPRESS with spacer element                           V925CMPR
 _  Enhanced MOVE EDITED with wildcard character                    V925MVED
 _  ESCAPE MODULE for inline subroutines                            V4-ESC
 _  Extended precision of results for arithmetic operations         V82EXPRE
 _  MARK option with CV variable in REINPUT statement              V82REINP
 _  Option REPOSITION in READ and ESCAPE TOP REPOSITION             V4-REPO
 _  STARTING FROM and REMAINDER POSITION in SEPARATE statement     V823SEPA
 _  SUBSTRING clause in DECIDE ON statement                        V82DECON
 _  SUBSTRING clause in MOVE ALL statement                         V82MOVAL
 _  Unicode (step 7) MOVE ENCODED                                  V42UENC
 _  Unicode (step 8) MOVE NORMALIZED                               V42UNORM

Search: ____________________________

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

Markieren Sie eine Zeile, um das entsprechende Beispielprogramm auszuführen.

Version auswählen

Jedes Beispielprogramm gehört zu der Version, in der die Funktionalität eingeführt wurde. Standardmäßig werden auf dem Bildschirm **Version Selection** alle Versionen angezeigt, wobei die neueste Version an erster Stelle steht.

```
16:11:34              ***** NATURAL SYSEXV UTILITY *****              2026-08-12
User LHU                     - Version Selection -                Library SYSEXV

 _  Natural Version 9.2.5
 _  Natural Version 9.2.4
 _  Natural Version 9.2.1
 _  Natural Version 9.1.1
 _  Natural Version 8.2.4
 _  Natural Version 8.2.3
 _  Natural Version 8.2
 _  Natural Version 4.2.6
 _  Natural Version 4.2
 _  Natural Version 4.1
 _  Natural Version 3.1
 _  Natural Version 2.3




Search: ________________________________

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Categ Vers  --    -     +     ++    Membr       Canc   ↩
```

Markieren Sie eine Version, um die zu dieser Version gehörenden Beispielprogramme anzuzeigen, zum Beispiel **Natural Version 9.2.5**:

```
16:12:43             ***** NATURAL SYSEXV UTILITY *****              2026-08-12
User LHU                  - Natural Version 9.2.5 -             Library SYSEXV

 _  Enhanced COMPRESS with spacer element                          V925CMPR
 _  Enhanced hexadecimal constants                                 V925HXNW
 _  Enhanced MOVE EDITED with wildcard character                   V925MVED
 _  System variable *CPU-TIMEX                                     V925CPUX









Search: ____________________________

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Categ Vers  --    -     +     ++    Membr       Canc  ↩
```

Markieren Sie eine Zeile, um das entsprechende Beispielprogramm auszuführen.

Member auswählen

Der Bildschirm **Member Selection** zeigt eine Liste der Beispielprogramme, sortiert nach den Namen der Member in alphabetischer Reihenfolge.

```
14:58:57              ***** NATURAL SYSEXV UTILITY *****              2026-08-12
User LHU                      - Member Selection -                Library SYSEXV

 _  V2-CMPMI   Option (PM=I) in COMPRESS and MOVE statement
 _  V2-CMPR1   COMPRESS option FULL
 _  V2-CMPR2   COMPRESS option NUMERIC
 _  V2-CMPR3   COMPRESS option ALL
 _  V2-CMPR4   COMPRESS option SUBSTRING
 _  V2-ERR     Error information with API USR2006N
 _  V2-MASK    Character '/' in MASK option
 _  V2-MOVE    MOVE option RIGHT JUSTIFIED
 _  V2-RDESC   READ option DESCENDING
 _  V2-RISN    READ option STARTING WITH ISN
 _  V2-RVAR    READ option VARIABLE SEQUENCE
 _  V2-VAR     Environment and date related system variables
 _  V2-WRITE   Time variables in WRITE or DISPLAY statement
 _  V2-YDF     Year 2000, MOVE with field parameter DF
 _  V2-YDFOU   Year 2000, INPUT with session parameter DFOUT

Search: ________________________________

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

Markieren Sie eine Zeile, um das entsprechende Beispielprogramm auszuführen.

XVIII SYSMAIN Utility

Mit dem Dienstprogramm („Utility“) SYSMAIN können Sie Funktionen zur Pflege von Objekten ausführen, z.B. Kopieren, Verschieben, Ersetzen und Löschen.

Allgemeine Informationen zu SYSMAIN

SYSMAIN aufrufen und beenden

Menüfunktionen und Kommandos benutzen

Programmierobjekte bearbeiten

Debug-Umgebungen bearbeiten

Fehlermeldungen bearbeiten

Profile bearbeiten

Rules bearbeiten

DDMs bearbeiten

Predict Sets bearbeiten

Schlüsselwörter und Variablen in Direktkommandos

An SYSMAIN absetzbare spezielle Kommandos

Statusmeldungen und Fehlerbenachrichtigungen bearbeiten

Besondere Hinweise für Administratoren

69 Allgemeine Informationen zu SYSMAIN

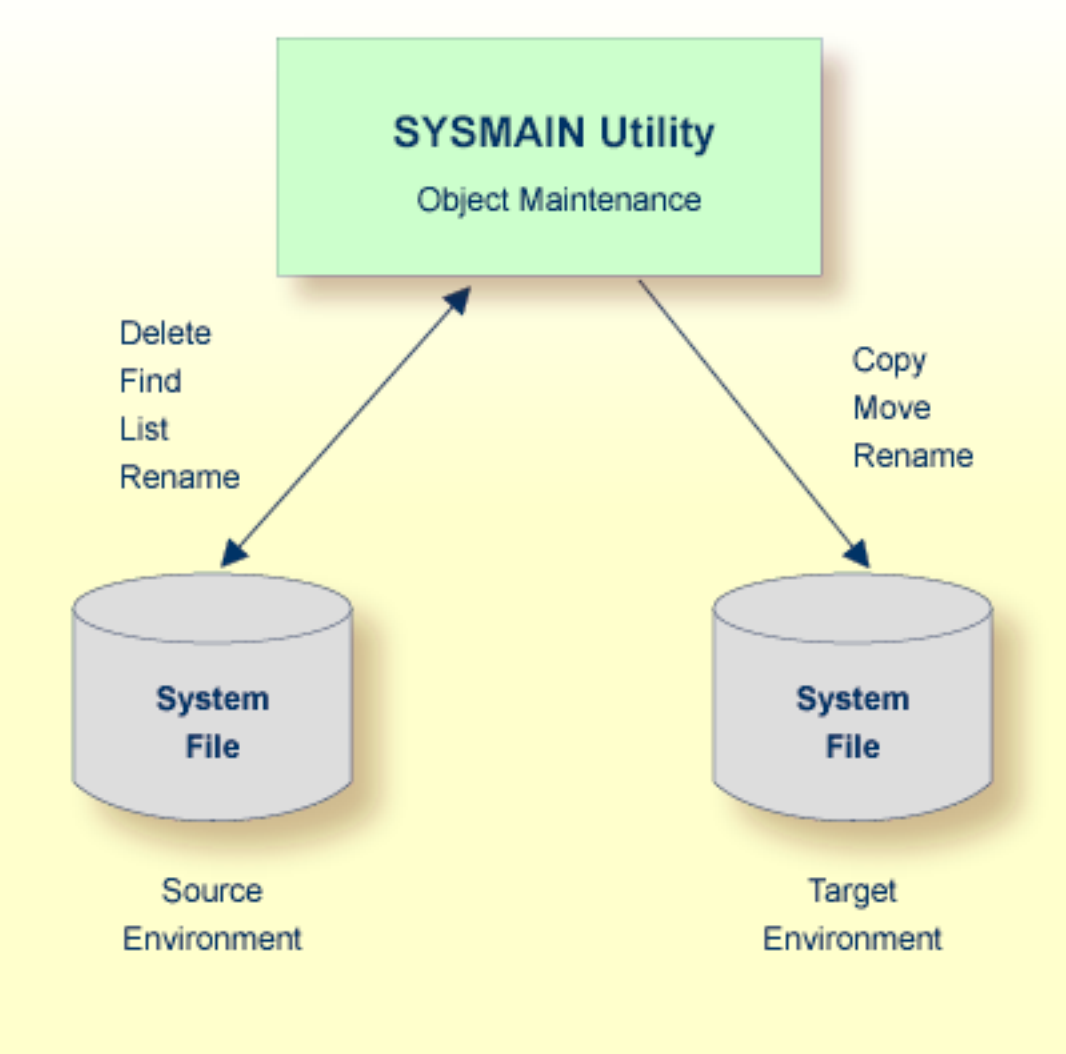
- Grundsätzliches zu den SYSMAIN-Funktionen 662
- Objekttypen und Speicherort 663
- Übersicht über die Funktionen 663

Das Dienstprogramm SYSMAIN dient zur Pflege von Natural-Objekten innerhalb einer Natural-Umgebung oder über verschiedene Umgebungen hinweg.

In diesem Kapitel finden Sie grundlegende Informationen zu den Funktionen der Objektpflege und zu den Natural-Objekttypen, die mit SYSMAIN bearbeitet werden können.

Grundsätzliches zu den SYSMAIN-Funktionen

Das folgende Diagramm zeigt eine Übersicht über die SYSMAIN-Funktionen zur Objektpflege:



Das Dienstprogramm SYSMAIN kopiert (**Copy**) oder verschiebt (**Move**) Natural-Objekte aus einer Quellumgebung (Source Environment) in eine Zielumgebung (Target Environment) und führt Objektoperationen wie das Löschen (**Delete**), Suchen (**Find**) und Auflisten (**List**) in einer Quellum-

gebung durch. Die Funktion zum Umbenennen (**Rename**) kann sowohl in einer Quell- als auch in einer Zielumgebung ausgeführt werden. Im folgenden Abschnitt finden Sie eine Übersicht über die Funktionen und die Natural-Objekte, für die sie gelten. Die Funktionen des Dienstprogramms SYSMAIN sind online und im Batch-Modus verfügbar.

Quell- oder Zielumgebung ist eine FNAT-, FUSER- oder FDIC-Systemdatei, die in einer Adabas-Datenbank oder einem VSAM-Dateisystem enthalten ist. Natural-Objekte in der FNAT- oder FUSER-Systemdatei können in Libraries enthalten sein, wie im folgenden Abschnitt beschrieben.

Objekttypen und Speicherort

Die Natural-Objekttypen, die mit SYSMAIN verwaltet werden können, sind in der folgenden Tabelle aufgeführt. Der Speicherort eines Natural-Objekts hängt vom Objekttyp ab, wie in der folgenden Tabelle angegeben:

Objekte/Subfiles	Beschreibung	Speicherort
Programmierobjekte	Alle Natural-Objekttypen, die in einer Natural-Bibliothek gespeichert sind, außer Debug-Umgebungen und Fehlermeldungen.	Libraries in FUSER und FNAT
Debug-Umgebungen	Debug-Umgebungen für Online-Programmtests.	Library in FUSER
Fehlermeldungen	Kurze und erweiterte (lange) Natural-Systemfehlermeldungen und benutzerdefinierte Fehlermeldungen.	Libraries in FUSER und FNAT
Profile	Editorprofile, Map-Profile, Geräteprofile und Parameterprofile.	FNAT
Rules	Automatische und freie Rules.	FDIC
DDMs	Datendefinitionsmodule.	FDIC
Predict Sets	Eingespeicherte Sets, die mit dem Systemkommando `LIST XREF` oder Predict erstellt wurden. Nur verfügbar, wenn Predict installiert ist.	FDIC

Übersicht über die Funktionen

Die von SYSMAIN bereitgestellten Funktionen sind in der folgenden Tabelle aufgeführt. In der Tabelle ist auch angegeben, welche Funktion für jeden Natural-Objekttyp gültig ist. Einzelheiten zu den einzelnen Funktionen finden Sie unter ***Beschreibung der Funktionen*** in *Menüfunktionen und Kommandos benutzen*.

Funktion	Programmier-objekt	Debug-Umgebung	Fehler-meldung	Profil	Rule	DDM	Predict Set
Copy Objekt von einer Systemdatei in eine andere kopieren.	x	x	x	x	x	x	x
Delete Objekt aus einer Systemdatei löschen.	x	x	x	x	x	x	x
Find Objekt in einer Systemdatei finden.	x		x				
List Einen Bereich von Objekten in einer Systemdatei anzeigen.	x	x	x	x	x	x	x
Move Objekt von einer Systemdatei in eine andere übertragen.	x	x	x	x	x	x	x
Rename Einem Objekt einen neuen Namen oder eine neue Nummer (je nach Typ) geben und es (optional) in eine neue Systemdatei übertragen.	x	x	x	x	x		x

70 SYSMAIN aufrufen und beenden

- SYSMAIN online oder im Batch aufrufen 666
- SYSMAIN mittels Anwendungsprogrammierschnittstelle aufrufen 667
- SYSMAIN beenden 668

Dieser Abschnitt beschreibt, wie Sie das Dienstprogramm SYSMAIN online, im Batch oder über eine Anwendungsprogrammierschnittstelle aufrufen und beenden.

SYSMAIN online oder im Batch aufrufen

Die folgenden Anweisungen beschreiben die Methoden zum Aufrufen des SYSMAIN-Dienstprogramms über ein Direktkommando (Online oder Batch) oder über Menüfunktionen.

Um SYSMAIN online aufzurufen:

- Geben Sie in einer beliebigen Library das folgende Natural-Systemkommando ein:

```
SYSMAIN
```

Oder:

Rufen Sie aus dem Natural-Hauptmenü (**Natural Main Menu**) das Menü **Maintenance and Transfer Utilities** auf und wählen Sie **Transfer Objects to Other Libraries**.

Es erscheint ein SYSMAIN-Hauptmenü (**Main Menu**) ähnlich dem folgenden Beispiel:

```
 16:33:22            ***** NATURAL SYSMAIN UTILITY *****           2022-09-16
 User SAG                        - Main Menu -

             Code  Object                              Code  Function

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

Object Code .. A                      Function Code .. _




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

Die aktuelle Einstellung der Systemvariablen `*LIBRARY-ID` wird an SYSMAIN übergeben und als Standard Source Library für die Verarbeitung von Programmierobjekten und Debug-Umgebungen verwendet.

Um SYSMAIN im Batch aufzurufen:

- Benutzen Sie das folgende Direktkommando:

```
SYSMAIN
```

gefolgt von einer oder mehreren Kommandozeichenfolgen. Siehe auch *Direktkommandos absetzen*.

SYSMAIN mittels Anwendungsprogrammierschnittstelle aufrufen

MAINUSER ist eine Anwendungsprogrammierschnittstelle (API), mit der SYSMAIN-Funktionen direkt aus einem beliebigen vom Benutzer geschriebenen Objekt (z.B. aus einer Subroutine, einem Programm oder Subprogramm) ausgeführt werden können, ohne die normalen Schritte des Aufrufs von SYSMAIN zu durchlaufen.

Nach Abschluss der Verarbeitung der SYSMAIN-Funktionen wird das Dienstprogramm beendet und die Kontrolle an das Programm, Subprogramm oder die Subroutine zurückgegeben, von dem/der aus die Anforderung gestellt wurde.

MAINUSER wird als katalogisiertes Objekt vom Typ Subprogramm in der System Library SYSMAIN bereitgestellt. MAINUSER kann sowohl im Online- als auch im Batch-Modus verwendet werden.

Anmerkung: MAINUSER darf sich *nicht* in einer User Library befinden. Kopieren Sie ihn daher in die Library SYSTEM in der Systemdatei FNAT oder FUSER oder in eine beliebige Library mit Präfix SYS, die als Steplib für die Anwendung dient.

MAINUSER aufrufen

MAINUSER wird mit dem Statement CALLNAT und den entsprechenden Parametern aufgerufen (siehe *Statements*-Dokumentation). MAINUSER darf nicht aus der Library SYSMAIN heraus aufgerufen werden.

Um MAINUSER aufzurufen:

- Setzen Sie ein `CALLNAT`-Statement ab, das die folgenden Syntaxelemente enthält:

```
CALLNAT 'MAINUSER' command error message library
```

wobei die variablen Werte die folgenden Parameter bezeichnen:

Parameter	Natural-Datenformat/Länge	Erläuterung
command	A250	Die Direktkommandozeichenfolge, die von SYSMAIN ausgeführt werden soll.
error	N4	Der Rückgabecode, der von SYSMAIN am Ende der Verarbeitung ausgegeben wird, um eine normale Beendigung der Verarbeitung oder einen Fehler anzuzeigen.
message	A72	Die Meldung, die dem online angezeigten Fehler entspricht.
library	A8	Die Library, die SYSMAIN enthält. Wenn nicht angegeben, ist die Vorgabe SYSMAIN.

Ein Beispiel für eine aufrufbare Routine ist das Programm `MAINCALL` in der Library SYSMAIN.

SYSMAIN beenden

Um SYSMAIN zu beenden:

- Geben Sie in der Kommandozeile eines beliebigen SYSMAIN-Bildschirms eines der folgenden Direktkommandos ein:

```
.
```

(ein Punkt)

oder

```
END
```

oder

```
QUIT
```

Oder:

PF3 (Exit) drücken, ggf. wiederholt.

Oder:

Verwenden Sie im Batch-Modus eines der folgenden Direktkommandos:

```
END
```

oder

```
QUIT
```

Siehe auch ***Direktkommandos absetzen*** im Abschnitt *Menüfunktionen und Kommandos benutzen*.

⚠ **Wichtig:** Beenden Sie das Dienstprogramm SYSMAIN nicht mit dem Terminalkommando `%%`, da die Umgebung möglicherweise nicht korrekt zurückgesetzt wird.

71 Menüfunktionen und Kommandos benutzen

- Menüfunktionen ausführen 672
- Kommandos ausführen 675
- Beschreibung der Funktionen 677
- Funktionsverarbeitung und Reporting 685
- SYSMAIN-Online-Hilfe 690

Eine SYSMAIN-Funktion kann entweder im Menü- oder im Kommandomodus ausgeführt werden.

In diesem Kapitel wird beschrieben, wie Sie ein SYSMAIN-Menü oder ein Direktkommando benutzen und welche Optionen bei der Ausführung einer SYSMAIN-Funktion zur Verfügung stehen.

Menüfunktionen ausführen

Im Menümodus führen Sie eine SYSMAIN-Funktion aus, indem Sie Codes für den zu bearbeitenden Objekttyp und die auszuführende Funktion eingeben. Alternativ zur Eingabe eines Funktionscodes können Sie auch eine entsprechend belegte PF-Taste drücken.

Um eine SYSMAIN-Menüfunktion auszuführen:

- Geben Sie im Hauptmenü (**Main Menu**) des SYSMAIN-Dienstprogramms im Feld **Object Code** den aus einem Buchstaben bestehenden Code ein, der den gewünschten Objekten entspricht (im Beispiel unten: `A` für Programmierobjekte), und im Feld **Function Code** den aus einem Buchstaben bestehenden Code, der der gewünschten Funktion entspricht (im Beispiel unten: `C` für Kopieren). Alternativ zur Eingabe des Funktionscodes können Sie auch die PF-Taste drücken, die der Funktion entspricht (hier: `PF4`); siehe auch die Liste der PF-Tasten.

```
 12:41:50            ***** NATURAL SYSMAIN UTILITY *****           2013-04-05
 User SAG                        - Main Menu -

             Code  Object                              Code  Function

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

Object Code .. A                       Function Code .. C




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

Es wird dann das entsprechende, objekttypspezifische Menü angezeigt, in diesem Beispiel das Menü **Copy Programming Objects**:

```
17:28:52            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
User SAG                 - Copy Programming Objects -


                    Code  Function

                      A   Copy All/Individual Objects
                      C   Copy only Cataloged Objects
                      S   Copy only Saved Objects
                      W   Copy only Stowed Objects
                      ?   Help
                      .   Exit

          Code ...... A           Sel. List ... Y
  Object  Name ...... *________   Type ........ ________________
                                  Set Number .. __     XREF .. N
  Source  Library ... OLDLIB__    Database .... 10___  File .. 50___
  Target  Library ... NEWLIB__    Database .... 10___  File .. 60___
  Options Replace ... N           Criteria .... N


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

Die in einem objekttypspezifischen Menü angebotenen Felder hängen von der ausgeführten SYSMAIN-Funktion ab. Felder, die für eine bestimmte Funktion nicht zutreffen, werden im zugehörigen Bildschirm nicht angezeigt.

Die Felder werden zur Angabe von Objektauswahlkriterien und Bearbeitungsoptionen verwendet. Erläuterungen zu den Feldern finden Sie in den objekttypspezifischen Kapiteln dieser Dokumentation.

PF-Tasten benutzen

Mit den PF-Tasten können Sie Funktionen des SYSMAIN-Menüs ausführen. PF-Tasten, die innerhalb eines Menüs nicht gültig sind, werden für dieses Menü nicht angezeigt. Die PF-Tasten sind in der nachstehenden Tabelle zusammengefasst:

PF-Taste	Name	Funktion
PF1	Help	**Hilfe** Anzeige der Online-Hilfe in Abhängigkeit von der aktuellen Cursorposition. Befindet sich der Cursor im Feld **Object Code** oder **Function Code**, wird die allgemeine SYSMAIN-Hilfe angezeigt. Befindet sich der Cursor in einem anderen Feld, wird die feldspezifische Hilfe angezeigt. Siehe auch ***SYSMAIN Online-Hilfe***.
PF2	Menu	**Menü** Anzeige des Hauptmenüs (**Main Menu**) des SYSMAIN-Dienstprogramms.
PF3	Exit	**Beenden** Rückkehr zum vorherigen Bildschirm. Wenn Sie PF3 im Hauptmenü (**Main Menu**) des SYSMAIN-Dienstprogramms drücken, wird SYSMAIN beendet.
PF4	Copy	**Kopieren** Ausführen der Kopierfunktion (**Copy**) für das angegebene Objekt.
PF5	Del	**Entfernen** Ausführen der Löschfunktion (**Delete**) für das angegebene Objekt.
PF6	Find	**Suchen** Gilt nur bei Programmierobjekten und Fehlermeldungen. Ausführen der Suchfunktion (**Find**) für das angegebene Objekt.
PF7	List	**Liste** Ausführen der Funktion **List** für das angegebene Objekt.
PF8	Move	**Verschieben** Ausführen der Funktion **Move** für das angegebene Objekt.
PF9	Ren	**Umbenennen** Gilt nicht bei DDMs. Ausführen der Funktion **Rename** für das angegebene Objekt aus.
PF10	Fsec	Ruft den Security-Bildschirm zum Angeben der Adabas-Sicherheitsinformationen für die FSEC-Systemdatei auf, wenn Natural Security installiert ist.
PF11	Fdic	Gilt nicht bei Debug-Umgebungen, Fehlermeldungen und Profilen. Gilt nur bei Programmierobjekten, Rules und DDMs. Ruft den Security-Bildschirm zum Angeben der Adabas-Sicherheitsinformationen für die FDIC-Systemdatei auf. Wenn Predict installiert ist, können Sie auch einen Predict Set-Benutzer angeben, wenn Sie die Funktionen **Copy**, **Delete**, **Move** oder **Rename** (Kopieren, Löschen, Verschieben oder Umbenennen) bei Programmierobjekten ausführen.

PF-Taste	Name	Funktion
PF12	Fnat	Gilt nicht bei Debug-Umgebungen, Fehlermeldungen, Predict Sets und Profilen. Ruft den Security-Bildschirm auf, um die Systemdatei-Informationen der Systemdateien FNAT und/oder FUSER anzugeben.

Kommandos ausführen

Dieser Abschnitt enthält Anleitungen zum Ausführen einer SYSMAIN-Funktion mit Hilfe eines Direktkommandos oder eines Systemkommandos innerhalb des SYSMAIN-Dienstprogramms.

Ein Direktkommando wird verwendet, um eine SYSMAIN-Funktion entweder im Online- oder im Batch-Modus auszuführen.

Im Batch-Modus wird automatisch ein Report erstellt, der den Status der verarbeiteten Objekte anzeigt. Dieser Report kann auch im Online-Modus angezeigt werden. Siehe ***Online Report-Modus***.

- Direktkommandos absetzen
- SYSMAIN-Kommandozeile benutzen

Direktkommandos absetzen

Direktkommandos bestehen aus einer Reihe von Schlüsselwörtern, die Parameter darstellen. Ein Punkt (.) kennzeichnet das Ende eines Kommandos. Wenn dieses Zeichen irgendwo in einer Kommandozeichenfolge erkannt wird, werden alle nachfolgenden Daten ignoriert.

Die für Direktkommandos geltende Syntax ist in den objekttypspezifischen Abschnitten der SYSMAIN Utility-Dokumentation beschrieben. Die entsprechenden Schlüsselwörter und Variablen werden in ***Schlüsselwörter und Variablen in Direktkommandos*** erläutert. Beispiele für Direktkommandos finden Sie in den objekttypspezifischen Abschnitten, in der SYSMAIN-Online-Hilfe und auf einem Auswahl-Bildschirm (siehe den Beispielbildschirm in ***Selektive Verarbeitung***).

Um ein Direktkommando online abzusetzen:

- Geben Sie an einer beliebigen Natural-Eingabeaufforderung Folgendes ein:

```
SYSMAIN
```

  gefolgt von einer Kommandozeichenfolge, wobei jedes Schlüsselwort durch ein Leerzeichen anstelle des Trennzeichens abgegrenzt werden kann.

  Oder:

Geben Sie in der Kommandozeile eines beliebigen SYSMAIN-Menüs eine Direktkommandozeichenfolge ein, bei der jedes Schlüsselwort durch ein Leerzeichen anstelle des Begrenzungszeichens getrennt werden kann.

Oder:

Rufen Sie aus einem Natural-Objekt heraus das Subprogramm `MAINUSER` mit der Direktkommandozeichenfolge als Parameter auf, wobei jedes Schlüsselwort durch ein Leerzeichen anstelle des Begrenzungszeichens getrennt werden kann.

Um ein Direktkommando im Batch auszuführen:

1 Benutzen Sie das Kommando `SYSMAIN` und geben Sie eine Kommandozeichenfolge auf eine der beiden folgenden Arten an:

1. Die Kommandozeichenfolge folgt dem `SYSMAIN`-Kommando in derselben Eingabezeile; jedes Schlüsselwort in der Kommandozeichenfolge kann durch ein Leerzeichen anstelle des Begrenzungszeichens abgegrenzt werden.
2. Die Kommandozeichenfolge folgt dem `SYSMAIN`-Kommando in der nächsten Eingabezeile. Jedes Schlüsselwort in der Kommandozeichenfolge muss durch das Begrenzungszeichen und nicht durch ein Leerzeichen abgegrenzt werden.

   Wenn die Direktkommandozeichenfolge länger als eine Zeile ist, müssen die Zeichen `CF` (siehe auch den Session-Parameter `CF`, der in *CF - Steuerzeichen für Terminalkommandos* in der *Parameter-Referenz*-Dokumentation beschrieben ist) an das Ende der Zeile gesetzt werden, um mit dem Direktkommando in der nächsten Zeile fortzufahren.

2 Wenn Sie nach dem/den `SYSMAIN`-Kommandos andere Natural-Befehle ausführen wollen, müssen Sie SYSMAIN zunächst mit dem Direktkommando `END` oder `QUIT` beenden.

SYSMAIN-Kommandozeile benutzen

In der Kommandozeile eines beliebigen SYSMAIN-Menüs können Sie eines der folgenden Kommandos eingeben:

- Ein Direktkommando zur Ausführung einer SYSMAIN-Funktion.
- Ein spezielles Kommando an das SYSMAIN-Dienstprogramm, das in ***An SYSMAIN absetzbare spezielle Kommandos*** beschrieben wird.
- Ein Systemkommando. Wenn der Befehl nicht eindeutig als Systemkommando identifizierbar ist, müssen ihm zwei Schrägstriche (//) vorangestellt werden. Siehe auch **Bemerkungen** in *MAINEX05 - User-Exit-Routine zur Verifizierung von Direktkommandos*.

Beschreibung der Funktionen

Die Funktionen, die in einem SYSMAIN-Menü oder als Direktkommando zur Verfügung stehen, werden im folgenden Abschnitt beschrieben. Zu jeder Funktion, die in einem Menü angeboten wird, gibt es ein entsprechendes Direktkommando mit demselben Namen. Ausnahme: **Help**.

Die für Direktkommandos geltende Syntax finden Sie in den objekttypspezifischen Abschnitten dieser Dokumentation.

Die speziellen Kommandos, die an das SYSMAIN-Dienstprogramm abgesetzt werden können, finden Sie unter *An SYSMAIN absetzbare spezielle Kommandos*.

Funktion/Kommando	Erläuterung
Copy	**Kopieren** Kopiert ein Natural-Objekt oder ein Predict Set aus einer Quellumgebung in eine Zielumgebung. Das Objekt bleibt in der Quellumgebung unverändert. Wenn die Zielumgebung bereits ein Objekt mit dem gleichen Namen (oder im Falle einer Fehlermeldung oder eines Predict Set, mit der gleichen Nummer) wie das zu kopierende Objekt enthält, wird das angegebene Objekt nicht kopiert. Mit der Option Replace/Ersetzen (siehe *Option Replace (Ersetzen) verwenden*) können Sie ein Objekt in der Zielumgebung überschreiben.
Delete	**Löschen** Löscht ein Natural-Objekt oder ein Predict Set aus einer Quellumgebung. Bei der **automatisierten** Online-Verarbeitung (siehe *Funktionsverarbeitung und Reporting*) wird ein Bestätigungsfenster angezeigt, in dem Sie die Möglichkeit haben, die Funktion fortzusetzen oder abzubrechen. Bei Fehlermeldungen: Wenn Natural Security installiert ist, ist die Löschfunktion bei Systemfehlermeldungen nicht erlaubt. Wenn Sie versuchen, eine Systemfehlermeldung zu löschen, wird die folgende Natural-Systemmeldung angezeigt: `4897 : Invalid error application specified` (Ungültige Fehleranwendung angegeben).
Find	**Suchen** Gilt nur bei Programmierobjekten und Fehlermeldungen. Findet ein oder mehrere Programmierobjekte oder Fehlermeldungen in einer Quellumgebung. Während der Online-Verarbeitung wird ein Fenster mit der aktuell durchsuchten Library angezeigt. Im Menümodus bietet die Suchfunktion auch die Möglichkeit, Libraries aufzulisten und auszuwählen oder die Anzahl der in einer Auswahlliste angezeigten Objekte zu

<table>
<tr><th>Funktion/Kommando</th><th>Erläuterung</th></tr>
<tr><td></td><td>reduzieren. Siehe Auflisten und Auswählen von Libraries bzw. Auswahlliste benutzen > Um eine Auswahlliste zu verkürzen.</td></tr>
<tr><td>List</td><td>Auflisten

Zeigt einen Bereich von Natural-Objekten oder Predict Sets in einer Quellumgebung an.

Bei Programmierobjekten, Debug-Umgebungen und Fehlermeldungen: Im Menümodus bietet die Auflistfunktion auch die Möglichkeit, Libraries aufzulisten und auszuwählen oder die Anzahl der in einer Auswahlliste angezeigten Objekte zu reduzieren. Siehe Auflisten und Auswählen von Libraries bzw. Auswahlliste benutzen > Um eine Auswahlliste zu verkürzen.

Bei Programmierobjekten: Im Batch-Modus können Sie sich mit dem Direktkommando <code>LISTLIB</code> eine Liste der Library-Namen anzeigen lassen.</td></tr>
<tr><td>Move</td><td>Verschieben

Überträgt ein Natural-Objekt oder ein Predict Set aus einer Quellumgebung in eine Zielumgebung. Das Objekt wird in der Quellumgebung gelöscht und in der Zielumgebung hinzugefügt. Wenn die Zielumgebung bereits ein Objekt mit demselben Namen (oder im Falle einer Fehlermeldung mit derselben Nummer) wie das zu verschiebende Objekt enthält, wird das angegebene Objekt nicht verschoben.

Während der automatisierten Online-Bearbeitung wird ein Bestätigungsfenster angezeigt, das Ihnen die Möglichkeit gibt, die Funktion fortzusetzen oder abzubrechen.

Mit der Ersetzungsoption (siehe Option Replace (Ersetzen) verwenden) können Sie das Objekt in der Zielumgebung überschreiben.</td></tr>
<tr><td>Rename</td><td>Umbenennen

Das Ergebnis der Umbenennungsfunktion hängt vom Typ des aufgerufenen Objekts ab:
<table><tr><th>Objekttyp</th><th>Ergebnis der Umbenennungsfunktion</th></tr><tr><td>Error Message</td><td>Fehlermeldung: Es wird eine neue Nummer zugewiesen.</td></tr><tr><td>Predict Set</td><td>Es wird eine neue Nummer und/oder ein neuer Benutzer und/oder eine neue Library zugewiesen.</td></tr><tr><td>Alle anderen</td><td>Das Objekt wird umbenannt.</td></tr></table>Beachten Sie, dass die Funktion auf die folgenden Arten verwendet werden kann:

1. Sie führen die Umbenennungsfunktion bei einem Objekt in der Quellumgebung aus.
2. Sie führen die Umbenennungsfunktion bei einem Objekt in der Quellumgebung aus und kopieren/verschieben es in eine andere (d.h. Ziel-) Umgebung. Beachten</td></tr>
</table>

Funktion/Kommando	Erläuterung
	Sie, dass die Umbenennungsfunktion jeweils nur für eine Umgebung ausgeführt werden kann. Die Umbenennungsfunktion löscht das ursprüngliche Objekt in der Quellumgebung. Deshalb werden Sie gefragt, ob Sie das ursprüngliche Objekt beibehalten wollen. Wenn das ursprüngliche Objekt beibehalten werden soll, wird es nicht gelöscht. Wenn in der Zielumgebung bereits ein Objekt mit dem gleichen Namen (oder im Falle einer Fehlermeldung oder eines Predict Set mit der gleichen Nummer) wie das angegebene Objekt vorhanden ist, wird die Umbenennungsfunktion nicht ausgeführt. In diesem Fall können Sie stattdessen die Option Replace (Ersetzen) verwenden (siehe *Option Replace (Ersetzen) verwenden*), um das Objekt in der Zielumgebung zu überschreiben. Bei **automatisierter** Verarbeitung kann die Umbenennungsfunktion keine Objektbereiche verarbeiten. Verwenden Sie stattdessen die **Selektive Verarbeitung**. Bei Fehlermeldungen: Wenn Sie einen Bereich von Fehlermeldungen innerhalb einer einzelnen (Quell-) Library neu nummerieren, dürfen sich die Werte des Bereichs nicht überschneiden. Es ist zum Beispiel nicht möglich, die Fehlernummern 1 - 6 in die neuen Fehlernummern 5 - 10 umzubenennen. Ein Bereich von Fehlermeldungen kann mit **automatisierter** Verarbeitung umbenannt werden. Wenn große Bereiche von Fehlermeldungen verarbeitet werden, kann die Verarbeitung der Meldungen erhebliche Ressourcen erfordern. In solchen Fällen ist die Verarbeitung im Stapelverarbeitungsmodus möglicherweise vorzuziehen.
Help	**Hilfe** Bietet Hilfeinformationen zu SYSMAIN. Siehe *SYSMAIN Online-Hilfe*. Die Hilfefunktion ist nur in einem SYSMAIN-Menü verfügbar. Ein entsprechendes Direktkommando gibt es nicht.
Exit	**Beenden** Beendet das SYSMAIN-Dienstprogramm. PF3 (Exit) beendet SYSMAIN ebenfalls, wenn Sie die Taste im Hauptmenü des SYSMAIN-Dienstprogramms drücken.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Option Replace (Ersetzen) verwenden

- Auflisten und Auswählen von Libraries

Option Replace (Ersetzen) verwenden

Wenn die Zielumgebung bereits ein Objekt mit demselben Namen wie das zu kopierende, zu verschiebende oder umzubenennende Objekt enthält, wird das angegebene Objekt nicht verarbeitet und die Verarbeitung mit dem nächsten Objekt fortgesetzt. Mit der **Replace**-Option können Sie diese Standardfunktionalität außer Kraft setzen und das Objekt in der Zielumgebung überschreiben.

Wenn ein Programmierobjekt ersetzt wird, wird es auch aus dem Natural Buffer Pool gelöscht. Eventuell vorhandene Cross-Referenz-Datensätze werden ebenfalls gelöscht, wenn Predict installiert ist.

Um die Replace-Option im Kommandomodus zu aktivieren:

- Geben Sie in der Kommandozeichenfolge das Schlüsselwort `REPLACE`.

Um die Replace-Option im Menümodus zu aktivieren:

1. Geben Sie in einem SYSMAIN-Menü im Feld **Replace** ein `Y` ein.
2. Führen Sie eine Funktion aus. Wenn die Option S**el. List** (Auswahlliste) auf `N` gesetzt wurde, erscheint ein Fenster, in dem Sie wählen können, ob Sie jeden Ersetzungsvorgang bestätigen möchten, bevor er ausgeführt wird: Geben Sie ein `Y` ein, um jeden Ersetzungsvorgang zu bestätigen, oder drücken Sie ENTER, um die Verarbeitung ohne Bestätigung fortzusetzen.
3. Wenn Sie ein `Y` zur Bestätigung einer Ersetzung eingegeben haben, erscheint für jedes zu ersetzende Objekt ein Fenster, in dem Sie eines der folgenden Zeichen eingeben können:
   - Bei Eingabe von `Y` wird das im Fenster angegebene Objekt ersetzt,
   - bei `N` wird das im Fenster angegebene Objekt nicht ersetzt (dies ist die Standardeinstellung)
   - Ein Punkt (.) beendet die Funktionsverarbeitung. Alternativ können Sie auch PF3 drücken.

Auflisten und Auswählen von Libraries

Diese Option gilt nur bei Programmierobjekten, Debug-Umgebungen und Fehlermeldungen.

Wenn Sie die Auflistfunktion verwenden, können Sie eine Auswahlliste der Libraries aufrufen, die das (oder die) angegebene(n) Objekt(e) enthalten. Dies gilt auch für die Suchfunktion, wenn das angegebene Objekt in mehreren Libraries enthalten ist.

- Liste von Libraries für Programmierobjekte aufrufen
- Liste von Libraries für Debug-Umgebungen aufrufen

- Liste von Libraries für Fehlermeldungen aufrufen

Liste von Libraries für Programmierobjekte aufrufen

Um eine Liste von Libraries für Programmierobjekte aufzurufen:

- Geben Sie in einem **List**-Menü im Feld **Library** einen Bereich von Library-Namen ein (siehe auch *Angabe eines Namensbereichs*) und im Feld **Name** einen Objektnamen oder einen Namensbereich.

  Wenn Sie einen Stern (*) in das Feld **Library** eingeben (nur **List**-Funktion), wird eine Liste aller in der angegebenen Systemdatei verfügbaren Libraries angezeigt.

  In dem unten gezeigten Beispiel eines **Library Selection**-Bildschirms werden alle Libraries, deren Namen mit `LIB` beginnen, ausgewählt:

```
 21:23:03            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
 User SAG                   -   Library Selection  -

LIST ALL * IN LIB* WHERE DBID 10 FNR 32

  C Library   S/C     C Library   S/C     C Library   S/C     C Library   S/C
  - --------  ---     - --------  ---     - --------  ---     - --------  ---
  _ LIB1      S/C     _ LIB2      S/C     _ LIB3        C     _ LIB4      S/C
  _ LIB5        S     _ LIB6      S/C     _ LIB7      S/C     _ LIB8      S/C
  _ LIB9      S/C     _ LIB10     S/C     _ LIB11     S/C     _ LIB12     S/C
  _ LIB13     S       _ LIB14     S/C     _ LIB15       C     _ LIB16     S/C
  _ LIB17     S/C     _ LIB18     S/C     _ LIB19     S/C     _ LIB20     S/C









                                              Object Start Value: *

             Enter options (above), or '?' (Help) or '.' (Exit): _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren               Canc
```

  In der Spalte **Library** werden alle Libraries aufgelistet, die dem angegebenen Bereich entsprechen. Die Spalte **S/C** zeigt an, ob eine Library gespeicherte (Quellcode-)Objekte und/oder katalogisierte Objekte enthält.

  Eine Liste der Zeilenkommandos, die auf dem Bildschirm **Library Selection** verfügbar sind, finden Sie unter *Auswahllisten für Programmierobjekte*.

Wenn Sie die Suchfunktion (**Find**) verwenden, wird der Bildschirm **Library Selection** angezeigt, wenn das angegebene Objekt in mehreren Libraries gefunden wird.

Der Bildschirm **Library Selection** erscheint auch, wenn Sie einen **Find Selection**- oder **List Selection**-Bildschirm durch Drücken von PF3 (Quit) verlassen.

Liste von Libraries für Debug-Umgebungen aufrufen

Um eine Liste von Libraries für Debug-Umgebungen aufzurufen:

- Im Menü **Debug Environments**:

  Geben Sie in das Feld **Code** ein `L` und in das Feld **Environment Name** den Namen einer Debug-Umgebung oder einen Bereich von Namen und in das Feld **Source Library** einen Bereich von Library-Namen ein (siehe auch *Angabe eines Namensbereichs*).

  Wenn Sie in den Feldern **Environment Name** und **Source Library** einen Stern (*) eingeben, wird eine Liste aller Libraries angezeigt, die Debug-Umgebungen in der angegebenen Systemdatei enthalten.

  In dem unten gezeigten Beispiel eines **Library Selection**-Bildschirms wurden alle Libraries, deren Namen mit `L` beginnen, ausgewählt:

```
09:55:08             ***** NATURAL SYSMAIN UTILITY *****            2005-08-10
 User SAG                      -   Library Selection  -
LIST DEBUG * IN LIB* FROM DBID 10 FNR 32


  C Library           C Library           C Library           C Library
  - --------          - --------          - --------          - --------
  _ LIB1              _ LIB5              _ LIB9              _ LIB11
  _ LIB20

                                                  Object Start Value ... *
              Enter options (above), or '?' (Help) or '.' (Exit): _

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

In der Spalte **Library** werden alle Libraries aufgelistet, die die angegebene(n) Debug-Umgebung(en) enthalten. Eine Liste der Zeilenkommandos, die auf dem Bildschirm **Library Selection** verfügbar sind, finden Sie unter *Auswahllisten für Debug-Umgebungen*.

Liste von Libraries für Fehlermeldungen aufrufen

Um eine Liste von Libraries für Fehlermeldungen aufzurufen:

- Wählen Sie eine der folgenden Methoden:

1. Geben Sie in einem **List**-Menü in das Feld **No. From** eine Fehlermeldungsnummer ein (im untenstehenden Beispiel `1`) oder geben Sie einen Nummernbereich an, indem Sie eine Startnummer in das Feld **No. From** und eine Endnummer in das Feld **No. To** eingeben.

   Geben Sie im Feld **Library** einen Bereich von Library-Namen ein (siehe auch *Angabe eines Namensbereichs*).

   Es erscheint ein **Library Help**-Fenster ähnlich dem folgenden Beispiel:

```
14:02:47            ***** NATURAL SYSMAIN UTILITY *****          2005-08-10
User SAG                  - List Error Message Texts -
                               +---------------------------+
                               !  --- Library Help ---     !
                     Code  Fun !       - Source -          !
                               !                           !
                       A   Lis !   1   system messages     !
                       E   Lis !   2   LIB1                !
                       S   Lis !   3   LIB2                !
                       ?   Hel !   4   LIB3                !
                       .   Exi !   5   LIB4                !
                               !   6   LIB5                !
            Code ...... A      !   7   LIB6                !
                               !   8   LIB7                !
 Error      No. From .. 1___   !                           !
 Source     Library ... LIB*_  !                         ! 0___ FNR .. 32___
                               !   Enter selection or      !
                               !    '.' to Exit: 8_        !
                               +---------------------------+


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

   Wählen Sie im Fenster eine Library aus, indem Sie die Nummer eingeben, die der gewünschten Library entspricht, wie im obigen Beispiel gezeigt, in dem `8` eingegeben wurde, um `LIB7` auszuwählen. Wenn Sie die richtige FNAT-Systemdatei angegeben haben,

können Sie auch Systemfehlermeldungen auswählen. Drücken Sie bei Bedarf die Eingabetaste, um an das Ende der Liste zu gelangen.

2. In einem **Find**-Menü geben Sie in das Feld **Number** eine Fehlermeldungsnummer und in das Feld **Library** einen Bereich von Library-Namen ein (siehe auch *Angabe eines Namensbereichs*).

   Für den angegebenen Library-Bereich (hier: `LIB*`) wird ein **Find Selection**-Bildschirm ähnlich dem folgenden Beispiel angezeigt:

```
 19:09:34           ***** NATURAL SYSMAIN UTILITY *****          2005-08-10
 User SAG                      - Find Selection -

FIND ERR 1 TYPE A FROM LIB* LANG * WHERE DBID 10 FNR 32


 C   Library   Error Message Text(s) for Error number: 1                Type
 -   --------  ----------------------------------------------------------  ----
 _    LIB1      Short message of Error 1: wrong input value.                S
 _    LIB2      Short message of Error 1: incorrect syntax.                 S
 _    LIB3      Short message of Error 1: undefined variable.               S
 _    LIB4      Short message of Error 1: timeout error.                    S
 _    LIB5     Short message of Error 1: initialization failed.           S/E
 _    LIB6      Short message of Error 1: invalid format.                  S/E
 _    LIB7      Short message of Error 1: wrong password.                  S/E
 _    LIB8      Short message of Error 1: input missing.                   S/E
 _    LIB9      Short message of Error 1: undefined keyword.               S/E
 _    LIB10     Short message of Error 1: invalid command.                  S



        Enter options (above), or '?' (Help) or '.' (Exit): __
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del         List  Move  Ren               Canc
```

In der Spalte **Library** werden alle Libraries aufgelistet, die die angegebene(n) Fehlermeldung(en) enthalten (hier: `1`). Die Spalte **Type** zeigt an, ob die kurze (`S`) und/oder die erweiterte (`E`) Fehlermeldung für die angegebene Meldung verfügbar ist.

Eine Liste der auf dem Bildschirm **Find Selection** verfügbaren Zeilenkommandos finden Sie unter *Auswahllisten für Fehlermeldungen*.

Funktionsverarbeitung und Reporting

Es gibt zwei Arten der Verarbeitung, die bei der Ausführung einer SYSMAIN-Funktion verwendet werden können: selektive Verarbeitung und automatisierte Verarbeitung.

Dieser Abschnitt beschreibt die beiden Arten der Funktionsverarbeitung und wie Sie einen Statusbericht über die Funktionsverarbeitung erhalten.

- Automatisierte Verarbeitung
- Selektive Verarbeitung
- Auswahlliste benutzen
- Online-Report-Modus

Automatisierte Verarbeitung

Die automatisierte Verarbeitung ist die vorgegebene Art der Verarbeitung, wenn Sie online im Kommandomodus arbeiten. Im Batch-Modus wird automatisierte Verarbeitung standardmäßig verwendet.

Bei der automatisierten Verarbeitung handelt es sich um eine Online- oder Batch-Funktionalität, die Objekte ohne Anzeige einer Auswahlliste verarbeitet. Dies erfordert wenig oder gar keine Eingabe/Ausgabe am Terminal, nachdem eine Funktion ausgewählt wurde.

Bei der automatisierten Online-Verarbeitung wird der Status der einzelnen Objekte nicht angezeigt, sondern es wird nach Abschluss der Verarbeitung eine entsprechende Meldung angezeigt. Wenn eine der folgenden Fehlermeldungen des Natural-Systems angezeigt wird, wurden einige oder alle der angegebenen Objekte nicht verarbeitet:

4867	Für diesen Anforderung wurde nichts gefunden.
4810	Alle Daten wurden nach diesen Auswahlkriterien abgelehnt.
4893	Normaler Abschluss, aber einige Daten wurden abgelehnt.

Der erweiterte (lange) Meldungstext des Fehlers NAT4810 listet Gründe auf, warum ein Objekt möglicherweise nicht verarbeitet wurde. Die Gründe, warum ein Objekt nicht verarbeitet werden konnte, werden auch im Abschnitt ***Statusmeldungen und Fehlerbenachrichtigungen bearbeiten*** behandelt.

Der Batch-Modus oder die selektive Verarbeitung sollte verwendet werden, wenn es notwendig ist, den Status jedes Objekts nach seiner Verarbeitung zu sehen.

Bei Bedarf können Sie im Online-Betrieb einen Batch-Report mit dem Verarbeitungsstatus aller Objekte anfordern, wie unter ***Online Report-Modus*** beschrieben.

Selektive Verarbeitung

Die selektive Verarbeitung ist die vorgegebene Art der Verarbeitung, wenn Sie im Menümodus arbeiten.

Die selektive Bearbeitung ist eine Online-Funktionalität, die eine Auswahlliste mit allen Objekten anzeigt, die den angegebenen Auswahlkriterien entsprechen. Eine Auswahlliste bietet die folgenden Optionen:

- Einzelne oder mehrere Objekte für die weitere Verarbeitung auswählen.
- Vor der Bearbeitung eines Objektes eine zusätzliche Funktion (z.B. Anzeige des Quelltextes) ausführen. Siehe *Auswahlliste benutzen*).
- Einzelne Meldungen über den Bearbeitungsstatus jedes Objekts bekommen (z.B. `Moved` oder `Copied`).
- Anzeige des Direktkommandos, das den im SYSMAIN-Menü eingegebenen Eingabewerten entspricht (siehe das **Beispiel eines Auswahlbildschirms** unten). Dies kann bei der Benutzung von Direktkommandos anstelle von Menüfunktionen hilfreich sein.

Auswahlliste benutzen

Dieser Abschnitt enthält Anleitungen zum Aufrufen einer Auswahlliste, um einzelne oder mehrere Objekte weiter zu verarbeiten. Außerdem wird in diesem Abschnitt beschrieben, wie Sie eine Auswahl verkürzen können, wenn Sie die **List**- oder **Find-**Funktion verwenden. Diese Option ist bei großen Libraries nützlich, um die Anzahl der angezeigten Objekte zu begrenzen.

Um eine Anweisungsliste online zu benutzen:

1 **Im Menümodus:** Geben Sie in einem SYSMAIN-Menü im Feld **Sel. List** ein `Y` ein (dies ist die Standardeinstellung), um die selektive Bearbeitung zu aktivieren. Die Eingabe eines `N` deaktiviert die selektive Bearbeitung.

Oder:

Im Kommandomodus: Fügen Sie das Schlüsselwort `HELP` in die With-Klausel (*`with-clause`*) des Direktkommandos ein oder geben Sie ein Fragezeichen (`?`) unmittelbar nach dem Objektnamen ein. Siehe auch die Syntax-Diagramme in den objekttypspezifischen Abschnitten der SYSMAIN Dienstprogramm-Dokumentation.

Wenn die selektive Verarbeitung aktiviert wurde, erscheint ein **Selection**-Fenster ähnlich dem folgenden Beispiel:

```
 11:05:22            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
 User SAG                    -    Copy   Selection  -

COPY ALL * WITH XREF N FROM OLDLIB WHERE DBID 10 FNR 50 TO NEWLIB WHERE DBID
10 FNR 60

C  Name     Type   S/C  Message        C  Name     Type   S/C  Message
-  -------- ------ ---  ------------   -  -------- ------ ---  ------------
_  AA       Copycd S/C                 _  G0000001 Global S
_  G0000002 Global S/C                 _  G0000003 Global S/C
_  L0000001 Local  S/C                 _  L0000002 Local  S/C
_  L0000003 Local  S                   _  M0000001 Map    S
_  M0000002 Map    S/C                 _  M0000003 Map    S/C
_  P0000001 Progrm S                   _  P0000002 Progrm   C
_  P0000003 Progrm S/C                 _  P0000004 Progrm S/C
_  P0000005 Progrm S                   _  P0000006 Progrm S/C
_  P0000007 Progrm S/C                 _  S0000001 Subpgm S/C
_  S0000002 Subpgm S/C                 _  S0000003 Subpgm S
_  T0000001 Text   S                   _  T0000002 Text
                                                  Listed Library: OLDLIB

              Enter options (above), or '?' (Help) or '.' (Exit) _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren               Canc
```

Die Liste enthält die Namen und die Typen der angegebenen Objekte und zeigt an, ob gespeicherte Objekte (`S`) und/oder katalogisierte Objekte (`C`) vorhanden sind.

In den Zeilen oberhalb der Auswahlliste (im obigen Beispiel hervorgehoben) wird das SYSMAIN-Direktkommando angezeigt, das den von Ihnen in den Menüfeldern eingegebenen Werten entspricht.

Dieses Kommando entspricht der Kommandosyntax, die beim Arbeiten im Kommandomodus gilt, wobei einige Schlüsselwörter optional sind, wie in den Syntax-Diagrammen in den objekttypspezifischen Abschnitten der SYSMAIN-Dokumentation gezeigt.

2 Wählen Sie ein oder mehrere Objekte für die weitere Bearbeitung aus:

Geben Sie in der Spalte **C** (Kommando) neben dem/den gewünschten Objekt(en) eines der in den objekttypspezifischen Abschnitten dieser Dokumentation beschriebenen Zeilenkommandos ein.

3 Drücken Sie ENTER, um ein oder mehrere Zeilenkommandos auszuführen.

Die Zeilenkommandos werden in alphabetischer Reihenfolge der angegebenen Objektnamen ausgeführt, wobei Kommandos, die eine SYSMAIN-Pflegefunktion ausführen (z.B. Löschen), immer zuletzt ausgeführt werden.

Wurde für ein Objekt *keine* SYSMAIN-Pflegefunktion (z.B. Anzeigen von Quellcode) ausgeführt, können Sie für ein beliebiges Objekt erneut ein Zeilenkommando eingeben. Sobald jedoch

eine SYSMAIN-Pflegefunktion für ein oder mehrere Objekte ausgeführt wurde, sind die Felder der Spalte **C** nicht mehr eingabebereit.

Wenn ein Zeilenkommando ausgeführt wurde, wird der Status des/der bearbeiteten Objekts/Objekte in der Spalte **Message** (Meldung) angezeigt, wie im folgenden Beispiel dargestellt:

```
 11:05:22            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
 User SAG                    -    Copy   Selection  -

COPY ALL * WITH XREF N FROM OLDLIB WHERE DBID 10 FNR 50 TO NEWLIB WHERE DBID
10 FNR 60

C  Name     Type   S/C  Message        C  Name     Type   S/C  Message
-  -------- ------ ---  ------------   -  -------- ------ ---  ------------
_  AA       Copycd S/C  Copied         _  G0000001 Global S    Copied
_  G0000002 Global S/C  Copied         _  G0000003 Global S/C  Copied
_  L0000001 Local  S/C  Copied         _  L0000002 Local  S/C  Copied
_  L0000003 Local  S    Replaced       _  M0000001 Map    S    Replaced
_  M0000002 Map    S/C  Replaced       _  M0000003 Map    S/C  Replaced
_  P0000001 Progrm S                   _  P0000002 Progrm   C
_  P0000003 Progrm S/C                 _  P0000004 Progrm S/C
_  P0000005 Progrm S                   _  P0000006 Progrm S/C
_  P0000007 Progrm S/C                 _  S0000001 Subpgm S/C
_  S0000002 Subpgm S/C                 _  S0000003 Subpgm S
_  T0000001 Text   S                   _  T0000002 Text
                                                  Listed Library: OLDLIB

              Enter options (above), or '?' (Help) or '.' (Exit) _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren               Canc
```

Mögliche Meldungen siehe ***Status-Meldungen*** im Abschnitt *Statusmeldungen und Fehlerbenachrichtigungen bearbeiten*.

Um eine Auswahlliste zu verkürzen:

- Geben Sie in einem **List Selection**- oder **Find Selection**-Bildschirm im Feld **Reposition to** den Namen eines Objekts ein oder geben Sie einen Namensbereich an (siehe ***Angabe eines Namensbereichs***) und drücken Sie ENTER.

  Wenn Sie einen Namen angegeben haben, beginnt die Liste nun mit dem angegebenen Namen. Wenn Sie einen Namensbereich angegeben haben, werden in der Liste nur noch Objekte innerhalb dieses Bereichs angezeigt.

  Die Option **Reposition to** ist nur im Online-Modus gültig. Sie ist kein Auswahlkriterium für die **List**-Funktion.

Online-Report-Modus

Der Online-Report-Modus kann verwendet werden, um statt einer Auswahlliste einen SYSMAIN-Batch-Report online abzurufen. Ein Online-Report listet alle Objekte auf, die von einer SYSMAIN-Funktion betroffen waren, und gibt an, welche Aktion an jedem dieser Objekte durchgeführt wurde.

Um einen Batch-Report online zu benutzen:

1 Geben Sie in einem SYSMAIN-Menü in das Feld **Sel. List** (Auswahlliste) ein `N` ein, um die automatisierte Verarbeitung zu aktivieren.

2 Geben Sie in der Kommandozeile eines der folgenden Direktkommandos ein:

```
BATCH
```

oder

```
BAT
```

Es erscheint eine Meldung, die bestätigt, dass der Batch-Modus aktiviert wurde.

3 Bei Bedarf können Sie eine Hardcopy des Berichts anfordern, indem Sie das folgende Terminalkommando eingeben:

```
%H
```

(Siehe auch *%H - Hardcopy-Ausgabe* in der *Terminalkommandos*-Dokumentation.)

4 Führen Sie eine SYSMAIN-Funktion aus. SYSMAIN verarbeitet nun die Funktion wie im Batch-Modus. Daher wird nur das Ergebnis jeder Aktion in einem reportartigen Format ausgegeben, wie im folgenden Beispiel gezeigt:

```
10:50:30              ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
User SAG                    - Copy   Function -

COPY ALL * WITH XREF N FROM TESTLIB1 WHERE DBID 10 FNR 30 TO TESTLIB2 WHERE
DBID 10 FNR 40




Saved     Progrm TEST1    has now been Copied
Saved     Progrm TEST2    has now been Copied
Saved     Progrm TEST3    has now been Copied
Stowed    Progrm TEST4    has now been Copied
Stowed    Progrm TEST5    has now been Copied
Stowed    Progrm TEST6    has now been Copied
Stowed    Progrm TEST7    has now been Copied
Saved     Record TEST8    has now been Copied
Saved     Text   TEST9    has now been Copied
Cataloged Progrm TEST10   has now been Copied
Saved     Progrm TEST11   has now been Copied
Stowed    Progrm TEST12   has now been Copied
MORE
Stowed    Progrm TEST13   has now been Copied
Cataloged Progrm TEST14   has now been Copied
Stowed    Progrm TEST15   has now been Copied
```

5 Bei Bedarf können Sie die Bearbeitung der Funktion unterbrechen, indem Sie an einer MORE-Eingabeaufforderung ein System- oder Terminalkommando eingeben. Andernfalls drücken Sie ENTER, bis Sie das Ende der Liste erreicht haben und kehren zum SYSMAIN-Menü zurück, wo eine Meldung die erfolgreiche Ausführung der Funktion bestätigt.

SYSMAIN-Online-Hilfe

Die SYSMAIN-Online-Hilfe bietet Informationen zu allen Funktionen von SYSMAIN, einschließlich detaillierter Erklärungen der Syntax der Direktkommandos und Beispiele für Direktkommandos.

Zusätzlich zur Hilfe bietet SYSMAIN individuelle Informationen zu jedem Eingabefeld, das in jedem SYSMAIN-Bildschirm verfügbar ist.

Um die SYSMAIN-Hilfethemen aufzurufen:

1 Positionieren Sie im Hauptmenü des SYSMAIN-Dienstprogramms (**SYSMAIN Utility Main Menu**) den Cursor im Feld **Object Code** oder **Function Code** und drücken Sie PF1 (Hilfe) oder geben Sie ein Fragezeichen (?) ein.

Das Hilfemenü (**Help Menu**) des Dienstprogramms SYSMAIN wird ähnlich wie im folgenden Beispiel angezeigt. Es enthält eine Liste von Hilfethemen:

```
15:54:28            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
User SAG                        - Help Menu -
                                                              Help Name SHT-0001

                 Code  Topic

                   A   SYSMAIN General Overview
                   C   Command Mode
                   E   Environment Definition
                   F   Functions / Commands
                   S   Security Environment
                   .   Exit









           Code ... _

Select a function code.

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

2 Geben Sie in das Feld **Code** den aus einem Buchstaben bestehenden Code ein, der dem gewünschten Hilfethema entspricht.

3 Drücken Sie ENTER.

Es erscheint ein Ergebnisbildschirm mit Informationen zu dem gewählten Hilfethema, oder es wird ein weiteres Menü mit weiteren Hilfethemen aufgerufen, die Ihre Suche eingrenzen.

Um die Hilfe zu einem Eingabefeld aufzurufen:

- Positionieren Sie den Cursor in das betreffende Feld und drücken Sie PF1 (Hilfe) oder geben Sie ein Fragezeichen (?) ein und drücken Sie ENTER.

  Es erscheint ein Fenster mit feldspezifischen Anleitungen und ggf. einer Liste der gültigen Eingabewerte.

72 Programmierobjekte bearbeiten

- Felder in den Menüs für Programmierobjekte 694
- Profilparameter RECAT verwenden 697
- Auswahllisten für Programmierobjekte 697
- Besonderheiten bei XRef 702
- Angabe zusätzlicher Kriterien 704
- Direktkommando-Syntax für Programmierobjekte 706

Programmierobjekte können mit allen SYSMAIN-Funktionen be- bzw. verarbeitetet werden. Dazu gehören die folgenden Natural-Objekttypen: Programm, Subprogramm, Subroutine, Copycode, Helproutine, Map (Maske), Local Data Area (LDA, lokaler Datenbereich), Global Data Area (GDA, globaler Datenbereich), Parameter Data Area (PDA, Parameterdatenbereich), Class, Text, Recording, Natural Command Processor, Dialog, Function, ISPF Macro, Report, Adapter und Resource.

Programmierobjekte werden in den Systemdateien entsprechend dem Namen der Library, in der sie enthalten sind, gespeichert: wenn die Library mit SYS beginnt (außer der Library SYSTEM), werden die Objekte in der Systemdatei FNAT gespeichert. In allen anderen Libraries werden die Objekte in der Systemdatei FUSER gespeichert.

Dieses Kapitel beschreibt die Menüfunktionen und die Optionen der Auswahlliste, die zur Ausführung einer SYSMAIN-Funktion bei Programmierobjekten zur Verfügung stehen, sowie die Syntax, die bei der Verwendung von Direktkommandos gilt.

Felder in den Menüs für Programmierobjekte

Die Menüs für Programmierobjekte (**Programming Objects**) enthalten alle SYSMAIN-Funktionen, die für die Verarbeitung von Programmierobjekten erforderlich sind. Die Felder, die in einem Menü vorhanden sein können, sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung	
Recat: ON	Zeigt an, dass der Profilparameter `RECAT` auf `ON` gesetzt wurde. Siehe *Profilparameter RECAT verwenden*.	
Code	Gibt an, ob ein gespeichertes (Quellcode-)Objekt und/oder ein katalogisiertes Objekt verarbeitet werden soll:	
	`A`	Jedes Objekt, das als gespeichertes und/oder katalogisiertes Objekt existiert, wird bearbeitet.
	`C`	Jedes Objekt, das als katalogisiertes Objekt vorhanden ist, wird bearbeitet.
	`S`	Jedes Objekt, das als gespeichertes Objekt vorhanden ist, wird bearbeitet.
	`W`	Nur ein Objekt, das sowohl als gespeichertes als auch als katalogisiertes Objekt vorhanden ist, wird bearbeitet. Ausnahmen sind Copycode, Text und Recording, die nicht katalogisiert werden können. Sie werden jedoch in die Verarbeitung einbezogen, wenn diese Option angegeben ist.
Sel. List	Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird:	

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td></td><td><table>
<tr><td>Y</td><td>Ja: Die selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von Objekten wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten, Felder und Zeilenkommandos in einer Auswahlliste verfügbar sind, erfahren Sie unter Auswahllisten für Programmierobjekte.</td></tr>
<tr><td>N</td><td>Nein: Die selektive Bearbeitung ist ausgeschaltet.</td></tr>
</table></td></tr>
<tr><td>Name</td><td>Der Name des zu verarbeitenden Objekts oder ein Namensbereich: siehe auch Angabe eines Namensbereichs.

Die Standardeinstellung ist ein Stern (*), was bedeutet, dass alle Namen ausgewählt werden.</td></tr>
<tr><td>New Name</td><td>Neuer Name: Der Name, der einem Objekt gegeben werden soll, wenn es mit der Funktion Rename umbenannt wird.</td></tr>
<tr><td>Type</td><td>Typ: Der Code, der dem/den zu bearbeitenden Objekttyp(en) entspricht, z. B. P für Programm und M für Map. Siehe TYPE-Angabe. Sie können einen oder mehrere Codes in beliebiger Reihenfolge eingeben. Wenn Sie z.B. PAM eingeben, werden Programme, Parameterdatenbereiche und Maps verarbeitet.</td></tr>
<tr><td>Set Number</td><td>Die Nummer des gespeicherten Sets, das mit der Option XRef Save Set des Kommandos LIST XREF erstellt wurde. Sie können alle SYSMAIN-Verarbeitungsfunktionen auf die in diesem Set enthaltenen Objekte anwenden.

Wenn eine gültige Nummer angegeben wird, geht SYSMAIN von einem Predict Set aus. Wenn keine Nummer angegeben wird, wird eine normale Objektverarbeitung angenommen. Sie können eine Library und eine Benutzerkennung für den Predict Set angeben, indem Sie die Felder Set Library und Set User im Fenster Additional Criteria benutzen, das unter Angabe zusätzlicher Kriterien beschrieben wird.</td></tr>
<tr><td>XREF</td><td>Zeigt an, ob XRef-Daten, die in Predict-Systemdateien gespeichert sind, verarbeitet werden sollen.<table>
<tr><td></td><td></td></tr>
<tr><td>N</td><td>Nein: XRef-Daten werden nicht verarbeitet, außer bei Verwendung der Löschfunktion (Delete). Wenn ein katalogisiertes Objekt gelöscht oder ersetzt wird, löscht SYSMAIN immer alle vorhandenen XRef-Daten für dieses Objekt. Dies ist die Standardeinstellung.</td></tr>
<tr><td>Y</td><td>Ja: Alle XRef-Daten werden verarbeitet.</td></tr>
<tr><td>S</td><td>Speziell: Ein angegebenes Objekt wird unabhängig davon verarbeitet, ob es Cross-Referenz-Daten hat oder nicht. Alle vorhandenen XRef-Daten werden verarbeitet.</td></tr>
<tr><td>F</td><td>Force/Erzwingen: Alle XRef-Daten werden verarbeitet und das Objekt muss in Predict dokumentiert sein.</td></tr>
<tr><td>D</td><td>Dokumentiert: Das Objekt muss in Predict dokumentiert sein. Alle vorhandenen XRef-Daten werden verarbeitet.</td></tr>
</table>Weitere Einzelheiten siehe Besonderheiten bei XRef.</td></tr>
</table>

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Library</td><td>Der Name einer Quell- oder Ziel-Library.

Die Quell-Library enthält das zu verarbeitende Objekt. Die Ziel-Library ist die Library, in die das Objekt kopiert oder verschoben werden soll, oder in der das Objekt umbenannt wird.

Siehe auch Auflisten und Auswählen von Libraries.</td></tr>
<tr><td>Database</td><td>Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.

Die Quelldatenbank enthält die Library und die Systemdatei, in der das zu bearbeitende Objekt gespeichert ist. Die Zieldatenbank enthält die Library und die Systemdatei, in die das Objekt kopiert oder verschoben werden soll, oder in der das Objekt umbenannt wird.

Gültige Datenbankkennungen: 1 bis 65535.</td></tr>
<tr><td>File</td><td>Datei: Die Dateinummer (FNR) einer Quell- oder einer Ziel-Systemdatei (FNAT oder FUSER).

Die Quelldatei enthält die Library, in der das zu bearbeitende Objekt gespeichert ist. Die Zieldatei enthält die Library, in die das Objekt kopiert oder verschoben werden soll, oder in der das Objekt umbenannt wird.

Gültige Dateinummern: 1 bis 65535.</td></tr>
<tr><td>Replace</td><td>Ersetzen: Gibt an, ob ein Objekt bei der Benutzung der Funktionen Verschieben (Move), Kopieren (Copy) oder Umbenennen (Rename) ersetzt werden soll:
<table>
<tr><td></td><td></td></tr>
<tr><td>Y</td><td>Ja: Ein in der Zielumgebung vorhandenes Objekt mit demselben Namen wird ersetzt.</td></tr>
<tr><td>N</td><td>Nein: Ein gleichnamiges Objekt, das in der Zielumgebung existiert, wird nicht ersetzt. Dies ist die Standardeinstellung.</td></tr>
<tr><td></td><td></td></tr>
</table>
Siehe auch Option Replace (Ersetzen) verwenden.</td></tr>
<tr><td>Criteria</td><td>Kriterien: Ruft das Fenster Additional Criteria auf, in dem Sie zusätzliche Objektauswahlkriterien angeben können. Mögliche Werte:
<table>
<tr><td></td><td></td></tr>
<tr><td>N</td><td>Nein: Das Fenster Additional Criteria wird nicht aufgerufen. Dies ist die Standardeinstellung.</td></tr>
<tr><td>Y</td><td>Ja: Das Fenster Additional Criteria wird aufgerufen.</td></tr>
<tr><td></td><td></td></tr>
</table>
Siehe auch Angabe zusätzlicher Kriterien.</td></tr>
</table>

Profilparameter RECAT verwenden

Wenn der Profilparameter `RECAT` beim Start der Sitzung auf `ON` gesetzt wurde, wird dies in den Menüs der Programmierobjekte (**Programming Objects**) und in den Auswahlbildschirmen (**Selection**) angezeigt.

Bei der Verwendung von SYSMAIN-Funktionen mit `RECAT=ON` gelten die normalen dynamischen Rekatalogisierungsregeln, wie sie unter *RECAT - Dynamisches Rekatalogisieren* in der *Parameter-Referenz*-Dokumentation beschrieben sind. Dies bedeutet Folgendes:

- Wenn ein Objekt sowohl als gespeichertes Objekt (saved) als auch als katalogisiertes Objekt (cataloged) existiert, können weder das gespeicherte noch das katalogisierte Objekt unabhängig voneinander verarbeitet werden.
- Wenn ein Objekt nur als katalogisiertes Objekt existiert, können Sie mit ihm keine SYSMAIN-Funktion ausführen, die nur katalogisierte Objekte verarbeitet (es erscheint die Fehlermeldung `Invalid request with dynamic recatalog`/Ungültige Anforderung mit dynamischer Rekatalogisierung) oder ein katalogisiertes Objekt auf einem **Selection**-Bildschirm auswählen.

Bei der automatisierten Verarbeitung wird jedes Objekt, das diesen Regeln nicht entspricht, ignoriert und die Verarbeitung mit dem nächsten Objekt fortgesetzt.

Auswahllisten für Programmierobjekte

Wenn die selektive Verarbeitung aktiviert wurde, wird auf einem **Selection**-Bildschirm eine Auswahlliste mit allen Programmierobjekten angezeigt, die den angegebenen Auswahlkriterien entsprechen.

Dieser Abschnitt beschreibt die Spalten und Felder eines **Selection**-Bildschirms und die Zeilenkommandos, die für die weitere Verarbeitung eines Programmierobjekts zur Verfügung stehen:

- Spalten und Felder

- Zeilenkommandos

Spalten und Felder

Die folgenden Spalten und Felder werden auf einem **Selection**-Bildschirm angezeigt:

Spalte/Feld	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Name	Der Name des Programmierobjekts, das den angegebenen Auswahlkriterien entspricht.
Type	Der Code, der dem Typ des Objekts entspricht, wie er in *TYPE-Angabe* aufgeführt ist.
S/C	Das Objekt, das für das Programmierobjekt existiert: ein gespeichertes/Quellcodeobjekt (S) und/oder ein katalogisiertes Objekt (C).
Message	Die Meldung, die den Verarbeitungsstatus eines Programmierobjekts angibt. Mögliche Meldungen: Siehe *Status-Meldungen*.
Listed Library	Aufgelistete Library: Der Name der Library, die das/die ausgewählte(n) Programmierobjekt(e) enthält.

Zeilenkommandos

In der Spalte C (Command) eines **Selection**-Bildschirms kann eines der folgenden Zeilenkommandos eingegeben werden:

Zeilen-kommando	Funktion
A	Jedes in der Spalte **S/C** (Saved/Cataloged) aufgelistete Objekt verarbeiten, d.h. gespeicherte/Quellcodeobjekte (S) und/oder katalogisierte Objekte (C). Dieses Kommando gilt nicht für die Such- (**Find**) oder Auflistfunktion (**List**).
B	Löscht das katalogisierte Objekt aus dem Natural Buffer Pool. Das Löschen des/der angegebenen Objekts/Objekte muss durch die Eingabe von DELETE in einem Fenster bestätigt werden, das erscheint, nachdem Sie das/die Objekt(e) angegeben und ENTER gedrückt haben.
C	Verarbeitet nur das katalogisierte Objekt, auch wenn es ein entsprechendes gespeichertes Objekt gibt. Wenn C für ein Objekt angegeben wird, das nur als gespeichertes Objekt existiert, tritt ein Fehler auf. Dieses Kommando gilt nicht für die Such- (**Find**) oder Auflistfunktion (**List**).
D	Gilt nur bei einem **Library Selection**-Bildschirm. Siehe auch *Auflisten und Auswählen von Libraries*). Zeigt eine Kurzliste der Objekte an, die in der angegebenen Library enthalten sind. Die in der Liste enthaltenen Informationen (Name, Typ, Quellcode-/katalogisiertes Objekt) sind identisch mit den Informationen, die auf dem **Selection**-Bildschirm angezeigt werden, der unter *Selektive Verarbeitung* in *Menüfunktionen und Kommandos benutzen* gezeigt wird.

Zeilen-kommando	Funktion
H	Erstellt eine Hardcopy des gespeicherten (Quellcode-)Objekts. Der Quellcode des angegebenen Objekts wird gedruckt und auf dem Bildschirm angezeigt.
I	Verzeichnisinformationen eines Objekts anzeigen. Dieses Kommando ist vergleichbar mit dem Systemkommando `LIST DIR object-name`. Details zu den angezeigten Verzeichnisinformationen finden Sie unter *Directory-Informationen anzeigen* in der *Systemkommandos*-Dokumentation.
L	Anzeige des Quellcodes eines gespeicherten (Quellcode-)Objekts. Dieses Kommando entspricht dem Systemkommando `LIST object-name`. Zu den Kommandos, die von einem Quellcode-Bildschirm aus ausgeführt werden können, siehe *Source-Liste* in der *Systemkommandos*-Dokumentation.. Ausnahmen: Das Kommando `EXPAND` ist nicht verfügbar. Das Kommando `ZOOM` kann nicht verwendet werden, wenn das anzuzeigende Quellobjekt in einer Steplib Library enthalten ist.
	Auf einem **Library Selection**-Bildschirm (siehe auch ***Auflisten und Auswählen von Libraries***): Sie können eine erweiterte Liste der in der Library enthaltenen Objekte anzeigen. Zusätzlich zu den Informationen, die mit dem Zeilenkommando `D` angezeigt werden, liefert die erweiterte Liste Informationen des Objektverzeichnisses: Programmiermodus, Natural-Version, Benutzerkennung, Datum und Uhrzeit wann gespeichert/katalogisiert.
R	Anzeige des Langnamens eines Natural-Objekts vom Typ Ressource.
	Auf einem **Library Selection**-Bildschirm (siehe auch ***Auflisten und Auswählen von Libraries***): Überprüfen der Verwendung von externen Subroutinen: Zeigt für die angegebene Library und den Objektbereich die Objekte an, für die ein katalogisiertes Objekt existiert, und gibt an, ob ein Objekt auf ein externes Subprogramm verweist. Für jedes Objekt, das auf ein externes Subprogramm verweist, wird der Name des externen Subprogramms und der Name des katalogisierten Objekts angezeigt, das für dieses Subprogramm existiert, falls verfügbar. Für ein Objekt vom Typ Subroutine wird der Name des entsprechenden Alias-Langnamens angezeigt.
S	Verarbeitet nur das gespeicherte (Quellcode-)Objekt, auch wenn es ein entsprechendes katalogisiertes Objekt gibt. Wenn `S` bei einem Objekt angegeben wird, das nur als katalogisiertes Objekt existiert, tritt ein Fehler auf. Dieses Kommando gilt nicht für die Such- (**Find**) oder Auflistfunktion (**List**).
	Auf einem **Library Selection**-Bildschirm (siehe auch ***Auflisten und Auswählen von Libraries***): Auflistung aller in der angegebenen Library enthaltenen externen Subroutinen anhand ihrer Alias-Langnamen und/oder der Kurznamen ihrer entsprechenden katalogisierten Objekte.
X	Gilt nur, wenn Natural Connection und Entire Connection installiert sind. Gespeicherte(s) (Quellcode-)Objekt(e) auf einen PC herunterladen: siehe auch ***Quellcodeobjekte auf einen PC herunterladen***.

Zeilen-kommando	Funktion
Z	Größen berechnen: Überprüft die Größen der gespeicherten (Quellcode-)Objekte und der katalogisierten Objekte, z. B. die DATSIZE-, ESIZE- und MCG-Größe.

Quellcodeobjekte auf einen PC herunterladen

Die Download-Option gilt nur, wenn Natural Connection und Entire Connection installiert sind.

Um ein oder mehrere Quellobjekte herunterzuladen:

1 Bevor Sie SYSMAIN aufrufen:

- Definieren Sie die Arbeitsdateien (Work Files) 6 und 7 als PC-Arbeitsdateien.
- Aktivieren Sie nach dem Sitzungsstart die PC-Verbindung durch Eingabe des folgenden Terminalkommandos:

```
%+
```

(Siehe auch *%+ und %- - Natural Connection ein-/ausschalten* in der *Terminalkommandos*-Dokumentation).

2 Rufen Sie einen beliebigen **Selection**-Bildschirm auf.

3 Geben Sie neben dem oder den Objekten, die Sie herunterladen möchten, das folgende Zeilenkommando ein:

```
X
```

Drücken Sie ENTER.

Es erscheint ein **PC Download Options**-Fenster ähnlich dem folgenden Beispiel:

```
 13:23:15            ***** NATURAL SYSMAIN UTILITY *****           2005-08-10
 User SAG                    -    Copy   Selection  -

COPY ALL * WITH XREF N FROM OLDLIB WHERE DBID 10 FNR 32 TO NEWLIB WHERE
DBID 10 FNR 32

 C  Name     Type   S/C  Message        C  Name     Type   S/C  Message
 -  -------- ------ ---  ------------   -  -------- ------ ---  ------------
 X  SUB1     Subpgm S                    X  SUB2     Subpgm S/C
 X  SUB3     Subpgm S/C                  X  SUB4     Subpgm S
 X  SUB5     Subpgm S/C                  X  SUB6     Subpgm S/C
+----------------- PC Download Options ------------------+ S/C
!   Specify the relevant PC options                      ! S/C
!                                                        ! S
! Drive ...... C                                        ! S/C
! Path ....... SOURCES/SUBPROGRAMS                      ! S/C
! Extension .. NS*                                      ! S
!                                                        ! S
!                                                        ! S
!                                                        ! ed Library: OLDLIB
! Warning: This will overwrite any existing objects with !
!          the same path, name and extension.            ! Exit) _
+--------------------------------------------------------+ -PF10--PF11--PF12---
      Help  Menu  Exit  Copy  Del   Find  List  Move  Ren              Canc
```

Geben Sie in das Feld **Drive** (Laufwerk) den Namen des PC-Laufwerks ein, auf das das Objekt bzw. die Objekte heruntergeladen werden sollen. Die Standardeinstellung ist `C`.

Geben Sie in das Feld **Path** (Pfad) den Namen des PC-Verzeichnisses/Unterverzeichnisses ein, in das das oder die Objekt(e) heruntergeladen werden sollen (im obigen Beispiel das Unterverzeichnis `SUBPROGRAMS` im Verzeichnis `SOURCES`). Geben Sie einen Schrägstrich (/) als Trennzeichen zwischen einem Verzeichnis und einem Unterverzeichnis sowie zwischen Unterverzeichnissen ein. Wenn das angegebene Verzeichnis/Unterverzeichnis nicht existiert, erhalten Sie eine entsprechende Fehlermeldung.

Geben Sie im Feld **Extension** (Erweiterung) die Erweiterung der Textdatei an, in die der Quellcode des Objekts geladen werden soll. Wenn Sie `NS*` angeben, wird der Stern (*) durch den angegebenen Objekttyp ersetzt (siehe ***TYPE-Angabe***).

Beispiel: Ein Subprogramm mit dem Namen `SUB1` wird in eine Datei mit dem Namen `SUB1.NSN` geladen.

4 Drücken Sie ENTER, um das oder die Objekte zu laden.

In der Nachrichtenzeile erscheint die folgende Meldung: `4824:Requested option(s) processed successfully` (Angeforderte Optionen erfolgreich ausgeführt).

Wenn die Spalte **Message** (Meldung) auf dem Bildschirm angezeigt wird, erscheint neben dem bzw. den heruntergeladenen Objekten außerdem die Meldung `Exported` (exportiert).

5 Falls gewünscht, können Sie mit dem Herunterladen weiterer Objekte aus dieser Auswahlliste fortfahren.

Das Fenster PC **Download Options** wird für weitere Downloads vom aktuellen **Selection**-Bildschirm nicht mehr angezeigt. Dieses Fenster erscheint nur, wenn das Zeilenkommando `X` zum ersten Mal nach dem Aufrufen eines **Selection**-Bildschirms oder nach der Ausführung des Direktkommandos `SET PC` abgesetzt wird.

Die Einstellungen im Fenster **PC Download Options** bleiben aktiv, bis Sie SYSMAIN beenden.

Besonderheiten bei XRef

Alle Cross-Referenz-Daten (XRef), die in der Predict-Systemdatei gespeichert sind, können mit SYSMAIN verarbeitet werden. Die Option **XREF** gibt an, ob SYSMAIN XRef-Daten verarbeiten soll. XRef-Daten werden immer gelöscht, wenn die Funktion **Delete** oder **Replace** für ein katalogisiertes Objekt ausgeführt wird.

Wenn Predict nicht installiert wurde, setzen Sie die Option **XREF** auf `N`, so dass keine Validierung von Predict-Dateien durchgeführt wird. Wenn die Option **XREF** auf `Y` gesetzt ist und die verwendete(n) FDIC-Datei(en) keine gültige(n) Predict-Datei(en) ist (sind), wird eine Fehlermeldung zurückgegeben.

Die Regeln für das Setzen der **XREF**-Option sind die gleichen wie die von Natural Security geforderten. In einer Umgebung ohne Natural Security gibt es jedoch keine Einschränkungen.

Es gibt folgende Einstellungen für **XREF**:

- XREF auf N gesetzt
- XREF auf Y oder F gesetzt
- XREF auf F gesetzt
- XREF auf S gesetzt
- XREF auf D gesetzt
- XREF-Fehler

XREF auf N gesetzt

Wenn die Option **XREF** auf `N` gesetzt ist, werden keine XRef-Daten verarbeitet, aber in Situationen, in denen ein katalogisiertes Programmierobjekt gelöscht oder ersetzt wird, löscht SYSMAIN die XRef-Daten. Die Ziel-Predict-Systemdatei wird entsprechend den aktuellen Einstellungen der Quell- oder Ziel-FDIC-Systemdatei bestimmt. Die Voreinstellung ist der Wert, der dem Profilparameter `FDIC` (siehe *FDIC - Predict-Systemdatei* in der *Parameter-Referenz*-Dokumentation) zu Beginn der Natural-Sitzung zugewiesen wurde.

XREF auf Y oder F gesetzt

Wenn die Option **XREF** auf `Y` oder `F` gesetzt ist, werden während der Verarbeitung die folgenden Aktionen durchgeführt:

- SYSMAIN prüft, ob XRef-Daten bereits in der Quell-Predict-Systemdatei vorhanden sind.
- Wenn die Option **Replace** aktiv ist (auf `Y` gesetzt) und ein Programmierobjekt aus der Zielumgebung gelöscht werden soll, werden die XRef-Daten in der Ziel-Predict-Systemdatei gelöscht.
- Wenn ein Programmierobjekt in eine neue Umgebung kopiert wird, werden die XRef-Daten des Programmierobjekts aus der Quell-Predict-Systemdatei in die Ziel-Predict-Systemdatei kopiert. Der Library-Name wird entsprechend geändert und im Falle der **Rename**-Funktion wird auch der Objektname geändert.
- Wurde die **Move**-Funktion (Verschieben) angefordert, werden die XRef-Daten des Programmierobjekts aus der Quell-Predict-Systemdatei gelöscht.

XREF auf F gesetzt

Wenn die Option **XREF** auf `F` gesetzt ist, prüft SYSMAIN zusätzlich, ob für das Programmierobjekt (nur Programm, Subroutine, Subprogramm, Map oder Helproutine) ein Predict-Programmeintrag in der Ziel-Predict-Systemdatei definiert ist. Wenn nicht, wird die Verarbeitung des Objekts abgebrochen.

XREF auf S gesetzt

Wenn die Option **XREF** auf `S` gesetzt ist, gilt der Sonderfall, dass eine Reihe von angegebenen Objekten mit entsprechenden XRef-Daten verarbeitet wird, unabhängig davon, ob alle Objekte Cross-Referenzdaten haben oder nicht: Die Objekte, die Cross-Referenzdaten haben, werden mit ihren Cross-Referenzdaten verarbeitet, und die Objekte, die keine haben, werden ebenfalls verarbeitet.

XREF auf D gesetzt

Wenn die Option **XREF** auf `D` gesetzt ist, prüft SYSMAIN, ob für das Programmierobjekt (nur Programm, Subroutine, Subprogramm, Map oder Helproutine) ein Predict-Programmeintrag in der Ziel-Predict-Systemdatei definiert ist. Ist dies nicht der Fall, wird die Verarbeitung des Objekts abgebrochen. Objekte, die Cross-Referenzdaten haben, werden mit ihren Cross-Referenzdaten verarbeitet, und die Objekte ohne Cross-Referenzdaten werden ebenfalls verarbeitet.

XREF-Fehler

Wenn bei der SYSMAIN-Verarbeitung von XRef-Daten eine der folgenden Inkonsistenzen auftritt, wird die gesamte Verarbeitung für das Objekt oder die Funktion abgebrochen und eine Fehlermeldung angezeigt:

- Der Wert der Option **XREF** in Natural Security ist `F` oder `Y`, und Sie haben den Wert `Y` bzw. `N` angegeben.
- Die Option **XREF** ist auf `F` oder `D` gesetzt, und SYSMAIN findet in Predict keinen dokumentierten Programmeintrag für das bearbeitete Objekt.
- Es wurde eine ungültige Predict-Datei angegeben.
- Der Wert der Option **XREF** in Natural Security ist `F`, `D` oder `Y` und Sie haben den Wert `S` angegeben.
- Der Wert der Option **XREF** in Natural Security ist `D` und Sie haben den Wert `N` angegeben.

Angabe zusätzlicher Kriterien

Zusätzlich zu den Auswahlkriterien, die in den Eingabefeldern eines **Programming Object**-Menüs angegeben sind, können Sie Objekte nach Datum/Uhrzeit, Benutzerkennung und Terminalkennung auswählen, das bzw. die sich auf ihre Speicherung oder Katalogisierung bezieht.

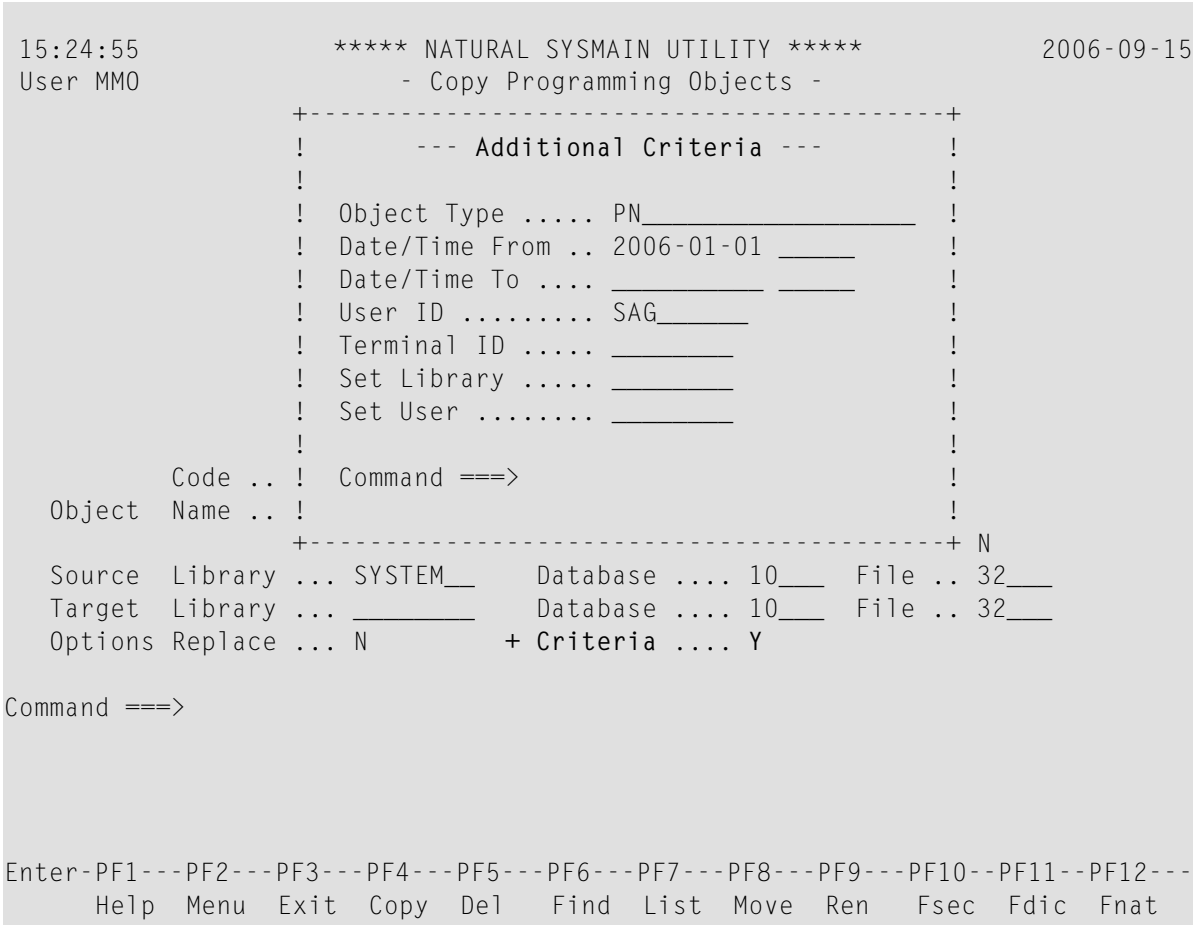
Sie können auch die Benutzerkennung und die Library für einen Predict Set angeben. Diese Option gilt nicht für die **List**- und **Find**-Funktionen.

Sie können z.B. nur die Objekte auswählen, die an einem bestimmten Tag zwischen 8:00 und 12:00 Uhr von einem bestimmten Benutzer an einem bestimmten Terminal katalogisiert wurden, d.h. die Verarbeitung der Objekte nach den Auswahlkriterien basiert auf allen ausgewählten Kriterien als Ganzes, nicht auf jeder einzelnen Bedingung.

Um zusätzliche Auswahlkriterien anzugeben:

1 Ersetzen Sie im Feld **Criteria** des Menüs **Programming Objects** den Wert `N` (Standardeinstellung) durch `Y`.

  Es erscheint das Fenster **Additional Criteria**, zum Beispiel:



Geben Sie die gewünschten Auswahlkriterien ein. Wenn im Fenster **Additional Criteria** im Feld **Object Type** ein oder mehrere Objekttyp-Codes eingegeben wurden, ist das Feld **Type** im Menü **Programming Objects** mit denselben Objekttyp-Codes vorbelegt.

2 Wenn Sie zum Menü **Programming Objects** zurückkehren, zeigt ein Pluszeichen (+) vor dem Feld **Criteria** an, dass zusätzliche Objektauswahlkriterien angegeben wurden. Im obigen Beispiel die Benutzerkennung `SAG`.

Das Pluszeichen (+) wird nicht angezeigt, wenn nur das Feld **Object Type** einen Eintrag enthält, da dieser Eintrag bereits im Feld **Type** im Menü **Programming Objects** erscheint.

Direktkommando-Syntax für Programmierobjekte

In diesem Abschnitt wird die Syntax erläutert, die bei der Ausführung einer SYSMAIN-Funktion bei Programmierobjekten unter Verwendung von Direktkommandos im Online- oder Batch-Modus gilt. Allgemeine Anleitungen zur Verwendung von Direktkommandos finden Sie unter ***Kommandos ausführen***.

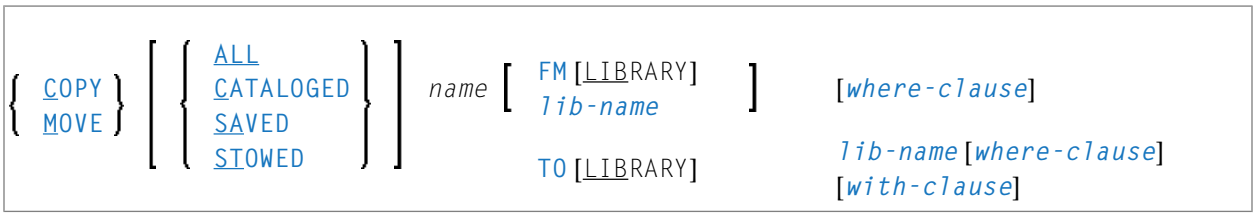
Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den nachfolgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel (*with-clause*) ist bei allen Befehlen identisch.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- FIND, LIST und LISTLIB
- RENAME
- where-clause
- with-clause

COPY und MOVE



Beispiele:

```
COPY PROG1 FM TESTORD TO ORDERS DBID 1 FNR 6 REP

C PGM* WITH REP TYPE PNS FM PRODLIB TO TESTLIB

M PROG1 FM OLDLIB TO NEWLIB

MOVE STOWED * TO NEWLIB WHERE DBID 100 FNR 160 FMDATE 2007-01-01 FM OLDLIB WITH ↩
XREF Y
```

DELETE

```
DELETE [ { ALL       } ]  name [ IN [LIBRARY]  ] [where-clause] [with-clause]
       [ { CATALOGED }  ]       [ lib-name    ]
       [ { SAVED     }  ]
       [ { STOWED    }  ]
```

Beispiele:

```
DELETE C M> IN LIB ORDERS

D * IN TESTLIB DBID 1 FNR 5 NAME SYSNAT

D SA * IN LIBTEST TYPE GLA

D * TYPE PM IN TESTORD FMDATE 2007-01-01 TODATE 2007-04-30
```

FIND, LIST und LISTLIB

```
{ FIND    } [ { ALL       } ]  name [ IN [LIBRARY]  ] [where-clause] [with-clause]
{ LIST    }   { CATALOGED }  ]       [ lib-name    ]
{ LISTLIB }   { SAVED     }  ]
              { STOWED    }  ]
```

Anmerkung: Das Direktkommando `LISTLIB` ist nur im Batch-Modus verfügbar und dient dazu, eine Liste von Library-Namen zu erhalten.

Beispiele:

```
FIND SAVED MENU IN TESTLIB

FIND STOWED MAINMENU IN SYS* WHERE DBID 1 FNR 5

F ALL PROG2 IN PROD* FNR 27 DBID 1

LIST * IN TESTLIB

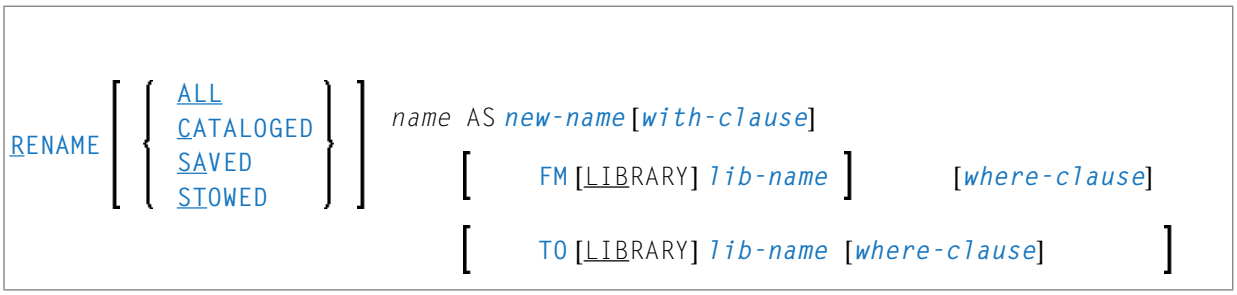
LIST DT* IN TESTLIB

L SAVED TEST* IN TESTLIB TYPE PNS FNR 6

L SA TEST* TYPE PM IN TESTLIB FNR 6 DBID 2 FMDATE 2007-01-01

LISTLIB ALL MENU IN SYS* DBID 10 FNR 44
```

RENAME



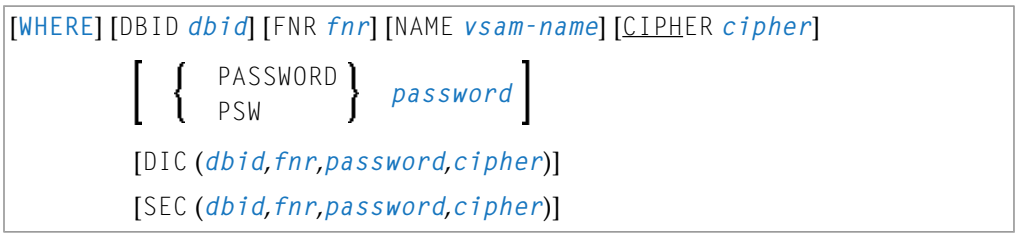
Beispiele:

```
RENAME PGM1 AS PROG1 FM TESTLIB

R PGM1 AS PROG1 FM TESTLIB DBID 1 FNR 5 TO PRODLIB DBID 2 FNR 6

R PGM* TYPE PS RCOP FM TESTLIB TO PRODLIB
```

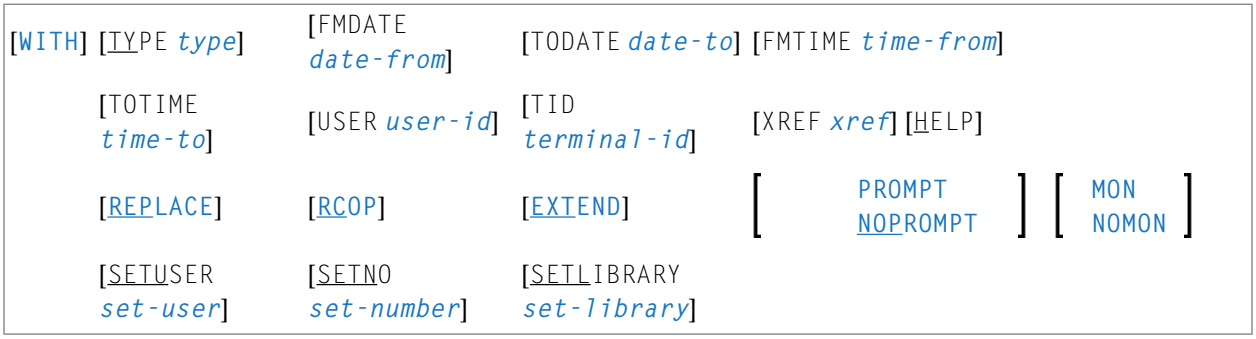
where-clause



Trennzeichen

Kommas müssen als Trennzeichen zwischen den Werten nach dem Schlüsselwort `SEC` verwendet werden, oder wenn ein Wert fehlt. Zum Beispiel: `SEC (10,,secret,2a)`. Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf Komma (`,`) gesetzt wurde, verwenden Sie einen Schrägstrich (`/`) als Trennzeichen zwischen den Werten.

with-clause



73 Debug-Umgebungen bearbeiten

- Felder im Menü Debug Environments 710
- Auswahllisten für Debug-Umgebungen 712
- Direktkommando-Syntax für Debug-Umgebungen 713

Alle SYSMAIN-Funktionen, mit Ausnahme der Funktion **Find**, können in Debug-Umgebungen ausgeführt werden.

Die Angabe der Debug-Umgebung muss immer mit der Datenbankkennung (DBID) und der Dateinummer (FNR) der entsprechenden FUSER-Systemdatei übereinstimmen.

Dieses Kapitel beschreibt die Menüfunktionen und Auswahllistenoptionen, die zur Ausführung einer SYSMAIN-Funktion in Debug-Umgebungen zur Verfügung stehen, sowie die Syntax, die bei der Verwendung von Direktkommandos gilt.

Verwandtes Thema:

Debug Environment Maintenance (Verwaltung der Debug-Umgebung) in der *Debugger*-Dokumentation

Felder im Menü Debug Environments

Das Menü **Debug Environments** enthält alle SYSMAIN-Funktionen, die für die Bearbeitung von Debug-Umgebungen erforderlich sind. Die Felder in diesem Menü sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung
Code	Gibt die auszuführende Funktion an. Siehe ***Beschreibung der Funktionen***:
C	Copy - Debug-Umgebung kopieren.
D	Delete - Debug-Umgebung löschen.
L	List - Debug-Umgebung auflisten.
M	Move - Debug-Umgebung verschieben.
R	Rename - Debug-Umgebung umbenennen.
	Anmerkung: Wenn eine Debug-Umgebung von einer Library in eine andere verschoben oder kopiert wurde, müssen die Breakpoints und Watchpoints an die neue Library angepasst werden. Einzelheiten siehe *Debug-Umgebung in verschiedenen Libraries verwalten* in der *Debugger*-Dokumentation.
Sel. List	Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird:
Y	Ja: Die selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von Debug-Umgebungen wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten, Felder und Zeilenkommandos in einer Auswahlliste zur Verfügung stehen, erfahren Sie unter ***Auswahllisten für Debug-Umgebungen***.
N	Nein: Die selektive Verarbeitung ist ausgeschaltet. Name

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Name</td><td>Der Name der zu bearbeitenden Debug-Umgebung oder ein Bereich von Namen: siehe auch Angabe eines Namensbereichs.

Die Standardeinstellung ist ein Stern (*), was bedeutet, dass alle Namen ausgewählt werden.</td></tr>
<tr><td>New Name</td><td>Neuer Name: Der Name, der einer Debug-Umgebung gegeben werden soll, wenn sie mit der Funktion Rename umbenannt wird.</td></tr>
<tr><td>Library</td><td>Der Name einer Quell- oder Ziel-Library.

Die Quell-Library enthält die zu bearbeitende Debug-Umgebung. Die Ziel-Library ist die Library, in die die Debug-Umgebung kopiert oder verschoben werden soll, oder in die die Debug-Umgebung umbenannt wird.

Siehe auch Auflisten und Auswählen von Libraries.</td></tr>
<tr><td>Database</td><td>Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.

Die Quelldatenbank enthält die Library und die Systemdatei, in der die zu bearbeitende Debug-Umgebung gespeichert ist. Die Zieldatenbank enthält die Library und die Systemdatei, in die die Debug-Umgebung kopiert oder verschoben werden soll, oder in der die Debug-Umgebung umbenannt wird.

Gültige Datenbankkennungen: 1 bis 65535.</td></tr>
<tr><td>File</td><td>Datei: Die Dateinummer (FNR) einer Quell- oder einer Ziel-Systemdatei (FNAT oder FUSER).

Gültige Dateinummern: 1 bis 65535.

Die Quelldatei enthält die Library, in der die zu bearbeitende Debug-Umgebung gespeichert ist. Die Zieldatei enthält die Library, in die die Debug-Umgebung kopiert oder verschoben werden soll, oder in der die Debug-Umgebung umbenannt wird.</td></tr>
<tr><td>Replace</td><td>Ersetzen: Gibt an, ob eine Debug-Umgebung bei der Verwendung der Funktionen Move (Verschieben), Copy (Kopieren) oder Rename (Umbenennen) ersetzt werden soll:
<table>
<tr><td></td><td></td></tr>
<tr><td>Y</td><td>Ja: Eine Debug-Umgebung mit demselben Namen, die in der Zielumgebung existiert, wird ersetzt.</td></tr>
<tr><td>N</td><td>Nein: Eine Debug-Umgebung mit demselben Namen, die in der Zielumgebung existiert, wird nicht ersetzt. Dies ist die Standardeinstellung.</td></tr>
</table>
Siehe auch Option Replace (Ersetzen) verwenden.</td></tr>
</table>

Auswahllisten für Debug-Umgebungen

Wenn die selektive Verarbeitung aktiviert wurde, wird auf einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen Debug-Umgebungen angezeigt, die den angegebenen Auswahlkriterien entsprechen.

Dieser Abschnitt beschreibt die auf einem Auswahlbildschirm enthaltenen Spalten und Felder sowie die Zeilenkommandos, die zur weiteren Bearbeitung einer Debug-Umgebung zur Verfügung stehen:

- Spalten und Felder
- Zeilenkommandos

Spalten und Felder

Die folgenden Spalten und Felder werden auf einem Auswahlbildschirm angezeigt:

Spalte/Feld	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Environment	Umgebung: Der Name der Debug-Umgebung, die den angegebenen Auswahlkriterien entspricht.
Message	Meldung: Die Meldung, die den Verarbeitungsstatus einer Debug-Umgebung angibt. Mögliche Meldungen siehe *Status-Meldungen*.
Listed Library	Aufgelistete Library: Der Name der Library, die die ausgewählte(n) Debug-Umgebung(en) enthält.

Zeilenkommandos

Eines der folgenden Zeilenkommandos kann in der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) eingegeben werden:

Zeilen-kommando	Funktion
A	Verarbeitet die Debug-Umgebung. Dieses Zeilenkommando ist auf dem **List Selection**-Bildschirm nicht verfügbar. Auf einem **List Selection**-Bildschirm können Sie nur einen Punkt (.) eingeben, um den Bildschirm zu verlassen.
L	Zeigt eine Liste der Debug-Umgebungen an, die in der angegebenen Library enthalten sind. Gilt nur bei einem **Library Selection**-Bildschirm (siehe auch *Auflisten und Auswählen von Libraries*).

Direktkommando-Syntax für Debug-Umgebungen

In diesem Abschnitt wird die Syntax für die Ausführung von SYSMAIN-Funktionen in Debug-Umgebungen mit Hilfe von Direktkommandos im Online- oder Batch-Modus beschrieben. Allgemeine Anleitungen zur Verwendung von Direktkommandos siehe ***Schlüsselwörter und Variablen in Direktkommandos***.

Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel (*with-clause*) ist bei allen Kommandos identisch.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- LIST
- RENAME
- where-clause
- with-clause

COPY und MOVE

```
{ COPY }
{ MOVE } DEBUG   name  FM [LIBRARY] lib-name [where-clause]
                       TO [LIBRARY] lib-name [where-clause] [with-clause]
```

Beispiele:

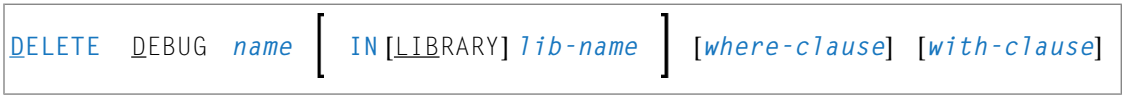
```
COPY D ENV FM TESTLIB WHERE DBID 1 FNR 5 TO PRODLIB WHERE DBID 2 FNR 5 WITH REP

C DEBUG ENV FM TESTLIB FNR 6 TO PRODLIB FNR 7 REP

MOVE DEBUG ENV FM OLDLIB WHERE DBID 1 FNR 5 TO NEWLIB WHERE DBID 2 FNR 5

M DEBUG ENV FM OLDLIB FNR 6 TO NEWLIB FNR 7 REP
```

DELETE

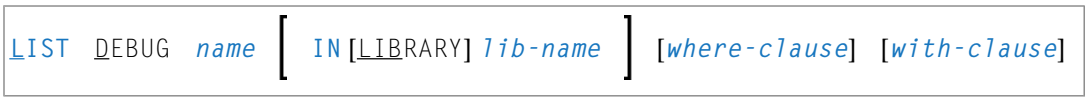


Beispiele:

```
DELETE DEBUG U* IN TESTLIB FNR 150
```

```
D DEBUG TEST* IN TESTLIB IN DBID 177 FNR 205
```

LIST

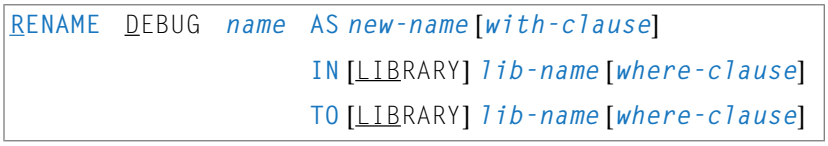


Beispiele:

```
LIST DEBUG ENV* IN TESTLIB DBID 1 FNR 5
```

```
L D DT* IN TESTLIB DBID 10
```

RENAME



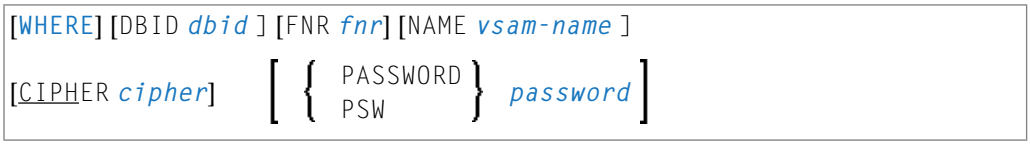
Beispiele:

```
RENAME D OLDENV AS NEWENV IN TESTLIB RCOP
```

```
R DEBUG OLDENV AS NEWENV IN TESTLIB DBID 1 FNR 4 TO PRODLIB DBID 1 FNR 5
```

```
R DEBUG OLDENV AS NEWENV IN TESTLIB FNR 4 TO PRODLIB FNR 5 REPLACE RCOP
```

where-clause



with-clause



74 Fehlermeldungen bearbeiten

- Felder in den Menüs für Fehlermeldungen 718
- Auswahllisten für Fehlermeldungen 720
- Fehlermeldungen neu nummerieren 722
- Sprachen angeben 723
- Direktkommando-Syntax für Fehlermeldungen 723

Alle SYSMAIN-Funktionen können bei benutzerdefinierten und bei Natural-System-Fehlermeldungen angewendet werden. Fehlermeldungen können von einer Library in eine andere verschoben werden, oder die Sprachtexte der einzelnen Fehlermeldungen können kopiert (**Copy**), verschoben (**Move**) oder ersetzt (**Replace**) werden. Darüber hinaus ist es möglich, eine einzelne Fehlermeldung oder einen Bereich von Fehlermeldungen mit Hilfe der Umbenennungsfunktion (**Rename**) neu zu nummerieren.

Wenn Natural Security installiert ist, dürfen nur autorisierte Benutzer die Fehlermeldungen des Natural-Systems bearbeiten.

Fehlermeldungen werden je nach Typ in den Systemdateien gespeichert:

- Fehlermeldungen des Natural-Systems werden in der Systemdatei FNAT gespeichert.
- Benutzerdefinierte Fehlermeldungen in der Systemdatei FUSER oder FNAT gespeichert.

Dieser Abschnitt beschreibt die Menüfunktionen und Auswahllistenoptionen, die für die Ausführung einer SYSMAIN-Funktion bei Fehlermeldungen zur Verfügung stehen, sowie die Syntax, die bei der Benutzung von Direktkommandos gilt.

Verwandtes Thema:

SYSERR Utility-Dokumentation.

Felder in den Menüs für Fehlermeldungen

Die **Error Message Texts**-Menüs enthalten alle SYSMAIN-Funktionen, die für die Bearbeitung von Fehlermeldungen erforderlich sind. Die Felder, die in einem Menü vorhanden sein können, sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung	
Code	Gibt an, ob eine kurze und/oder eine erweiterte (lange) Fehlermeldung verarbeitet werden soll. Mögliche Werte:	
	A	Jede Fehlermeldung, für die eine kurze und/oder eine erweiterte (lange) Meldung vorliegt, wird verarbeitet.
	E	Jede Fehlermeldung, zu der eine erweiterte (lange) Fehlermeldung vorliegt, wird verarbeitet. Es werden nur erweiterte Fehlermeldungen mit zugehörigen kurzen Fehlermeldungen verarbeitet. Eine erweiterte Fehlermeldung kann nicht in eine Zielumgebung übertragen werden, wenn keine entsprechende kurze Fehlermeldung in der Zielumgebung vorhanden ist.

Feld	Erläuterung	
	S	Jede Fehlermeldung, zu der eine kurze Fehlermeldung vorhanden ist, wird verarbeitet.
Sel. List	Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird:	
	Y	Ja: Selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von Fehlermeldungen wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten und Zeilenkommandos auf einer Auswahlliste zur Verfügung stehen, erfahren Sie unter *Auswahllisten für Fehlermeldungen*.
	N	Nein: Die selektive Verarbeitung ist ausgeschaltet.
No. From or **Number**	Nummer von oder Nummer: Die Nummer der zu verarbeitenden Fehlermeldung bzw. die Anfangsnummer eines Nummernkreises, wenn im Feld **No. To** eine Endnummer angegeben ist. Siehe auch *Fehlermeldungen neu nummerieren*.	
No. To	Nummer bis: Die Endnummer eines zu bearbeitenden Nummernbereichs von Fehlermeldungen, wenn im Feld **No. From** eine Startnummer eingegeben wird. Siehe auch *Fehlermeldungen neu nummerieren*.	
New From	Neu von: Gilt für die Funktion **Rename** (Umbenennen). Die neue Nummer, die einer bestehenden Fehlermeldung gegeben werden soll, oder die Startnummer eines Bereichs neuer Nummern, die einem Bereich bestehender Fehlermeldungen gegeben werden sollen. Siehe auch *Fehlermeldungen neu nummerieren*.	
New To	Neu bis: Gilt für die Funktion **Rename** (Umbenennen). Die Endnummer eines Bereichs neuer Fehlermeldungsnummern, die einem Bereich vorhandener Fehlermeldungen zugewiesen werden. Siehe auch *Fehlermeldungen neu nummerieren*.	
Library	Der Name einer Quell- oder Ziel-Library oder eines Bereichs von Namen. Wenn Sie Natural System-Fehlermeldungen verarbeiten wollen, lassen Sie das Feld **Library** leer. Die Quell-Library enthält die zu verarbeitende(n) Fehlermeldung(en). Die Ziel-Library ist die Library, in die die Fehlermeldung kopiert oder verschoben werden soll, oder in der die Fehlermeldung neu nummeriert wird. Siehe auch *Auflisten und Auswählen von Libraries*.	
Lang.	Sprache: Der/die Code(s) der Sprache(n), in dem/denen die Fehlermeldung verfügbar ist. Die Sprachen können durch eine beliebige Kombination von Sprachcodes angegeben werden. Informationen darüber, welcher Sprachcode welcher Sprache zugeordnet ist, finden Sie unter *Sprachcode-Zuordnungen* in **LANGUAGE* in der *Systemvariablen*-Dokumentation. Für die erweiterten (langen) Fehlermeldungen des Natural-Systems ist nur der Sprachcode 1 verfügbar. Geben Sie einen Stern (*) ein, um Fehlermeldungen in allen vorhandenen Sprachen auszuwählen. Siehe auch *Sprachen angeben*.	

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Database</td><td>Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.

Die Quelldatenbank enthält die Library und die Systemdatei, in der die zu bearbeitende Fehlermeldung gespeichert ist. Die Zieldatenbank enthält die Library und die Systemdatei, in die die Fehlermeldung kopiert (Copy) oder verschoben (Move) werden soll, oder in der die Fehlermeldung neu nummeriert (Rename) wird.

Gültige Datenbankkennungen: 1 bis 65535.</td></tr>
<tr><td>File</td><td>TDatei: Die Dateinummer (FNR) einer Quell- oder Zielsystemdatei.

Gültige Dateinummern: 1 bis 65535.

Die Quelldatei enthält die Library, in der die zu bearbeitende Fehlermeldung gespeichert ist. Die Zieldatei enthält die Library, in die die Fehlermeldung kopiert (Copy) oder verschoben (Move) werden soll, oder in der die Fehlermeldung neu nummeriert (Rename) wird.</td></tr>
<tr><td>Replace</td><td>Ersetzen: Gibt an, ob eine Fehlermeldung bei der Funktion Verschieben (Move), Kopieren (Copy) oder (Rename) ersetzt werden soll:
<table><tr><td></td><td></td></tr><tr><td>Y</td><td>Ja: Eine Fehlermeldung mit derselben Nummer, die in der Zielumgebung vorhanden ist, wird ersetzt.</td></tr><tr><td>N</td><td>Nein: Eine Fehlermeldung mit der gleichen Nummer, die in der Zielumgebung vorhanden ist, wird nicht ersetzt. Dies ist die Standardeinstellung.</td></tr><tr><td></td><td></td></tr></table>Siehe auch Option Replace (Ersetzen) verwenden.</td></tr>
</table>

Auswahllisten für Fehlermeldungen

Wenn die selektive Verarbeitung aktiviert ist, wird auf einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen Fehlermeldungen angezeigt, die den angegebenen Auswahlkriterien entsprechen.

Dieser Abschnitt beschreibt, welche Spalten ein Auswahlbildschirm enthält und welche Zeilenkommandos zur Weiterverarbeitung einer Fehlermeldung zur Verfügung stehen:

- Spalten

- Zeilenkommandos

Spalten

Die folgenden Spalten werden auf einem Auswahlbildschirm (**Selection**) angezeigt:

Spalte	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Error	Fehler: Der Name der Fehlermeldung, die den angegebenen Auswahlkriterien entspricht.
Error Message Text	Fehlermeldungstext: Der Text der kurzen Fehlermeldung.
Type	Der Typ der Fehlermeldung: `S` (kurz) und/oder `E` (erweitert/lang).
Lang	Sprache: Der erste Sprachcode in fortlaufender oder alphabetischer Reihenfolge, der für die Fehlermeldung existiert. Wenn Sie auf einem Auswahlbildschirm (**Selection**) alle zu einer Fehlermeldung existierenden Sprachen anzeigen wollen, geben Sie das Zeilenkommando `L` ein oder verwenden Sie die Auflistfunktion (**List**).
Message	Meldung: Die Meldung, die den Bearbeitungsstatus einer Fehlermeldung angibt. Diese Spalte erscheint nur, wenn ein Zeilenkommando bei einer Fehlermeldung ausgeführt wurde. Mögliche Meldungen siehe *Status-Meldungen*.

Zeilenkommandos

In der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) kann eines der folgenden Zeilenkommandos eingegeben werden:

Zeilenkommando	Funktion
`A`	Verarbeitet jeden in der Spalte **Type** aufgeführten Meldungstyp. Der Typ kann sein: kurz (`S`), erweitert/lang (`E`) oder kurz und erweitert (`S/E`).
`E`	Verarbeitet die erweiterte (lange) Meldung, wenn die entsprechende kurze Meldung in der Zielumgebung vorhanden ist. Wenn `E` bei einer Meldung angegeben wird, die nur als Kurzmeldung existiert, wird ein Fehler zurückgegeben.
`S`	Verarbeitet alle Kurzmeldungen.
`L`	Überprüfen einer Fehlermeldung, bevor sie verarbeitet wird. Die kurze und/oder erweiterte (lange) Fehlermeldung wird für alle vorhandenen Sprachen angezeigt, je nach der zuvor im Menü **Error Message Texts** (Fehlermeldungstexte) angegebenen Funktion. Dieses Zeilenkommando kann auch auf einem **Library Selection**-Bildschirm verwendet werden (siehe auch *Auflisten und Auswählen von Libraries*).

Fehlermeldungen neu nummerieren

Sie können eine einzelne Fehlermeldung oder einen Bereich von Fehlermeldungen neu nummerieren.

Um eine einzelne Fehlermeldung neu zu nummerieren:

1 Geben Sie in das Feld **No. From** die Nummer der Fehlermeldung ein, die Sie neu nummerieren möchten, und geben Sie in das Feld **New From** die neue Fehlermeldungsnummer ein.

2 Geben Sie in das Feld **Source Library** den Namen der Library ein, die die umzunummerierende Fehlermeldung enthält.

Soll die umnummerierte Fehlermeldung in einer anderen Library abgelegt werden, geben Sie in das Feld **Target Library** den Library-Namen ein.

Um einen Bereich von Fehlermeldungen neu zu nummerieren:

1 Geben Sie in das Feld **No. From** die Anfangsnummer des neu zu nummerierenden Bereichs von Fehlermeldungen und in das Feld **No. To** die Endnummer ein.

Geben Sie in das Feld **New From** die Anfangsnummer eines neuen Bereichs von Fehlermeldungen und in das Feld **New To** die Endnummer ein.

Die Anzahl der Fehlermeldungen, die durch den Bereich in den Feldern **No. From** und **Nr. To** der Quell-Library angegeben wird, muss mit der Anzahl des Bereichs in den Feldern **New From** und **New To** der Ziel-Library übereinstimmen. Es ist beispielsweise nicht möglich, die Fehlermeldungsnummern `1 - 6` als neue Fehlermeldungsnummern `7 - 10` zu nummerieren.

Wenn Sie einen Bereich von Fehlermeldungen innerhalb einer einzigen Library neu nummerieren möchten, dürfen sich die Werte des Bereichs nicht überschneiden. Es ist beispielsweise nicht möglich, die Fehlermeldungsnummern `1 - 6` als neue Meldungsnummern `5 - 10` zu nummerieren.

2 Geben Sie in das Feld **Source Library** den Namen der Library ein, die die neu zu nummerierenden Fehlermeldungen enthält.

Wenn die neu nummerierten Fehlermeldungen in einer anderen Library abgelegt werden sollen, geben Sie einen Namen in das Feld **Target Library** ein.

Siehe auch die Beispiele für Bereichsangaben unter ***RENAME*** in *Direktkommando-Syntax für Fehlermeldungen*.

Sprachen angeben

Bei der Angabe von Sprachen ist Folgendes zu beachten:

- Wenn die Auflistungsfunktion (**List**) ausgeführt wird und der angegebene Sprachcode bei der Fehlermeldung nicht existiert, verwendet SYSMAIN die in der Systemvariablen `*LANGUAGE` angegebene Standardsprache (siehe **LANGUAGE* in der *Systemvariablen*-Dokumentation).
- Wenn bei einer Quellfehlermeldung ein Stern (*) angegeben wird, überschreiben alle für diese Fehlermeldung definierten Sprachcodes alle für die Zielfehlermeldung definierten Sprachcodes.

  Beispiel: Wenn die Quellfehlermeldung nur in den Sprachen `1`, `2` und `3` existiert und für die Zielfehlermeldung nur die Sprachen `1`, `4` und `6` definiert sind, existiert die resultierende Zielfehlermeldung nach Ausführung einer Kopierfunktion (**Copy**) nur in den Sprachen `1`, `2` und `3`.
- Wenn die Sprachen als einzelne Codes angegeben sind, wird jede Ausprägung des Sprachcodes einzeln verarbeitet.

  Beispiel: Wenn die Quellfehlermeldung die Sprachen `1`, `2` und `3` enthält und die Sprachcodes auf `123` gesetzt sind, und wenn die Zielfehlermeldung die Sprachen `1`, `4` und `6` enthält und die Sprachcodes ebenfalls auf `123` gesetzt sind, enthält die resultierende Zielfehlermeldung nach Ausführung einer Kopierfunktion (**Copy**) die Sprachen `1`, `2`, `3`, `4` und `6`, aber nur die englische Zielfehlermeldung (Sprache 1) wird durch den englischen Text der Quellfehlermeldung überschrieben.
- Wenn für die Quellfehlermeldung ein einziger Sprachcode und für die Zielfehlermeldung mehrere Sprachcodes angegeben sind, ist die resultierende Zielfehlermeldung nach Ausführung der Kopierfunktion (**Copy**) in der ersten für das Ziel angegebenen Sprache.

Direktkommando-Syntax für Fehlermeldungen

Dieser Abschnitt zeigt die Syntax, die bei der Ausführung einer SYSMAIN-Funktion für Fehlermeldungen mit Direktkommandos im Online- oder Batch-Modus gilt. Allgemeine Anleitungen zur Verwendung von Direktkommandos finden Sie im Abschnitt ***Kommandos ausführen***.

Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den folgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*`where-clause`*) und der With-Klausel (*`with-clause`*) ist für alle Kommandos gleich.

Wichtig: Bei Systemfehlermeldungen geben Sie als *lib-name* NATURAL-SYSTEM oder NATURAL-SYS an.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- FIND
- LIST
- RENAME
- where-clause
- with-clause

COPY und MOVE

```
{ COPY }
{ MOVE } ERROR  number  [THRU number]
                        FM [LIBRARY] lib-name [where-clause]
                        TO [LIBRARY] lib-name [where-clause] [with-clause]
```

Beispiele:

```
COPY ERROR 1 FM ACCOUNTS TO ACCOUNTS1 REP WITH TYPE A
```

```
C ERROR 1 THRU 50 FM ACCT WHERE DBID 1 FNR 10 LANG 123456 TO ACCT WHERE DBID 5 FNR ↩
26 LANG 234567 WITH REP HELP
```

```
MOVE E 200 THRU 210 FM ACCT FNR 10 LANG 123 TO ACCT LANG 123 TYPE S
```

```
M E 376 TYPE E FM ACCT LANG E TO ACCT LANG G
```

DELETE

```
DELETE  ERROR  number  [THRU number]
                      [   IN [LIBRARY] lib-name   ]    [where-clause] [with-clause]
```

Beispiele:

```
DELETE ERROR 1 THRU 10 IN LIBRARY ACCT WHERE DBID 1 FNR 2 PSW GUESS CIPH 137561 ↩
WITH TYPE E MON HELP
```

```
D E 100 IN ACCT
```

FIND

```
FIND  ERROR  number  [  IN [LIBRARY] lib-name  ]  [where-clause]  [with-clause]
```

Beispiele:

```
FIND E 4280 IN A* MON
```

```
F ERROR 10 IN LIB ACCT WHERE DBID 1 FNR 3 WITH TYPE E
```

LIST

```
LIST  ERROR  number  [THRU number]  [  IN [LIBRARY]          ]  [where-clause] [with-clause]
                                    [  lib-name             ]
```

Beispiele:

```
LIST E 1 THRU 10 IN ACCT
```

```
L ERROR 100 THRU 150 IN LIB ACCT WHERE DBID 12 FNR 5
```

RENAME

Wenn innerhalb einer einzelnen (Quell-)Library ein Bereich von Fehlermeldungen neu nummeriert wird, dürfen sich die Wertebereiche nicht überschneiden. Siehe *Beispiele für ungültige Nummernbereiche*.

```
RENAME   ERROR  number  [THRU number] AS new-number
                        [THRU new-number] [with-clause]
                        IN [LIBRARY] lib-name [where-clause]
                        TO [LIBRARY] lib-name [where-clause]
```

Beispiele:

```
RENAME ERR 1 AS 101 IN ACCT
```

```
R ERROR 1 THRU 100 AS 101 THRU 200 IN CLAIMS
```

```
R ERROR 101 THRU 200 AS 1 THRU 100 IN CLAIMS
```

```
RENAME ERROR 1 THRU 50 AS 11 THRU 60 WITH TYPE A REP HELP MON RCOP IN LIBRARY ACCT ↩
WHERE DBID 1 FNR 2 TO LIB ACCOUNT WHERE FNR 3
```

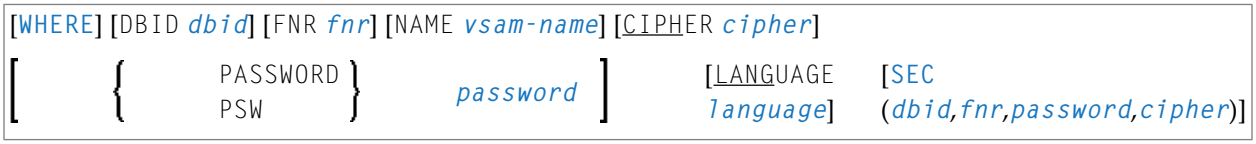
Beispiele für ungültige Nummernbereiche:

Die folgenden Beispiele sind *ungültig*, weil sich die Nummernbereiche überlappen:

```
R ERROR 1 THRU 100 AS 51 THRU 150 IN CLAIMS

R ERROR 101 THRU 200 AS 51 THRU 150 IN CLAIMS
```

where-clause



Trennzeichen

Kommas müssen als Trennzeichen zwischen den Werten nach dem Schlüsselwort `SEC` verwendet werden, oder wenn ein Wert fehlt. Zum Beispiel: `SEC (10,,secret,2a)`. Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf Komma (`,`) gesetzt wurde, verwenden Sie einen Schrägstrich (`/`) als Trennzeichen zwischen den Werten.

with-clause



75 Profile bearbeiten

- Felder im Menü Profiles 728
- Auswahllisten für Profile 729
- Direktkommando-Syntax für Profile 731

Alle SYSMAIN-Funktionen mit Ausnahme der Suchfunktion (**Find**) können mit den folgenden Profiltypen ausgeführt werden: Geräteprofile (Device), Editorprofile, Map-Profile und Parameterprofile. Geräteprofile, Editor- und Map-Profile werden mit dem Programmeditor bzw. dem Masken-Editor (*Map-Editor*) erstellt. Parameterprofile werden mit dem Dienstprogramm SYSPARM erstellt.

Dieses Kapitel beschreibt die Menüfunktionen und die Optionen der Auswahlliste (**Selection**), die für die Ausführung einer SYSMAIN-Funktion bei Profilen zur Verfügung stehen, sowie die Syntax, die bei der Verwendung von Direktkommandos gilt.

Felder im Menü Profiles

Das **Profiles**-Menü enthält alle SYSMAIN-Funktionen, die für die Bearbeitung von Profilen erforderlich sind. Die Felder des Menüs sind in der folgenden Tabelle beschrieben:

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Code</td><td>Gibt die auszuführende Funktion an. Siehe Beschreibung der Funktionen:
<table>
<tr><td></td><td></td></tr>
<tr><td>C</td><td>Copy - Profil kopieren.</td></tr>
<tr><td>D</td><td>Delete - Profil löschen.</td></tr>
<tr><td>L</td><td>List - Profil auflisten.</td></tr>
<tr><td>M</td><td>Move - Profil verschieben.</td></tr>
<tr><td>R</td><td>Rename - Profil umbenennen.</td></tr>
</table></td></tr>
<tr><td>Sel. List</td><td>Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird:
<table>
<tr><td></td><td></td></tr>
<tr><td>Y</td><td>Ja: Selektive Verarbeitung ist aktiviert. Bei der Bearbeitung von Profilen wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten und Zeilenkommandos in einer Auswahlliste verfügbar sind, erfahren Sie unter Auswahllisten für Profile.</td></tr>
<tr><td>N</td><td>Nein: Die selektive Verarbeitung ist deaktiviert.</td></tr>
</table></td></tr>
<tr><td>Name</td><td>Der Name des zu bearbeitenden Profils oder ein Bereich von Namen: siehe auch Angabe eines Namensbereichs.</td></tr>
<tr><td>New Name</td><td>Neuer Name: Der Name, der einem Profil gegeben wird, wenn es mit der Funktion Rename umbenannt wird.</td></tr>
<tr><td>Type</td><td>Der Typ des zu bearbeitenden Profils:
<table>
<tr><td></td><td></td></tr>
<tr><td>D</td><td>Es werden nur Geräteprofile (Device) verarbeitet.</td></tr>
<tr><td>E</td><td>Es werden nur Editor-Profile verarbeitet.</td></tr>
<tr><td>M</td><td>Es werden nur Map-Profile bearbeitet.</td></tr>
</table></td></tr>
</table>

Feld	Erläuterung	
	P	Es werden nur Parameterprofile bearbeitet.
	* oder leeres Feld	Es werden alle Profile bearbeitet. Dies ist die Standardeinstellung.
	Die Typen D, E, M und P können in beliebiger Kombination verwendet werden.	
Database	Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank. Die Quelldatenbank enthält die Systemdatei, in der das zu bearbeitende Profil gespeichert ist. Die Zieldatenbank enthält die Systemdatei, in die das Profil kopiert (**Copy**) oder verschoben (**Move**) werden soll, oder in der das Profil umbenannt (**Rename**) wird. Gültige Datenbankkennungen: 1 bis 65535.	
File	Datei: Die Dateinummer (FNR) einer Quell- oder Zielsystemdatei. Gültige Datenbankkennungen: 1 bis 65535. Die Quelldatei enthält das zu bearbeitende Profil. Die Zieldatei ist die Datei, in die das Profil kopiert (**Copy**) oder verschoben (**Move**) werden soll, oder in der das Profil umbenannt (**Rename**) wird.	
Name (unter **New Name**)	Gilt nur für VSAM-Dateien. Der DDNAME/FCT-Eintrag für die Nummer der Quell- oder Zieldatei.	
Replace	Ersetzen: Gibt an, ob ein Profil beim Verschieben (**Move**) , Kopieren (**Copy**) oder Umbenennen (**Rename**) ersetzt werden soll:	
	Y	Ja: Ein Profil mit demselben Namen, das in der Zielumgebung existiert, wird ersetzt.
	N	Nein: Ein Profil mit demselben Namen, das in der Zielumgebung existiert, wird nicht ersetzt. Dies ist die Standardeinstellung.
	Siehe auch *Option Replace (Ersetzen) verwenden*.	

Auswahllisten für Profile

Wenn die selektive Verarbeitung aktiviert wurde, wird auf einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen Profilen angezeigt, die die angegebenen Auswahlkriterien erfüllen.

- Spalten

- Zeilenkommandos

Spalten

Die folgenden Spalten werden auf einem Auswahlbildschirm (**Selection**) angezeigt:

Spalte	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Profile	Der Name des Profils, das die angegebenen Auswahlkriterien erfüllt.
Type	Der Typ des Profils: `D` Device-Profil. `E` Editor-Profil. `M` Map-Profil. `P` Parameterprofil.
Message	Meldung: Die Meldung, die den aktuellen Bearbeitungsstatus eines Profils angibt. Mögliche Meldungen siehe *Status-Meldungen*.

Zeilenkommandos

Die Zeilenkommandos, die in der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) eingegeben werden können, sind in der folgenden Tabelle aufgeführt. Die Zeilenkommandos `E`, `M`, `D` und `P` können in beliebiger Kombination eingegeben werden.

Zeilen-kommando	Funktion
`A`	Verarbeitet Device-Profile, Editorprofile, Map-Profile und/oder Parameterprofile.
`D`	Es werden nur Device-Profile verarbeitet, selbst wenn auch Editor-, Map- und Parameterprofile aufgeführt sind. Wenn `D` bei einem Profil angegeben wird, das nicht als Device-Profile existiert, tritt ein Fehler auf.
`E`	Es werden nur Map-Profile verarbeitet, auch wenn Geräte-, Map- und Parameter-Profile aufgelistet sind. Wenn `E` für ein Profil angegeben wird, das nicht als Editor-Profil existiert, wird ein Fehler zurückgegeben.
`M`	Es werden nur Map-Profile verarbeitet, auch wenn Device-, Editor- und Parameterprofile aufgelistet sind. Wenn `M` für ein Profil angegeben wird, das nicht als Map-Profil vorhanden ist, wird ein Fehler zurückgegeben.
`L`	Zeigt den Inhalt eines Parameterprofils an.
`P`	Es werden nur Parameterprofile verarbeitet, auch wenn Device-, Editor- und Map-Profile aufgeführt sind. Wenn `P` für ein Profil angegeben wird, das nicht als Parameterprofil existiert, wird ein Fehler zurückgegeben.

Direktkommando-Syntax für Profile

Dieser Abschnitt zeigt die Syntax, die bei der Ausführung einer SYSMAIN-Funktion für Profile mit Direktkommandos im Online- oder Batch-Modus gilt. Allgemeine Anleitungen zur Benutzung von Direktkommandos finden Sie unter ***Kommandos ausführen***.

Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den nachfolgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel (*with-clause*) ist bei allen Kommandos identisch.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- LIST
- RENAME
- where-clause
- with-clause

COPY und MOVE

```
{ COPY }
{ MOVE } PROFILE  name  [with-clause] FM [where-clause] TO [where-clause]
```

Beispiele:

```
COPY PRO USER1 TYPE E FM DBID 1 FNR 5 TO DBID 2 FNR 5

C PROFILE USER TYPE MED REP FM FNR 6 TO FNR 7

COPY PRO USER1 TYPE P FM DBID 10 FNR 44 TO DBID 3 FNR 7

MOVE PRO USER1 TYPE E FM DBID 1 FNR 5 TO DBID 2 FNR 5

M PROFILE USER1 TYPE MED REP FM FNR 6 TO FNR 7
```

DELETE

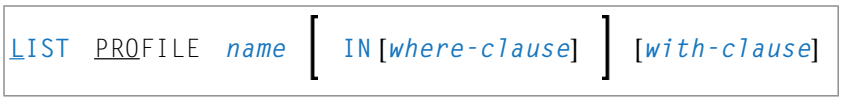


Beispiele:

```
DELETE PRO U* TYPE DM
```

```
D PRO TEST* IN DBID 177 FNR 205
```

LIST

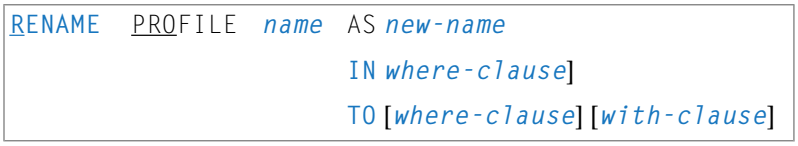


Beispiele:

```
LIST PRO USER* IN DBID 1 FNR 5
```

```
L PRO DT* TYPE E
```

RENAME



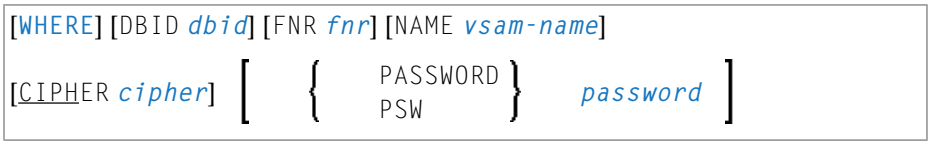
Beispiele:

```
RENAME PROFILE USER1 AS USER2 RCOP
```

```
R PRO USER1 AS USER2 DBID 1 FNR 4 TO DBID 1 FNR 5
```

```
R PRO USER1 AS NEWUSER IN FNR 4 TO FNR 5 REPLACE RCOP
```

where-clause



with-clause



76 Rules bearbeiten

- Felder im Menü Rules 736
- Auswahllisten für Rules 738
- Direktkommando-Syntax für Rules 739

Alle SYSMAIN-Funktionen mit Ausnahme der Suchfunktion (**Find**) können auf automatisierte und freie Rules angewendet werden.

Wenn Predict installiert ist, empfehlen wir Ihnen, Predict anstelle von SYSMAIN für die Verarbeitung von Rules zu verwenden, weil Natural nicht alle von Predict bereitgestellten Informationen zu Rules (z.B. Format und Verifikationstyp) verarbeitet.

Die Angabe der Rules-Umgebung muss immer mit der Datenbankkennung (DBID) und der Dateinummer (FNR) der entsprechenden FDIC-Systemdatei übereinstimmen.

In diesem Kapitel werden die Menüfunktionen und Auswahllistenoptionen beschrieben, die zur Ausführung einer SYSMAIN-Funktion bei Rules zur Verfügung stehen, sowie die Syntax, die bei der Verwendung von Direktkommandos gilt.

Felder im Menü Rules

Das Menü **Rules** enthält alle SYSMAIN-Funktionen, die für die Bearbeitung von Rules erforderlich sind. Die Felder des Menüs sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung
Code	Gibt die auszuführende Funktion an. Siehe *Beschreibung der Funktionen*: `C` Copy - Rule kopieren. `D` Delete - Rule löschen. `L` List - Rule auflisten. `M` Move - Rule verschieben. `R` Rename - Rule umbenennen.
Sel. List	Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird: `Y` Ja: Selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von Rules wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten und Zeilenkommandos in einer Auswahlliste verfügbar sind, erfahren Sie unter *Auswahllisten für Rules*. `N` Nein: Die selektive Verarbeitung ist deaktiviert.
Name	Der Name der zu verarbeitenden Rule oder ein Namensbereich: siehe auch *Angabe eines Namensbereichs*.
New Name	Neuer Name: Der Name, der einer Rule gegeben werden soll, wenn sie mit der Funktion **Rename** umbenannt wird.
Type	Der Typ der zu verarbeitenden Rule:

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td></td><td>
<table>
<tr><td></td><td></td></tr>
<tr><td>A</td><td>Es werden nur automatische Rules verarbeitet.</td></tr>
<tr><td>F</td><td>Es werden nur freie Rules verarbeitet.</td></tr>
<tr><td>AF, * oder leeres Feld</td><td>Es werden alle Rules verarbeitet: automatische und/oder freie Rules. Dies ist die Standardeinstellung. Datenbank Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.</td></tr>
</table>
</td></tr>
<tr><td>Database</td><td>Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.

Die Quelldatenbank enthält die FDIC-Systemdatei, in der die zu verarbeitende Rule gespeichert ist. Die Zieldatenbank enthält die FDIC-Systemdatei, in die die Rule kopiert (Copy) oder verschoben (Move) werden soll, oder in der die Rule umbenannt wird.

Gültige Datenbankkennungen: 1 bis 65535.</td></tr>
<tr><td>File</td><td>Datei: Die Dateinummer (FNR) einer Quell- oder Ziel-FDIC-Systemdatei.

Gültige Dateinummern: 1 bis 65535.

Die Quelldatei enthält die zu verarbeitende Rule. Die Zieldatei ist die Datei, in die die Rule kopiert (Copy) oder verschoben (Move) werden soll, oder in der die Rule (Rename) umbenannt wird.</td></tr>
<tr><td>Name
(neben File)</td><td>Gilt nur für VSAM-Dateien.

Der DDNAME/FCT-Eintrag für die Nummer der Quell- oder Zieldatei.</td></tr>
<tr><td>Replace</td><td>Ersetzen: Gibt an, ob eine Rule bei der Verwendung der Funktionen Verschieben, Kopieren oder Umbenennen ersetzt werden soll:
<table>
<tr><td></td><td></td></tr>
<tr><td>Y</td><td>Ja: Eine Rule mit demselben Namen, die in der Zielumgebung existiert, wird ersetzt.</td></tr>
<tr><td>N</td><td>Nein: Eine Rule mit demselben Namen, die in der Zielumgebung existiert, wird nicht ersetzt. Dies ist die Standardeinstellung.</td></tr>
<tr><td></td><td></td></tr>
</table>
Siehe auch Option Replace (Ersetzen) verwenden.</td></tr>
</table>

Auswahllisten für Rules

Wenn die selektive Verarbeitung aktiviert wurde, wird auf einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen Rules angezeigt, die die angegebenen Auswahlkriterien erfüllen.

Dieser Abschnitt beschreibt die Spalten eines Auswahlbildschirms und die Zeilenkommandos, die zur weiteren Bearbeitung einer Rule zur Verfügung stehen:

- Spalten
- Zeilenkommandos

Spalten

Die folgenden Spalten werden in einem Auswahlbildschirm (**Selection**) angezeigt:

SpalteSpalte	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Rule Name	Der Name der Rule, die den angegebenen Auswahlkriterien entspricht.
Type	Der Typ der Rule: Freie Rule (`Free`) oder automatische Rule (`Auto`).
Ver. Type	Der Verifizierungstyp: Ein Attribut der Rule, das direkt aus der in Predict definierten Verifizierung übernommen wird. Der Wert `Unknown` (unbekannt) bedeutet, dass keine entsprechenden Verifizierungsinformationen in Predict verfügbar sind. Gültige Werte finden Sie in der *Predict*-Dokumentation.
Format	Der Formattyp: Ein Attribut der Rule, das direkt aus der in Predict definierten Verifizierung übernommen wird. Der Wert `Unknown` (unbekannt) zeigt an, dass keine entsprechenden Verifikationsinformationen in Predict verfügbar sind. Gültige Werte finden Sie in der *Predict*-Dokumentation.
Message	Meldung Die Nachricht, die den aktuellen Verarbeitungsstatus einer Rule angibt. Mögliche Meldungen siehe ***Status-Meldungen***.

Zeilenkommandos

In der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) kann eines der folgenden Zeilenkommandos eingegeben werden:

Zeilen-kommando	Funktion
`A`	Verarbeitet die angegebene Rule.
`F`	Gilt nur für automatische Rules. Es werden alle DDMs und Felder angezeigt, denen die angegebene Rule zugewiesen ist.

Zeilen-kommando	Funktion
H	Erzeugt einen Ausdruck des gespeicherten (Quellcode-)Objekts einer Rule. Der Quellcode der angegebenen Rule wird gedruckt und auf dem Bildschirm angezeigt.
I	Verzeichnisinformationen der Rule anzeigen. Dieses Kommando ist vergleichbar mit dem Systemkommando `LIST DIR` . Details zu den angezeigten Verzeichnisinformationen finden Sie unter *Directory-Informationen anzeigen* in der *Systemkommandos*-Dokumentation.
L	Anzeige des Quellcodes einer Rule.

Direktkommando-Syntax für Rules

Dieser Abschnitt zeigt die Syntax, die bei der Ausführung einer SYSMAIN-Funktion bei Rules unter Verwendung von Direktkommandos im Online- oder Batch-Modus gilt. Allgemeine Anleitungen zur Benutzung von Direktkommandos finden Sie unter ***Kommandos ausführen***.

Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den nachfolgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

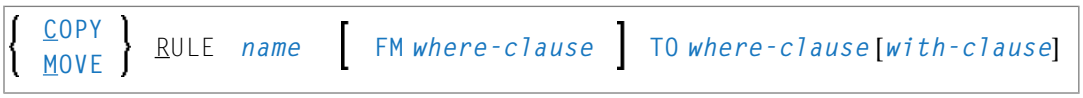
Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel)*with-clause*) ist bei allen Kommandos identisch.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- LIST
- RENAME
- where-clause

- with-clause

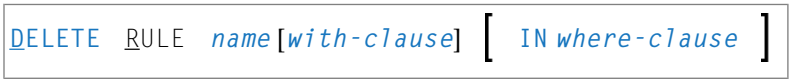
COPY und MOVE



Beispiele:

```
C RULE TESTRULE FM FNR 20 TO FNR 24 REPLACE
COPY R C< FM FNR 20 TO FNR 24
M RULE TESTRULE FM FNR 20 TO FNR 24 REPLACE
MOVE RULE C< FM FNR 20 TO FNR 24
```

DELETE



Beispiel:

```
D RULE DEMORULE IN DBID 12 FNR 27
```

LIST



Beispiel:

```
L RULE * DBID 1 FNR 5
```

RENAME



Beispiele:

```
RENAME R OLDRULE AS NEWRULE FM DBID 1 FNR 4 TO DBID 1 FNR 5
R RULE OLDRULE AS NEWRULE FM FNR 4 TO FNR 5 REPLACE RCOP
```

where-clause

```
[WHERE] [DBID dbid] [FNR fnr] [NAME vsam- name] [CIPHER cipher]
      [ { PASSWORD } password ]
        { PSW      }
      [DIC (dbid,fnr,password,cipher)]
      [SEC (dbid,fnr,password,cipher)]
```

Trennzeichen

Kommas müssen als Trennzeichen zwischen den Werten nach dem Schlüsselwort `SEC` verwendet werden, oder wenn ein Wert fehlt. Zum Beispiel: `SEC (10,,secret,2a)`. Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf Komma (`,`) gesetzt wurde, verwenden Sie einen Schrägstrich (`/`) als Trennzeichen zwischen den Werten.

with-clause

```
[WITH] [TYPE type] [REPLACE] [RCOP] [ PROMPT   ] [ MON   ] [HELP]
                                    [ NOPROMPT ] [ NOMON ]
```

77 DDMs bearbeiten

- Felder im Menü DDMs 744
- Auswahllisten für DDMs 746
- Direktkommando-Syntax für DDMs 747

Alle SYSMAIN-Funktionen mit Ausnahme der Suchfunktion (**Find**) und Umbenennungsfunktion (**Rename**) können bei Datendefinitionsmodulen (DDMs) ausgeführt werden.

Die Angabe der DDM-Umgebung muss immer mit der Datenbankkennung (DBID) und der Dateinummer (FNR) der entsprechenden FDIC-Systemdatei übereinstimmen.

Dieses Kapitel beschreibt die Menüfunktionen und Auswahllistenoptionen, die zur Ausführung einer SYSMAIN-Funktion auf DDMs zur Verfügung stehen, sowie die Syntax, die bei der Verwendung von Direktkommandos gilt.

Felder im Menü DDMs

Das Menü **DDMs** enthält alle SYSMAIN-Funktionen, die für die Bearbeitung von DDMs erforderlich sind. Die Felder in diesem Menü sind in der folgenden Tabelle beschrieben:

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>Code</td><td>Gibt die auszuführende Funktion an. Siehe Beschreibung der Funktionen:<table><tr><td></td><td></td></tr><tr><td>C</td><td>Copy - DDM kopieren.</td></tr><tr><td>D</td><td>Delete - DDM löschen.</td></tr><tr><td>L</td><td>List - DDM auflisten.</td></tr><tr><td>M</td><td>Move- DDM verschieben.</td></tr></table></td></tr>
<tr><td>Sel. List</td><td>Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird:<table><tr><td></td><td></td></tr><tr><td>Y</td><td>Ja: Selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von DDMs wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung.

Welche Spalten und Zeilenbefehle auf einer Auswahlliste verfügbar sind, erfahren Sie unter Auswahllisten für DDMs.</td></tr><tr><td>N</td><td>Nein: Die selektive Verarbeitung ist ausgeschaltet.</td></tr></table></td></tr>
<tr><td>Name</td><td>Der Name des zu verarbeitenden DDMs oder ein Bereich von Namen. Siehe auch Angabe eines Namensbereichs.</td></tr>
<tr><td>DDM DBID</td><td>In diesem Feld können Sie nur DDMs auswählen, die unter der angegebenen Datenbankkennung (DBID) katalogisiert wurden.

Gültige Datenbankkennungen: 1 bis 65535.

Wenn dieses Feld leer oder auf 0 (Null) gesetzt ist, wird die DBID nicht als Auswahlkriterium verwendet.</td></tr>
</table>

<table>
<tr><th>Feld</th><th>Erläuterung</th></tr>
<tr><td>DDM FNR</td><td>Dieses Feld kann verwendet werden, um nur DDMs auszuwählen, die unter der angegebenen Dateinummer (FNR) katalogisiert wurden.

Gültige Dateinummern: 1 bis 65535.

Wenn dieses Feld leer oder auf 0 (Null) gesetzt ist, wird die FNR nicht als Auswahlkriterium verwendet.</td></tr>
<tr><td>Database</td><td>Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank.

Die Quelldatenbank enthält die FDIC-Systemdatei, in der der zu verarbeitende DDM gespeichert ist. Die Zieldatenbank enthält die FDIC-Systemdatei, in die das DDM kopiert oder verschoben werden soll.

Gültige Datenbankkennungen: 1 bis 65535.</td></tr>
<tr><td>File</td><td>Die Dateinummer (FNR) einer Quell- oder Ziel-FDIC-Systemdatei.

Gültige Dateinummern: 1 bis 65535.

Die Quelldatei enthält das zu bearbeitende DDM. Die Zieldatei ist die Datei, in die das DDM kopiert oder verschoben werden soll.</td></tr>
<tr><td>Name
(neben File)</td><td>Gilt nur bei VSAM-Dateien.

Der DDNAME/FCT-Eintrag für die Nummer der Quell- oder Zieldatei.</td></tr>
<tr><td>Replace</td><td>Ersetzen: Gibt an, ob ein DDM bei der Verwendung der Verschiebe- oder Kopierfunktion ersetzt werden soll:<table><tr><td></td><td></td></tr><tr><td>Y</td><td>Ja: Ein DDM mit demselben Namen, das in der Zielumgebung vorhanden ist, wird ersetzt.</td></tr><tr><td>N</td><td>Nein: Ein DDM mit demselben Namen, das in der Zielumgebung vorhanden ist, wird nicht ersetzt. Dies ist die Standardeinstellung.</td></tr><tr><td></td><td></td></tr></table>Siehe auch Option Replace (Ersetzen) verwenden.</td></tr>
<tr><td>Del.NSC-Def.</td><td>Dieses Feld erscheint nur in einer Natural Security-Umgebung.

Es zeigt die Löschung von Natural Security-Definitionen an.

Wenn ein DDM aus einer Quellumgebung gelöscht oder in eine neue Umgebung verschoben wird und andere FSEC-Systemdateien angegeben wurden, können Sie in diesem Feld angeben, ob die DDM-Definition in der Quell-FSEC-Systemdatei gelöscht werden soll oder nicht.

Mögliche Werte:<table><tr><td></td><td></td></tr></table></td></tr>
</table>

Feld	Erläuterung	
	Y	Ja: Die DDM-Definition in Quell-FSEC-Systemdatei wird gelöscht.
	N	Nein: Die DDM-Definition in der Quell-FSEC-Systemdatei wird beibehalten. Dies ist die Standardeinstellung.

Auswahllisten für DDMs

Wenn die selektive Verarbeitung aktiviert wurde, wird auf einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen DDMs angezeigt, die den angegebenen Auswahlkriterien entsprechen.

Dieser Abschnitt beschreibt die Spalten eines Auswahlbildschirms (**Selection**) und die Zeilenkommandos, die für die weitere Bearbeitung eines DDMs zur Verfügung stehen.

- Spalten
- Zeilenkommandos

Spalten

Die folgenden Spalten und Felder werden auf einem Auswahlbildschirm (**Selection**) angezeigt:

Spalte	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
DDM Name	Der Name des DDM, der den angegebenen Auswahlkriterien entspricht.
DBID	Die Datenbankkennung (DBID), unter der das DDM katalogisiert wurde. Wenn keine DBID oder 0 (Null) angegeben ist, wurde das DDM unter der aktuellen DBID katalogisiert.
FNR	Die Dateinummer (FNR), unter der das DDM katalogisiert wurde.
Cat Date	Das Datum, an dem das DDM katalogisiert wurde.
Time	Die Zeit, zu der das DDM katalogisiert wurde.
User ID	Die Benutzerkennung des Benutzers, der das DDM katalogisiert hat.
Message	Meldung: Die Meldung, die den Bearbeitungsstatus eines DDM angibt. Diese Spalte erscheint nur, wenn das Zeilenkommando A für ein DDM ausgeführt wurde. Mögliche Meldungen siehe *Status-Meldungen*.

Die Spalten **Cat Date**, **Time** (Uhrzeit) und **User ID** (Datum, Zeit und Benutzerkennung der Katalogisierung) sind leer, wenn das DDM unter einer älteren Version von Natural katalogisiert wurde, die diese Informationen nicht liefert.

Zeilenkommandos

In der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) kann eines der folgenden Zeilenkommandos eingegeben werden:

Zeilen-kommando	Funktion
A	Verarbeitet das angegebene DDM. Dieses Zeilenkommando ist nicht bei der Auflistfunktion verfügbar.
L	Listet das angegebene DDM auf. Ausführliche Informationen zu den angezeigten Felddefinitionen finden Sie unter *Spalten für Feld-Attribute* in der *DDM-Editor (SYSDDM Utility)*-Dokumentation.
R	Listet alle automatischen Rules auf, die mit dem angegebenen DDM verknüpft sind. Das Zeilenkommando `R` ruft den Bildschirm **Rule Usage for DDM/Fields** (Rules-Verwendung für DDM/Felder) auf, der anzeigt, ob ein DDM eine Rule verwendet oder nicht. Wenn eine Rule verwendet wird, werden der Name der Rule und das DDM-Feld, auf das verwiesen wird, auf dem Bildschirm angezeigt.
X	Gilt nur, wenn Natural Connection und Entire Connection installiert sind. DDM(s) auf einen PC herunterladen. Ausführliche Anleitungen finden Sie unter ***Objekte auf einen PC herunterladen***.

Direktkommando-Syntax für DDMs

In diesem Abschnitt wird die Syntax für die Ausführung einer SYSMAIN-Funktion bei DDMs mit Direktkommandos im Online- oder Batch-Modus erläutert. Allgemeine Anleitungen zur Benutzung von Direktkommandos finden Sie im Abschnitt ***Kommandos ausführen***.

Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den nachfolgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel (*with-clause*) ist bei allen Kommandos identisch.

Anmerkung: Aus Kompatibilitätsgründen können Sie in Direktkommandos für DDMs statt des Schlüsselworts `DDM` das Schlüsselwort `VIEW` (oder dessen Kurzform `V`) verwenden.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- LIST
- where-clause
- with-clause

COPY und MOVE

```
{ COPY } DDM name [ FM where-clause ] TO where-clause [with-clause]
{ MOVE }
```

Beispiele:

```
C DDM PERSONNEL FM FNR 20 TO FNR 24 REPLACE
```

```
COPY DDM C< FM FNR 20 TO FNR 24
```

```
M DDM PERSONNEL FM FNR 20 TO FNR 24 REPLACE
```

```
MOVE DDM C< FM FNR 20 TO FNR 24
```

DELETE

```
DELETE DDM name [with-clause] [ IN where-clause ]
```

Beispiel:

```
D DDM FINANCE IN DBID 12 FNR 27
```

LIST

```
LIST DDM name [ IN where-clause ] [with-clause]
```

Beispiel:

```
L DDM * IN DBID 1 FNR 5
```

where-clause

```
[WHERE] [DBID dbid] [FNR fnr] [NAME vsam-name] [CIPHER cipher]
        [ { PASSWORD } password ]
        [ { PSW      }          ]
        [DIC (dbid,fnr,password,cipher)]
        [SEC (dbid,fnr,password,cipher)]
```

Trennzeichen

Kommas müssen als Trennzeichen zwischen den Werten nach dem Schlüsselwort `SEC` verwendet werden, oder wenn ein Wert fehlt. Zum Beispiel: `SEC (10,,secret,2a)`. Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf Komma (`,`) gesetzt wurde, verwenden Sie einen Schrägstrich (`/`) als Trennzeichen zwischen den Werten.

with-clause

```
[WITH] [ { DDMDBID } DDM-dbid ] [ { DDMFNR } DDM-fnr ] [REPLACE] [ PROMPT   ] [ MON   ] [HELP]
       [ { DDBID  }           ] [ { DFNR  }           ]          [ NOPROMPT ] [ NOMON ]
```

78 Predict Sets bearbeiten

- Felder im Menü Predict Sets 752
- Auswahllisten für Predict Sets 753
- Direktkommando-Syntax für Predict Sets 754

Dieses Kapitel beschreibt die Menüs und Direktkommandos zur Bearbeitung von Predict Sets. Die folgenden Themen werden behandelt:

Anmerkungen:

1. Alle SYSMAIN-Funktionen außer der Suchfunktion (**Find**) können auf Predict Sets angewendet werden.
2. Die Angabe der Umgebung für Predict Sets muss immer mit der Datenbankkennung (DBID) und der Dateinummer (FNR) der entsprechenden FDIC-Systemdatei übereinstimmen.
3. Bei der Verarbeitung von Predict Sets wird der Software AG Editor verwendet. Daher müssen Sie den *Editor Buffer Pool* verwenden oder den Natural-Profilparameter `EDPSIZE` setzen.

Felder im Menü Predict Sets

Das Menü **Predict Sets** enthält alle SYSMAIN-Funktionen zur Bearbeitung von Predict Sets. Die Felder des Menüs sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung
Code	Gibt die auszuführende Funktion an. Siehe *Beschreibung der Funktionen* C Copy - Predict Set kopieren. D Delete - Predict Set löschen. L List - Predict Set auflisten. M Move - Predict Set verschieben. R Rename - Predict Set umbennen.
Sel. List	Gibt an, ob selektive oder automatisierte Verarbeitung verwendet wird: Y Ja: Selektive Verarbeitung ist aktiviert. Bei der Verarbeitung von Predict Sets wird eine Auswahlliste angezeigt. Dies ist die Standardeinstellung. Welche Spalten und Zeilenkommandos in einer Auswahlliste verfügbar sind, finden Sie unter *Auswahllisten für Predict Sets*. N Nein: Die selektive Verarbeitung wird deaktiviert und die automatisierte Verarbeitung wird aktiviert.
Set No. from	Set-Nummer von: Die Nummer des zu verarbeitenden Predict Sets oder die Startnummer eines Bereichs von Predict Sets, wenn im Feld **No. to** Nummer bis) eine Endnummer eingegeben wurde.
New No.	Neue Nummer: Die (neue) Nummer eines Predict Set, bei dem die Umbenennungsfunktion (**Rename**) durchgeführt werden soll.
Library	Der Name einer Quell- oder Ziel-Library, die dem/den Predict Set(s) zugewiesen ist. Es ist nicht möglich, Bereiche anzugeben.

Feld	Erläuterung
Database	Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank für die Library, die dem/den Predict Set(s) zugewiesen ist.
File	Datei: Die Dateinummer (FNR) einer Quell- oder Zieldatenbank für die Library, die dem/den Predict Set(s) zugewiesen ist.
User	Benutzer: Die Quell- oder Ziel-Benutzerkennung, die dem/den Predict Set(s) zugewiesen ist. Es ist nicht möglich, Bereiche anzugeben.
FDIC Database	FDIC-Datenbank: Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank. Die Quelldatenbank enthält die FDIC-Systemdatei, in der das zu verarbeitende Predict Set gespeichert ist. Die Zieldatenbank enthält die FDIC-Systemdatei, in die das Predict Set kopiert (**Copy**) oder verschoben (**Move**) werden soll, oder in der die Funktion Umbenennen (**Rename**) ausgeführt wird. Gültige Datenbankkennungen: `1` bis `65535`.
FDIC File	FDIC-Datei: Die Dateinummer (FNR) einer Quell- oder Ziel-FDIC-Systemdatei. Die Quelldatei enthält das zu verarbeitende Predict Set. Die Zieldatei ist die Datei, in die das Predict Set kopiert (**Copy**) oder verschoben (**Move**) werden soll, oder in der die Funktion Umbenennen (**Rename**) ausgeführt wird. Gültige Dateinummern: `1` bis `65535`.
Replace	Ersetzen: Gibt an, ob ein Predict Set bei der Verwendung der Verschiebe- (**Move**), Kopier- (**Copy**) oder Umbenennungsfunktion (**Rename**) ersetzt werden soll: Y Ja: Ein Predict Set in der Zielumgebung mit derselben Nummer für dieselbe Library und denselben Benutzer wird ersetzt. N Nein: Keine Ersetzung. Dies ist die Standardeinstellung. Siehe auch *Option Replace (Ersetzen) verwenden* .

Auswahllisten für Predict Sets

Wenn die selektive Verarbeitung aktiviert wurde, wird in einem Auswahlbildschirm (**Selection**) eine Auswahlliste mit allen Predict Sets angezeigt, die den angegebenen Auswahlkriterien entsprechen.

Dieser Abschnitt beschreibt die Spalten eines Auswahlbildschirms (Selection) und die Zeilenkommandos, die für die weitere Bearbeitung eines Predict Set zur Verfügung stehen:

- Spalten

- Zeilenkommandos

Spalten

Die folgenden Spalten und Felder werden auf einem Auswahlbildschirm (**Selection**) angezeigt:

Spalte	Erläuterung
C	Eingabefeld für Zeilenkommandos (siehe unten).
Set	Die Nummer des Predict Set, das den angegebenen Auswahlkriterien entspricht.
Criteria	Die Kriterien (Text des Predict-Kommandos), die für die Erstellung des Predict Set verwendet wurden.
Members	Anzahl der Namen der Natural-Objekte, die im Predict Set enthalten sind.
Date	Das Datum, an dem das Predict Set erstellt wurde.
Time	Die Uhrzeit, zu der das Predict Set erstellt wurde.

Zeilenkommandos

In der Spalte **C** (Command) eines Auswahlbildschirms (**Selection**) kann eines der folgenden Zeilenkommandos eingegeben werden:

Zeilen-kommando	Erläuterung
A	Verarbeitet das angegebene Predict Set.
L	Zeigt den Inhalt (Namen und Typen) des angegebenen Predict Set an.

Direktkommando-Syntax für Predict Sets

Dieser Abschnitt zeigt die Syntax, die bei der Ausführung einer SYSMAIN-Funktion bei einem Predict Set unter Verwendung von Direktkommandos im Online- oder Batch-Modus gilt. Allgemeine Anleitungen zur Verwendung von Direktkommandos finden Sie unter ***Kommandos ausführen***.

Erläuterungen zu den Schlüsselwörtern und Werten von Variablen

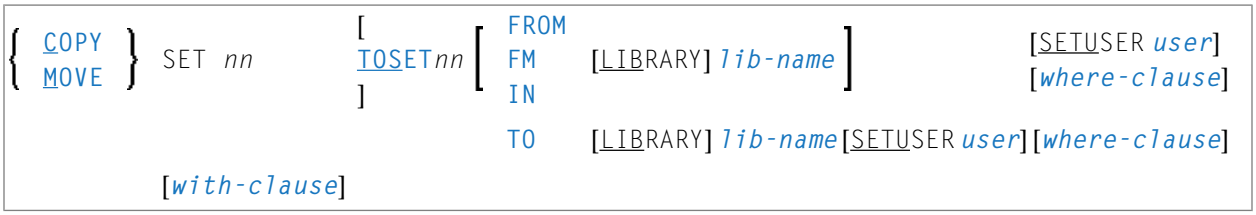
Erläuterungen zu den Schlüsselwörtern und Variablenwerten, die in den nachfolgenden Syntax-Diagrammen verwendet werden, finden Sie unter ***Schlüsselwörter und Variablen in Direktkommandos***. Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Where-Klausel (*where-clause*) und der With-Klausel (*with-clause*) ist bei allen Kommandos identisch.

In diesem Abschnitt werden die folgenden Themen behandelt:

- COPY und MOVE
- DELETE
- LIST
- RENAME
- where-clause
- with-clause

COPY und MOVE



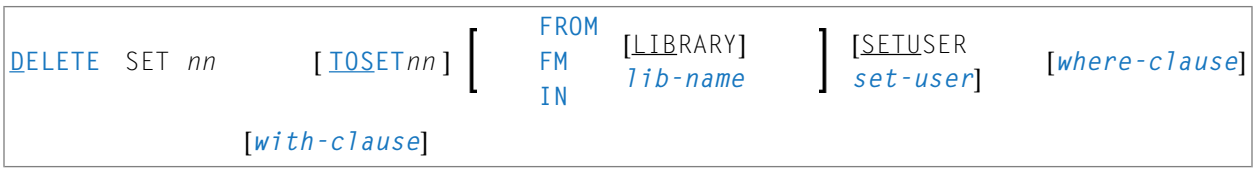
Beispiele:

```
COPY SET 1 TOSET 99 FROM LIBRARY SRCLIB SETUSER UID1 WHERE DBID 10 FNR 32 DIC (10/460)

  TO LIBRARY TGTLIB SETUSER UID2 WHERE DBID 10 FNR 110 DIC (10/1460) WITH REPLACE

COPY SET 1 TOS 99 FM SRCLIB SETU UID1 TO TGTLIB SETU UID2 REPLACE
```

DELETE



Beispiel:

```
D SET 17 IN SRCLIB DBID 10 FNR 32
```

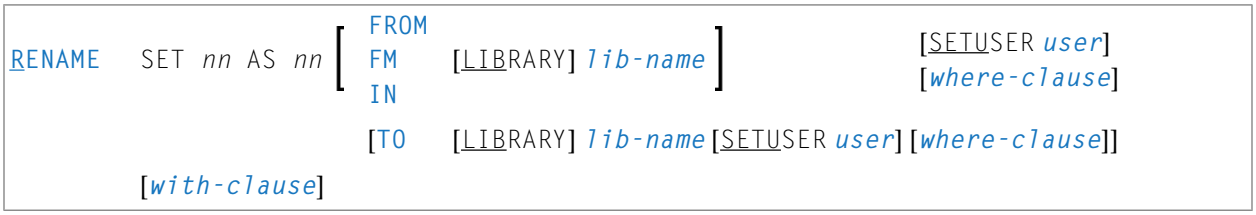
LIST



Beispiel:

```
LIST SET 11 TOS 44 IN LIB SRCLIB DBID 10 FNR 32
```

RENAME



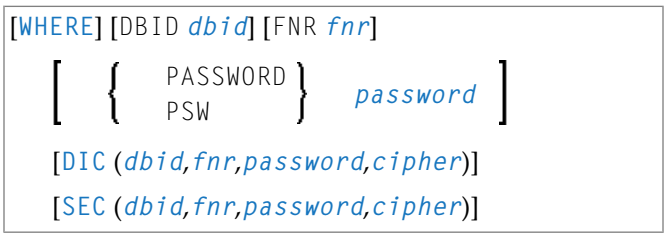
Beispiel:

```
RENAME SET 17 AS 83 FM SRCLIB SETUSER UID1 WHERE DBID 10 FNR 32 DIC (10/1460) TO ↩
TGTLIB  ↩

                SETUSER UID1 WHERE DBID 10 FNR 32 DIC (10/1460) WITH RCOP REP

RENAME SET 11 AS 11 FM LIB1 SETUSER UID1 TO LIB1 SETUSER UID2 WITH RCOP'
```

where-clause



Trennzeichen

Kommas müssen als Trennzeichen zwischen den Werten nach dem Schlüsselwort `SEC` verwendet werden, oder wenn ein Wert fehlt. Zum Beispiel: `SEC (10,,secret,2a)`. Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf Komma (`,`) gesetzt wurde, verwenden Sie einen Schrägstrich (`/`) als Trennzeichen zwischen den Werten.

with-clause



79 Schlüsselwörter und Variablen in Direktkommandos

- Beschreibung der Schlüsselwörter 760
- Angabe eines Namensbereichs 772

In diesem Kapitel werden alle Schlüsselwörter und Variablen beschrieben, die bei der Verwendung von Direktkommandos im Online- oder Batch-Modus relevant sind. Jedes Schlüsselwort stellt einen Parameter dar, der zur Angabe von Objektauswahlkriterien oder zur Einstellung einer Option für den auszuführenden Befehl verwendet wird. Falls angegeben, muss bei einem Schlüsselwort ein Variablenwert angegeben werden.

Die Symbole in den Syntax-Diagrammen entsprechen den Syntax-Symbolen, die für Systemkommandos verwendet werden. Diese Symbole werden unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erläutert.

Die Syntax der Direktkommandos, auf die sich die Schlüsselwörter beziehen, einschließlich der Einzelheiten zu der in diesem Abschnitt erwähnten Where-Klausel (*where-clause*) und With-Klausel (*with-clause*), finden Sie in den objekttypspezifischen Abschnitten dieser Dokumentation.

Beschreibung der Schlüsselwörter

In diesem Abschnitt werden die in einem Direktkommando verwendeten Schlüsselwörter und die entsprechenden Variablenwerte (falls erforderlich) erläutert.

Die Schlüsselwörter sind in alphabetischer Reihenfolge aufgeführt. Kursiv gedruckte Buchstaben stehen für Variablenwerte, die bei einem Schlüsselwort angegeben werden müssen. Zu jedem Variablenwert werden das Natural-Datenformat und die Länge angegeben.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
ALL oder A	*name*	A9	Gilt nur für Programmierobjekte. Der Name des zu bearbeitenden Objekts oder ein Bereich von Namen (siehe auch ***Angabe eines Namensbereichs***). Es werden alle gespeicherten (Quellcode-)Objekte und/oder katalogisierten Objekte bearbeitet.
AS	*new-name*	A8 oder A12	Gilt nicht für DDMs. Der neue Name, der einem Objekt gegeben wird, wenn es mit dem Kommando RENAME umbenannt wird. Format/Länge A12 gilt nur für Debug-Umgebungen.
	new-number	N4	Für Fehlermeldungen: Die neue Nummer, die einer Fehlermeldung zugewiesen wird, oder die Anfangsnummer eines Bereichs neuer Nummern, die einem Bereich vorhandener Fehlermeldungen bei Verwendung des Kommandos RENAME zugewiesen werden.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
	new-set-number	N2	Für Predict Sets: Die neue Nummer, die einem Predict Set bei Verwendung des Kommandos RENAME gegeben werden soll.
CATALOGED	*name*	A9	Gilt nur für Programmierobjekte. Der Name des zu bearbeitenden katalogisierten Objekts oder ein Bereich von Namen (siehe auch ***Angabe eines Namensbereichs***).
CIPHER	*cipher*	A8	Der Adabas-Chiffrierschlüssel einer Quell- und/oder Zielsystemdatei, der in einer Where-Klausel (*where-clause*) verwendet wird.. Für Rules und DDMs: Anstelle von CIPHER kann auch die entsprechende DIC-Angabe verwendet werden. Wenn ein Chiffrierschlüssel (*cipher*) zweimal angegeben wird, wird der zuletzt angegebene verwendet.
DBID	*dbid*	N5	Die Datenbankkennung (DBID) einer Quell- oder Zieldatenbank. Die Quell-Datenbank enthält die Systemdatei, in der das zu bearbeitende Objekt gespeichert ist. Die Ziel-Datenbank enthält die Systemdatei, in die das Objekt kopiert oder verschoben werden soll oder in der das Objekt gegebenenfalls umbenannt (oder im Falle einer Fehlermeldung umnummeriert) wird. Gültige DBIDs: 1 bis 65535. Wenn keine DBID oder Dateinummer (FNR) angegeben ist und SYSMAIN mit dem Systemkommando SYSMAIN oder dem Subprogramm MAINUSER aufgerufen wird (siehe auch ***SYSMAIN mittels Anwendungsprogrammierschnittstelle aufrufen***), gilt Folgendes: Es wird immer die DBID und FNR der Systemdatei verwendet, aus der SYSMAIN aufgerufen wurde. Beispiel: Wenn Sie SYSMAIN aus einer Library aufrufen, die in der Systemdatei FUSER enthalten ist, werden die DBID und FNR dieser Datei verwendet. Für Rules und DDMs: Anstelle von DBID kann auch die entsprechende DIC-Angabe verwendet werden. Wenn *dbid* zweimal angegeben wird, wird die zuletzt angegebene verwendet.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
DDM oder VIEW	*name*	A32	Gilt nur für DDMs. Der Name des zu verarbeitenden DDMs oder eines Namensbereichs. Siehe auch ***Angabe eines Namensbereichs***.
DDMDBID oder DDBID	*ddm-dbid*	N5	Gilt nur für DDMs. Die DDM-Datenbankkennung (DBID): Alle DDMs, die unter der angegebenen DBID katalogisiert wurden, werden verarbeitet. Gültige DDM-DBIDs: 0 bis 65535. Wenn kein Wert oder 0 (Null) angegeben wird, wird die DDM-DBID nicht geprüft.
DDMFNR oder DFNR	*ddm-fnr*	N5	Gilt nur für DDMs. Die DDM-Dateinummer (FNR): Alle DDMs, die unter der angegebenen FNR katalogisiert wurden, werden verarbeitet. Gültige DDM-FNRs: 0 bis 65535. Wenn kein Wert oder 0 (Null) angegeben wird, wird die DDM-FNR nicht geprüft.
DEBUG	*name*	A12	Gilt nur für Debug-Umgebungen. Der Name der zu verarbeitenden Debug-Umgebung oder ein Bereich von Namen. Siehe auch ***Angabe eines Namensbereichs***.
DIC	*dbid* *fnr* *password* *cipher*	A80	Gilt nur nicht für Fehlermeldungen und Profile. Gibt die Umgebung der FDIC-Quell- und/oder Ziel-Systemdatei an: Datenbankkennung (*dbid*), Dateinummer (*fnr*), Adabas-Passwort (*password*) und Adabas-Chiffrierschlüssel (*cipher*). Für Rules und DDMs: Die Angaben DBID, FNR, CIPHER und PASSWORD können anstelle der entsprechenden DIC-Angaben verwendet werden oder umgekehrt. Wenn ein Element zweimal angegeben wird, wird das zuletzt angegebene verwendet.
ERROR	*number*	N4	Gilt nur für Fehlermeldungen. Die Nummer der zu verarbeitenden Fehlermeldung oder die Anfangsnummer eines Nummernbereichs, wenn THRU angegeben ist.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
EXTEND	-	-	Gilt nur für Programmierobjekte und das Kommando LIST oder FIND, wenn es im Batch-Modus verwendet wird. Wenn EXTEND *nicht* angegeben ist, wird eine Kurzliste der in der angegebenen Library enthaltenen Objekte angezeigt. Die Kurzliste enthält den Namen und den Typ des Objekts und gibt an, ob ein Quellcodeobjekt und/oder ein katalogisiertes Objekt existiert. Wenn EXTEND angegeben ist, wird eine erweiterte Liste der in der angegebenen Library enthaltenen Objekte angezeigt. Zusätzlich zu den Informationen, die angezeigt werden, wenn EXTEND nicht angegeben ist, enthält die erweiterte Liste Informationen aus dem Objektverzeichnis: Programmiermodus, Natural-Version, Benutzerkennung, Datum und Uhrzeit der Speicherung/Katalogisierung und den Quellcode des Objekts (falls vorhanden).
FROM oder FM oder IN	*lib-name*	A8	Für Programmierobjekte, Predict Sets, Debug-Umgebungen und Fehlermeldungen: Gibt eine Quell-Library an. Für Profile, Rules, und DDMs: Fügt eine Where-Klausel (*where-clause*) ein.
FMDATE oder FMDD	*date-from*	A10	Gilt nur für Programmierobjekte. Das Startdatum eines Zeitraums: Alle Objekte, die an oder nach dem angegebenen Datum gespeichert oder katalogisiert wurden, werden verarbeitet. Wird mit TODATE kein Endedatum angegeben, werden alle Objekte ab dem angegebenen Datum zur Bearbeitung ausgewählt. Die Angabe eines Datums muss entsprechend der Einstellung des Profilparameters DTFORM (siehe *DTFORM - Datumsformat* in der *Parameterreferenz*-Dokumentation) erfolgen, wie sie in der oberen rechten Ecke eines Menü-Bildschirms SYSMAIN angezeigt wird. Die Standardeinstellung ist das internationale Format *YYYY-MM-DD* (*YYYY* = Jahr, *MM* = Monat, *DD* = Tag), z.B. 2005-08-20.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
FMTIME oder FMTM oder FMTT	*time-from*	A5	Gilt nur für Programmierobjekte und wenn FMDATE angegeben ist. Gibt eine Startzeit an: Alle Objekte, die zum oder nach dem angegebenen Zeitpunkt (und Datum) gespeichert oder katalogisiert wurden, werden verarbeitet. Die Uhrzeit muss im Format *HH*:*II* (*HH* = Stunden, *II* = Minuten) angegeben werden, z. B. 11:33.
FNR	*fnr*	N5	Die Dateinummer (FNR) einer Quell- oder einer Ziel-Systemdatei (FNAT, FDIC oder FUSER). Die Quell-Systemdatei enthält das zu bearbeitende Objekt. Die Ziel-Systemdatei ist die Systemdatei, in die das Objekt kopiert oder verschoben werden soll, oder in der das Objekt gegebenenfalls umbenannt (oder im Falle einer Fehlermeldung neu nummeriert) wird. Gültige Dateinummern: 1 bis 65535. Wenn keine Datenbankkennung (DBID) oder Dateinummer (FNR) angegeben ist und SYSMAIN mit dem Systemkommando SYSMAIN oder dem Subprogramm MAINUSER aufgerufen wird (siehe auch ***SYSMAIN mittels Anwendungsprogrammierschnittstelle aufrufen***), gilt folgendes: Es werden immer die DBID und FNR der Systemdatei verwendet, aus der SYSMAIN aufgerufen wurde. Beispiel: Wenn Sie SYSMAIN aus einer Library aufrufen, die in der Systemdatei FUSER enthalten ist, werden die DBID und FNR dieser Datei verwendet. Für Rules und DDMs gilt: Anstelle von FNR kann auch die entsprechende DICDIC-Angabe verwendet werden. Wenn *fnr* zweimal angegeben wird, wird die zuletzt angegebene verwendet.
HELP oder ?	-	-	Aktiviert die online-**selektive Verarbeitung**. Sie können entweder das Schlüsselwort HELP in die With-Klausel (*with-clause*) aufnehmen oder ein Fragezeichen (?) als letztes Zeichen eines Objektnamens eingeben.
LANGUAGE	*language*	A9	Gilt nur für Fehlermeldungen. Der Sprachcode für die zu bearbeitende Fehlermeldung. Die Sprachen können durch eine beliebige Kombination von Sprachcodes angegeben werden. Informationen darüber,

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
			welcher Sprachcode welcher Sprache zugeordnet ist, finden Sie unter *Sprachcode-Zuordnungen* in **LANGUAGE* in der *Systemvariablen*-Dokumentation. Siehe auch ***Sprachen angeben***. Sie können einen Stern (*) verwenden, um alle vorhandenen Sprachen der zu bearbeitenden Fehlermeldungen auszuwählen.
`LIBRARY` oder `APPLIC` oder `APL`	*`lib-name`*	A8	Gilt nur für Programmierobjekte, Predict Sets, Debug-Umgebungen und Fehlermeldungen. Ein optionales Schlüsselwort, das den Namen einer Quell- oder Ziel-Library angibt. Wenn Sie das Schlüsselwort und den entsprechenden Wert weglassen, wird die Library, bei der Sie sich angemeldet haben, bevor Sie SYSMAIN aufgerufen haben, für die Verarbeitung verwendet. Die Quell-Library enthält das zu verarbeitende Objekt. Die Ziel-Library ist die Library, in die das Objekt kopiert oder verschoben werden soll oder in der das Objekt umbenannt wird (oder im Falle einer Fehlermeldung neu nummeriert wird). Bei Systemfehlermeldungen ist als *`lib-name`* `NATURAL-SYSTEM` oder `NATURAL-SYS` anzugeben. *`lib-name`* muss unmittelbar nach den Schlüsselwörtern `FROM` und `TO` angegeben werden. Wenn das optionale Schlüsselwort `LIBRARY` verwendet wird, muss es zwischen `FROM` oder `TO` und *`lib-name`* angegeben werden.
`MON` oder `NOMON` oder `MONOFF`	-	-	Gilt nicht im Batch-Modus. Aktiviert (`MON`) oder deaktiviert (`NOMON` oder `MONOFF`) die Verfolgung (Tracing) der aktuellen Aktivität in SYSMAIN. Während der Bearbeitung werden Sie darüber informiert, welches Objekt gelesen, gelöscht, geändert, hinzugefügt wird und ob ein Fehler auftritt. Bei Programmierobjekten werden Sie auch darüber informiert, welche Maßnahme mit den XRef-Daten durchgeführt wird. Diese Funktion ist nur in TP-Umgebungen wirksam, die im nicht-konversationellen Modus laufen können.
`NAME`	*`vsam-name`*	A8	Der DDNAME/FCT-Eintrag für die Quell- oder Zieldateinummer.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
PROMPT oder NOPROMPT	-	-	Gilt nicht im Batch-Modus. Aktiviert (PROMPT) oder deaktiviert (NOPROMPT) die SYSMAIN-Eingabeaufforderungen. Bei NOPROMPT wird kein Bestätigungsbildschirm angezeigt. So werden Sie beispielsweise vor jedem Löschvorgang von SYSMAIN zur Bestätigung aufgefordert.
PASSWORD oder PSW	*password*	A8	Das Adabas-Passwort einer Quelldatei und/oder einer Zielsystemdatei, die in einer Where-Klausel (*where-clause*) verwendet wird.. Für Rules und DDMs gilt: Anstelle von PASSWORD kann auch die entsprechende DIC-Angabe verwendet werden. Wenn das Passwort zweimal angegeben wird, wird das zuletzt angegebene verwendet.
PROFILE	*name*	A8	Gilt nur für Profile. Der Name des zu bearbeitenden Profils oder ein Bereich von Namen. Siehe auch ***Angabe eines Namensbereichs***.
RCOP	-	-	Gibt an, dass eine Kopie des umzubenennenden Objekts erstellt werden soll.
REPLACE	-	-	Aktiviert die Option replace, die in einer With-Klausel (*with-clause*) verwendet wird. Ein Objekt mit dem gleichen Namen in der Zielumgebung wird durch das zu bearbeitende Objekt ersetzt. **Anmerkung:** Wenn ein Programmierobjekt ersetzt wird, wird es auch aus dem Natural Buffer Pool gelöscht; eventuell vorhandene Cross-Reference Records werden ebenfalls gelöscht, wenn Predict installiert ist.
RULE	*name*	A32	Gilt nur für Rules. Der Name der zu verarbeitenden Rule oder ein Bereich von Namen. Siehe auch ***Angabe eines Namensbereichs***.
SAVED	*name*	A9	Gilt nur für Programmierobjekte. Der Name des gespeicherten (Quellcode-)Objekts, das bearbeitet werden soll, oder ein Namensbereich. Siehe auch ***Angabe eines Namensbereichs***.
SEC	*dbid* *fnr* *password* *cipher*	A80	Gilt nicht für Profile. Gibt die Umgebung der FSEC-Quell- und/oder Ziel-Systemdatei an: Datenbankkennung (*dbid*),

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
			Dateinummer (*fnr*), Adabas-Passwort (*password*) und Adabas-Chiffrierschlüssel (*cipher*).
SET	*Set-number*	N2	Gilt nur für Predict Sets. Die Nummer des Predict Set oder die Startnummer eines Bereichs von Predict Sets, die verarbeitet werden sollen.
SETNO	*set-number*	N2	Gilt nur für Programmierobjekte. Die Nummer des beibehaltenen Predict Set, das mit der Option `Predict XRef save set` des Kommandos `LIST XREF` erstellt wurde. Sie können alle SYSMAIN-Verarbeitungsfunktionen bei den in diesem Set enthaltenen Objekten anwenden. Wenn eine gültige Nummer angegeben wird, geht SYSMAIN von einem Predict Set aus. Wenn keine Nummer angegeben wird, wird die normale Objektverarbeitung angenommen.
SETLIBRARY	*set-library*	A8	Gilt nur für Programmierobjekte. Aktiviert die Option zum Überschreiben der Library-Angabe für ein Predict Set als Teil der Security für Predict-Dateien. `SETLIBRARY` wird nur ausgewertet, wenn für `SETNO` eine gültige Nummer angegeben wurde.
SETUSER	*set-user*	A8	Für Programmierobjekte: Aktiviert die Option, die Angabe der Benutzerkennung für ein Predict Set als Teil der Security für Predict-Dateien zu überschreiben. `SETUSER` wird nur ausgewertet, wenn für `SETNO` eine gültige Nummer angegeben wurde. Für Predict Sets: Die Angabe der Benutzerkennung der zu verarbeitenden Predict Sets. Bei Anwendung auf die Quellangabe wird sie als Auswahlkriterium verwendet. Wenn sie auf die Zielangabe angewendet wird, wird sie als neuer Wert verwendet. Es ist nicht möglich, Bereiche anzugeben.
STOWED oder BOTH	*name*	A9	Gilt nur für Programmierobjekte. Der Name eines Objekts (oder ein Namensbereich), für das das gespeicherte (Quellcode) und das katalogisierte Objekt bearbeitet werden sollen (siehe auch ***Angabe eines Namensbereichs***). Es wird nur ein Objekt bearbeitet, das

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
			sowohl als gespeichertes Objekt als auch als katalogisiertes Objekt existiert. Ausnahmen sind Copycode, Text und Recording, die nicht katalogisiert werden können. Sie werden jedoch in die Verarbeitung einbezogen, wenn diese Option angegeben ist.
THRU	*number* oder *new-number*	N4	Gilt nur für Fehlermeldungen. Die Endnummer eines Bereichs von zu verarbeitenden Fehlermeldungsnummern, wenn eine Startnummer mit AS angegeben wird.
TID	*terminal-ID*	A8	Gilt nur für Programmierobjekte. Eine Terminalkennung: Alle Objekte, die mit dem angegebenen Terminal gespeichert oder katalogisiert wurden, werden verarbeitet.
TO	*lib-name*	A8	Für Programmierobjekte, Predict Sets, Debug-Umgebungen und Fehlermeldungen: Gibt eine Ziel-Library an. Für Profile, Rules und DDMs: Fügt eine Where-Klausel (*where-clause*) ein.
TODATE oder TODD	*date-to*	A10	Gilt nur für Programmierobjekte. Das Enddatum eines Zeitraums: Alle Objekte, die an oder vor dem angegebenen Datum gespeichert oder katalogisiert wurden, werden bearbeitet. Ein Startdatum kann mit FMDATE angegeben werden. Ein Datum muss entsprechend der Einstellung des Profilparameters DTFORM (siehe *DTFORM - Datumsformat* in der *Parameter-Referenz*-Dokumentation), wie in der oberen rechten Ecke eines Menübildschirms SYSMAIN angezeigt. Die Standardeinstellung ist das internationale Format *YYYY-MM-DD* (*YYYY* = Jahr, *MM* = Monat, *DD* = Tag), z.B. 2005-08-20.
TOSET	*Set-number*	N2	Gilt nur für Predict Sets. Die letzte Nummer eines Bereichs von Predict Sets, die verarbeitet werden sollen.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
TOTIME oder TOTT oder TOTM	*time-to*	A5	Gilt nur für Programmierobjekte und wenn TODATE angegeben ist. Die Endzeit eines Zeitraums: Alle Objekte, die zu oder vor dem angegebenen Zeitpunkt (und Datum) katalogisiert oder gespeichert wurden, werden bearbeitet. Die Zeit muss im Format *HH*:*II* (*HH* = Stunden, *II* = Minuten), z.B. 11:33, angegeben werden.
TYPE	*type*	-	Der Typ des zu verarbeitenden **Programmierobjekts**, der **Fehlermeldung**, des **Profils** oder der **Rule**, wie unten in ***TYPE-Angabe*** aufgeführt.
USER oder USR	*user-id*	A8	Gilt nur für Programmierobjekte. Eine Benutzerkennung: Alle Objekte, die von dem angegebenen Benutzer gespeichert oder katalogisiert wurden, werden verarbeitet.
WHERE	*where-clause*	-	Ein optionales Schlüsselwort, das den Beginn einer Where-Klausel (*where-clause*) angibt. Die Where-Klausel muss immer auf das FROM- oder TO-Schlüsselwort und den *lib-name* (falls zutreffend) folgen. Die Angabe der Schlüsselwörter und Werte innerhalb der Klausel kann in beliebiger Reihenfolge erfolgen. Einzelheiten finden Sie in der Syntax des Direktkommandos in den objekttypspezifischen Abschnitten dieser Dokumentation.
WITH	*with-clause*	-	Ein optionales Schlüsselwort, das den Beginn einer With-Klausel (*with-clause*) angibt. Die Schlüsselwörter und Werte der With-Klausel können in beliebiger Reihenfolge angegeben werden, und die With-Klausel kann an einer beliebigen Stelle innerhalb der Direktkommandozeichenfolge stehen, außer an den ersten drei Positionen. Einzelheiten finden Sie in der Syntax des Direktkommandos in den objekttypspezifischen Abschnitten dieser Dokumentation.
XREF	F oder N oder S oder	A1	Gilt nur für Programmierobjekte. Gibt an, ob Cross-Referenz-Daten (XRef), die in Predict-Systemdateien gespeichert sind, verarbeitet werden sollen.

Schlüsselwort	Wert	Natural-Daten-Format/Länge	Erläuterung
	Y oder D		Sie können einen der folgenden Werte angeben: F: Alle XRef-Daten werden verarbeitet und das Objekt muss in Predict dokumentiert sein. N: XRef-Daten werden nicht verarbeitet, außer bei Verwendung des DELETE-Kommandos. Wenn ein katalogisiertes Objekt gelöscht oder ersetzt wird, löscht SYSMAIN immer alle vorhandenen XRef-Daten zu diesem Objekt. S: Ein angegebenes Objekt wird verarbeitet, unabhängig davon, ob es XRef-Daten hat oder nicht. Y: Alle XRef-Daten werden verarbeitet. D: Das Objekt muss in Predict dokumentiert sein. Alle vorhandenen XRef-Daten werden verarbeitet. Weitere Einzelheiten siehe *Besonderheiten bei XRef*.

TYPE-Angabe - Programmierobjekte

Natural-Datenformat/Länge: A20

In der folgenden Tabelle sind alle gültigen Objekttypcodes für Programmierobjekte aufgeführt:

Code	Objekttyp
P	Programm
N	Subprogramm
S	Subroutine
M	Map (Maske)
H	Helproutine
0	ISPF Macro
3	Dialog
5	Processor
A	Parameter-Datenbereich (PDA)

Code	Objekttyp
G	Globaler Datenbereich (GDA)
L	Lokaler Datenbereich (LDA)
C	Copycode
T	Text
R	Report
Z	Recording
4	Class
7	Function
8	Adapter
9	Resource
*	Alle Programmierobjekttypen

TYPE-Angabe - Fehlermeldungen

Natural-Datenformat/Länge: A1

In der folgenden Tabelle sind alle gültigen Typcodes für Fehlermeldungen aufgeführt:

Code	Typ
S	Kurze Fehlermeldung
E	Erweiterte (lange) Fehlermeldung
A	Alle Fehlermeldungstypen: Kurze und/oder erweiterte Fehlermeldungen.

TYPE-Angabe - Profile

Natural-Datenformat/Länge: A3

In der folgenden Tabelle sind alle gültigen Typcodes für Profile aufgeführt:

Code	Typ
E	Editor-Profil
D	Device-Profil
M	Map-Profil
P	Parameter-Profil
*	Alle Profiltypen.

TYPE-Angabe - Rules

Natural-Datenformat/Länge: A2

In der folgenden Tabelle sind alle gültigen Typcodes für Rules aufgeführt:

Code	Typ
A	Automatische Rule
F	Freie Rule
AF	Alle Rule-Typen: automatische und/oder freie Rules.

Angabe eines Namensbereichs

Alle SYSMAIN-Funktionen bieten die Möglichkeit, entweder einen Namen oder einen Bereich von Namen für die zu bearbeitenden Objekte anzugeben. Darüber hinaus können Sie im Menümodus auf dem Bildschirm **Find Selection** oder **List Selection** einen Namen oder einen Namensbereich angeben, um die Anzahl der angezeigten Objekte zu begrenzen. Siehe auch ***Um eine Auswahlliste zu verkürzen*** in *Auswahlliste benutzen*.

Wenn Sie die **Find**- oder **List**-Funktion bei Programmierobjekten verwenden, können Sie auch einen Bereich von Library-Namen angeben. Dasselbe gilt, wenn Sie die **List**-Funktion bei Debug-Umgebungen oder die **Find**-Funktion bei Fehlermeldungen verwenden. Die Angabe von Library-Bereichen kann sich jedoch negativ auf die Antwortzeit auswirken, je nachdem, wie oft die Auswahlkriterien auftreten.

Die gültigen Schreibweisen für Namensbereiche sind im Folgenden aufgeführt, wobei *value* (Wert) für eine beliebige Kombination aus einem oder mehreren Zeichen steht:

Eingabe	Ausgewählte Elemente
*	Alle Elemente.
*value**	Alle Elemente, deren Namen mit *value* beginnen. Beispiel: AB* Ausgewählt: AB, AB1, ABC, ABEZ Nicht ausgewählt: AA1, ACB
value>	Alle Elemente, deren Namen größer oder gleich *value* sind. Beispiel: AB> Ausgewählt: AB, AB1, BBB, ZZZZZZZ Nicht ausgewählt: AA1, AAB
value<	Alle Elemente, deren Namen kleiner oder gleich *value* sind. Beispiel: AX<

Eingabe	Ausgewählte Elemente
	Ausgewählt: AB, AWW, AX Nicht ausgewählt: AXA, AY

80 An SYSMAIN absetzbare spezielle Kommandos

Es gibt Kommandos, die an das SYSMAIN-Dienstprogramm abgesetzt werden können, um Funktionen auszuführen, die mit dem Betrieb des Dienstprogramms selbst zusammenhängen, oder um die Security für Natural-Systemdateien zu definieren.

Kommando	Funktion
ADAON oder NOADA oder ADAOFF	Aktiviert (ADAON) oder deaktiviert (NOADA oder ADAOFF) die Error Trap-Funktion, um abnormale Datenbankfehler zu Debugging-Zwecken abzufangen (nur online mit Programmierobjekten anwendbar).
BATCH oder NOBATCH	Schaltet das Dienstprogramm SYSMAIN in den Batch-Modus (BATCH), wobei alle Verarbeitungen so durchgeführt werden, als ob SYSMAIN im Batch-Modus laufen würde. Bei Verwendung der automatisierten Online-Verarbeitung wird dann ein Batch-Report angezeigt. Siehe auch ***Online-Report-Modus***. NOBATCH schaltet das SYSMAIN-Dienstprogramm zurück in den Online-Modus.
CLEAR	Löscht den aktuellen Arbeitsbereich. Diese Funktion kann sinnvoll sein, wenn der Quellcode eines großen Objekts im Arbeitsbereich enthalten ist und das SYSMAIN-Dienstprogramm deshalb einen größeren Buffer-Erweiterungsbereich (ESIZE) benötigt.
DISPLAY	Anzeige des erweiterten (langen) Meldungstextes zu dem aufgetretenen Fehler.
FINDFIRST	Stoppt die Suche nach Programmierobjekten FIND-Kommando, wenn die erste Library gefunden wird, die das angegebene Objekt enthält.
FINDALL	Bewirkt, dass das FIND-Kommando alle Libraries nach Programmierobjekten durchsucht.
MON oder NOMON oder MONOFF	Aktiviert (MON) oder deaktiviert (NOMON oder MONOFF) die Verfolgung (Tracing) der aktuellen Aktivität in SYSMAIN. Siehe auch MON in *Schlüsselwörter und Variablen in Direktkommandos*.

Kommando	Funktion
PROMPT oder NOPROMPT	Aktiviert (PROMPT) oder deaktiviert (NOPROMPT) die SYSMAIN-Eingabeaufforderungen. Siehe auch PROMPT in *Schlüsselwörter und Variablen in Direktkommandos*.
SET	Ruft das Kommando-Hilfe-Fenster (**Command Help**) auf, in dem alle speziellen SYSMAIN-Kommandos erklärt werden.
SET FDIC	Ruft ein Fenster auf, in dem Sie Adabas-Security-Informationen zu der Predict-Systemdatei aufrufen können. Dies bezieht sich auf den Profilparameter FDIC (siehe FDIC - Predict System File in der Parameter-Referenzdokumentation). Im Batch-Modus können Sie mit dem Schlüsselwort SEC in einer Where-Klausel (*where-clause*) Sicherheitsinformationen angeben, wie in der Syntax des Direktkommandos in den objekttypspezifischen Abschnitten dieser Dokumentation beschrieben. Siehe auch ***Besondere Hinweise für Administratoren***.
SET FNAT	Ruft ein Fenster auf, in dem Sie Adabas-Sicherheitsinformationen für die SYSMAIN-Quell- und Ziel-Systemdateien angeben können. Im Batch-Modus können Sie mit dem Schlüsselwort SEC in einer Where-Klausel (*where-clause*) Sicherheitsinformationen angeben, wie in der Syntax des Direktkommandos in den objekttypspezifischen Abschnitten dieser Dokumentation beschrieben. Siehe auch ***Besondere Hinweise für Administratoren***.
SET FSEC	Ruft ein Fenster auf, in dem Sie die Adabas-Security-Informationen für die Systemdatei FSEC angeben können, wenn Natural Security installiert ist. Dies bezieht sich auf den Profilparameter FSEC (siehe *FSEC - Natural Security-Systemdatei* in der *Parameter-Referenz* -Dokumentation). Siehe auch ***Besondere Hinweise für Administratoren***.
SET PC	Gilt nur, wenn Natural Connection und Entire Connection installiert sind. Aktiviert die PC-Verbindung. Diese Einstellung kann zeitweise mit den Terminalkommandos %+ und %- geändert werden (siehe auch *Natural Connection ein-/ausschalten* in der *Terminalkommandos*-Dokumentation). SET PC bewirkt dann, dass SYSMAIN den Status der PC-Verbindung erneut prüft.
STATUS	Anzeige der aktuellen Werte von SYSMAIN-Variablen, die für den Support wichtig sind.
TOTAL	Ruft das Fenster **Results of Function** auf, in dem die Verarbeitung der zuletzt ausgeführten SYSMAIN-Funktion überprüft wird. Die folgenden Informationen werden für gespeicherte (Quellcode) und katalogisierte Objekte angezeigt:

Read	Gesamtzahl der Objekte, die tatsächlich gelesen wurden, basierend auf der Angabe des Objektnamens.
Rejected	Gesamtzahl der gelesenen Objekte, die anschließend aufgrund der angegebenen Auswahlkriterien abgelehnt wurden. Siehe auch ***Ablehnung von Objekten und Ursache***.
Processed	Gesamtzahl der Objekte, die verarbeitet wurden.

Kommando	Funktion	
	Added	Gesamtzahl der neuen Objekte, die der Zielumgebung hinzugefügt wurden.
	Updated	Gesamtzahl der vorhandenen Objekte, die aktualisiert wurden. (Wenn möglich, versucht SYSMAIN, vorhandene Objekte zu aktualisieren, anstatt sie zu löschen und neue hinzuzufügen.)
	Deleted	Gesamtzahl der Objekte, die entweder aus der Quell- oder aus der Zielumgebung gelöscht wurden, je nach Funktion und Einstellung der Ersetzungsoption.
	Replaced	Gesamtzahl der Objekte, die in der Zielumgebung ersetzt wurden.
	Not Repl.	Gesamtzahl der Objekte, die in der Zielumgebung *nicht* ersetzt wurden.
	Recs.Read:	Gesamtzahl der Datensätze, die gelesen wurden.

.
 	
oder
`END`
 	
oder
`QUIT`
	Beendet das Dienstprogramm SYSMAIN.	

81 Statusmeldungen und Fehlerbenachrichtigungen bearbeiten

- Ablehnung von Objekten und Ursache 780
- Status-Meldungen 782
- SYSMAIN-Fehlerbenachrichtigung 786

Dieses Kapitel beschreibt mögliche Gründe für die Ablehnung von Objekten während der Funktionsverarbeitung, die nach der Verarbeitung angezeigten Statusmeldungen und die Fehlerbenachrichtigung während der Verarbeitung.

Ablehnung von Objekten und Ursache

Wenn bei der Ausführung einer SYSMAIN-Funktion ein oder mehrere Objekte gefunden wurden, die den angegebenen Auswahlkriterien entsprechen, aber einige oder alle dieser Objekte zur weiteren Verarbeitung abgelehnt wurden, tritt einer der folgenden Natural-Systemfehler auf:

Nummer	Fehlertext
4867	`Nothing found for this request.`
	Zu dieser Anforderung wurde nichts gefunden.
4810	`All data rejected by these selection criteria.`
	Daten wurden aufgrund dieser Auswahlkriterien abgelehnt.
4893	`Normal completion, but some data were rejected.`
	Normale Erledigung, aber einige Daten wurden abgelehnt.

Sie können das SYSMAIN-Kommando `TOTAL` (siehe *An SYSMAIN absetzbare spezielle Kommandos*) benutzen, um den individuellen Status einer Anfrage zu überprüfen.

Mögliche Gründe für die Ablehnung von Objekten sind im folgenden Abschnitt beschrieben.

- Ungültige Angaben zu Objekttyp, Datum/Uhrzeit, Benutzer oder Terminal
- Identischer Name in der Zielumgebung
- XREF und User Exits
- Keine kurze oder erweiterte Fehlermeldung
- Library-Einschränkungen

Ungültige Angaben zu Objekttyp, Datum/Uhrzeit, Benutzer oder Terminal

- Ein Objekt wurde ausgewählt und dann abgelehnt, weil der Objekttyp für die angegebene Art der Verarbeitung nicht gültig war. Bei der Verarbeitung von Programmierobjekten werden beispielsweise alle Maps abgelehnt, wenn die Verarbeitung auf Objekte vom Typ Programm oder Subroutine beschränkt wurde.
- Ein Objekt wurde ausgewählt und dann abgelehnt, weil das Datum oder die Uhrzeit, zu der das angegebene Objekt gespeichert oder katalogisiert wurde, nicht in dem angegebenen Datums- oder Zeitbereich liegt.
- Ein Objekt wurde ausgewählt und dann abgelehnt, weil die Benutzerkennung oder Terminal-Kennung, unter der das angegebene Objekt gespeichert oder katalogisiert wurde, nicht mit der angegebenen Benutzerkennung oder Terminal-Kennung übereinstimmte.

- Ein katalogisiertes Objekt wurde ausgewählt und dann abgelehnt, weil der Profilparameter RECAT auf ON gesetzt war und es kein gespeichertes (Quellcode-)Objekt gab, das dem katalogisierten Objekt entsprach. Siehe auch *Profilparameter RECAT verwenden*.
- Es wurde (nur) ein gespeichertes Objekt ausgewählt und dann abgelehnt, weil der Profilparameter RECAT auf ON gesetzt war und die Zielumgebung bereits ein katalogisiertes Objekt mit demselben Namen enthielt. Siehe auch *Profilparameter RECAT verwenden*.

Identischer Name in der Zielumgebung

- Ein Objekt wurde ausgewählt und dann abgelehnt, weil die Zielumgebung bereits ein Objekt mit demselben Namen enthielt und die Ersetzungsoption nicht aktiviert war (Feld **Replace** auf N gesetzt oder Schlüsselwort REPLACE nicht angegeben).
- Ein katalogisiertes Programmierobjekt vom Typ S (Subprogramm) wurde ausgewählt und dann abgelehnt, weil der Name der externen Subroutine mit dem Namen einer anderen Subroutine in der Ziel-Library identisch war.
- Ein katalogisiertes Programmierobjekt vom Typ 4 (Class) wurde ausgewählt und dann zurückgewiesen, weil der Name der externen Class mit dem Namen einer anderen Class in der Ziel-Library identisch war.

XREF und User Exits

- Die Option **XREF** war aktiviert (Feld XREF nicht auf N gesetzt oder Schlüsselwort XREF angegeben) und es waren keine XRef-Daten für das angegebene Programmierobjekt vorhanden.
- Eine User-Exit-Routine war aktiv, und bei der Verarbeitung des Objekts wurde ein Rückgabe-Code ungleich Null zurückgegeben.

Keine kurze oder erweiterte Fehlermeldung

- Eine erweiterte (lange) Fehlermeldung wurde ausgewählt und dann abgelehnt, weil keine entsprechende kurze Fehlermeldung in der Quell-Library vorhanden war.
- Eine erweiterte (lange) Fehlermeldung wurde ausgewählt, konnte aber nicht verarbeitet werden, weil es in der Zielumgebung keine entsprechende kurze Fehlermeldung gab.
- Eine kurze Fehlermeldung wurde zum Verschieben (**Move**), Löschen (**Delete**) oder Umnummerieren (**Rename**) ausgewählt, konnte aber nicht verarbeitet werden, weil der entsprechende erweiterte Fehlertext nicht in den Auswahlkriterien enthalten war. Zu einer erweiterten Fehlermeldung muss immer eine entsprechende kurze Fehlermeldung vorhanden sein.

Library-Einschränkungen

Es wurde eine Library angegeben, die von Predict Application Control/Predict Application Audit kontrolliert wird, und das Objekt kann nicht von SYSMAIN behandelt werden.

Status-Meldungen

Die folgende Tabelle beschreibt die Statusmeldungen, die in der Spalte **Message** eines Auswahlbildschirms oder in einem Batch-Report (siehe auch *Online Report-Modus*) angezeigt werden können, nachdem eine SYSMAIN-Funktion auf einem Objekt ausgeführt wurde. Die erwähnten Zeilenkommandos finden Sie unter Auswahllisten in den objekttypspezifischen Abschnitten dieser Dokumentation.

Meldung	Erläuterung	Funktion abgeschlossen?
`Class Exists`	Gilt nur für Programmierobjekte. Das katalogisierte Objekt einer Class hatte einen externen Class-Namen, der bereits von einem anderen katalogisierten Objekt einer Class in der Zielumgebung verwendet wurde.	Nein
`Copied`	Eines der folgenden Zeilenkommandos wurde von einem **Copy/Rename Selection**-Bildschirm ausgeführt: `A`, oder `A`, `C` oder `S` für Programmierobjekte, oder `A`, `E` oder `S`für Fehlermeldungen, oder `A`, `D`, `E`, `M` oder `P` für Profile.	Ja
`DB Error: nnn`	Für das Objekt wurde bei der Verarbeitung ein Datenbankfehler zurückgegeben.	Nein
`Deleted`	Eines der folgenden Zeilenkommandos wurde von einem **Delete Selection**-Bildschirm ausgeführt: `A`, oder `A`, `C` oder `S` für Programmierobjekte, oder `A`, `E` oder `S`für Fehlermeldungen, oder `A`, `D`, `E`, `M` oder `P` für Profile.	Ja
`Dev exists`	Gilt nur für Profile. Device existiert: Die Option **Replace** (Ersetzen) wurde auf `N` gesetzt und das Zeilenkommando `A` oder `D` wurde von einem **Copy/Move/Rename Selection**-Bildschirm (Kopieren/Verschieben/Umbenennen) ausgeführt. Das Device-Profil mit demselben Namen in der Zielumgebung wurde *nicht* ersetzt.	Nein
`Directory`	Verzeichnis: Das Zeilenkommando `I` (Verzeichnis anzeigen) wurde von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja

Meldung	Erläuterung	Funktion abgeschlossen?
`Edt exists`	Gilt nur für Profile. Editorprofil: Die **Replace**-Option wurde auf `N` gesetzt und das Zeilenkommando `A` oder `E` wurde von einem **Copy/Move/Rename Selection**-Bildschirm aus ausgeführt. Das gleichnamige Editorprofil in der Zielumgebung wurde *nicht* ersetzt.	Nein
`Err: NAT2999`	Gilt nur für Programmierobjekte. Ein katalogisiertes Objekt wurde bearbeitet, wobei die Option **XREF** auf `F` gesetzt war. In der für das Objekt angegebenen FDIC-Systemdatei existiert kein Predict-Eintrag.	Nein
`Err: NAT4852`	Eine Natural Security-Verletzung ist aufgetreten.	Nein
`Exit: nnn`	Eine Benutzer-Exit-Routine war aktiv, und der Exit lieferte einen Rückgabecode ungleich Null (*nnn* = der Rückgabecode). Siehe auch ***User-Exit-Routinen***.	Nein
`Exported`	Gilt nur für Programmierobjekte und DDMs. Das Zeilenkommando X (PC-Download) wurde für ein Quellcodeobjekt von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja
`Ext Exists`	Gilt nur für Fehlermeldungen. Es wurde versucht, nur die kurze Fehlermeldung einer Fehlermeldung zu löschen, für die eine lange Fehlermeldung existiert. Dies hätte zu einer erweiterten (langen) Fehlermeldung ohne entsprechende kurze Meldung geführt. Dies ist in Natural nicht zulässig.	Nein
`File Listed`	Gilt nur für Rules. Das Zeilenkommando `F` (DDMs/Felder anzeigen) wurde bei einer automatischen Rule von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja
`In Use`	In Verwendung: Bei der Natural `UPDATE/READ`-Verarbeitung eines Objekts wurde ein Adabas-Antwortcode 145 zurückgegeben.	Nein
`Invalid`	Gilt nur für Debug-Umgebungen. Für eine der auf einem Auswahlbildschirm (**Selection**) aufgelisteten Debug-Umgebungen wurde ein ungültiges Zeilenkommando eingegeben.	Nein
`Listed`	Gilt nicht für DDMs und Debug-Umgebungen. Das Zeilenkommando `L` (Quellcode anzeigen) wurde von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja
`Map exists`	Gilt nur für Profile. Die Option **Replace** wurde auf `N` gesetzt und das Zeilenkommando `A` oder `M` wurde von einem Auswahlbildschirm (**Selection**) für **Copy/Move/Rename** ausgeführt. Das gleichnamige Map-Profil in der Zielumgebung wurde *nicht* ersetzt.	Nein

Meldung	Erläuterung	Funktion abgeschlossen?
`Moved`	Eines der folgenden Zeilenkommandos wurde von einem **Move Selection**-Bildschirm ausgeführt: `A`, oder `A`, `C` oder `S` für Programmierobjekte, oder `A`, `E` oder `S` für Fehlermeldungen, oder `A`, `D`, `E`, `M` oder `P` für Profile.	Ja
`Must be Auto`	Gilt nur für Rules. Das Zeilenkommando `F` (DDMs/Felder anzeigen) wurde bei einer freien Rule von einem Auswahlbildschirm (**Selection**) aus ausgeführt. Das Zeilenkommando `F` kann nur bei automatischen Rules ausgeführt werden.	Nein
`Name Erro`	Gilt nur für Debug-Umgebungen. Die Funktion **Rename** wurde verwendet, aber der angegebene neue Name wurde als ungültig erkannt. Entweder wurde kein neuer Name für die Auswahl angegeben, oder der angegebene Name enthielt entweder ungültige Sonderzeichen oder begann nicht mit einem alphabetischen Zeichen.	Nein

`NBP Deleted`
oder
`Ignored`
	Gilt nur für Programmierobjekte. Das Zeilenkommando `B` (Objekt aus dem Bufferpool löschen) wurde von einem Auswahlbildschirm (**Selection**) aus ausgeführt. Die nach Abschluss der Verarbeitung zurückgegebene Meldung lautet entweder `NBP Deleted` (NBP Gelöscht) oder `Ignored` (Ignoriert), je nachdem, ob das Löschen aus dem Bufferpool für das/die angegebene(n) Objekt(e) bestätigt wurde oder nicht.	Ja/Nein

`No Lang 1`	Gilt nur für Fehlermeldungen. Für die erweiterten (langen) Natural-System-Fehlermeldungen ist nur der Sprachcode `1` (Englisch) verfügbar. Es wurde versucht, eine erweiterte Natural-System-Fehlermeldung zu kopieren, und Sprachcode `1` wurde nicht als Sprache angegeben.	Nein
`no Short Err`	Gilt nur für Fehlermeldungen. Es wurde eine erweiterte (lange) Fehlermeldung zur Weiterverarbeitung ausgewählt, aber die Zielfehlermeldungsnummer hat keine entsprechende kurze Fehlermeldung.	Nein
`No Xref`	Gilt nur für Programmierobjekte. Ein katalogisiertes Objekt wurde bearbeitet und die Option **XREF** war auf `Y` oder `F` gesetzt. In der für das Objekt angegebenen FDIC-Systemdatei sind keine XRef-Daten vorhanden.	Nein
`Not Found`	Gilt nur für Fehlermeldungen und Profile. Bei der Verarbeitung ist ein Fehler in der Update-Logik aufgetreten und die angeforderte Fehlermeldung oder das angeforderte Profil konnte nicht gefunden werden. Dies bedeutet, dass die angegebene Fehlermeldung oder das angegebene Profil in der Zeit zwischen Auswahl und Update gelöscht wurde.	Nein

Meldung	Erläuterung	Funktion abgeschlossen?

`Not Replaced`
 	
oder
 	
`Not Repld`
	Die **Replace**-Option wurde auf `N` gesetzt und eines der folgenden Zeilenkommandos wurde von einem **Copy/Move/Rename Selection**-Bildschirm aus ausgeführt:
	`A`, oder
	`A`, `C` oder `S` für Programmierobjekte, oder
	`A`, `E` oder `S`für Fehlermeldungen, oder
	`A`, `D`, `E`, `M` oder `P` für Profile.
	
	Das Objekt mit dem gleichen Namen in der Zielumgebung wurde *nicht* ersetzt.
		Nein

`Parm exists`	Gilt nur für Profile. Die **Replace**-Option wurde auf `N` gesetzt und das Zeilenkommando `A` oder `P` wurde von einem **Copy/Move/Rename Selection**-Bildschirm aus ausgeführt. Das Parameterprofil mit demselben Namen in der Zielumgebung wurde *nicht* ersetzt.	Nein
`Printed`	Gilt nur für Programmierobjekte und Rules. Das Zeilenkommando H (Hardcopy erzeugen) wurde von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja

`Renamed`
 	
oder
 	
`Copied`
	Gilt nur für Debug-Umgebungen und Fehlermeldungen.
	
	Eines der folgenden Zeilenkommandos wurde von einem **Rename Selection**-Auswahlbildschirm ausgeführt: `A` für Debug-Umgebungen, oder `A`, `E` oder `S` für Fehlermeldungen.
	
	Die Meldung, die nach Abschluss der Verarbeitung zurückgegeben wird, lautet entweder `Renamed` (Umbenannt) oder `Copied` (Kopiert), je nachdem, ob die Option zur Beibehaltung des ursprünglichen Objekts angegeben wurde.
		Ja

`Renamed as`
 	
oder
 	
`Copied as`
	Gilt nur für Programmierobjekte.
	
	Das Zeilenkommando `A`, `C` oder `S` wurde von einem **Rename Selection**-Bildschirm aus ausgeführt. Die nach Abschluss der Verarbeitung zurückgegebene Meldung lautet entweder `Renamed as` (Umbenannt in) oder `Copied as` (Kopiert in), je nachdem, ob die Option zur Beibehaltung des ursprünglichen Objekts angegeben wurde.
		Ja

`Replaced`	Die **Replace**-Option wurde auf `Y` gesetzt und einer der folgenden Zeilenbefehle wurde von einem **Copy/Move/Rename Selection**-Bildschirm aus ausgeführt: `A`, oder `A`, `C` oder `S` für Programmierobjekte, oder `A`, `E` oder `S`für Fehlermeldungen, oder `A`, `D`, `E`, `M` oder `P` für Profile. Das Objekt mit dem gleichen Namen in der Zielumgebung wurde ersetzt.	Ja

Meldung	Erläuterung	Funktion abgeschlossen?
`Sized`	Gilt nur für Programmierobjekte. Das Zeilenkommando `Z` (Größe berechnen) wurde von einem Auswahlbildschirm (**Selection**) aus ausgeführt.	Ja
`Subrtn Exists`	Gilt nur für Programmierobjekte. Das katalogisierte Objekt einer Subroutine hatte einen externen Subroutinenamen, der bereits von einer anderen katalogisierten Subroutine in der Zielumgebung verwendet wurde.	Nein
`Updated`	Gilt nur für Fehlermeldungen. Der Text in der angegebenen Sprache war für die gewählte Fehlermeldung vorher nicht vorhanden, und SYSMAIN hat die Fehlermeldung mit dem neuen Sprachtext aktualisiert.	Ja
`Src locked`	Gilt nur für Programmierobjekte. Das Zeilenkommando `A` oder `S` wurde von einem **Copy/Move/Rename/Delete Selection**-Bildschirm aus ausgeführt. Das Objekt wurde nicht verarbeitet, weil der Quellcode des betreffenden Objekts gesperrt war.	Nein

SYSMAIN-Fehlerbenachrichtigung

SYSMAIN versucht im Falle eines Laufzeitfehlers während der Verarbeitung immer eine Wiederherstellung zu erreichen. Diese Funktion wird automatisch aktiviert und verwendet die Systemvariable `*ERROR-TA` (siehe auch **ERROR-TA* in der *Systemvariablen*-Dokumentation). Diese Funktion wird deaktiviert, wenn SYSMAIN normal beendet wird.

Wenn ein Terminalkommando (siehe auch *%% und %. - Aktive Operation unterbrechen* in der *Terminalkommandos*-Dokumentation) zum Beenden von SYSMAIN verwendet, wird dies als abnormales Beenden betrachtet, und die Systemvariable `*ERROR-TA` wird nicht zurückgesetzt. Sie kann zurückgesetzt werden, indem SYSMAIN erneut aufgerufen und normal beendet wird. Falls Sie die Systemvariable `*ERROR-TA` gesetzt haben, setzt SYSMAIN sie beim Beenden auf den zuvor zugewiesenen Wert zurück.

Wenn in Bezug auf die Auswahlkriterien ungültige Daten angegeben wurden, wird eine Fehlermeldung in der Meldungszeile ausgegeben. Wenn Sie sich über die Bedeutung der kurzen Fehlermeldung unsicher sind, können Sie mit dem Spezialkommando `DISPLAY` (siehe auch *An SYSMAIN absetzbare spezielle Kommandos*) eine Anzeige des entsprechenden erweiterten (langen) Fehlermeldungstextes aktivieren.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Fehler bei der Dateneingabe
- Fehler bei der Verarbeitung

Fehler bei der Dateneingabe

Wenn in Bezug auf die Auswahlkriterien ungültige Daten angegeben wurden, wird in der Meldungszeile eine Fehlermeldung angezeigt. In einigen Fällen wird die Online-Hilfe für bestimmte Einträge aufgerufen. Dadurch erhalten Sie genauere Informationen zu dem Fehler.

Tritt ein Fehler im Batch-Modus auf, wird eine Fehlermeldung und die entsprechende Fehlernummer ausgegeben und das SYSMAIN-Dienstprogramm beendet.

Fehler bei der Verarbeitung

Wenn Sie eine Anforderung stellen, die einen Verarbeitungsfehler verursacht, gibt SYSMAIN einen Fehlerbericht aus, wie im folgenden Beispiel gezeigt:

```
16:51:08          *** SYSMAIN Error Report ***          2005-08-10

The following internal error occurred while processing the
SYSMAIN function xxxxxx (cc):

   Error in field specification for IF SELECTION statement.


Error Number .. eeee
Program ....... pppppppp
Status Code ... s          Status ...... tttttttt
Line .......... llll       Level ....... vv
Device ........ dddddddd
User ID ....... uuuuuuuu   User Name ... nn...nn
```

Die in dem Fenster enthaltenen Informationen sind nützlich, um die Fehlerursache zu analysieren.

Die Werte im obigen Fenster haben die folgenden Bedeutungen:

Feld	Wert	Erläuterung
SYSMAIN function	*xxxxxx*	Die SYSMAIN-Funktion, die gerade ausgeführt wird.
	cc	Ein interner Statuscode, der für den Support nützlich ist. Die folgenden Codes können angezeigt werden: A Automatisierte Verarbeitung. D XRef-Daten werden gelöscht. E Fehler bei der Verarbeitung (Flag für SYSMAIN). F Statuseinstellung, wenn XRef-Daten verarbeitet werden.

Feld	Wert	Erläuterung
		G Statuseinstellung, wenn XRef-Daten verarbeitet werden. H Verarbeitung der Auswahlliste. I Option wird gerade verarbeitet. S Einzelobjektverarbeitung. T SYSMAIN-Abbruch durch Kommandoprozessor. V Statuseinstellung, wenn XRef-Daten verarbeitet werden. X SYSMAIN-Abbruch durch den Kommandoprozessor. Y Es ist ein Validierungsfehler aufgetreten, eine erneute Anzeige sollte folgen. Z Es ist ein Validierungsfehler aufgetreten, eine erneute Anzeige sollte folgen.
Error Number	*eeee*	Entspricht der Systemvariablen *ERROR-NR (siehe *Systemvariablen* -Dokumentation).
Program	*pppppppp*	Entspricht der Systemvariablen *PROGRAM (siehe *Systemvariablen* -Dokumentation).
Status Code	*s*	Die Art des Fehlers. Mögliche Codes: C Fehler bei der Kommandoverarbeitung. L Anmeldefehler. O Objektzeitfehler. S Nicht korrigierbarer Syntaxfehler.
Line	*llll*	Entspricht der Systemvariablen *ERROR-LINE (siehe *Systemvariablen* -Dokumentation).
Device	*dddddddd*	Entspricht der Systemvariablen *DEVICE (siehe *Systemvariablen* -Dokumentation).
User ID	*uuuuuuuu*	Entspricht der Systemvariablen *USER (siehe *Systemvariablen* -Dokumentation).
Status	*tttttttt*	Entspricht der Systemvariablen *ERROR-TA (siehe *Systemvariablen* -Dokumentation).
Level	*vv*	Entspricht der Systemvariablen *LEVEL (siehe *Systemvariablen* -Dokumentation).
User Name	*nn...nn*	Entspricht der Systemvariablen *USER-NAME (siehe *Systemvariablen* -Dokumentation).

Wenn ein Verarbeitungsfehler auftritt, notieren Sie die Informationen im Fenster und drücken Sie ENTER. Das Dienstprogramm SYSMAIN versucht, zum letzten aktiven Menü-Bildschirm zurückzukehren, wobei die Datenwerte der Eingabefelder unverändert bleiben.

Wird in dem Fenster das spezielle Kommando `DISPLAY` eingegeben (siehe auch *An SYSMAIN absetzbare spezielle Kommandos*), wird der erweiterte (lange) Meldungstext zu dem aufgetretenen Fehler angezeigt.

Tritt während der Batch-Verarbeitung ein Verarbeitungsfehler auf, gibt das Dienstprogramm SYSMAIN die entsprechende Fehlermeldung aus und wird beendet.

Bestimmte Benutzerfehler können ebenfalls zur Anzeige des Fensters führen. Obwohl SYSMAIN versucht, alle Fehler während der Auswertung abzufangen, ist dies nicht immer ganz erfolgreich. Wenn ein Benutzer beispielsweise verlangt, dass ein DDM von einer Umgebung in eine andere kopiert wird, aber eine ungültige Datenbankkennung (DBID) angibt, versucht SYSMAIN, auf diese Datenbank zuzugreifen. Es wird ein Adabas-Antwortcode 148 zurückgegeben, und die Transaktion `SYSMAIN ERROR` wird aufgerufen und das Fenster angezeigt. In ähnlicher Weise kann eine ungültige Datei zu einer Reihe von Fehlern führen, die von der Datenbank gesendet werden.

In Situationen, in denen ein Adabas-Antwortcode `9` zurückgegeben wird, schreibt SYSMAIN eine Nachricht, die Sie über den Fehler informiert, und beginnt die Verarbeitung wieder bei der letzten Funktion oder dem letzten Menü. Wenn eine bestimmte Anforderung nicht erledigt wurde, können Sie davon ausgehen, dass der Antwortcode `9` zu einem `BACKOUT TRANSACTION` zur letzten nicht abgeschlossenen Transaktion führte.

82 Besondere Hinweise für Administratoren

- File Security 792
- Natural Security 793
- User-Exit-Routinen 794

In diesem Abschnitt werden die Security-Aspekte des Dienstprogramms SYSMAIN und der für SYSMAIN bereitgestellten User-Exit-Routinen beschrieben.

File Security

File Security (Dateisicherheit, d. h. Kennwörter und Chiffrierschlüssel) bezieht sich auf die Security, die für eine Systemdatei in einer Adabas- oder VSAM-Umgebung definiert wurde. Wenn für eine Systemdatei File Security definiert wurde, müssen Sie ein Passwort, einen Chiffrierschlüssel und/oder einen VSAM-Namen für die Quell- und/oder Ziel-Systemdatei angeben, bevor Sie eine SYSMAIN-Funktion ausführen. Andernfalls wird Adabas oder VSAM eine entsprechende Fehlermeldung ausgeben. Für die Standard-Systemdateien, die der Natural-Sitzung beim Start des SYSMAIN-Dienstprogramms zugeordnet sind, müssen Sie keine Security-Informationen angeben.

Um Passwörter und Chiffriercodes anzugeben:

1 Rufen Sie von einem beliebigen Menü des SYSMAIN-Dienstprogramms aus ein Security-Fenster für die gewünschte Systemdatei auf, indem Sie, wie in der folgenden Tabelle angegeben, entweder eine PF-Taste oder ein spezielles Kommando verwenden:

Systemdatei	Kommando	PF-Taste	Betroffene Objekte / Daten
FUSER, FNAT	`SET FNAT`	PF12	- Programmierobjekte - Debug-Umgebungen - Fehlermeldungen - Profile
FDIC	`SET FDIC`	PF11	- Rules - DDMs - XRef-Informationen
FSEC	`SET FSEC`	PF10	- Natural Security-Profil

Das Security-Fenster, das für die angegebene Systemdatei erscheint, sieht ähnlich aus wie das unten gezeigte Beispiel für die Systemdateien FUSER und FNAT:

```
+-------------------------------------------------+
!   --- Security for the Natural System Files ---   !
!                                                   !
!  Specify the password(s), cipher(s) and VSAM FCT  !
!  name(s) for the source/target file(s) below:     !
!                                                   !
!      - Source -               - Target -          !
! Library .... OLDLIB       Library .... NEWLIB     !
! Database ... 10           Database ... 10         !
! File ....... 32           File ....... 32         !
!                                                   !
! Password ...              Password ...            !
! Cipher .....              Cipher .....            !
! VSAM Name .. ________     VSAM Name .. ________   !
+-------------------------------------------------+
```

2 Geben Sie im Fenster das/die entsprechende(n) Passwort/Passwörter, den/die Chiffrierschlüssel und/oder den VSAM-Namen für die gewünschte Quell- und/oder Ziel-Systemdatei ein.

Anmerkung: Das Feld **Library** ist nur für die Verarbeitung von Programmierobjekten, Debug-Umgebungen oder Fehlermeldungen relevant.

Sobald die File Security definiert ist, verwendet das Dienstprogramm SYSMAIN diese Security-Informationen für alle nachfolgenden Verarbeitungen. Wenn Sie dann die Standard-Security-Informationen (die Sie bei der Initialisierung der Sitzung erhalten haben) verwenden möchten, müssen Sie das entsprechende Security-Fenster erneut aufrufen und die Felder für Passwort, Chiffrierschlüssel und/oder VSAM-Name löschen. Die Passwörter und Chiffrierschlüssel werden nicht angezeigt, d.h. auch wenn die Felder leer zu sein scheinen, sollten sie wieder gelöscht werden.

Natural Security

Bei der Verwendung des Dienstprogramms SYSMAIN in einer Umgebung mit Natural Security müssen zwei Aspekte berücksichtigt werden:

- Definition der Natural Security-Umgebung

- Einschränkung der Verwendung von SYSMAIN unter Natural Security

Definition der Natural Security-Umgebung

Die Quell- und Ziel-Libraries können sich in derselben Natural Security-Umgebung oder in zwei verschiedenen Natural Security-Umgebungen befinden. Diese Umgebungen müssen für das Dienstprogramm SYSMAIN definiert werden.

Die Definition der zu verwendenden Natural Security-Umgebung(en) erfolgt mit dem speziellen Kommando `SET FSEC`.

Standardmäßig werden die aktuellen FSEC-Einstellungen verwendet, die beim Start der Natural-Sitzung zugewiesen wurden. Wenn Sie diese Einstellungen (im Fenster **Security for Natural Security (FSEC) Files**) ändern, bleiben sie so lange gültig, bis sie durch den nächsten `SET FSEC`-Vorgang geändert werden. Im Batch- oder Direktkommando-Modus sollte das Schlüsselwort `SEC` verwendet werden, um die File Security und die Zuweisungen der Anforderung anzugeben.

Sobald die Quell- und Zielumgebungen festgelegt sind, verifiziert SYSMAIN sowohl die Quell- als auch die Ziel-Library mit Natural Security. (Quell- und/oder Ziel-Library und -Datei müssen der Datenbankkennung (DBID) und der Dateinummer (FNR) entsprechen, die im Security-Profil der Library angegeben sind. Sind diese Werte nicht angegeben, werden Standardwerte aus dem Security-Profil übernommen).

Einschränkung der Verwendung von SYSMAIN unter Natural Security

Die Verwendung des SYSMAIN-Dienstprogramms selbst kann eingeschränkt werden oder die Verwendung der Quell- und Ziel-Libraries, die mit dem SYSMAIN-Dienstprogramm behandelt werden sollen, kann eingeschränkt werden. Die Verwendung von SYSMAIN-Funktionen, die über die Anwendungsprogrammierschnittstelle MAINUSER aufgerufen werden, kann separat kontrolliert werden. Einzelheiten hierzu finden Sie unter *Protecting Utilities* in der *Natural Security*-Dokumentation.

User-Exit-Routinen

Die User-Exit-Routinen des SYSMAIN-Dienstprogramms werden verwendet, um Informationen über jedes verarbeitete Objekt zu liefern oder die Verarbeitung von Funktionen zu steuern. Eine User-Exit-Routine ist ein Natural-Subprogramm, das mit einem `CALLNAT`-Statement aufgerufen wird.

Die Quellcodes der Subprogramme und die von ihnen verwendeten Datenbereiche sind als Quellcodeobjekte unter dem Namen `SM-UX-`*nn* (*nn* = `01` bis `11`) in der Library SYSMAIN abgelegt. Um eine User-Exit-Routine verfügbar zu machen, müssen Sie das entsprechende Quellcodeobjekt unter dem Namen `MAINEX`*nn* katalogisieren, und zwar entweder in der Library SYSMAIN oder in einer ihrer Steplibs.

Anmerkung: Die Namen von Quellcodeobjekten und katalogisierten Objekten von User-Exit-Routinen sind unterschiedlich, um sicherzustellen, dass das Überschreiben der Quellcodeobjekte durch eine Update-Installation die katalogisierten Objekte nicht beeinflusst.

Sie können jede der User-Exit-Routinen nach Bedarf ändern oder erweitern.

Die Verwendung dieser Exits führt je nach Code-Logik zu einem zusätzlichen Verwaltungsaufwand für das SYSMAIN-Dienstprogramm. Es ist daher immer notwendig, die Kontrolle an SYSMAIN zurückzugeben, wenn die Exit-Verarbeitung abgeschlossen ist.

Da das SYSMAIN-Dienstprogramm ET-Logik mit Adabas-Dateien verwendet, kann die Verwendung von User-Exit-Routinen das Transaktionszeitlimit (Adabas-Parameter `TT`) verlängern. Außerdem sollte die Definition der Adabas-Transaktion nicht verändert werden, d.h. Sie sollten keine `ET`/`BT`-Befehle oder `END`/`BACKOUT TRANSACTION`-Statements absetzen. Für das Absetzen aller `END TRANSACTION`-Statements ist SYSMAIN zuständig. Eine Ausnahme von dieser Regel besteht dann, wenn ein Benutzer die normale Beendigung einer SYSMAIN-Funktion mit den User-Exit-Routinen abbricht. In diesem Fall müssen Sie vor der Beendigung ein `BACKOUT TRANSACTION` absetzen.

Wenn der Rückgabecode auf einen Wert ungleich Null gesetzt wird, hat dies Vorrang vor einem von SYSMAIN ausgegebenen Fehler. Wenn ein Fehler von einem Exit empfangen wird, wird er in das Message-Feld gestellt und je nach Bedarf angezeigt oder gedruckt. Die Ausnahme ist die automatisierte Verarbeitung, da die Verarbeitung mit einem Minimum an Terminal-Ein-/Ausgaben abgeschlossen wird.

Die einzelnen User-Exit-Routinen werden im folgenden Abschnitt beschrieben.

- MAINEX01 - Erste User-Exit-Routine für die Objektabfrage
- MAINEX02 - Zweite User-Exit-Routine für die Objektabfrage
- MAINEX03 - User-Exit-Routine zur Abfrage von Anforderungen
- MAINEX04 - User-Exit-Routine zur Änderung von Dateizuordnungen
- MAINEX05 - User-Exit-Routine zur Verifizierung von Direktkommandos
- MAINEX06 - User-Exit-Routine zur SYSMAIN-Initialisierung
- MAINEX07 - User-Exit-Routine zum Beenden von SYSMAIN
- MAINEX08 - User-Exit-Routine bei Nothing Found (Nichts gefunden) im Batch-Modus
- MAINEX09 - User-Exit-Routine für abnormale Beendigung im Batch-Modus
- MAINEX10 - User-Exit-Routine bei Kommandofehlern im Batch-Modus

- MAINEX11 - User-Exit-Routine zum Setzen spezieller Flags für SYSMAIN

MAINEX01 - Erste User-Exit-Routine für die Objektabfrage

Funktion	Abfrage der aktuellen Werteinstellungen der mit einem Objekt verbundenen Datenelemente, *bevor* das Objekt von SYSMAIN verarbeitet wird.
Bemerkungen	Jedes an MAINEX01 übergebene Objekt kann zurückgewiesen werden, indem der Parameter `RESP-CODE` auf einen Wert ungleich Null gesetzt wird. Wenn eine zusätzliche Logik ausgeführt werden soll, darf sich die Transaktion *nicht* im End of Transaction-Status befinden, so dass kein `END TRANSACTION`- oder `BACKOUT TRANSACTION`-Statement ausgegeben werden sollte. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`PARM-AREA1 (A250)` SYSMAIN-Parameterbereich (feste Werte). `PARM-AREA2 (A250)` SYSMAIN-Parameterbereich (variable Werte). `RESP-CODE (B1)` Rückgabecode, der an SYSMAIN zurückgegeben wird. **Anmerkung:** Nur der Parameter `RESP-CODE` kann geändert werden.
Lokaler Datenbereich (LDA)	SM-UX-L

MAINEX02 - Zweite User-Exit-Routine für die Objektabfrage

Funktion	Abfrage der aktuellen Werteinstellungen der mit einem Objekt verbundenen Datenelemente, *nachdem* das Objekt von SYSMAIN verarbeitet wurde.
Bemerkungen	Jedes an MAINEX02 übergebene Objekt kann zurückgewiesen werden, indem der Parameter `RESP-CODE` auf einen Wert ungleich Null gesetzt wird. Wenn eine zusätzliche Logik ausgeführt werden soll, darf die Transaktion *nicht* den End of Transaction-Status erreicht haben, so dass kein `END TRANSACTION`- oder `BACKOUT TRANSACTION`-Statement ausgegeben werden sollte. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`PARM-AREA1 (A250)` SYSMAIN-Parameterbereich (feste Werte). `PARM-AREA2 (A250)` SYSMAIN-Parameterbereich (variable Werte). `RESP-CODE (B1)` Rückgabecode, der an SYSMAIN zurückgegeben wird. **Anmerkung:** Nur der Parameter `RESP-CODE` kann geändert werden.
Lokaler Datenbereich (LDA)	SM-UX-L

MAINEX03 - User-Exit-Routine zur Abfrage von Anforderungen

Funktion	Abfrage jeder an SYSMAIN gerichteten Anforderung in Form eines Direktkommandos oder einer online im Menümodus eingegebenen Information. MAINEX03 erhält die Kontrolle, *bevor* SYSMAIN das Kommando verarbeitet.
Bemerkungen	Jedes an MAINEX03 übergebene Kommando kann zurückgewiesen werden, indem der Parameter `RESP-CODE` auf einen Wert ungleich Null gesetzt wird. Zusätzliche Logik kann hinzugefügt werden, aber es liegt in Ihrer Verantwortung, alle notwendigen `END TRANSACTION`-Anforderungen an die Datenbank zu stellen. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`PARM-AREA (A250)` Kommandozeichenfolge `RESP-CODE (B1)` Rückgabecode, der an SYSMAIN zurückgegeben wird. **Anmerkung:** Nur der Parameter `RESP-CODE` kann geändert werden.

MAINEX04 - User-Exit-Routine zur Änderung von Dateizuordnungen

Funktion	Überschreiben des Datenbank-, Datei-Passworts und Chiffrierschlüssels für die Natural-Systemdatei(en).
Bemerkungen	MAINEX04 wird aufgerufen, *bevor* eine Anforderung von SYSMAIN verarbeitet oder validiert wird. Wenn die Kontrolle an MAINEX04 übergeben wird, befinden Sie sich im End-of-Transaction-Status. Daher müssen Sie den `RESP-CODE`-Parameter auf einen Wert ungleich Null setzen, wenn Sie die Anforderung ablehnen wollen. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`PARM-AREA (A250)` SYSMAIN-Parameterbereich. `RESP-CODE (B1)` Rückgabecode, der an SYSMAIN zurückgegeben wird.
Lokaler Datenbereich (LDA)	SM-UX-L4

MAINEX05 - User-Exit-Routine zur Verifizierung von Direktkommandos

Funktion	Überprüfung eines Direktkommandos, das während der Online-Verarbeitung von SYSMAIN eingegeben wurde. Darüber hinaus können die Sonderzeichen, die zur Erkennung eines Systembefehls verwendet werden, überschrieben werden.
Bemerkungen	MAINEX05 wird aufgerufen, *bevor* ein in SYSMAIN eingegebenes Direktkommando verarbeitet wird. Mit MAINEX05 können Sie z.B. alle `SET` **Kommandos** abfragen (siehe *An SYSMAIN absetzbare spezielle Kommandos*) und auch verhindern, dass sie abgesetzt werden. Sie können diese Kommandos überprüfen und zurückweisen, indem Sie im Parameter `RESP-CODE` einen Wert ungleich Null zurückgeben. Sie befinden sich im End of Transaction-Status, wenn die Kontrolle an MAINEX05 übergeben wird. Einem in SYSMAIN eingegebenen Systemkommando müssen zwei Schrägstriche (//) vorangestellt werden; siehe auch ***SYSMAIN-Kommandozeile benutzen***. Mit MAINEX05 können Sie zu diesem Zweck zwei weitere Sonderzeichen definieren: Weisen Sie dazu dem Parameter `CMD-DEL` die gewünschten Zeichen zu. Wenn `CMD-DEL` auf Leerzeichen gesetzt ist, verwendet SYSMAIN den Standardwert aus zwei Schrägstrichen (`//`). Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`COMMAND (A68)` Aktuelles in SYSMAIN abgesetztes Kommando. `CMD-DEL (A3)` Sonderzeichen für Systemkommandos. `RESP-CODE (B1)` Rückgabecode, der an SYSMAIN zurückgegeben wird.

MAINEX06 - User-Exit-Routine zur SYSMAIN-Initialisierung

Funktion	Erlangung der Kontrolle bei der Initialisierung einer SYSMAIN-Sitzung.
Bemerkungen	MAINEX06 wird zu Beginn der SYSMAIN-Sitzung aufgerufen, wobei Sie einige der SYSMAIN-Standardeinstellungen außer Kraft setzen können, wie z.B. die Aufforderung zur Bestätigung einer Anforderung wie Löschen, Verschieben oder Ersetzen eines Objekts. Alle Parameter werden überprüft. Falls sie ungültig sind, werden die Standardeinstellungen verwendet. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter-Datenbereich (PDA)	SM-UX-L6

MAINEX07 - User-Exit-Routine zum Beenden von SYSMAIN

Funktion	Erlangung der Kontrolle beim Beenden einer SYSMAIN-Sitzung.
Bemerkungen	MAINEX07 wird beim Beenden einer SYSMAIN-Sitzung aufgerufen, um zu entscheiden, ob die Kontrolle bei SYSMAIN verbleiben soll oder nicht.
Parameter	`USER-AREA (A50)` Bereich zur freien Verwendung.

MAINEX08 - User-Exit-Routine bei Nothing Found (Nichts gefunden) im Batch-Modus

Funktion	Festlegen der weiteren Verarbeitung, wenn für ein Kommando im Batch-Modus keine Objekte gefunden werden.
Bemerkungen	MAINEX08 wird aufgerufen, wenn für ein bestimmtes Kommando, das im Batch-Modus ausgeführt wird, keine Objekte gefunden werden, die den angegebenen Kriterien entsprechen. Wenn dies der Fall ist, kann, muss aber nicht, die Kontrolle an SYSMAIN zurückgegeben werden. Wenn die Kontrolle an SYSMAIN zurückgegeben wird, setzt SYSMAIN die Verarbeitung mit dem nächsten Kommando fort.
Parameter	`CMD (A250)` Kommandozeichenfolge

MAINEX09 - User-Exit-Routine für abnormale Beendigung im Batch-Modus

Funktion	Bestimmt die zu ergreifende Maßnahme im Falle eines Fehlers im Batch-Modus.
Bemerkungen	MAINEX09 wird aufgerufen, wenn die SYSMAIN-Verarbeitung im Batch-Modus zu einem Fehler führt. In diesem Fall kann, muss aber nicht, die Kontrolle an SYSMAIN zurückgegeben werden. Wenn die Kontrolle an SYSMAIN zurückgegeben wird, wird SYSMAIN mit dem Bedingungscode 45 beendet. **Anmerkung:** Die Fehler NAT4810, NAT4818, NAT4867, NAT4868 und NAT4893 können von dieser User-Exit-Routine nicht behandelt werden.
Parameter	`CMD (A250)` Kommandozeichenfolge `ERROR-CODE (N4)` Nummer des Fehlers, der zum Abbruch geführt hat.

MAINEX10 - User-Exit-Routine bei Kommandofehlern im Batch-Modus

Funktion	Bestimmt die zu ergreifende Maßnahme im Falle eines Kommandofehlers im Batch-Modus.
Bemerkungen	MAINEX10 wird aufgerufen, wenn ein Fehler in einem SYSMAIN-Kommando im Batch-Modus festgestellt wird. In diesem Fall kann, muss aber nicht, die Kontrolle an SYSMAIN zurückgegeben werden. Wenn die Kontrolle an SYSMAIN zurückgegeben wird, setzt SYSMAIN die Verarbeitung mit dem nächsten Kommando fort.
Parameter	`CMD (A250)` Kommandozeichenfolge `ERROR-CODE (N4)` Nummer des Fehlers, der zum Abbruch geführt hat.

MAINEX11 - User-Exit-Routine zum Setzen spezieller Flags für SYSMAIN

Funktion	User-Exit-Routine zum Setzen spezieller Einstellungen.
Bemerkungen	MAINEX11 wird zu Beginn der SYSMAIN-Sitzung aufgerufen, wo Sie einige spezielle SYSMAIN-Flags setzen können, wie z.B. die Anzeige von MAINUSER-Meldungen im Batch. Siehe das Quellcodeobjekt der User-Exit-Routine (SM-UX-11) zu den verfügbaren Flags. Die Kontrolle muss an SYSMAIN zurückgegeben werden.
Parameter	`FLAGS (A250)` Flag-String (neu definiert).

XIX SYSNCP Utility

83 SYSNCP Utility

- Einführung in das Dienstprogramm SYSNCP 804
- Dienstprogramm SYSNCP aufrufen 812
- Auswahl des Prozessors 813
- Header-Daten 814
- Verwaltung der Schlüsselwörter 826
- Verwaltung der Funktionen 832
- Laufzeit-Aktionen 838
- Kommandoprozessor katalogisieren 843
- Administrative SYSNCP-Funktionen 844
- Sitzungsprofil 851

Das Dienstprogramm (Utility) SYSNCP ermöglicht die Definition und Wartung eines Natural-Kommandoprozessor unter Verwendung des Natural-Objekttyps Command Processor (NCP).

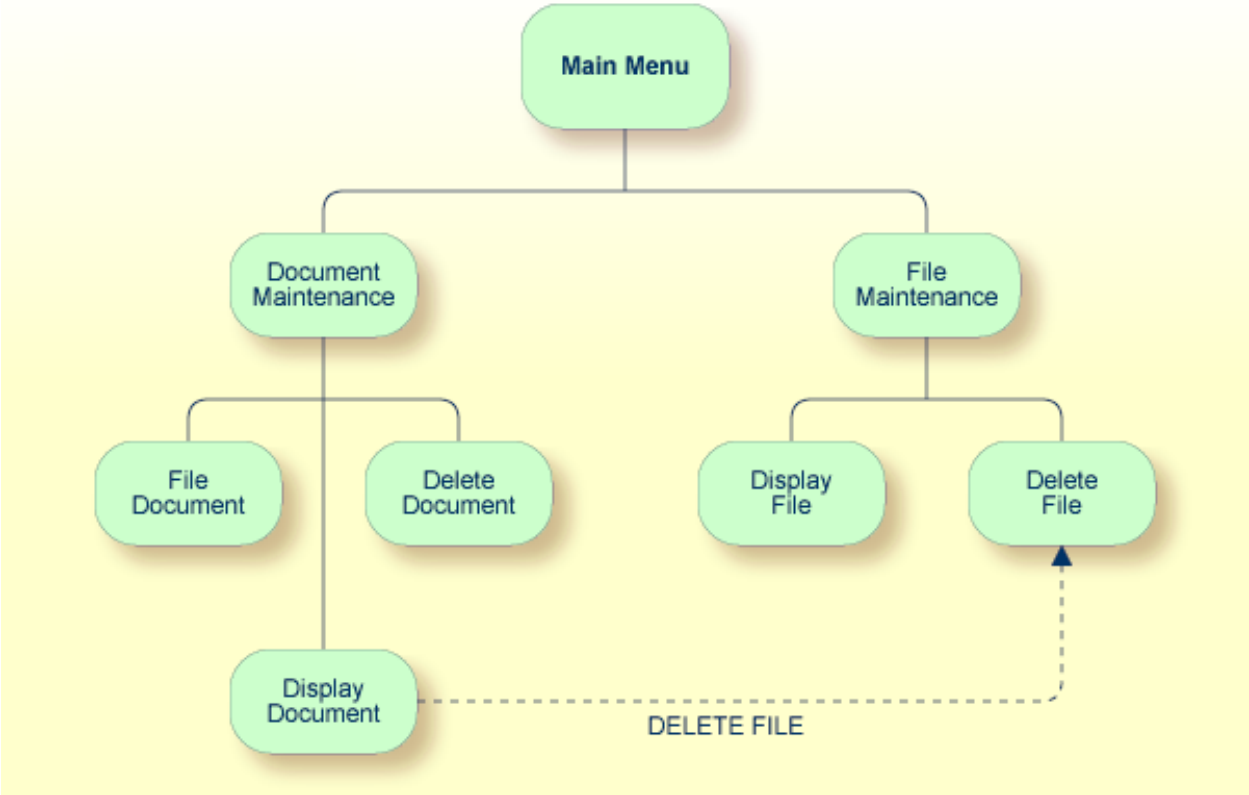
Ein Natural-Kommandoprozessor umfasst zwei Komponenten: Verwaltung und Laufzeit.

- **Verwaltung**
  Das Dienstprogramm SYSNCP ist der Verwaltungsteil, der alle Funktionen zur Definition und Steuerung der Navigation innerhalb einer Anwendung bereitstellt.
- **Laufzeit**
  Das Statement `PROCESS COMMAND` bildet den Laufzeitteil, der zum Aufrufen von Natural-Programmen dient.

  Sobald ein Kommando-Prozessor mit der Natural-Utility `SYSNCP` erstellt worden ist, kann er aus einem Natural-Programm mit dem Statement `PROCESS COMMAND` aufgerufen werden. Siehe *Statements*-Dokumentation.

Einführung in das Dienstprogramm SYSNCP

Anwendungen, in denen Benutzer unter Verwendung von Direktkommandos von einer Aktivität zu einer anderen wechseln können, sind weitaus benutzerfreundlicher als solche, die den Benutzer zwingen, durch hierarchische Menüstrukturen zu einer gewünschten Aktivität zu navigieren.



Die obige Abbildung verdeutlicht den Vorteil der Verwendung von Direktkommandos. In einer Anwendung, in der hierarchische Menüstrukturen die Grundlage für die Navigation bilden, müsste ein Benutzer, der von der Funktion **Display Document** (Dokument anzeigen) zur Funktion **Delete File** (Akte löschen) gelangen möchte, über den Dokumentenzweig zum Hauptmenü zurückkehren und dann den Zweig **File** (Akte) aufrufen. Dies ist eindeutig weniger effizient als der direkte Zugriff auf die Funktion **Delete File** aus der Funktion **Display Document** heraus.

Nachfolgend finden Sie Informationen über:

- Objektorientierte Datenverarbeitung
- Merkmale des Kommandoprozessors
- Komponenten des Kommandoprozessors
- Was ist ein Kommando?

- Einen Kommandoprozessor erstellen

Objektorientierte Datenverarbeitung

Der Natural-Kommandoprozessor dient zur Definition und Steuerung der Navigation innerhalb einer Anwendung. Er kann beispielsweise verwendet werden, um ein Kommando `DISPLAY DOCUMENT` zu definieren, das den direkten Zugriff auf die Funktion **Display Document** (Dokument anzeigen) ermöglicht. Wenn ein Benutzer diese Kommandozeichenfolge in die Kommandozeile eines Bildschirms (bei dem dieses Kommando erlaubt ist) eingibt, verarbeitet der Natural-Kommandoprozessor die Eingabe und führt die dem Kommando zugeordnete(n) Aktion(en) aus.

Im Gegensatz zu menügesteuerten Anwendungen bedeuten die mit dem Natural-Kommandoprozessor implementierten kommandogesteuerten Anwendungen einen großen Schritt in Richtung objektorientierte Datenverarbeitung.

Dieser Ansatz hat folgende Vorteile:

- Das Design einer Anwendung muss nicht davon abhängen, auf welchem Wege ein bestimmtes Ergebnis erreicht werden kann, sondern nur von dem gewünschten Ergebnis selbst. Damit wird das Design einer Anwendung nicht mehr durch den Prozessfluss innerhalb ihrer Komponenten beeinflusst.
- Die Verarbeitungseinheiten einer Anwendung werden unabhängig voneinander, was die Wartung der Anwendung einfacher, schneller und wesentlich effizienter macht.
- Anwendungen können einfach durch Hinzufügen unabhängiger Verarbeitungseinheiten erweitert werden. Die resultierenden Anwendungen sind daher nicht nur aus der Sicht des Endbenutzers einfach zu verwenden, sondern auch aus der Sicht des Programmierers einfacher zu erstellen.

Der Natural-Kommandoprozessor bietet die folgenden zusätzlichen Vorteile:

- **Weniger Programmieraufwand**
  Statt immer wieder lange und identisch aufgebaute Statement-Blöcke für die Verarbeitung von Kommandos zu programmieren, müssen Sie nur noch ein `PROCESS COMMAND`-Statement angeben, das den Kommandoprozessor aufruft. Die eigentliche Kommandoverarbeitung muss nicht mehr im Quellcode angegeben werden. Dadurch wird der Programmieraufwand erheblich reduziert.
- **Effizientere Kommandoverarbeitung**
  Da die Kommandoverarbeitung standardisiert und an einer zentralen Stelle definiert ist, kann die Arbeit bei der Erstellung und Pflege des kommandoverarbeitenden Teils einer Anwendung viel schneller und effizienter durchgeführt werden.
- **Verbesserte Performance**
  Bei der Entwicklung des Natural-Kommandoprozessors wurde besonders auf Performance-Aspekte geachtet: Er sorgt dafür, dass Kommandos durch Natural möglichst schnell verarbeitet werden und trägt so zur Verbesserung der Performance Ihrer Natural-Anwendungen bei.

Merkmale des Kommandoprozessors

Der Natural-Kommandoprozessor bietet zahlreiche Funktionen für eine effiziente und benutzerfreundliche Kommandoverarbeitung:

- **Flexibler Umgang mit Kommandos**
  Sie können Aliase (d.h. Synonyme für Schlüsselwörter) und Abkürzungen für häufig verwendete Kommandos definieren.
- **Automatische Prüfung auf Eindeutigkeit von abgekürzten Schlüsselwörtern**
  Der Kommandoprozessor vergleicht automatisch jedes Schlüsselwort, das Sie in SYSNCP angeben, mit allen anderen Schlüsselwörtern und ermittelt die Mindestanzahl an Zeichen in jedem Schlüsselwort, die zur eindeutigen Identifizierung des Schlüsselworts erforderlich sind. Das bedeutet, dass Benutzer bei der Eingabe von Kommandos in einer Anwendung jedes Schlüsselwort auf die Mindestlänge kürzen können, die der Kommandoprozessor benötigt, um es von anderen Schlüsselwörtern zu unterscheiden.
- **Lokale und globale Gültigkeit von Kommandos**
  Sie können in SYSNCP festlegen, ob die Aktion, die als Reaktion auf ein bestimmtes Kommando ausgeführt werden soll, unter allen Bedingungen gleich oder situationsabhängig sein soll. Sie können die Aktion z. B. davon abhängig machen, welches Programm zuvor ausgeführt wurde. Außerdem können Sie festlegen, dass ein Kommando unter einer Bedingung gültig, unter einer anderen aber ungültig ist.
- **Fehlerbehandlung bei ungültigen Kommandos**
  Sie können Ihre eigenen Fehlerbehandlungsroutinen an Kommandos anhängen oder fehlerhafte Eingaben durch Natural behandeln lassen.
- **Funktionale Sicherheit**
  Mit Natural Security können Sie bibliotheks- und benutzerspezifische Nutzungsbedingungen für die mit SYSNCP generierten Tabellen festlegen. So können Sie für Ihre Natural-Anwendungen bestimmte Funktionen oder Schlüsselwörter für einen bestimmten Benutzer zulassen oder verbieten. Dies wird als funktionale Sicherheit bezeichnet. Siehe auch den Abschnitt *Functional Security* in der *Natural Security*-Dokumentation.
- **Hilfetext**
  In SYSNCP können Sie einen Hilfetext an ein Schlüsselwort oder ein Kommando anhängen. Durch Angabe eines `PROCESS COMMAND ACTION TEXT`-Statements können Sie dann einen kommandospezifischen Hilfetext an das Programm zurückgeben.
- **Online-Tests der Kommandoverarbeitung**
  Wenn die Ausführung eines Kommandos nicht das gewünschte Ergebnis liefert, können Sie mit dem Statement `PROCESS COMMAND` (siehe *Statements*-Dokumentation) und den `EXAM*`-Beispielprogrammen (als Quellcode) aus der Library SYSNCP ermitteln, warum das Kommando nicht korrekt verarbeitet wurde. Die Endungen der `EXAM-*`-Programmnamen erscheinen als Abkürzungen in der oberen Randzeile der jeweiligen Aktionsfenster (z.B. `EXAM-C` erscheint als **C**).

Um einen Kommandoprozessor zur Laufzeit zu testen:

1. Geben Sie das Direktkommando `EXAM` ein, um alle Testprogramme aufzulisten. Das Fenster **Demonstrate PROCESS COMMAND Statement** wird angezeigt.
2. Geben Sie den Funktionscode `O` ein, um einen Prozessor zu öffnen.
3. Geben Sie den Namen des Prozessors ein.
4. Wählen Sie einen der aufgelisteten Funktionscodes (z.B. `C` für **CHECK**), um Kommando-Aktionen anzuwenden.
5. Geben Sie den Funktionscode `Q` ein, um den Prozessor zu schließen.

Komponenten des Kommandoprozessors

Der Natural-Kommandoprozessor besteht aus zwei Teilen: einem Entwicklungsteil und einem Laufzeitteil:

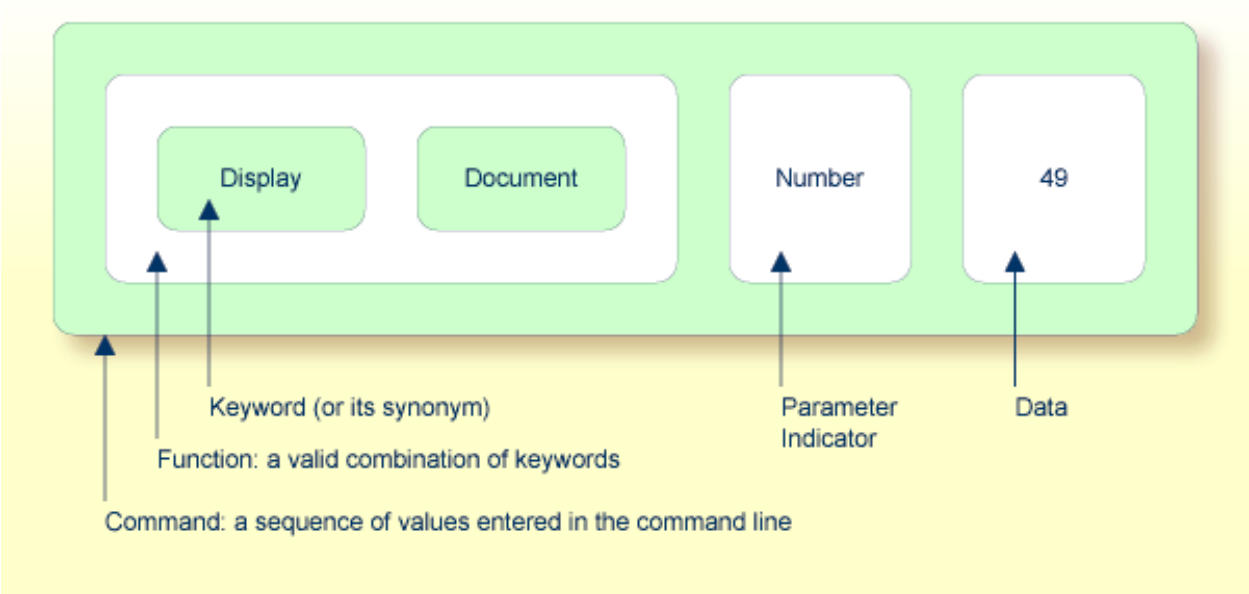
- **Entwicklungsteil**
  Der Entwicklungsteil besteht aus dem Dienstprogramm SYSNCP, das im vorliegenden Dokument beschrieben wird. Mit diesem Dienstprogramm definieren Sie Kommandos (wie unten beschrieben) und die Maßnahmen, die als Reaktion auf die Ausführung dieser Kommandos ausgeführt werden sollen. Aus Ihren Definitionen erzeugt SYSNCP Entscheidungstabellen, die festlegen, was geschieht, wenn ein Benutzer ein Kommando eingibt. Diese Tabellen sind in einem Natural Member vom Typ Processor enthalten.
- **Laufzeitteil**
  Der Laufzeitteil ist das Statement `PROCESS COMMAND`, das in der *Statements*-Dokumentation beschrieben ist. Dieses Statement wird verwendet, um den Kommandoprozessor innerhalb eines Natural-Programms aufzurufen. In diesem Statement geben Sie den Namen des Prozessors an, mit dem die Kommandoeingabe eines Benutzers an dieser Stelle verarbeitet werden soll.

Was ist ein Kommando?

Ein Kommando ist eine beliebige Folge von Werten, die in der Kommandozeile eingegeben und von einer Anwendung erkannt und verarbeitet wird. Kommandos können bis zu drei Elemente enthalten:

- **Funktion:**
  Ein oder mehrere gültige Schlüsselwörter. Zum Beispiel: `MENU` oder `DISPLAY DOCUMENT`.
- **Parameter-Indikator:**
  Optional. Ein Schlüsselwort, das die Kommandodaten einleitet.

- **Kommandodaten:**
  Informationen, die an eine Funktion übergeben werden sollen. Die Kommandodaten können alphanumerisch oder numerisch sein, z. B. der Name oder die Nummer der Datei, die angezeigt werden soll.



Kommandos werden immer ausgehend von einer Situation innerhalb einer Anwendung ausgeführt. Die Position, an der diese Situation erreicht wird, wird als Stelle (Location) bezeichnet. Kommandos bringen den Benutzer in einer Anwendung von einer Stelle zu einer anderen Stelle. Somit kann jedes Kommando als Vektor betrachtet werden:

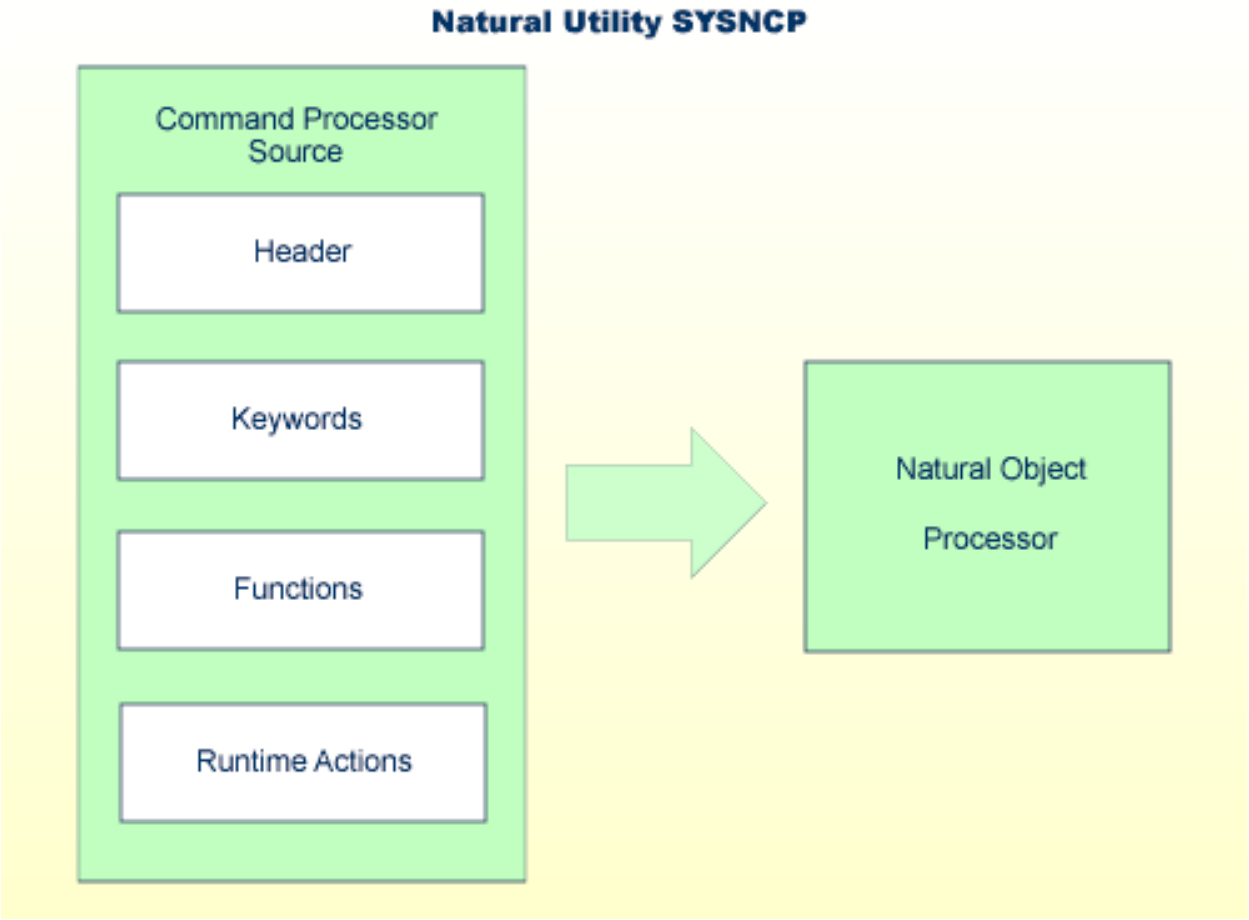


Die Stelle, von der aus ein bestimmtes Kommando abgesetzt werden kann, kann systemweit oder benutzerspezifisch eingeschränkt werden. Systemweit betrachtet können die in den Kommandos angegebenen Funktionen zum Beispiel lokal oder global sein. Eine globale Funktion kann von jeder Stelle aus ausgeführt werden, während eine lokale Funktion nur von bestimmten Stellen aus ausgeführt werden kann. Schlüsselwörter und Funktionen können jedoch eingeschränkt werden, wenn Natural Security in Ihrer Umgebung aktiv ist.

Einen Kommandoprozessor erstellen

Mit dem Dienstprogramm SYSNCP können Sie Kommandoprozessoren erstellen und pflegen. Ein Kommandoprozessor enthält Entscheidungstabellen, die festlegen, was geschieht, wenn ein Benutzer ein gültiges Kommando eingibt.

Das Erstellen eines Kommandoprozessors ist ein kumulativer Vorgang, der mehrere Schritte umfasst, von der Definition des Header, in dem die allgemeinen Vorgaben für den Prozessor festgelegt werden, über die Definition von Schlüsselwörtern und Funktionen bis hin zum Verknüpfen von Aktionen mit Funktionen. SYSNCP stellt spezielle Editoren zur Verfügung, um Schlüsselwörter, Funktionen und Aktionen zu definieren.



Das Endprodukt der Kommandoprozessor-Entwicklung ist ein komplexer Quellcode des Kommandoprozessors, der, wenn er katalogisiert wird, ein Natural-Objekt vom Typ Command Processor erzeugt. Wird dieses Objekt in einem Natural-Statement `PROCESS COMMAND` referenziert, dann wird das Laufzeitsystem des Natural-Kommandoprozessors angesprochen.

Im Folgenden finden Sie eine Zusammenfassung der Schritte, die zur Erstellung eines Kommandoprozessors erforderlich sind.

Um einen Kommandoprozessor zu erstellen, sind folgende Schritte erforderlich:

1 **Sitzungsprofil überprüfen und ändern**

SYSNCP selbst verwendet ein Sitzungsprofil, das verschiedene Parameter enthält, die steuern, wie SYSNCP bestimmte Aktionen ausführen und wie Informationen angezeigt werden sollen. Gewünschte Änderungen können vorgenommen und das resultierende Profil unter einer bestimmten Benutzerkennung gespeichert werden. Siehe Abschnitt *Sitzungsprofil*.

2 **Kommandoprozessor initialisieren**

Der Name des Kommandoprozessors und die Library, in der er gespeichert werden soll, werden angegeben.

3 **Globale Einstellungen definieren (Header)**

Es werden verschiedene globale Einstellungen für den Kommandoprozessor festgelegt. Zum Beispiel beschreibender Text für Schlüsselwörter während der Bearbeitung, minimale und maximale Länge für Schlüsselwörter, in welcher Reihenfolge Schlüsselwörter zur Laufzeit abgearbeitet werden sollen, Laufzeitfehlerbehandlung, und ob zur Laufzeit PF-Tasten zum Aufrufen von Funktionen benutzt werden können. Siehe dazu den Abschnitt *Header-Daten*.

4 **Schlüsselwörter definieren**

Jedes Schlüsselwort, das vom Kommandoprozessor verarbeitet werden soll, wird zusammen mit einer Angabe dazu definiert, ob das Schlüsselwort als erster, zweiter oder dritter Eintrag eines Kommandos eingegeben werden soll. Es können auch Schlüsselwort-Synonyme und Parameter-Indikatoren definiert werden. Zu jedem Schlüsselwort kann ein Benutzertext definiert werden. Dieser Text kann dann zur Laufzeit mit dem Statement `PROCESS COMMAND ACTION TEXT` gelesen werden. Siehe dazu den Abschnitt *Schlüsselwortverwaltung*.

5 **Funktionen definieren**

Funktionen werden durch Bestätigen von Schlüsselwortkombinationen definiert. Eine Funktion kann als lokal (kann nur von einer bestimmten Stelle innerhalb einer Anwendung aufgerufen werden) und/oder global (kann von überall innerhalb einer Anwendung aufgerufen werden) definiert werden. Siehe Abschnitt *Funktionsverwaltung*.

6 **Laufzeit-Aktionen definieren**

Hier werden die Aktionen festgelegt, die der Kommandoprozessor ausführen soll, wenn ein Kommando zur Laufzeit abgesetzt wird. Beispiele für Aktionen sind: Abrufen eines Natural-Programms, Platzieren eines Kommandos auf dem Natural-Stack, Platzieren von Daten auf dem Natural-Stack, Ändern des Inhalts der Kommandozeile. Siehe Abschnitt *Laufzeit-Aktionen*.

7 **Kommandoprozessor katalogisieren**

Der resultierende Quellcode wird als Natural-Objekt (Typ Processor) in der vorgesehenen Natural Library katalogisiert. Der Kommandoprozessor kann nun aus einem Natural-Pro-

gramm mit dem Statement PROCESS COMMAND aufgerufen werden. Siehe Abschnitt *Prozessoren katalogisieren*.

Dienstprogramm SYSNCP aufrufen

Um das Dienstprogramm SYSNCP aufzurufen:

- Geben Sie das Systemkommando `SYSNCP` ein.

Das Menü **Processor Source Maintenance** zur Pflege eines Kommandoprozessor-Quellcodes wird angezeigt:

```
18:22:53              ***** NATURAL SYSNCP UTILITY *****           2000-05-22
User SAG                 - Processor Source Maintenance -

                     Code  Function

                      S    Select Processor
                      N    Create New Processor
                      H    Modify Header
                      K    Define Keywords
                      F    Define Functions
                      R    Define Runtime Actions
                      C    Catalog Processor
                      A    Administrator Services
                      ?    Help
                      .    Exit

               Code .. _    Name .. SAGTEST_  Library .. SYSNCP__



Logon to SYSNCP accepted.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd   Exit  Last  List  Flip                                  Canc
```

Von diesem Menü aus können Sie alle Funktionen aufrufen, die zur Erstellung, Pflege und Verwaltung von Kommandoprozessoren erforderlich sind. Um eine Funktion aufzurufen, geben Sie den entsprechenden Codebuchstaben in das Feld **Code** ein.

Anmerkung: Wenn Sie das Dienstprogramm SYSNCP aufrufen oder SYSNCP neu starten, wird der User-Exit `NCP-USR1` zur dynamischen Anpassung aufgerufen. Siehe mitgeliefertes Programm `NCP-USR1` in der Natural System Library SYSNCP.

Hilfe

Hilfe zu einzelnen Eingabefeldern (und auch zu einigen Ausgabefeldern) im Dienstprogramm SYSNCP erhalten Sie, indem Sie den Cursor in das Feld setzen und PF1 drücken.

Auswahl des Prozessors

Mit der Funktion **Select Processor** erhalten Sie eine Liste aller vorhandenen Kommandoprozessor-Quellcode-Objekte mit zugehörigen Informationen. Wenn Natural Security installiert ist, werden nur die Quellcode-Objekte aufgelistet, die in einer Library katalogisiert werden können, bei der Sie sich anmelden dürfen. Diese Einschränkung gilt nicht für Benutzer, die den Status eines Administrators haben.

Um die Funktion Select Processor aufzurufen:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode `S` ein.

2 Drücken Sie ENTER.

Zu jedem Prozessor werden folgende Informationen angezeigt:

Name	Der Name des Kommandoprozessors.
Library	Der Name der Natural Library, für die ein Prozessor erstellt wird. Wenn der Prozessor katalogisiert wird, wird er in dieser Library gespeichert.
User ID	Die Kennung des Benutzers, der den Prozessor erstellt hat.
Date	Das Datum, an dem der Prozessor erstellt wurde.
Status	Das Entwicklungsstadium des Prozessors. Mögliche Statuswerte finden Sie unter ***Aktueller Status*** im Abschnitt *Header-Datensätze*.
Cat	Zeigt an, ob der Prozessor katalogisiert wurde.

Anmerkung: Mit dem User-Exit `NCP-SELX` (ausgeliefert in der Natural System Library SYSNCP) können Sie die Anzeige auf bestimmte Prozessoren beschränken.

3 Geben Sie im Feld **Ac** ein beliebiges Zeichen ein, um einen Prozessor auszuwählen.

Sie gelangen in das Menü **Processor Source Maintenance**, in dem der Name des ausgewählten Prozessors automatisch in das Feld **Name** gestellt wird.

Wenn Sie ein Fragezeichen (`?`) in das Feld **Ac** eingeben, wird ein Fenster angezeigt, das weitere mögliche Optionen auflistet.

Der Name und der Library-Name eines Kommandoprozessors können ein bis acht Zeichen lang sein. Er kann aus alphabetischen Großbuchstaben (A - Z), numerischen Zeichen (0 - 9) und den folgenden Sonderzeichen bestehen: "-", "/ ", "$", "&", "#", "+" und "_".

Header-Daten

Bei der Pflege von Header-Daten werden verschiedene globale Einstellungen für einen Kommandoprozessor definiert. Diese Definitionen zusammen werden als Header bezeichnet. Zum Anlegen und Ändern von Header-Daten stehen sieben Pflegebildschirme Verfügung. Die Header-Einstellungen für einen Kommandoprozessor können in jedem Stadium der Entwicklung geändert werden (siehe den folgenden Abschnitt). Nachdem die Einstellungen geändert wurden, wird der Status eines Kommandoprozessors immer auf Header gesetzt (siehe auch *Aktueller Status*).

Nachfolgend finden Sie Informationen über:

- Neuen Prozessor anlegen
- Header-Daten ändern - Allgemeines
- Schlüsselwort-Laufzeit-Optionen - Header 1
- Keyword Editor-Optionen - Header 2
- Verschiedene Optionen - Header 3
- Behandlung von Kommandodaten - Header 4
- Behandlung von Laufzeitfehlern - Header 5
- Statistikdaten - Header 6
- Status - Header 7

Neuen Prozessor anlegen

Um einen neuen Kommandoprozessor anzulegen:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode N (**Create New Processor**) ein, den Namen des zu erstellenden Kommandoprozessors und den Namen der Natural Library, in der der Kommandoprozessor später katalogisiert werden soll.

2 Drücken Sie ENTER.

Der erste **Header Maintenance**-Bildschirm wird angezeigt.

Der erste **Header Maintenance**-Bildschirm wird angezeigt. Dieser erste Bildschirm und die folgenden Bildschirme sind mit Vorschlagswerten gefüllt, die bearbeitet werden können.

Header-Daten ändern - Allgemeines

Mit der Funktion **Modify Header** können Sie einen bestehenden Header bearbeiten, d.h. die verschiedenen Header-Einstellungen für einen bestimmten Kommandoprozessor ändern.

Um einen bestehenden Header zu ändern:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode H (**Modify Header**) ein, den Namen des entsprechenden Kommandoprozessors und den Namen der Library, in der dieser Kommandoprozessor katalogisiert wurde.

2 Drücken Sie ENTER.

Das erste Bildschirmbild zur Pflege der Header-Daten wird angezeigt.

3 Sie können jedes beliebige Eingabefeld in den unten beschriebenen Bildschirmen ändern.

4 Drücken Sie ENTER, um die Änderungen zu bestätigen.

Für die Definition und Pflege eines Prozessor-Header stehen sieben verschiedene Bildschirme zur Verfügung (Informationen zur Definition eines Header siehe vorherigen Abschnitt).

Um zwischen den Bildschirmen hin- und her zu wechseln:

- Drücken Sie PF8 (vorwärts) oder PF7 (rückwärts).

Jeder der Bildschirme enthält die folgenden Informationen:

Name	Der Name des Kommandoprozessors.
Library	Der Name der Library, in der das resultierende Kommandoprozessor-Objekt nach dem Katalogisieren abgelegt werden soll.
DBID, FNR	Die Datenbankkennung und die Nummer der Datei, in der sich die angegebene Library befindet.
Created by	Die Benutzerkennung des Natural-Benutzers, der diesen Kommandoprozessor initiiert hat.
Date	Das Datum, an dem der Kommandoprozessor initiiert wurde.
Current Status	Der aktuelle Status des Kommandoprozessors: Init — Der Kommandoprozessor wurde initialisiert. Header — Der Header für den Kommandoprozessor wurde angelegt/geändert. Keysave — Schlüsselwörter wurden definiert und gespeichert. Keystow — Schlüsselwörter wurden überprüft und gespeichert (mit STOW). Function — Schlüsselwortkombinationen wurden definiert. Action — Laufzeit-Aktionen wurden definiert. Object — Der Kommandoprozessors wurde in Objektform erstellt.

	Frozen	Der Kommandoprozessor wurde eingefroren.
	Copied	Der Kommandoprozessor wurde kopiert.
	Error	Es wurde ein Fehler festgestellt.

Schlüsselwort-Laufzeit-Optionen - Header 1

Wenn Sie die Funktion **Modify Header** (wie oben beschrieben) wählen, wird der Bildschirm **Processor Header Maintenance 1** mit den **Keyword Runtime Options** angezeigt:

```
16:40:19                ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG                - Processor Header Maintenance 1 -

Modify Processor            Name SAGTEST  Library SYSNCP   DBID 10    FNR 32
Created by SAG      Date 2000-04-29                     Current Status Init

Keyword Runtime Options:
------------------------
   First  Entry used as ....... Action__________
   Second Entry used as ....... Object__________
   Third  Entry used as ....... Addition________

   Minimum Length ............. _1
   Maximum Length ............. 16
   Dynamic Length Adjustment .. -

   Keyword Sequence ........... 123_____
   Alternative Sequence ....... _________
   Local/Global Sequence ...... LG______

Processor Header with name SAGTEST for library SYSNCP has been added.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd   Exit  Last  List  Flip  -     +                      Canc
```

Auf diesem Bildschirm werden verschiedene Attribute eingegeben, die für die für den Kommandoprozessor definierten Schlüsselwörter gelten sollen.

Feld	Erläuterung
First Entry used as	**Erster Eintrag verwendet als** Ein beschreibender Text, der allen Schlüsselwörtern zugeordnet werden soll, die bei der Definition einer Schlüsselwortfolge als erster Eintrag (Eintragstyp 1) eingegeben werden. Soll z.B. das erste Schlüsselwort einer Schlüsselwortfolge die auszuführende Aktion darstellen (`DISPLAY`, `DELETE` usw.), so könnte in diesem Feld der beschreibende Text „Aktion“ eingegeben werden. Die ersten vier Zeichen des in diesem Feld eingegebenen Textes erscheinen unter der Spaltenüberschrift **Use** im Keyword Editor, wie im Abschnitt *Schlüsselwörter verwalten* beschrieben.
Second Entry used as	**Zweiter Eintrag verwendet als** Ein beschreibender Text, der mit allen Schlüsselwörtern verknüpft werden soll, die bei der Definition einer Schlüsselwortfolge als zweiter Eintrag (Eintragstyp 2) eingegeben werden. Soll z.B. das zweite Schlüsselwort einer Schlüsselwortfolge das zu verwendende Objekt (`DOCUMENT`, `FILE` usw.) darstellen, so könnte in diesem Feld der beschreibende Text "Objekt" eingetragen werden. Die ersten vier Zeichen des in diesem Feld eingegebenen Textes erscheinen unter der Spaltenüberschrift **Use** im Keyword Editor, wie im Abschnitt *Schlüsselwörter verwalten* beschrieben.
Third Entry used as	**Dritter Eintrag verwendet als** Ein beschreibender Text (`TITLE`, `PARAGRAPH` usw.), der mit allen Schlüsselwörtern verknüpft werden soll, die bei der Definition einer Schlüsselwortfolge als dritter Eintrag (Eintragstyp 3) eingegeben werden. Die ersten vier Zeichen des in diesem Feld eingegebenen Textes erscheinen unter der Spaltenüberschrift **Use** im Keyword Editor, wie im Abschnitt *Schlüsselwörter verwalten* beschrieben.
Minimum Length	**Minimale Länge** Die Minimale Länge, die bei der Definition eines Schlüsselworts zulässig ist. Gültige Werte sind 1 - 16 Zeichen. Der Standardwert ist ein Zeichen.
Maximum Length	**Maximale Länge** Die maximale Länge, die bei der Definition eines Schlüsselworts zulässig ist. Gültige Werte: 1 - 16 Zeichen. Der Standardwert ist 16 Zeichen.
Dynamic Length Adjustment	**Dynamische Längenanpassung** Folgende Werte sind zulässig: + Zur Laufzeit muss jedes Schlüsselwort vollständig eingegeben werden.

Feld	Erläuterung
	- Zur Laufzeit kann jedes Schlüsselwort abgekürzt werden, sofern es im Zusammenhang mit anderen Schlüsselwörtern eindeutig bleibt. S Die Anzahl der Zeichen, die für ein bestimmtes Schlüsselwort eingegeben werden müssen, ist bei der Schlüsselwortdefinition im Feld ML des Keyword Editors anzugeben, wie im Abschnitt *Schlüsselwörter verwalten* beschrieben.
Keyword Sequence	**Schlüsselwort-Reihenfolge** Die Reihenfolge, in der die Schlüsselworteinträge zur Laufzeit abgearbeitet werden sollen. Mögliche Werte sind `1`, `2`, `3` und `P` (für Parameterindikator); die Standardreihenfolge ist `12`, d.h. zuerst der erste Schlüsselworteintrag und dann der zweite Schlüsselworteintrag. Siehe auch **Feld E** im Abschnitt *Schlüsselwörter verwalten*.
Alternative Sequence	**Alternative Reihenfolge** Eine alternative Reihenfolge, in der die Schlüsselwörter zur Laufzeit abgearbeitet werden sollen, falls die (oben angegebene) Standardreihenfolge zur Laufzeit zu einem Fehler führt.
Local/Global Sequence	**Lokale/globale Reihenfolge** Diese Option gibt die Reihenfolge an, in der die Kommandovalidierung zur Laufzeit durchgeführt werden soll. Mögliche Werte sind: L Kommando soll als lokales Kommando validiert werden. G Kommando soll als globales Kommando validiert werden. Die Standard-Validierungsreihenfolge ist `LG`, was bedeutet, dass das Kommando zuerst als lokales Kommando und dann (falls erforderlich) als globales Kommando validiert wird.

Keyword Editor-Optionen - Header 2

Auf dem Bildschirm **Processor Header Maintenance 2** können weitere Schlüsselwortattribute eingegeben werden:

Feld	Erläuterung
Header 1 for User Text	**Header 1 / Header 2 für Benutzertext** Diese beiden Felder dienen der Eingabe eines beschreibenden Textes, der im Keyword Editor oberhalb der für **User Text** reservierten Spalte erscheint. Dieser Text wird auch zur Laufzeit ausgegeben, wenn die Option TEXT beim `PROCESS COMMAND`-Statement angegeben ist, siehe *Statements*-Dokumentation.
Header 2 for User Text	
Prefix Character 1	**Präfix-Zeichen 1** Dieses und die nächsten drei Felder werden verwendet, um Schlüsselwörter mit einem hexadezimalen Präfix zu versehen. Dies ermöglicht die Verarbeitung von

Feld	Erläuterung
	internen Schlüsselwörtern, die nicht über eine normale Tastatur dargestellt werden können. Wenn der Kommandoprozessor katalogisiert wird, werden alle Präfixzeichen in Schlüsselwörtern durch die angegebenen hexadezimalen Werte ersetzt. Wenn ein Nicht-Leerzeichen in eines der Felder für Präfix-Zeichen eingegeben wird, wird das angegebene Zeichen durch den Hexadezimalwert ersetzt, der im Feld **Hexadecimal Replacement** angegeben ist.
Hex. Replacement 1	**Hex. Ersetzung 1** Der in diesem Feld angegebene Wert ersetzt das im Feld **Prefix Character** angegebene Zeichen und wird zur Laufzeit als Präfix für ein Schlüsselwort verwendet.
Prefix Character 2	Wie bei **Präfixzeichen 1**, siehe oben.
Hex. Replacement 2	Wie bei **Hex. Replacement 1**, siehe oben.
Keywords in Upper Case	**Schlüsselwörter in Großbuchstaben** Diese Option gibt an, ob Schlüsselwörter im **Keyword Editor** und in der Anwendung in Großbuchstaben umgewandelt werden sollen: Y Schlüsselwörter, die im **Keyword Editor** eingegeben werden, werden automatisch in Großbuchstaben umgewandelt. In der Anwendung können die Endbenutzer die Schlüsselwörter in Groß- oder Kleinbuchstaben eingeben. N Im **Keyword Editor** eingegebene Schlüsselwörter werden nicht in Großbuchstaben umgewandelt. In der Anwendung müssen die Endbenutzer die Schlüsselwörter *exakt* so eingeben, wie sie im **Keyword Editor** erscheinen.
Unique Keywords	**Eindeutige Schlüsselwörter** Diese Option gibt an, ob Schlüsselwörter innerhalb des Prozessors eindeutig sein müssen. Y Jedes definierte Schlüsselwort muss innerhalb des Prozessors eindeutig sein, unabhängig von seinem Typ. N Jedes für einen bestimmten Schlüsselworttyp (`1`, `2`, `3` oder `P`) definierte Schlüsselwort muss eindeutig sein.

Verschiedene Optionen - Header 3

Im Bildschirm **Processor Header Maintenance 3** können verschiedene Optionen eingegeben werden:

Feld	Erläuterung
Invoke Action Editor	Diese Option gibt an, ob der Runtime Action Editor vom Function Editor aus aktiviert werden soll (siehe die Abschnitte *Runtime Action Editor* und *Define Functions*). Y Der Runtime Action Editor wird immer dann aufgerufen, wenn im Function Editor eine gültige Schlüsselwortkombination definiert ist. N Der Runtime Action Editor wird im Function Editor unterbunden. **Anmerkung:** Wenn Sie den User-Exit `NCP-REDM` (ausgeliefert in der Natural System Library SYSNCP) verwenden, sollten Sie diese Option auf `Y` setzen, da sonst ungültige Laufzeit-Aktionswerte nicht rechtzeitig erkannt werden und zu Laufzeitfehlern führen können.
Catalog User Texts	**Benutzertexte katalogisieren** Diese Option gibt an, ob Benutzertexte mit dem Kommandoprozessor katalogisiert werden sollen. Y Beim Katalogisieren des Kommandoprozessors werden Textteile der Editierzeile (Keyword Editor; siehe Abschnitt *Schlüsselwörter definieren*) und der Benutzertextteil der Aktionszeile (Runtime Action Editor) an das zugehörige Schlüsselwort oder die Funktion gebunden. Dieser Text kann dann zur Laufzeit mit der `TEXT`-Option des `PROCESS COMMAND`-Statements gelesen werden. N Texte werden nicht mit dem Kommandoprozessor katalogisiert und können zur Laufzeit nicht gelesen werden.
Security Prefetch	**Sicherheitsüberprüfung** Diese Option gibt an, ob die Sicherheitsüberprüfung beim ersten Aufruf des Kommandoprozessors zur Laufzeit oder bei jeder Kommandoauswertung durchgeführt werden soll. Y Wenn Natural Security installiert ist, wird die Sicherheitsprüfung bei allen Schlüsselwörtern durchgeführt, wenn der Prozessor aufgerufen wird. N Wenn Natural Security installiert ist, wird die Sicherheitsüberprüfung bei der Auswertung jedes Schlüsselworts durchgeführt. Bei Auswahl der Option `Y` wird die Sicherheitsprüfung nur einmal für alle Schlüsselwörter durchgeführt, wenn der Kommandoprozessor aufgerufen wird. Da der Prüfvorgang Zeit in Anspruch nimmt, ist die Auswertung des ersten Kommandos zur Laufzeit vergleichsweise langsam, während die Auswertung aller übrigen Kommandos vergleichsweise schnell ist. Umgekehrt ist bei Wahl der Option `N` die Auswertungszeit für

Feld	Erläuterung
	jedes Kommando immer gleich, da die Sicherheit bei jedem Schlüsselwort einzeln geprüft wird, bevor es ausgewertet wird.
Command Log Size	**Größe des Kommandoprotokolls** Zur Laufzeit verarbeitete Kommandos können vom Kommandoprozessor in einem Kommandoprotokollbereich gespeichert werden. Geben Sie im Eingabefeld den Speicherplatz in KB an, der für die Kommandoprotokollierung vorgesehen ist: 0 Der Kommandoprotokollierung wird kein Speicherplatz zugeordnet. Die Kommandoprotokollierung ist inaktiv. 1 1 KB Speicherplatz wird der Kommandoprotokollierung zugeordnet. Die Kommandoprotokollierung ist aktiv. Impliziter Schlüsselworteintrag.
Implicit Keyword Entry	**Impliziter Schlüsselworteintrag** Diese Option gibt an, ob ein Schlüsselwort vom Typ 1 als implizites Schlüsselwort für alle nachfolgenden Kommandos beibehalten werden soll. 1 Wird ein Kommando eingegeben, das nur ein Schlüsselwort vom Typ 2 enthält, übernimmt der Kommandoprozessor das zuletzt eingegebene Schlüsselwort vom Typ 1 als implizites Schlüsselwort. N Option ist deaktiviert.
Command Delimiter	**Kommando-Trennzeichen** Diese Option gibt das Zeichen an, das zur Trennung von Kommandos verwendet wird, wenn mehr als ein Kommando in der Kommandozeile angegeben ist. Zur Laufzeit wird nur das erste Kommando ausgeführt. Beispiel: DISPLAY CUSTOMER; MODIFY CUSTOMER; PRINT.
PF-Key may be Command	**PF-Taste kann Kommando sein** Diese Option gibt an, ob Kommandos PF-Tasten zugeordnet werden können: Wenn der Kommandoprozessor zur Laufzeit eine Kommandozeile vorfindet, die nur Leerzeichen enthält, prüft er, ob eine PF-Taste vom Benutzer gedrückt wurde. Mögliche Werte: A Der Bezeichner für diese PF-Taste (Systemvariable `*PF-NAME`) wird als Kommando verwendet. K Der Inhalt der Systemvariablen `*PF-KEY` wird als Kommando verwendet. Y Wenn `*PF-NAME` leer ist, wird stattdessen der Inhalt der Systemvariablen `*PF-KEY` verwendet. N PF-Tasten können nicht als Kommando verwendet werden, es wird der Natural-Fehler NAT6913 mit der Meldung `Command line not accepted` (Kommandozeile nicht akzeptiert) ausgegeben.

Feld	Erläuterung
	Weitere Informationen zu den Systemvariablen `*PF-NAME` und `*PF-KEY` finden Sie in der *Systemvariablen*-Dokumentation.

Behandlung von Kommandodaten - Header 4

Die Attribute, die auf dem Bildschirm **Processor Header Maintenance 4** einzugeben sind, legen fest, wie Kommandodaten für eine Funktion behandelt werden. Kommandodaten sind optional.

Verfügbare Optionen:

Feld	Erläuterung
Data Delimiter	**Datentrennzeichen** Legt das Zeichen fest, das den Daten vorangestellt wird. Das Standard-Datentrennzeichen ist die Raute (#). Beispiel: ADD CUSTOMER #123
Data Allowed	**Daten erlaubt** Legt fest, ob die Dateneingabe zur Laufzeit erlaubt ist. `N` Wenn Daten gefunden werden, tritt ein Laufzeitfehler auf. `D` Daten werden verworfen, wenn sie vorhanden sind. `S` Die Daten werden oben auf den Natural Stack gelegt. Es erfolgt keine Überprüfung. `Y` Daten werden geprüft und Schlüsselworteinträge vom Typ `P` (Parameterindikator) werden ausgewertet. Beispiel für `Y`: DISPLAY CUSTOMER NAME=SMITH
More than one Item Allowed	**Mehr als ein Element erlaubt** Gilt nur, wenn die Option **Data Allowed** auf `Y` gesetzt ist. Gibt an, ob mehr als eine Datenzeichenkette erlaubt ist. `N` Ein Laufzeitfehler tritt auf, wenn mehr als eine Zeichenkette gefunden wird. `D` Alle Daten nach der ersten Datenzeichenkette werden verworfen. `Y` Mehr als eine Datenzeichenkette ist erlaubt. Beispiel: ADD ARTICLE #111 #222 Solange die Eindeutigkeit gewährleistet ist, kann das Datentrennzeichen weggelassen werden. Beispiel: ADD ARTICLE 123
Maximum Length	**Maximale Länge eines Elements**

Feld	Erläuterung
of one Item	Gilt nur, wenn die Option **Data Allowed** auf `Y` gesetzt ist. Gibt die maximal zulässige Anzahl an Zeichen für eine Zeichenkette an. Wenn das angegebene Maximum überschritten wird, tritt ein Laufzeitfehler auf. Gültiger Bereich: `1 - 99`.
Item Must be Numeric	**Element muss numerisch sein** Gilt nur, wenn die Option **Data Allowed** auf `Y` gesetzt ist. Gibt an, ob jeder Datenwert ein Ganzzahlwert sein muss. `Y` Die Dateneingabe muss ein positiver ganzzahliger Wert sein. Wenn nicht, tritt ein Laufzeitfehler auf. N Die Daten können von beliebigem Typ sein.
Put to Top of Stack	**Auf den Natural-Stack legen** Gilt nur, wenn die Option **Data Allowed** auf `Y` gesetzt ist. Gibt an, wo die Daten platziert werden sollen. `Y` Die Daten werden oben auf dem Natural-Stack abgelegt. `1 - 9` Die Daten werden in der *n*-ten Ausprägung des DDM-Feldes `RESULT-FIELD` abgelegt. Wenn die Ausprägung bereits als Ergebnis einer Laufzeitaktion gefüllt wurde, wird sie überschrieben.
If Error, Drop all Data	**Bei Fehler, alle Daten verwerfen** Gilt nur, wenn die Option **Data Allowed** auf `Y` oder `N` gesetzt ist. Legt die Reaktion auf einen Datenauswertungsfehler fest: `Y` Wenn bei der Auswertung der Daten ein Fehler auftritt, werden die Daten verworfen und die Verarbeitung fortgesetzt. `N` Wenn bei der Datenauswertung ein Fehler auftritt, wird die Kontrolle an den Error Handler übergeben (wie unten beschrieben).

Behandlung von Laufzeitfehlern - Header 5

Die im Bildschirm **Processor Header Maintenance 5** einzugebenden Attribute legen fest, wie Laufzeitfehler behandelt werden sollen:

Feld	Erläuterung
General Error Program	**Allgemeines Fehlerprogramm** Der Name des Programms, das die Kontrolle erhalten soll, wenn während der Laufzeit-Verarbeitung durch den Kommandoprozessor ein Fehler festgestellt wird. Der Natural Stack enthält die folgenden Informationen, wenn dieses Programm aufgerufen wird:

Feld	Erläuterung
	Fehlernummer (N4) Zeilennummer (N4) Status (A1) Programmname (A8) Ebene (Level) (N2) Wenn kein Fehlerprogramm und keine spezielle Fehlerbehandlung angegeben ist (siehe unten), wird das Programm mit dem Namen aufgerufen, der in der Natural-Systemvariablen `*ERROR-TA` enthalten ist. Andernfalls wird eine Natural-Systemfehlermeldung ausgegeben.
Keyword not found	**Schlüsselwort nicht gefunden** Zeigt an, ob eine Aktion angegeben wurde, die ausgeführt werden soll, wenn ein Schlüsselwort nicht gefunden werden konnte.
Keyword missing	**Schlüsselwort fehlt** Zeigt an, ob eine Aktion angegeben wurde, die ausgeführt werden soll, wenn der Schlüsselworttyp fehlt.
Keyword Sequence Error	**Schlüsselreihenfolge fehlerhaft** Gibt an, ob eine Aktion angegeben wurde, die im Falle eines Fehlers in der Reihenfolge der Schlüsselwörter ausgeführt werden soll.
Command not defined	**Kommando nicht definiert** Zeigt an, ob eine Aktion angegeben wurde, die im Falle eines nicht definierten Kommandos ausgeführt werden soll.
Data disallowed	**Unzulässige Daten** Zeigt an, ob eine Aktion angegeben wurde, die im Falle nicht erlaubter Daten ausgeführt werden soll.
Data Format/Length Error	**Datenformat-/Längenfehler** Zeigt an, ob eine Aktion angegeben wurde, die im Falle eines Format-/Längenfehlers ausgeführt werden soll.
General Security Error	**Allgemeiner Sicherheitsfehler** Gibt an, ob eine Aktion angegeben wurde, die ausgeführt werden soll, wenn bei einer allgemeinen Sicherheitsüberprüfung ein Fehler festgestellt wird.
Keyword Security Error	**Schlüsselwort-Sicherheitsfehler** Zeigt an, ob eine Aktion angegeben wurde, die ausgeführt werden soll, wenn bei einer Schlüsselwort-Sicherheitsprüfung ein Fehler festgestellt wird.
Command Security Error	**Kommando-Sicherheitsfehler**

Feld	Erläuterung
	Zeigt an, ob eine Aktion angegeben wurde, die ausgeführt werden soll, wenn bei einer Kommando-Sicherheitsprüfung ein Fehler festgestellt wird.

Statistikdaten - Header 6

Der Bildschirm **Processor Header Maintenance 6** enthält nur Ausgabefelder, die statistische Daten über die für einen Kommandoprozessor angegebenen Schlüsselwörter liefern.

Die folgenden statistischen Informationen werden bereitgestellt:

Feld	Erläuterung
Entry *n* Keywords	**Eintrag *n* Schlüsselwörter** Die Anzahl der im Kommandoprozessor definierten Schlüsselwörter des Typs *n* (ohne Synonyme).
Entry *n* Keywords + Synonyms	**Eintrag *n* Schlüsselwörter und Synonyme** Die Summe der Schlüsselwörter vom Typ *n* und ihrer zugewiesenen Synonyme.
Highest IKN for Entry *n*	**Höchste IKN für Eintrag *n*** Die höchste interne Schlüsselwortnummer für das Schlüsselwort des Typs *n*.
Possible Combinations	**Mögliche Kombinationen** Die Anzahl der möglichen Kombinationen für definierte Schlüsselwörter.
Cataloged Functions	**Katalogisierte Funktionen** Die Anzahl der aktuell katalogisierten Schlagwortkombinationen.

Status - Header 7

Der Bildschirm **Processor Header Maintenance 7** enthält nur Ausgabefelder, die die Zeit und das Datum angeben, wann Teile des Kommandoprozessors ausgeführt oder geändert wurden.

Verwaltung der Schlüsselwörter

Schlüsselwörter sind die Grundelemente für die Definition von Funktionen. Bevor die Definition von Schlüsselwörtern möglich ist, müssen die Header Maintenance-Datensätze angelegt werden (siehe Abschnitt *Header-Daten*).

- Schlüsselwörter definieren
- Editor-Kommandos im Keyword-Editor
- Kommandos zur Positionierung
- Zeilenkommandos

Schlüsselwörter definieren

Die in den Kommandos verwendeten Schlüsselwörter werden mit der Funktion **Define Keywords** und dem **Keyword Editor** erstellt. Der Keyword Editor ähnelt den normalen Natural-Editoren mit dem Unterschied, dass die Zeilen des Editors in separate Felder aufgeteilt sind. Die meisten **Editor-Kommandos** (siehe entsprechenden Abschnitt) und die **Zeilenkommandos** (siehe entsprechenden Abschnitt), die im Natural-Programm-Editor verwendet werden, können auch im Keyword Editor verwendet werden.

Um den Keyword-Editor aufzurufen:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode K (**Define Keywords**) ein.

2 Drücken Sie ENTER.

Der Bildschirm **Keyword Editor** wird angezeigt:

Hier wurden bereits mehrere Schlüsselwörter definiert, die als Beispiele für diesen Abschnitt dienen.

```
09:42:39                    - SYSNCP Keyword Editor -                 2000-05-04
Modify Keywords               Name SAGTEST   Library SYSNCP   DBID 10   FNR 32

I Line E Use  Keyword          IKN   ML Comment
- ---- - ---- ---------------- ----  -- ----------------------------------------
     1 1 Acti MENU             1004   1
     2 1 Acti DISPLAY          1002   2
     3 S Syno SHOW             1002   1
     4 1 Acti DELETE           1001   2
     5 S Syno PURGE            1001   1
     6 S Syno ERASE            1001   1
     7 1 Acti FILE             1003   4
     8 P Parm NAME             4002   2
     9 2 Obje FILE             2001   4
    10 P Parm NUMBER           4001   2
    11 2 Obje DOCUMENT         2003   2
    12 1 Acti INFORMATION      1005   1
    13
    14
- ---- - ---- ----- All ------ ----  -- ----------------------------------------

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd   Exit  Last  List  Flip  -1    +1    Top   Bot   Info  Canc
```

Geben Sie im **Keyword Editor** alle Schlüsselwörter ein, die Sie in Ihrer Kommandosprache verwenden möchten. Diese können in beliebiger Reihenfolge eingegeben werden, mit Ausnahme von Synonymen, die unmittelbar auf die Schlüsselwörter folgen müssen, mit denen sie verbunden sind. Weisen Sie jedem Schlüsselwort einen Typ zu, der angibt, zu welchem Teil der Kommandosyntax das Schlüsselwort gehört. Die Regeln der Kommandosyntax für einen Kommandoprozessor werden im Header des Prozessors angegeben, siehe ***Keyword Runtime Options - Header 1*** im Abschnitt *Header-Daten*. Sie können z.B. angeben, ob ein Schlüsselwort vom Typ 1 (an erster Stelle in einem Kommando), Typ 2, Typ 3, ein Synonym für ein anderes Schlüsselwort oder ein Parameterindikator sein soll.



Anmerkung: Die mit dem Natural-Kommandoprozessor erstellte Kommandosprache unterliegt strengen Syntaxanforderungen. Die Vielfalt an Schlüsselworttypen und -abfolgen, Synonymen und anderen Elementen ermöglicht jedoch die Erstellung einer Vielzahl von Kommandos für anspruchsvolle Anwendungen.

Im obigen Beispiel sind die Schlüsselwörter `DELETE` und `DISPLAY` als Schlüsselwörter des Typs 1 definiert. Wie im Header des Prozessors angegeben, bezeichnen diese Schlüsselwörter Aktionen. Das Schlüsselwort `DOCUMENT` ist als Schlüsselwort des Typs 2 definiert und bezeichnet ein Objekt. Das Schlüsselwort `FILE` hingegen ist sowohl als Schlüsselwort des Typs 1 als auch des Typs 2 definiert und kann daher eine Aktion (ablegen) oder ein Objekt (Akte) bezeichnen, je nachdem, wo es im Kommando steht. Es ist möglich, die beiden Schlüsselworttypen zu Kommandos zu kombinieren, z. B. `DELETE FILE` und `FILE DOCUMENT`.

Sie können die von Ihnen eingegebenen Schlüsselwörter speichern, indem Sie das Kommando SAVE oder STOW in der Kommandozeile eingeben. Das Kommando STOW speichert nicht nur die Schlüsselwortdefinitionen in Quellcodeform, sondern führt auch eine Konsistenzprüfung durch. Sobald ein Schlüsselwort erfolgreich gespeichert wurde, erhält es eine interne Schlüsselwortnummer (IKN), die zur Laufzeit zur Auswertung eines Kommandos verwendet wird. Synonyme sind immer mit einem Master-Schlüsselwort verknüpft und nehmen immer die IKN ihres Masters an.

Jede Zeile im Keyword Editor enthält die folgenden Felder:

Feld	Erläuterung
I	**Information** Ausgabefeld. Ein Informationsfeld, das die folgenden Werte enthalten kann: E Error: Zeigt an, dass ein Definitionsfehler festgestellt wurde. X Zeile ist mit X gekennzeichnet. Y Zeile ist mit Y gekennzeichnet. Z Zeile ist sowohl mit X als auch mit Y gekennzeichnet. S Scan: Suchwert in dieser Zeile gefunden.
Line	**Zeile** Ausgabefeld. Die Zeilennummer des Editors.
E	**Eingabe** Gibt den Eingabetyp für ein Schlüsselwort an, d.h. die Position, an der das Schlüsselwort in einem Kommando eingegeben werden soll: erste, zweite oder dritte Position, Synonym oder Parameterindikator. Im obigen Beispiel eines **Keyword Editor**-Bildschirms hat das Schlüsselwort DELETE den Eingabetyp 1 und DOCUMENT den Typ 2. Mit diesen Schlüsselwörtern kann das Kommando DELETE DOCUMENT definiert werden. Das Feld akzeptiert jedes der folgenden Zeichen als Eingabe: 1 Das in dieser Zeile definierte Schlüsselwort ist als erster Titem (Typparameter) in einer Kommandosequenz zu verwenden. 2 Das in dieser Zeile definierte Schlüsselwort ist als zweiter Titem in einer Kommandosequenz zu verwenden. 3 Das in dieser Zeile definierte Schlüsselwort ist als dritter Titem in einer Kommandosequenz zu verwenden. S Das in dieser Zeile definierte Schlüsselwort ist als Synonym für das vorangehende Schlüsselwort mit dem Titem-Typ 1, 2, 3 oder P zu verwenden. P Das in dieser Zeile definierte Schlüsselwort ist als Parameterindikator in einer Kommandosequenz zu verwenden.

Feld	Erläuterung
	* In dieser Zeile ist kein Schlüsselwort zu definieren. Stattdessen ist die Zeile nur als Kommentarzeile zu verwenden. ? Dieses Symbol ist ein Ausgabewert, der auf eine ungültige Schlüsselwortangabe hinweist.
Use	**Verwendung** Ausgabefeld. Der angezeigte Wert wird durch den im vorhergehenden Feld **E** eingegebenen Wert bestimmt: 1 - 3 Die ersten vier Zeichen des im Prozessor-Header angegebenen Benutzertextes für den ersten, zweiten bzw. dritten Schlüsselworteintrag werden angezeigt. Siehe auch ***Keyword Editor-Optionen - Header 2*** im Abschnitt Header-Daten. S SYNO, die Abkürzung für Synonym, wird angezeigt P PARM, die Abkürzung für Parameterindikator, wird angezeigt.
Keyword	**Schlüsselwort** Geben Sie das zu definierende Schlüsselwort ein. Eingebettete Leerzeichen sind nicht zulässig. Wenn Sie im Processor Header festgelegt haben, dass Schlüsselwörter nur in Großbuchstaben eingegeben werden können, werden Schlüsselwörter immer in Großbuchstaben umgesetzt, unabhängig davon, wie sie eingegeben werden. Andernfalls wird die Großschreibung beibehalten. Die maximale und minimale Länge von Schlüsselwörtern hängt von den Einstellungen im Header ab (Standard: 1 - 16 Zeichen). Schlüsselwörter müssen eindeutig sein, sofern im Header nicht anders angegeben. Schlüsselwort-Präfixe können verwendet werden, wie unter ***Keyword Editor-Optionen - Header 2*** im Abschnitt *Header-Daten* beschrieben.
IKN	**Interne Schlüsselwortnummer** Ausgabefeld. Die interne Schlüsselwortnummer (IKN) ist eine Kennung, die jedem gültigen Schlüsselwort zugewiesen wird. IKNs sind nützlich bei Tests und zur Fehlersuche. Sie werden nur zugeordnet, wenn ein Schlüsselwort erfolgreich gespeichert wurde (mit dem Kommando STOW, siehe ***Editor-Kommandos***). Jedem Schlüsselwort wird eine eindeutige IKN zugewiesen, mit Ausnahme von Synonymen, die die IKN ihres Hauptbegriffs erhalten (siehe das obige Beispiel im Schlüsselwort-Editor: DISPLAY und SHOW).
ML	**Mindestlänge** Eingabe- und Ausgabefeld zur Angabe der Mindestlänge eines Schlüsselworts. Das Feld ist ein Eingabefeld, wenn S im Feld **Dynamic Length Adjustment** (Dynamische Längenanpassung) im Prozessor-Header angegeben wird, wie in *Schlüsselwort-Laufzeit-Optionen - Header 1*, *Header-Daten* beschrieben. In diesem Fall müssen Sie die Anzahl der Zeichen angeben, die für das Schlüsselwort eingegeben werden müssen. Bei allen anderen Eingaben enthält dieses Feld die Mindestanzahl der Zeichen eines Schlüsselworts, die ein Benutzer angeben muss, um eine Verwechslungsmöglichkeit mit anderen Schlüsselwörtern zu vermeiden.

Feld	Erläuterung
	Im obigen Beispiel des **Keyword-Editors** ist für das Schlüsselwort MENU nur die Eingabe von M erforderlich, während für das Schlüsselwort DISPLAY die Eingabe von DI erforderlich ist, um eine Verwechslungsmöglichkeit mit dem Schlüsselwort DELETE zu vermeiden.
Comment	**Kommentar** Sie können einen Freitext zu einem Schlüsselwort eingeben. Es gibt keine Eingabebeschränkungen. Der Benutzertext wird in den katalogisierten Kommandoprozessor aufgenommen, wenn das Feld **Catalog User Texts** (Benutzertexte katalogisieren) in der Header-Definition auf Y gesetzt ist, wie in *Verschiedene Options - Header 3*, *Header-Daten* beschrieben. Er kann zur Laufzeit mit der TEXT-Option des PROCESS COMMAND-Statements gelesen werden. Der oben in dieser Spalte erscheinende Überschriftentext wird durch die Felder **Header for User Text 1** und **Header for User Text 2** in der Header-Definition gesteuert.

Editor-Kommandos im Keyword-Editor

In der Kommandozeile des **Keyword-Editors** können Sie die folgenden Kommandos eingeben:

Kommando	Funktion
ADD	Hängt zehn Leerzeilen am Ende des Editors an.
CANCEL	Rückkehr zum **Processor Maintenance**-Menü.
CHECK	Prüft den Schlüsselwortquelltext auf Konsistenz.
EXIT	Rückkehr zum **Processor Maintenance**-Menü.
HELP	Zeigt gültige Escape-Zeichen und andere nützliche Prozessoreinstellungen an.
INFO	Zeigt Informationen zu dem Schlüsselwort an, auf dem sich der Cursor befindet.
LET	Macht alle Änderungen am aktuellen Bildschirm rückgängig, die seit dem letzten Drücken von ENTER vorgenommen wurden.
POINT	Platziert die Zeile, in der ein Zeilenkommando .N eingegeben wird, an den Anfang des aktuellen Bildschirms.
RECOVER	Stellt den Schlüsselwortquelltext wieder her, der vor dem letzten SAVE/STOW vorhanden war.
RESET	Löscht die aktuellen X- und Y-Zeilenmarkierungen.
SAVE	Der Schlüsselwortquelltext wird gespeichert.
SCAN	Sucht nach dem nächsten Auftreten des Scan-Wertes.
STOW	Der Schlüsselwortquelltext wird gespeichert und für gültige Schlüsselwörter werden interne Schlüsselwortnummern (IKNs) generiert.

Kommandos zur Positionierung

Die Positionierungskommandos des Editors sind die gleichen wie beim Natural-Programm-Editor. Weitere Informationen siehe *Editor-Kommandos zum Positionieren* in der *Editoren*-Dokumentation.

In der letzten Zeile des Editors befindet sich ein Ausgabefeld, das Sie darüber informiert, wo sich Ihre Anzeige im Editor befindet. Es werden die folgenden Ausgabewerte angezeigt:

Top	Editor ist zurzeit am oberen Rand des Schlüsselwort-Quelltextes.
Mid	Editor ist zurzeit in der Mitte des Schlüsselwort-Quelltextes.
Bot	Editor ist zurzeit am unteren Rand des Schlüsselwort-Quelltextes.
Emp	Editor ist zurzeit leer.
All	Der gesamte Quelltext ist auf dem aktuellen Bildschirm.

Zeilenkommandos

Die Zeilenkommandos im Keyword-Editor sind die gleichen wie im Natural-Programm-Editor, mit Ausnahme der Kommandos `.J` und `.S`, die nicht verwendet werden können.

Jedes Kommando wird beginnend im Feld **E** eingegeben; der restliche Teil des Kommandos wird im Feld **Keyword** eingegeben, wie in der folgenden Abbildung dargestellt:

```
09:42:39                    - SYSNCP Keyword Editor -                 2000-05-04
Modify Keywords               Name SAGTEST   Library SYSNCP   DBID 10   FNR 32

I Line E Use  Keyword          IKN   ML Comment
- ---- - ---- ---------------- ----  -- ----------------------------------------
     1 1 Acti MENU             1004   1
     2 1 Acti DISPLAY          1002   2
     3 S Syno SHOW             1002   1
     4 . Acti i(3)TE           1001   2
     5 S Syno PURGE            1001   1
```

Vorsicht: Achten Sie beim Verschieben (`.M`) oder Kopieren (`.C`) von Zeilen darauf, dass einzelne Schlüsselwörter immer zusammen mit ihren Synonymen verschoben oder kopiert werden.

Wenn Sie Zeilen löschen (`.D`), werden die entsprechenden Schlüsselwörter und alle Funktionen, die diese Schlüsselwörter enthalten, erst dann aus der Datenbank gelöscht, wenn Sie das Editorkommando `STOW` absetzen. Solange Sie das `STOW`-Kommando nicht absetzen, werden diese Funktionen im **Function Editor** weiterhin angezeigt.

Verwaltung der Funktionen

Funktionen setzen sich aus den im **Keyword Editor** eingegebenen Schlüsselwörtern zusammen. Bevor Funktionen definiert werden können, müssen die Schlüsselwörter erfolgreich mit `STOW` gespeichert werden (siehe Abschnitt *Schlüsselwörter verwalten*).

- Funktionen definieren
- Editor-Kommandos im Function Editor
- Direktkommando QUICK-EDIT
- Lokale und globale Funktionen
- Vorgehensweise bei der Validierung von Funktionen

Funktionen definieren

Benutzen Sie die Funktion **Define Functions** und den **Function Editor**, um Funktionen zu spezifizieren und gültige Kommandos zusammenzustellen, auf die von einer bestimmten Stelle aus zugegriffen werden kann.

Um den Function Editor aufzurufen:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode `F` (**Define Functions**) ein.

2 Drücken Sie ENTER.

Der Bildschirm **Function Editor** wird angezeigt.

Der **Function Editor** zeigt alle möglichen Kombinationen der Schlüsselwörter an, die im **Keyword Editor** mittels `STOW` gespeichert sind.

Der nachstehende Bildschirm zeigt den Function Editor mit den Schlüsselwörtern, die im Abschnitt *Schlüsselwörter verwalten* im **Keyword Editor**-Bildschirm als Beispiele verwendet wurden:

```
09:45:53               ***** NATURAL SYSNCP UTILITY *****            2000-05-04
User SAG                       - Function Editor -
Edit Global Combinations    Name SAGTEST  Library SYSNCP    DBID 10   FNR 32

Global
I Ac   Action           Object           Addition        Global Local Any Loc
- --   ---------------- ---------------- --------------- ------ ----- -------
       DELETE
       DELETE           DOCUMENT                                       Yes
       DELETE           FILE                                           Yes
       DISPLAY
       DISPLAY          DOCUMENT                                       Yes
       DISPLAY          FILE                                           Yes
       FILE
       FILE             DOCUMENT                                       Yes
       FILE             FILE                                           Yes
       INFORMATION                                          Yes
       INFORMATION      DOCUMENT
       INFORMATION      FILE
Repos: ________________ ________________ ________________ ------ ----- -------

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd   Exit  Last  List  Flip        +     Top    Loc  Loc+  Canc
```

Sie müssen jede Schlüsselwortkombination validieren, die Sie als gültige Funktion in Ihrer Anwendung festlegen wollen. Eine Schlüsselwortkombination kann als globale Funktion, lokale Funktion oder beides (Any Location) validiert werden. Eine globale Funktion kann von überall in einer Anwendung aufgerufen werden, während eine lokale Funktion nur von einer bestimmten Stelle innerhalb einer Anwendung aufgerufen werden kann.

Zwei Felder in der oberen linken Ecke dieses Bildschirms zeigen den aktuellen Validierungsmodus (lokal oder global) und die Stelle an, für die Schlüsselwortkombinationen derzeit validiert werden können. Auf dem obigen Bildschirm zeigt der Text `Edit Global Combinations` (Globale Kombinationen bearbeiten) an, dass der globale Modus aktiv ist. Wäre der lokale Modus aktiv, würde hier der Text `Edit Local Combinations` erscheinen. Auf dem obigen Bildschirm erscheint der Text `Global` unter diesem Text. Dies bedeutet, dass die globale Validierung für alle aufgeführten Kombinationen durchgeführt werden kann. Im lokalen Modus erscheint in diesem Feld der Name der Stelle, für die die lokale Validierung durchgeführt werden kann (z. B. `Local DISPLAY FILE`).

Der **Function Editor** enthält die folgenden Spalten:

Spalte	Erläuterung
I	**Information** Ausgabefeld. Die folgenden Werte werden als Ergebnis der Funktionsbearbeitung ausgegeben: E Laufzeit-Aktion bearbeitet. D Referenzierte Stellen angezeigt. V Validierung abgesetzt. R Validierung aufgehoben.
Ac	**Aktion** Zu ergreifende Maßnahme. Die folgenden Werte können eingegeben werden: VG Validierung als globale Funktion. VL Validierung als lokale Funktion. RG Validierung als globale Funktion aufheben. RL Validierung als lokale Funktion aufheben. DL Alle Funktionen anzeigen, die die angegebene Funktion als lokale Funktion referenzieren. EG Aufruf des **Runtime Action Editor** für eine globale Funktion (siehe ***Runtime Action Editor*** im Abschnitt *Laufzeit-Aktionen*). EL Aufruf des **Runtime Action Editor** für eine lokale Funktion (siehe ***Runtime Action Editor*** im Abschnitt *Laufzeit-Aktionen*). +G Aufruf des globalen Modus, so dass Sie alle globalen Funktionen pflegen können. +L Aufruf des lokalen Modus für die aktuelle Zeile, so dass Sie lokale Funktionen für diese Zeile pflegen können. IN Informationen über Schlüsselwörter in dieser Zeile.
Action Object Addition	**Aktion/Objekt/Hinzufügung** Diese drei Spalten dienen dazu, alle möglichen Kombinationen der aktuell definierten Schlüsselwörter anzuzeigen. Der Text, der oben in jeder Schlüsselwortspalte erscheint, wird durch die Felder **First Entry used as** (Erster Eintrag verwendet als), **Second Entry used as** (Zweiter Eintrag verwendet als) und **Third Entry used as** (Dritter Eintrag verwendet als) gesteuert, die im Prozessor-Header angegeben sind (siehe ***Keyword Runtime Options - Header 1*** im Abschnitt *Header-Daten*).
Global	**Globales Kommando** Wenn die Funktion als globales Kommando definiert wurde, erscheint in diesem Feld `Yes`.
Local	**Lokales Kommando** Wenn die Funktion als lokales Kommando definiert wurde, erscheint in diesem Feld `Yes` für die aktuelle Position (nur im lokalen Modus angezeigt).
Any Loc	Any Location (beliebige Stelle). Wenn die Funktion an einer anderen Stelle des Prozessors als lokales Kommando definiert wurde, erscheint in diesem Feld `Yes` für jede andere Stelle.

Editor-Kommandos im Function Editor

In der Kommandozeile des **Function Editor** können Sie die folgenden Kommandos eingeben:

Kommando	Funktion
ANY ON	Aktiviert die Spalte **Any Loc** (Beliebige Stelle).
ANY OFF	Deaktiviert die Spalte **Any Loc** (die Spalte wird mit Fragezeichen gefüllt). Dies ermöglicht ein schnelleres Blättern im Function Editor. Darüber hinaus ist das dritte Repositionierungsfeld verfügbar. Außerdem werden die Informationsfenster für die laufende Verarbeitung nicht angezeigt.
FIELD	Schlüsselwortspezifische Kombinationen anzeigen.
GLOBAL	Aktiviert den globalen Modus.
LOC	Positionierung zur nächsten Gruppe von Stellen.
LOC+	Positionierung um eine Stelle vorwärts.
SINGLE ON	Nur Funktionen mit einem Wort anzeigen.
SINGLE OFF	Alle möglichen Kombinationen anzeigen.
TOP	Positionierung an den Anfang der Liste.

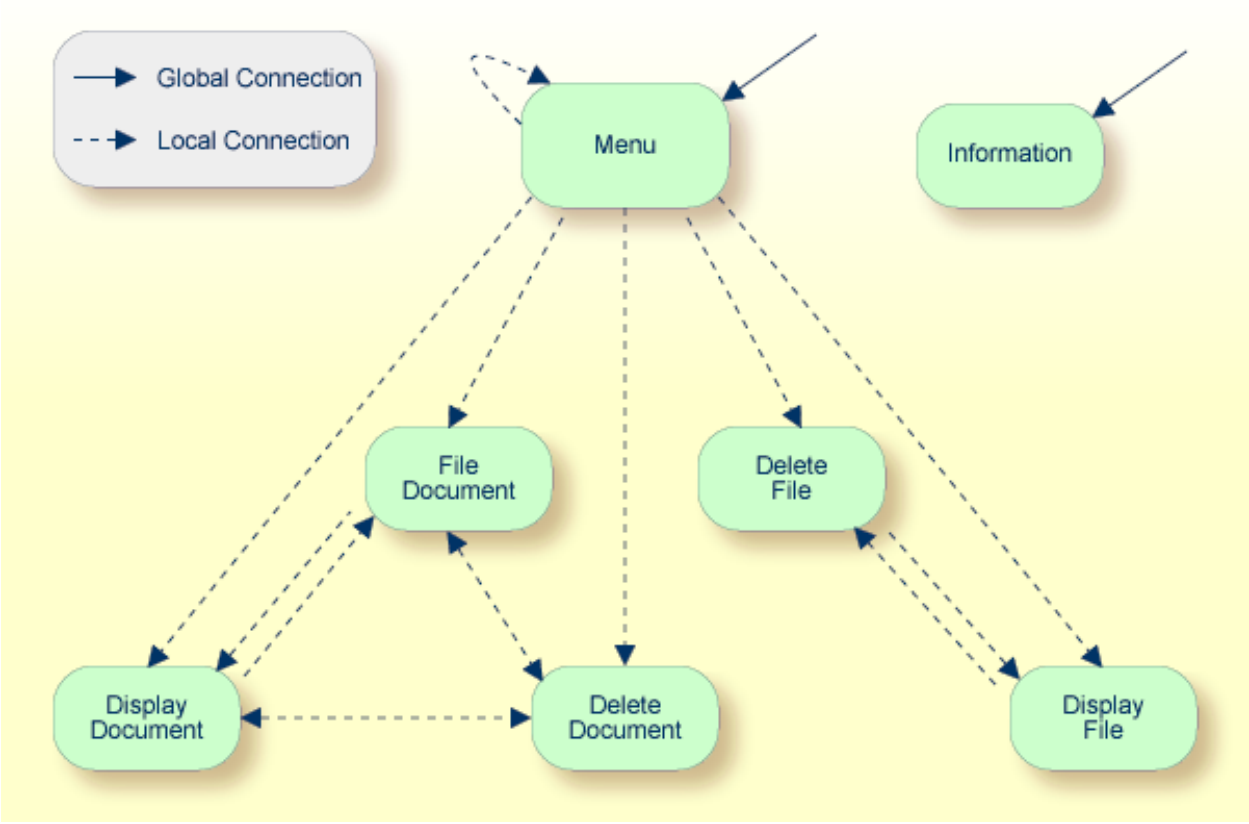
Direktkommando QUICK-EDIT

Das Direktkommando `QUICK-EDIT` ermöglicht die schnelle Definition lokaler/globaler Funktionen sowie der entsprechenden Laufzeit-Aktionen durch die direkte Eingabe von Schlüsselwörtern oder IKNs. Dies kann bei extrem großen Kommandoprozessoren hilfreich sein. Beachten Sie jedoch, dass die Stelle, von der aus das Kommando abgesetzt werden kann, nicht überprüft wird und die Navigation zur Laufzeit möglicherweise nicht korrekt funktioniert.

Lokale und globale Funktionen

Um das Konzept lokaler und globaler Funktionen zu verstehen, müssen Sie sich jede gültige Schlüsselwortkombination als eine Stelle in Ihrer Anwendung vorstellen (zum Beispiel eine Stelle namens **Display File**). Im **Function Editor** legen Sie fest, welche Kommandos von dieser Stelle aus abgesetzt werden können und von welchen Stellen aus diese Stelle mit dem Kommando `DISPLAY FILE` erreicht werden kann.

Lokale und globale Verbindungen in einer Beispielanwendung:



In der obigen Beispielanwendung sind die Stellen **Menu** und **Information** die einzigen Stellen, die als global bezeichnet wurden. Daher kann von allen anderen Stellen in der Anwendung direkt auf sie zugegriffen werden. Alle Stellen wurden als lokal für die Stelle **Menu** gekennzeichnet, mit Ausnahme von **Information**. Die einzige Möglichkeit, von der Stelle **Display File** zur Stelle **Display Document** zu gelangen, ist über **Menu**.

Vorgehensweise bei der Validierung von Funktionen

Der **Function Editor** arbeitet in zwei Modi: global und lokal. Im globalen Modus können Sie globale Funktionen validieren und im lokalen Modus können Sie globale und lokale Funktionen validieren. Der globale Modus ist der Standardmodus. Ob sich der Editor im globalen oder lokalen Modus befindet, erkennen Sie an dem Ausgabefeld oberhalb des Feldes **I** im Editor. Befindet sich der Editor im globalen Modus, so wird `Global` angezeigt. Befindet sich der Editor im lokalen Modus, so wird die Stelle (Location) angezeigt, für die lokale Funktionen validiert werden sollen. Die generelle Vorgehensweise für die Validierung globaler und lokaler Funktionen für eine Anwendung wird im Folgenden beschrieben.

Um globale und lokale Funktionen zu validieren:

1 Wenn sich der **Function Editor** im globalen Modus befindet, geben Sie `VG` (Validate Global) in das Feld **Ac** neben der entsprechenden Aktion ein, um alle globalen Funktionen zu validieren.

Drücken Sie ENTER.

Sie gelangen zum Bildschirm **Runtime Action Definition**.

2 Drücken Sie PF3, um zum **Function Editor** zurückzukehren.

`Yes` erscheint unter der Spaltenüberschrift **Global** neben den validierten Funktionen.

3 Geben Sie `+L` in das Feld **Ac** für jede globale Funktion ein, die im vorherigen Schritt validiert wurde, um in den lokalen Modus zu wechseln.

Drücken Sie ENTER.

4 Geben Sie `VL` (Validate Local) in das Feld **Ac** für jede Funktion ein, die als Stelle (Location) für diese globale Funktion dienen soll.

Drücken Sie ENTER.

Sie gelangen zum Bildschirm **Runtime Action Definition**.

5 Drücken Sie PF3, um zum **Function Editor** zurückzukehren.

`Yes` erscheint unter der Spaltenüberschrift **Local** neben den validierten Funktionen.

6 Um lokale Funktionen für eine *lokale* Stelle (Location) zu validieren:

Geben Sie `+L` (Aufruf des lokalen Modus) in das Feld **Ac** bei jeder Stelle (Location) ein, die im vorherigen Schritt validiert wurde, um alle lokalen Funktionen zu validieren, die von dieser Stelle aus benutzt werden sollen.

Drücken Sie ENTER.

7 Geben Sie `VL` (Validate Local) in das Feld **Ac** für jede Funktion ein, die als lokale Funktion für die aktuelle Stelle (Location) dienen soll.

8 Drücken Sie PF3, um zum **Function Editor** zurückzukehren.

`Yes` erscheint unter der Spaltenüberschrift **Local** neben den validierten Funktionen.

Anmerkung: Wenn im Header des Kommandoprozessors (**Processor Header Maintenance 3**) das Feld **Invoke Action Editor** auf **Y** gesetzt ist, wird zusätzlich für jede Aktion das Fenster **Runtime Action Definition** (siehe *Runtime Action Editor* im Abschnitt *Laufzeit-Aktionen*) angezeigt.

Laufzeit-Aktionen

Sobald gültige Schlüsselwortkombinationen als lokale oder globale Funktionen im **Function Editor** identifiziert worden sind, ist es möglich, jede Funktion mit einer oder mehreren Laufzeit-Aktionen zu verknüpfen. Laufzeit-Aktionen bestehen aus einem oder mehreren Schritten, die immer dann ausgeführt werden sollen, wenn eine Funktion aufgerufen wird.

Nachfolgend finden Sie Informationen zu:

- Laufzeit-Aktionen definieren
- Runtime Action Editor

Laufzeit-Aktionen definieren

In SYSNCP gibt es zwei verschiedene Stellen, an denen Sie Laufzeit-Aktionen definieren können: den **Function Editor** (siehe den Abschnitt *Function Maintenance*) und den **Result Editor**. Der **Result Editor** wird in diesem Abschnitt erläutert, einschließlich der Angabe von Laufzeit-Aktionen für eine Funktion.

Laufzeit-Aktionen definieren

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode **R** (**Define Runtime Actions**) ein.

2 Drücken Sie ENTER.

Um den **Result Editor** aufzurufen:

```
09:47:03              ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG                        - Result Editor -
List defined combinations   Name SAGTEST  Library SYSNCP   DBID 10    FNR 32


I Ac Location                        Command                          Result
- -- ------------------------------- ------------------------------- --------
     < Global >                       MENU                             KR
     < Global >                       INFORMATION                      SF
     DELETE FILE                      DISPLAY FILE                     SF
     DELETE DOCUMENT                  DISPLAY DOCUMENT                 SF
     DISPLAY FILE                     DELETE FILE                      SF
     DISPLAY DOCUMENT                 DELETE DOCUMENT                  SF
     DISPLAY DOCUMENT                 FILE DOCUMENT                    SF
     FILE DOCUMENT                    DELETE DOCUMENT                  SF
     FILE DOCUMENT                    DISPLAY DOCUMENT                 SF
     MENU                             DELETE FILE                      KCS
     MENU                             DELETE DOCUMENT                  KCCS
     MENU                             DISPLAY FILE                     KRCS
Repo _________________________________ _______________________________ --------

 Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Cmd   Exit  Last  List  Flip       +     Top   Loc-- Loc+  Canc
```

Der **Result Editor** enthält alle lokalen und globalen Funktionen, die im **Function Editor** angegeben wurden. Jede Zeile im Editor stellt die Stelle (Location) dar, von der aus ein Kommando abgesetzt werden kann (Feld **Location**), das Kommando selbst (Feld **Command**) und eine kurze Beschreibung der Aktion, die ausgeführt wird, wenn das Kommando abgesetzt wird (Feld **Result**).

Die Felder des Bildschirms werden in der nachstehenden Tabelle erläutert:

Feld	Erläuterung
I	**Information** Ausgabefeld. Informationen zu der zuletzt in dieser Zeile ausgeführten Aktion.
Ac	**Aktion** Auszuführende Aktion. Die folgenden Werte können eingegeben werden: DI Anzeige der Laufzeit-Aktionsdefinitionen für diese Funktion. ED Bearbeitung der Laufzeitaktionsdefinitionen für diese Funktion. PU Löschung dieser Funktion.
Location	**Stelle**

Feld	Erläuterung
	Ausgabefeld. Die Stelle innerhalb der Anwendung, von der aus das Kommando (siehe Feld **Command** weiter unten) ausgegeben werden kann. Wenn es sich um eine globale Funktion handelt, erscheint in diesem Feld `< Global >` (das Kommando kann von jeder Stelle aus ausgegeben werden).
Command	**Kommando** Ausgabefeld. Das Kommando. Der Inhalt der Felder **Location** und **Command** kann abgeschnitten werden, wenn sehr lange Schlüsselwörter verwendet werden.
Result	**Ergebnis** Enthält eine Kurzbeschreibung der Aktion, die ausgeführt werden soll, wenn das Kommando abgesetzt wird. Das erste Zeichen steht für die Information **Keep Location** (Stelle beibehalten, siehe folgenden Abschnitt). Alle anderen Zeichen finden Sie in der Tabelle *Runtime Action Definition* weiter unten.

Runtime Action Editor

Der **Runtime Action Editor** wird verwendet, um die Aktionen zu definieren, die ausgeführt werden sollen, wenn ein Kommando von einer bestimmten Stelle aus abgesetzt wird. Der Editor kann nur für Funktionen aufgerufen werden, die als globale oder lokale Funktionen definiert wurden. Der Editor kann entweder aus dem **Function Editor** oder aus dem **Result Editor** aufgerufen werden.

Um den Runtime Action Editor aus dem Function Editor aufzurufen:

1 Geben Sie im Feld **Ac** für globale Funktionen die Buchstaben `EG` (Edit Global) ein.

Oder:

Geben Sie im Feld **Ac** für lokale Funktionen die Buchstaben `EL` (Edit Local) ein.

2 Drücken Sie ENTER.

Um den Runtime Action Editor aus dem Result Editor aufzurufen

1 Geben Sie im Feld **Ac** die Buchstaben `ED` ein.

2 Drücken Sie ENTER.

Das Fenster **Runtime Action Definition** wird angezeigt:

```
                        Runtime Action Definition

Location .... DISPLAY DOCUMENT
Command ..... DELETE DOCUMENT

Keep Location .... S
Data allowed ..... Y   More than one .... N   Max. Length ...... 99
Numeric .......... N   TOP of STACK ..... Y   Error: Drop ...... Y

A  Runtime Action Definition
-  ------------------------------------------------------------------
F  DE-PGM_____________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
_  _____________________________________________________________________
```

Aktionen sind immer mit einem Ursprung und einem Ziel verbunden. Der Ursprung ist die Stelle (Location), von der aus das Kommando abgesetzt wird, und das Ziel ist das Kommando selbst. So ist es möglich, verschiedene Aktionen mit einem Kommando zu verknüpfen, je nach dem Kontext, in dem es verwendet wird.

Im **Runtime Action Editor** legen Sie außerdem fest, ob die Stelle (Location) nach der Ausführung der Aktionen gleich bleiben soll oder ob das Kommando selbst die neue aktuelle Stelle (Location) werden soll.

Aktionen werden durch Eingabe eines aus einem Buchstaben bestehenden Codes in der linken Spalte des Editors angegeben. Die Parameter, die zu einer Aktion gehören, werden in das Feld neben dem Code eingegeben. Wenn ein Schrägstrich und ein Stern (`/*`) in dieses Feld eingegeben werden, gelten alle nachfolgenden Eingaben als Kommentar. Wenn Sie einen erforderlichen Parameter auslassen, werden Sie zur Eingabe aufgefordert.

Die Reihenfolge, in der die Aktionen zur Laufzeit ausgeführt werden, richtet sich nach der Reihenfolge der Eingabe im Editor (von oben nach unten). Wenn also ein `FETCH` angegeben wird, werden alle darunter angegebenen Aktionen nicht ausgeführt.

Der Runtime Action Editor enthält die folgenden Felder:

Feld	Erläuterung
Location	**Stelle** Ausgabefeld. Die Stelle, von der aus das Kommando abgesetzt wird. Wenn die Funktion als global definiert ist, zeigt das Feld `<Global>` an.
Command	**Kommando** Ausgabefeld. Kommando, für das Aktionen angegeben werden sollen.
Keep Location	**Stelle beibehalten** Gibt an, ob die aktuelle oder eine neue Stelle aktiv sein soll, nachdem die Aktionen ausgeführt wurden. Ein Wert in diesem Feld wirkt sich nur auf Kommandos mit angegebener EXEC-Option aus. Mögliche Werte: K Aktuelle Stelle beibehalten. Die auszuführenden Aktionen wirken sich nur auf den aktuellen Ort aus. S Neue Stelle festlegen (global/lokal). Sobald die Aktionen ausgeführt sind, macht der Kommandoprozessor das Kommando zur neuen aktuellen Stelle. Jedes anschließend eingegebene Kommando muss entweder ein lokales Kommando dieser neuen Stelle oder ein globales Kommando sein. **Anmerkung:** Die definierten Aktionen selbst haben keinen Einfluss auf die Stelle, d.h. jede ausgeführte Aktion führt *nicht* zu einer Änderung der aktuellen Stelle.
Other Options	**Sonstige Optionen** Alle anderen Optionen beziehen sich auf die Behandlung von Parametern, die mit dieser Kommandosequenz übergeben werden. Weitere Informationen finden Sie unter *Behandlung von Kommandodaten - Header 4* im Abschnitt Header-Daten. Um die Header-Vorgaben dieser Optionen zu aktivieren, geben Sie einen Stern (*) ein.

Um Laufzeitaktionen zu definieren:

1 Rufen Sie das Fenster **Runtime Action Definition** (Laufzeit-Aktionsdefinition) wie oben beschrieben auf.

2 Geben Sie in das Feld **A** einen Aktionscode und in das gegenüberliegende Feld die entsprechende Aktion ein:

Code	Laufzeit-Aktionsdefinition
V	Standardwert. Es ist keine Laufzeit-Aktion angegeben.
T	Text, der zur Laufzeit mit der Option `TEXT` oder `GET` des Statements `PROCESS COMMAND` gelesen werden kann.
M	Kommandozeile ändern. Die Daten werden in die Kommandozeile gestellt.
C	Kommando. Dieses Kommando wird auf den Natural-Stack gelegt. Wird hier ein Stern (*) angegeben, wird der Name des Programms, das dieses `PROCESS COMMAND`-Statement

Code	Laufzeit-Aktionsdefinition
	ausgegeben hat, oben auf dem Natural-Stack abgelegt (`STACK TOP COMMAND '*PROGRAM'`). (*)
D	D Daten. Diese Daten werden oben auf dem Natural-Stack abgelegt. (*)
F	Natural-Programmname. Das Programm wird mit einem `FETCH`-Statement aufgerufen. (*)
S	Natural `STOP`-Statement. Das Statement wird zur Laufzeit ausgeführt. (*)
E	Der in dieser Zeile angegebene Wert soll sofort in die Systemvariable `*ERROR-NR` übertragen werden.
R	In das DDM-Feld `RETURN-CODE` wird ein Rückgabecode eingetragen. Siehe `PROCESS COMMAND`-Statement, Abschnitt *DDM COMMAND*.
1 bis 9	Ein Textstring. Dieser Wert wird in das multiple DDM-Feld `RESULT-FIELD` eingetragen. Siehe `PROCESS COMMAND`-Statement, Abschnitt *DDM COMMAND*.
*	Kommentarzeile.

(*) Diese Aktionen werden nur mit der Option `EXEC` des `PROCESS COMMAND`-Statements ausgeführt.

3 Drücken Sie PF3, um das Fenster **Runtime Action Definition** zu verlassen.

Anmerkung: Mit dem User-Exit `NCP-REAM` können Sie einige oder alle der oben genannten Codes verwenden. Mit dem User-Exit `NCP-REEM` können Sie die Zeile unterhalb der Überschrift der **Runtime Action Definition**-Tabelle ändern. Der User-Exit `NCP-REDM` ermöglicht es Ihnen, Standardwerte für Laufzeit-Aktionsdefinitionen zu definieren (wenn Sie diesen User-Exit verwenden, siehe auch Feld **Invoke Action Editor** im Abschnitt *Verschiedene Optionen - Header 3*). Alle oben genannten User-Exits werden in der Natural System Library SYSNCP ausgeliefert.

Kommandoprozessor katalogisieren

Nachdem Sie Laufzeit-Aktionen für alle Funktionen, die Sie in Ihrem Kommandoprozessor verwenden wollen, angegeben haben, sollten Sie den Kommandoprozessor katalogisieren. Durch das Katalogisieren eines Kommandoprozessors wird ein Natural-Objekt vom Typ Processor generiert.

Um einen Kommandoprozessor zu katalogisieren:

1 Geben Sie im Menü **Processor Maintenance** den Funktionscode `C` (**Catalog Processor**) ein, den Namen des Kommandoprozessors, der katalogisiert werden soll, und den Namen der Natural Library, in der der Kommandoprozessor katalogisiert werden soll.

2 Drücken Sie ENTER.

Anmerkung: Wenn bei Ihnen Natural Security installiert ist, muss die Verwendung Ihres Kommandoprozessors zugelassen werden. Weitere Informationen siehe *Functional Security* in der *Natural Security*-Dokumentation.

Administrative SYSNCP-Funktionen

SYSNCP bietet im Menü **Administrator Services** Funktionen für die Verwaltung von Kommandoprozessoren. Nur in Natural Security zugelassene Systemadministratoren sind zum Zugriff auf diese Dienste berechtigt.

Um die Administrator Services aufzurufen:

1 Geben Sie im Menü **Processor Source Maintenance** den Funktionscode **A** (**Administrator Services**) ein.

2 Drücken Sie ENTER.

Der Bildschirm **Administrator Services** wird angezeigt:

```
09:49:11              ***** NATURAL SYSNCP UTILITY *****          2000-05-04
User SAG                    - Administrator Services -

                     Code   Function

                      S     Select Processor
                      C     Copy Processor Source
                      D     Delete Processor Source
                      P     Print Source/Object/NCP-Buffer
                      U     Unload Processor to Work File 3
                      L     Load Processor from Work File 3
                      F     Freeze Processor Source
                      R     References from Natural Security
                      ?     Help
                      .     Exit

               Code .. _     Name .. SAGTEST_  Library .. SYSNCP__




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

Anmerkung: Wenn bei Ihnen kein Natural Security installiert ist, sollten Sie sich darüber im Klaren sein, dass alle anderen Benutzer ebenfalls den Status eines Administrators haben.

Nachfolgend finden Sie Informationen zu:

- Prozessor auswählen - Select Processor
- Prozessor-Quellcode kopieren - Copy Processor Source
- Prozessor-Quellcode löschen - Delete Processor Source
- Quellcode-Objekt, Objektmodul, NCP-Buffer drucken
- Prozessor entladen - Unload Processor
- Prozessor laden - Load Processor
- Prozessor-Quellcode einfrieren - Freeze Processor Source
- Referenzen von Natural Security

Prozessor auswählen - Select Processor

Siehe Abschnitt *Auswahl des Prozessors*.

Prozessor-Quellcode kopieren - Copy Processor Source

Beim Kopieren von Prozessor-Quellcode haben Sie die Wahl, den gesamten Prozessor oder nur ausgewählte Quellcodes (Header, Schlüsselwörter, Funktionen, Laufzeit-Aktionsdefinitionen) zu kopieren.

Um einen Kommandoprozessor zu kopieren:

1 Geben Sie im Menü **Administrator Services** den Funktionscode **C** ein.

2 Drücken Sie ENTER.

Das Fenster **Copy Processor Source** zur Angabe von Quell- und Ziel-Informationen erscheint:

```
                        Copy Processor Source

                Source           Target

Name ........ SAGTEST_          ________
Library ..... SYSNCP__          SYSNCP__
DBID ........ 10___             10___
FNR ......... 32___             32___
Password ....
Cipher Key ..

Replace ..... NO_
```

3 Geben Sie in die Felder in der Spalte **Source** (Quelle) den Namen des zu kopierenden Prozessors, die Library, die Datenbankkennung (DBID) und die Nummer der Datei (FNR) ein, in der der Prozessor gespeichert ist. Die Standardwerte entsprechen dem im Menü **Administrator Services** angegebenen Prozessor.

Geben Sie in den Feldern in der Spalte **Target** (Ziel) den Namen des Prozessors ein, in den kopiert werden soll, sowie die Library, die Datenbankkennung (DBID) und die Nummer der Datei (FNR), in die der Prozessor kopiert werden soll.

Geben Sie in das Feld **Cipher Key** (Chiffrierschlüssel) das entsprechende Passwort und/oder den Chiffrierschlüssel ein, wenn die Quell- und/oder die Ziel-Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist.

Geben Sie in das Feld **Replace** `YES` ein, wenn Sie einen Prozessor in der Ziel-Umgebung überschreiben wollen. Die Voreinstellung für dieses Feld ist `NO`.

4 Drücken Sie ENTER.

Das folgende Fenster wird zur Auswahl der Quellen angezeigt (Beispiel):

```
                        Copy Processor Source

 Mark  Copy                           Source  Target
 ----  -----------------------------  ------  ------
  _    Header ......................   yes      no
  _    Keywords ....................   yes      no

  _    Functions ...................   yes      no
       Runtime Action Definitions ..   no       no


Source Name SAGTEST     Library SYSNCP    DBID 10    FNR 32
Target Name TEST2       Library SYSNCP    DBID 10    FNR 32

Replace ... NO
```

5 Geben Sie in die Felder der Spalte **Mark** ein beliebiges Zeichen ein, um die Quellen auszuwählen, die Sie kopieren möchten.

6 Drücken Sie ENTER.

Prozessor-Quellcode löschen - Delete Processor Source

Mit dieser Funktion können Sie Prozessor-Quellcode-Objekte („Source-Objekte“) löschen.

Um einen Kommandoprozessor zu löschen:

1 Geben Sie im Menü **Administrator Services** den Funktionscode `D` ein.

2 Drücken Sie ENTER.

Das Fenster **Delete Processor Source** wird angezeigt.

3 Geben Sie den Namen des zu löschenden Prozessors, die Library, die Datenbankkennung und die Nummer der Datei an, in der der Prozessor gespeichert ist. Wenn die Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist, müssen Sie auch das entsprechende Passwort und/oder den Chiffrierschlüssel eingeben.

4 Drücken Sie ENTER.

Es erscheint das folgende Fenster zur Auswahl der zu löschenden Prozessor-Quellcode-Objekte:

```
                  Delete Processor Source

Mark  Delete                          Available
----  ----------------------------  ---------
 _    Header ......................    yes
 _    Keywords ....................    yes
 _    Functions ...................    yes
 _    Runtime Action Definitions ..    yes


Name SAGTEST   Library SYSNCP    DBID 10      FNR 32
```

In der Spalte **Available** rechts neben jedem Prozessor-Quellcode-Objekt (Header, Schlüsselwörter, Funktionen, Laufzeit-Aktionsdefinitionen) befindet sich ein Feld, das anzeigt, ob das Quellcode-Objekt existiert. Da die Erstellung von Kommandoprozessoren ein kumulativer Vorgang ist, können Sie ein Quellcode-Objekt nicht löschen, ohne alle Quellcode-Objekte zu löschen, die auf ihm basieren. So können Sie z.B. im obigen Bild das Quellcode-Objekt für die Funktionen nicht löschen, ohne auch das Quellcode-Objekt für die Laufzeit-Aktionsdefinitionen zu löschen.

5 Geben Sie in die Felder der Spalte **Mark** ein beliebiges Zeichen ein, um die als vorhanden (Available = `yes`) angezeigten Quellcode-Objekte auszuwählen, die Sie auswählen möchten.

6 Drücken Sie ENTER.

Quellcode-Objekt, Objektmodul, NCP-Buffer drucken

Zusätzlich zu den Prozessor-Quellcodes können Sie auch das Prozessor-Objektmodul und den NCP-Pufferspeicherinhalt drucken.

Um einen Kommandoprozessorelement zu drucken

1 Geben Sie im Menü **Administrator Services** den Funktionscode `P` ein.

2 Drücken Sie ENTER.

Das Fenster **Print Source/Object/NCP-Buffer** wird angezeigt.

3 Geben Sie den Namen des zu druckenden Prozessors sowie die Library, die Datenbankkennung und die Nummer der Datei an, in der der Prozessor gespeichert ist. Wenn die Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist, müssen Sie auch das entsprechende Passwort und/oder den Chiffrierschlüssel eingeben.

4 Drücken Sie ENTER.

5 Das folgende Fenster wird angezeigt, um die zu druckenden Elemente auszuwählen:

```
                Print Source/Object/NCP-Buffer

      Mark  Print                          Available
      ----  -----------------------------  ---------
       _    Header ......................  yes
       _    Keywords ....................  yes

       _    Functions ...................  yes
       _    Runtime Action Definitions ..  yes

       _    Processor Object ............  yes
            NCP-Buffer ..................  no

            Printer .....................  ________


      Name SAGTEST   Library SYSNCP    DBID 10      FNR 32
```

In der Spalte **Available** rechts neben jedem Prozessor-Quellcode-Objekt (Header, Schlüsselwörter, Funktionen, Laufzeit-Aktionsdefinitionen) befindet sich ein Feld, das anzeigt, ob das Objekt existiert.

Mögliche Eingabewerte für das Feld **Printer** sind die Kennung des logischen Druckers, `VIDEO` oder `SOURCE`. Siehe auch DEFINE PRINTER-Statement in der *Statements*-Dokumentation.

6 Markieren Sie in den entsprechenden **Mark**-Feldern die zu druckenden Objekte mit einem beliebigen Zeichen und geben Sie im Feld **Printer** den Namen des logischen Druckers oder den Wert `VIDEO` oder `SOURCE` ein.

7 Drücken Sie ENTER.

Prozessor entladen - Unload Processor

Um einen Kommandoprozessor zu entladen:

1 Geben Sie im Menü **Administrator Services** den Funktionscode `U` ein.

2 Drücken Sie ENTER.

Das Fenster **Unload Processor to Work File 3** wird angezeigt:

```
                Unload Processor to Work File 3

              Source            Target

Name ........ SAGTEST_
Library ..... SYSNCP__          SYSNCP__
DBID ........ 10___
FNR ......... 32___
Password ....
Cipher Key ..

Report ...... NO_
```

3 Geben Sie in den Feldern Source (Quelle) den Namen des zu entladenden Prozessors, die Library, die Datenbankkennung und die Nummer der Datei ein, in der der Prozessor zu finden ist. Der Standardwert ist der im Menü **Administrator Services** angegebene Prozessor. Geben Sie das entsprechende Passwort und/oder den Chiffrierschlüssel ein, wenn die Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist.

4 Geben Sie im Feld **Report** `YES` ein, wenn ein Bericht erstellt werden soll. Die Voreinstellung ist `NO`. Sie brauchen keine Dateierweiterung zu verwenden. Wenn Sie eine Erweiterung verwenden möchten, müssen Sie die Dateierweiterung `.sag` verwenden.

5 Drücken Sie ENTER.

Wenn der Prozessor entladen wird, werden alle Prozessor-Quellcode-Objekte (Header, Schlüsselwörter, Funktionen, Laufzeit-Aktionsdefinitionen) in die Arbeitsdatei (Work File) 3 geschrieben.

Anmerkung: Verwenden Sie den Object Handler (SYSOBJH Utility), um Kommandoprozessoren von einer Hardware-Plattform auf eine andere zu übertragen.

Prozessor laden - Load Processor

Um einen Kommandoprozessor zu laden:

1 Geben Sie im Menü **Administrator Services** den Funktionscode `L` ein.

2 Drücken Sie ENTER.

Das Fenster **Load Processor from Work File 3** wird angezeigt. Sie können damit Prozessoren aus der Arbeitsdatei (Work File) 3 in eine Natural Library laden:

```
        Load Processor from Work File 3

   Replace existing processors .. N
   Produce load report .......... NO_
```

3 Geben Sie in das Feld **Replace existing processors** (Vorhandene Prozessoren ersetzen) `Y` oder `N` ein (Standard ist `N`), um anzugeben, ob vorhandene Prozessoren mit demselben Namen durch den zu ladenden Prozessor ersetzt werden sollen.

4 Geben Sie in das Feld **Produce load report** (Ladebericht erstellen) `YES` (Standardwert ist `NO`) ein, wenn ein Bericht erstellt werden soll.

5 Drücken Sie ENTER.

Anmerkung: Die Eingaben für den Prozessornamen und die Library, in die der Prozessor geladen werden soll, werden aus der Arbeitsdatei übernommen.

Prozessor-Quellcode einfrieren - Freeze Processor Source

Sie können einen Kommandoprozessor in seinem aktuellen Zustand einfrieren, um zu verhindern, dass er von Benutzern weiter verändert wird.

Um einen Kommandoprozessor einzufrieren:

1 Geben Sie im Menü **Administrator Services** den Funktionscode `F` ein.

2 Drücken Sie ENTER.

Das Fenster **Freeze Processor Source** wird angezeigt.

3 Geben Sie den Namen des Kommandoprozessor, der eingefroren werden soll, sowie die Library, die Datenbankkennung und die Nummer der Datei, in der der Kommandoprozessor gespeichert ist, an. Wenn die Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist, müssen Sie auch das entsprechende Passwort und/oder den Chiffrierschlüssel eingeben.

4 Drücken Sie ENTER.

5 Geben Sie im Folgefenster mit Y oder N an, ob eine Änderung der Prozessor-Quellcode-Objekte erlaubt sein soll oder nicht. Die Voreinstellung ist Y.

6 Drücken Sie ENTER.

Referenzen von Natural Security

Die Funktion **References from Natural Security** ist nur verfügbar, wenn Natural Security in Ihrer Umgebung aktiv ist. Sie dient dazu, Referenzen für funktionale Sicherheit aus Natural Security zu löschen.

Wenn für einen Prozessor in Natural Security funktionale Sicherheit definiert ist, werden automatisch Referenzen erstellt. Diese Referenzen werden nicht in der Systemdatei FSEC, sondern in den Systemdateien FNAT/FUSER zusammen mit den Prozessor-Quellcode-Objekten gespeichert.

Um die Funktion References from Natural Security aufzurufen:

1 Geben Sie im Menü **Administrator Services** den Funktionscode R ein.

2 Drücken Sie ENTER.

Das Fenster **Delete References** erscheint.

3 Geben Sie den Namen des Prozessors, die Library, die Datenbankkennung und die Nummer der Datei an, in der der Prozessor gespeichert ist. Wenn die Datei durch ein Passwort und/oder einen Chiffrierschlüssel geschützt ist, müssen Sie auch das entsprechende Passwort und/oder den Chiffrierschlüssel eingeben.

4 Drücken Sie ENTER.

5 Im Folgefenster können Sie Hauptreferenzen, Funktionsreferenzen und Hilfsreferenzen löschen.

Weitere Informationen zur funktionalen Sicherheit von Kommandoprozessoren finden Sie im Abschnitt *Functional Security* in der *Natural Security*-Dokumentation.

Sitzungsprofil

Ein Sitzungsprofil ist eine Sammlung vom Benutzer definierbaren Voreinstellungen, die festlegen, wie die SYSNCP-Bildschirme aussehen oder wie SYSNCP auf Eingaben reagiert. In einem Sitzungsprofil können Sie z.B. festlegen, welchen Kommandoprozessor Sie als Standard für eine Sitzung verwenden wollen oder welche Farben Sie den Bildschirmattributen zuweisen wollen. In SYSNCP gibt es ein Standard-Sitzungsprofil namens STANDARD, das an alle neuen Benutzer ausgegeben wird. Sie können mehrere verschiedene Sitzungsprofile erstellen und diese nach Bedarf aktivieren.

Administratoren für SYSNCP können auf jedes Sitzungsprofil in SYSNCP zugreifen und es ändern. Andere Benutzer können auf alle Sitzungsprofile zugreifen, aber nur die Sitzungsprofile ändern, die unter ihrer Benutzerkennung erstellt wurden oder die den gleichen Namen wie ihre Benutzerkennung haben.

Um ein Sitzungsprofil zu erstellen oder zu ändern:

- Geben Sie das Kommando `PROFILE` in der Kommandozeile des Menüs **Processor Source Maintenance** ein.

  Der erste von drei Bildschirmen zur Pflege von Sitzungsprofilen wird angezeigt.

Nachfolgend finden Sie Informationen über:

- Name des Sitzungsprofils
- Session-Parameter - Profil 1
- Farbattribute - Profil 2
- Verschiedene Attribute - Profil 3

Name des Sitzungsprofils

Als Standardeinstellung für den Profilnamen wird das Profil `STANDARD` oder der Wert der Systemvariablen `*USER` verwendet.

Wenn Sie ein neues Sitzungsprofil definieren, sind die Parameter/Attribute voreingestellt. Sie können diese Standardwerte nach Bedarf ändern und speichern, indem Sie den neuen Namen eingeben und `PF5` drücken.

Das Feld **Session Profile Name** auf jedem Profil-Bildschirm ist sowohl ein Eingabe- als auch ein Ausgabefeld. Es ist also möglich, von jedem dieser Bildschirme aus ein anderes Profil zu definieren, zu lesen oder zu speichern, indem Sie dessen Namen in das Feld **Profile Name** eingeben und `PF5` bzw. `PF4` drücken.

Session-Parameter - Profil 1

Auf dem ersten Profilverwaltungs-Bildschirm können Sie die folgenden Felder ändern:

Feld	Erläuterung
Apply Terminal Control 1	**Terminalkommandos anwenden** In diese Felder können Sie die Parameter eines `SET CONTROL`-Statements eingeben, das von SYSNCP beim Start ausgegeben wird. Wenn Sie z.B. `Z` in eines der Felder eingeben, gibt SYSNCP das Statement `SET CONTROL 'Z'` aus. Dies entspricht dem Terminalkommando `%Z`, was bewirkt, dass der Arbeitsbereich des Editors gelöscht wird.
Apply Terminal Control 2	

Feld	Erläuterung
Default Processor Name	**Standard-Prozessorname** Der Name des Kommandoprozessors, der standardmäßig für diese Sitzung verwendet werden soll.
Default Processor Library	**Standard-Library** Die Natural Library, die zum Speichern eines Kommandoprozessors verwendet werden soll.
Cancel Reaction	**Reaktion abbrechen** Gibt an, ob eine Warnung ausgegeben werden soll, wenn die angeforderte Änderung nicht abgeschlossen ist und das Kommando `CANCEL` ausgegeben wird. Mögliche Werte: `W` Warnung ausgeben. `B` Änderung rückgängig machen und abbrechen, ohne eine Warnung auszugeben.
Clear Key Allowed	**Benutzung der Löschtaste** Legt fest, ob die Löschtaste erlaubt ist. Mögliche Werte: `N` Löschtaste nicht erlaubt. `Y` Löschtaste ist aktiv und hat die gleiche Wirkung wie das Kommando `CANCEL`.
Default Cursor Position	**Cursor-Standardposition** Legt die Standardposition des Cursors fest. Mögliche Werte: `1` Der Cursor wird standardmäßig im ersten Feld des Bildschirms positioniert. `C` Der Cursor wird standardmäßig in der Kommandozeile positioniert.
Exec/Display Last Command	**Letztes Kommando ausführen/anzeigen** Gibt die Aktion an, die als Ergebnis des Kommandos `LAST` ausgeführt werden soll. Mögliche Werte: E Letztes in der Kommandozeile eingegebenes Kommando ausführen. D Letztes in der Kommandozeile eingegebenes Kommando anzeigen.

Farbattribute - Profil 2

Auf dem zweiten Profilverwaltungs-Bildschirm können Sie verschiedenen Bildschirmattributen Farben zuordnen oder bestehende Farbzuordnungen überschreiben.

Durch die Angabe der folgenden Farbcodes können Sie die folgenden Farben zuordnen:

Code	Farbe
BL	Blau
GR	Grün
NE	Neutral
PI	Rosa
RE	Rot
TU	Türkis
YE	Gelb

Bezüglich Farbzuordnungen zu Bildschirmattributen siehe auch das Terminalkommando `%=` in der *Terminalkommandos*-Dokumentation.

Verschiedene Attribute - Profil 3

Die folgenden Attribute können auf dem dritten Profilverwaltungs-Bildschirm angegeben werden:

Feld	Erläuterung
Message Line Position	**Position der Meldungszeile** Die Zeile, in der die Meldungen angezeigt werden sollen. Es wird der Wert `21` empfohlen. Weitere Informationen siehe Terminalkommando `%M` in der *Terminalkommandos*-Dokumentation.
Text for PF5 Key	**Text für PF5-Taste** Die Funktionstaste PF5 ist für globalen (sitzungsweiten) Gebrauch reserviert. Der Text, der in der PF-Tastenzeile bei PF5 angezeigt werden soll, kann in diesem Feld eingegeben werden.
Command for PF5 Key	**Kommando für PF5-Taste** Die Funktionstaste PF5 ist für den globalen (sitzungsweiten) Gebrauch reserviert. In diesem Feld kann das Kommando eingegeben werden, das ausgeführt werden soll, wenn PF5 gedrückt wird.

Außerdem wird auf dem Bildschirm angezeigt, wann und von welchem Benutzer dieses Profil zuletzt geändert wurde.

XX Object Handler

Der Object Handler dient dazu, Natural- und Nicht-Natural-Objekte zur Verteilung in Natural-Umgebungen zu verarbeiten. Dies geschieht durch Entladen der Objekte in der Quellumgebung in Arbeitsdateien und Laden aus Arbeitsdateien in die Zielumgebung.

Die Dokumentation zum *Object Handler* gliedert sich in die folgenden Teile:

Allgemeine Informationen zum Object Handler	Den Object Handler im Batch- oder Online-Modus aufrufen, Natural Security anwenden.
Funktionen	Benutzung der Object Handler-Menüfunktionen: **Unload** (Entladen), **Load** (Laden), **Restart Load** (Laden erneut starten), **Scan** (Durchsuchen), **View** (Sichten/Anzeigen), **Find** (Suchen) und **Administrate** (Verwalten).
Objekte angeben (Object Specification)	Angabe der Objekte, die mit den Funktionen des Object Handler-Menüs verarbeitet werden sollen: **Natural-Library-Objekte (Natural Library Objects)**, **Natural-Systemfehlermeldungen (Natural System Error Messages)**, **Natural-Kommandoprozessoren (Natural Command Processors)**, **Datendefinitionsmodule (DDMs)**, **Natural-zugehörige Objekte (Natural-related Objects)**, Felddefinitionstabellen (**FDTs**).
Einstellungen (Settings)	Angabe von Options- und Parametereinstellungen für Object Handler-Menüfunktionen.
Arbeitspläne (Workplans)	Verwendung von Standardverfahren zur Ausführung von Object Handler-Funktionen..
Angabe von Name, Datum und Zeit	Konventionen für die Angabe von Namen, Datum, Zeit und Bereichen.
Arbeitsdateien (Work Files)	Vom Object Handler verwendete Arbeitsdateien.
Direktkommandos (Direct Commands)	Verwendung von Direktkommandos zur Ausführung von Object Handler-Funktionen.
Condition Codes und User Exit-Routinen	Condition Codes und User Exit-Routinen, die im Batch-Modus zur Verfügung stehen..
Werkzeuge (Tools)	Anzeigen von Statusinformationen und Einstellen von Trace- und Report-Optionen. Profileinstellungen.

Profileinstellungen (Profile Settings)	Einrichten eines Profils zur Definition individueller Standardeinstellungen und Standardvorgehensweisen.
Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler	Migration von den früher verwendeten Dienstprogrammen NATUNLD/NATLOAD und SYSTRANS zum Object Handler.

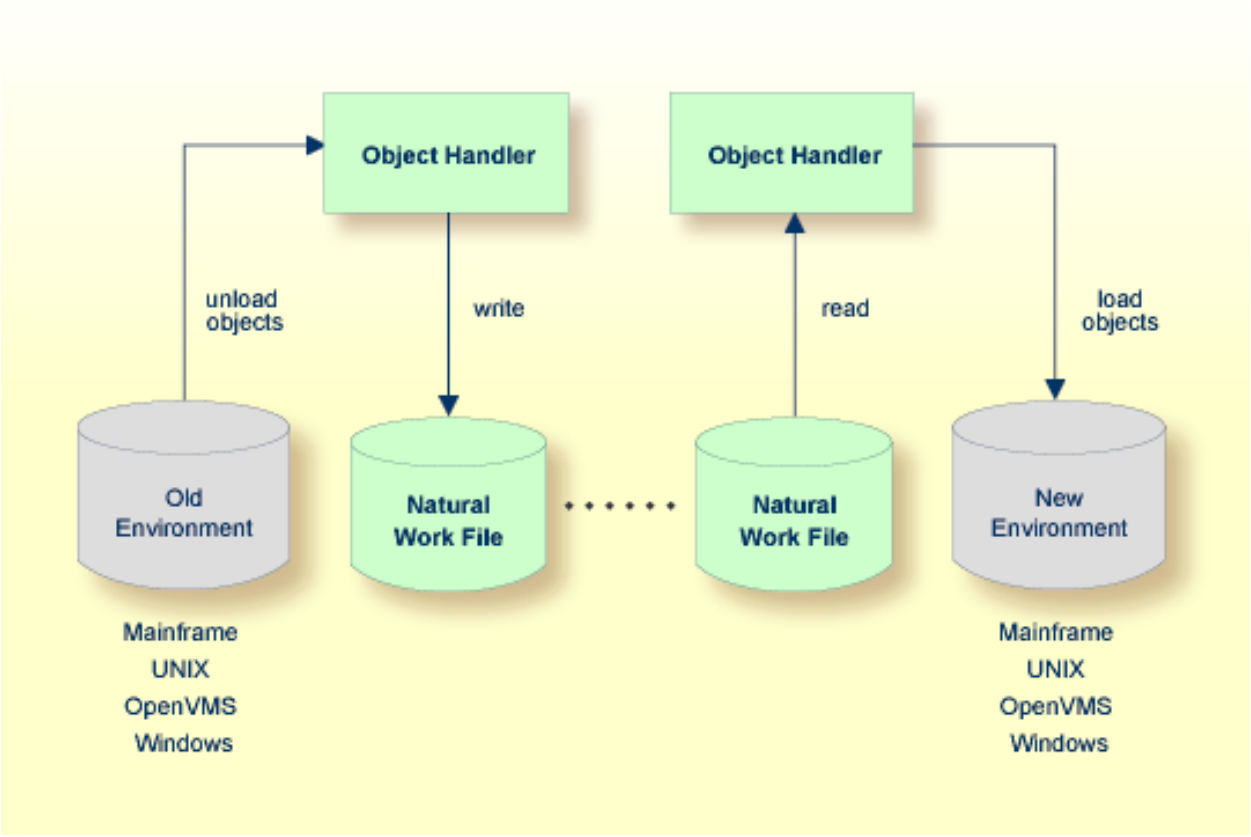
84 Allgemeine Informationen zum Object Handler

- Grundsätzliches zur Objektübertragung 858
- Object Handler aufrufen 860
- Batch- oder Direktkommandoaufrufe 862
- Object Handler-Kommandos aus einem Natural-Programm heraus absetzen 863
- Text-Objekte für Report-, Restart- und Trace-Daten 863
- Natural Security 864
- Standard-PF-Tasten 864

Der Object Handler besteht aus der **SYSOBJH Utility** (ein Dienstprogramm das sich in der Natural System Library SYSOBJH befindet) und der **Direktkommandoschnittstelle**. Darüber hinaus steht die **Anwendungsprogrammierschnittstelle OBJHAPI** zur Verfügung, über die Sie die Funktionen des Object Handlers aus einem Natural-Programm heraus ausführen können.

Grundsätzliches zur Objektübertragung

Die nachstehende Grafik veranschaulicht, wie der Object Handler Objekte überträgt, indem er sie aus der Quellumgebung in Arbeitsdateien entlädt und sie aus den Arbeitsdateien in die Zielumgebung lädt. Falls erforderlich, kann ein Anwendungsprotokoll wie z.B. FTP für die Übertragung von Arbeitsdateien von der Quell- zur Zielumgebung verwendet werden.



In diesem Abschnitt werden die folgenden Themen behandelt:

- Übertragungsumgebung und Dateisicherheit
- Vom Object Handler verarbeitete Objekte

- Formatierungsoptionen

Übertragungsumgebung und Dateisicherheit

Eine alte oder neue Umgebung ist eine FNAT-, FUSER- oder FDIC-Systemdatei, die in einer Adabas-Datenbank oder einem VSAM-Dateisystem auf einem Großrechner oder im Dateisystem auf einer Linux- oder Windows-Plattform enthalten ist. Natural-Objekte in der FNAT- oder FUSER-Systemdatei können, wie im folgenden Abschnitt beschrieben, in Libraries enthalten sein.

Die Dateisicherheit (File Security, d. h. Passwörter und Chiffrierschlüssel) bezieht sich auf die Sicherheit, die für eine Systemdatei in einer Adabas- oder VSAM-Umgebung definiert wurde. Wenn für eine Systemdatei Dateisicherheit definiert wurde, müssen Sie ein Passwort, einen Chiffrierschlüssel und/oder einen VSAM-Namen für die Quell- und/oder Zielsystemdatei angeben, bevor Sie eine Object Handler-Funktion ausführen können. Andernfalls wird Adabas oder VSAM eine entsprechende Fehlermeldung ausgeben. Für die Standard-Systemdateien, die der Natural-Sitzung beim Start des Object Handlers zugewiesen werden, brauchen Sie keine Sicherheitsinformationen anzugeben.

Vom Object Handler verarbeitete Objekte

Der Object Handler überträgt Natural-Quellcodeobjekte (auch gespeicherte Objekte (Saved Objects) genannt) und katalogisierte Objekte (Cataloged Objects), die in Natural-Libraries enthalten sind, Natural-Fehlermeldungen, Natural-Kommandoprozessor-Quellcodes, Natural-Datendefinitionsmodule (DDMs), Natural-zugehörige Objekte (Natural-Related Objects) und Adabas Field Definition Tables (FDTs).

Formatierungsoptionen

Sie können Daten in Binärformat oder in Textformat übertragen, je nach Quell- und Zielumgebung, in der die Objekte verarbeitet werden.

Binärformat kann für Quellcodeobjekte und katalogisierte Objekte (einschließlich DDMs), Fehlermeldungen, Natural-Kommandoprozessor-Quellcodes, Natural-zugehörige Objekte (Natural Related Objects) und Adabas FDTs verwendet werden.

Textformat gilt für Quellcodeobjekte, Natural-Kommandoprozessor-Quellcodes, Fehlermeldungen, DDMs und Adabas-FDTs. Zwischen Großrechner- und Linux/Windows-Plattformen können Sie nur Textdaten übertragen. Binärdaten können Sie zwischen identischen Plattformen übertragen.

Object Handler aufrufen

Um den Object Handler aufzurufen, können Sie entweder Menüfunktionen oder Direktkommandos verwenden.

Um den Object Handler online aus einer beliebigen Natural Library aufzurufen:

1 Geben Sie das folgende Systemkommando ein (online aus einer beliebigen Natural Library):

```
SYSOBJH
```

Je nach Eingabemodus wird entweder das Object Handler-Hauptmenü (**Main Menu**) oder das Object Handler-Kompaktmenü (**Compact Menu**) angezeigt. Siehe *Kompaktmodus*. Sie können den Eingabemodus entweder mit dem Kommando `SET INPUT` (beschrieben in *Direktkommandos*) oder durch Ändern des Parameters `INPUT-MODE` (beschrieben in *Profileinstellungen*) ändern.

Das Hauptmenü des Object Handlers bietet die folgenden Optionen:

- **Unload** (Entladen)
- **Load** (Laden)
- **Scan** (Durchsuchen)
- **View** (Sichten/Anzeigen)
- **Administrate** (Object Handler-Verwaltung)
- **Advanced user** (Expertenmodus)

```
10:20:34              ***** Natural Object Handler *****           2023-10-17
User XYZ                          - Main Menu -

     Select the desired function:

       _  Unload objects or a whole application from your Natural environment

       _  Load objects or an application into your Natural environment

       _  Scan work file contents

       _  View objects in the Natural environment

       _  Administrate the Object Handler environment,
                      process Workplans and direct commands

     Mark this field to avoid Object Handler wizards:

       _  Advanced user

Defaults from SYSOBJH profile (OBJHPROF) read.
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  Unld  Load  Scan  View  Admin       Cmds        Canc  ↩
 ↩
```

Beschreibungen dieser Funktionen und Informationen, wie Sie diese Funktionen im Expertenmodus oder mit Hilfe von Assistenten (Wizards) verwenden können, finden Sie im Abschnitt *Funktionen*

2 Wählen Sie eine Funktion aus, indem Sie eine der folgenden Vorgehensweisen wählen:

Geben Sie ein beliebiges Zeichen in das Eingabefeld neben dem Element ein, das der gewünschten Funktion entspricht.

Oder:

Wählen Sie die PF-Taste, die mit der gewünschten Funktion belegt ist.

Oder:

Geben Sie in der Kommandozeile das Object Handler-Kommando ein, das der gewünschten Funktion entspricht. Informationen zu den verfügbaren Kommandos finden Sie im Abschnitt *Direktkommandos*.

Um den Object Handler im Batch-Modus oder Direktkommmandomodus online aufzurufen:

- Geben Sie das Systemkommando `SYSOBJH` gefolgt von einem Direktkommando ein. Einzelheiten siehe *Batch- oder Direktkommandoaufrufe* und *Direktkommandos*.

Nach der Ausführung eines Direktkommandos können Sie entweder ein weiteres Direktkommando absetzen oder einen Punkt (.) eingeben, um den Object Handler zu beenden.

Batch- oder Direktkommandoaufrufe

Dem Object Handler können online oder im Batch-Modus mehrere Kommandos erteilt werden. Das letzte Kommando in der Kommandofolge muss ein Punkt (.) oder das Schlüsselwort `STOP`, `END`, `QUIT` oder `FIN` sein, wobei `FIN` die Natural-Sitzung beendet.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Batch-Modus
- Online-Modus

Batch-Modus

Die an den Object Handler abgesetzten Kommandos werden von der Standardeingabe gelesen. Jedes Kommando können Sie in maximal 20 Kommandoteile/Strings aufteilen, indem nach jedem Schlüsselwort oder Schlüsselwortwert Eingabebegrenzer (Session-Parameter `ID`) eingegeben. Jeder Kommandoteil/String darf maximal 248 Bytes lang sein.

Wenn das Kommando länger als eine Zeile ist, geben Sie am Ende jeder Zeile außer der letzten, die zum Kommando gehört, das mit dem Session-Parameter `CF` definierte Zeichen ein. Standardmäßig ist es das Prozentzeichen (%). Dies zeigt die Fortsetzung in der nächsten Zeile an. Dies ist jedoch nur möglich, wenn Sie das Kommando `SYSOBJH` selbst in einer Zeile angeben. Das heißt, Sie können `CF` nicht verwenden, wenn Sie SYSOBJH in der gleichen Zeile eingeben, in der ein mehrzeiliges Kommando beginnt.

Beispiel:

In diesem Beispiel wird davon ausgegangen, dass der Session-Parameter `ID` auf Komma (,) gesetzt ist.

```
SYSOBJH
UNLOAD PROG* LIBRARY MYLIB1, OBJTYPE N,%
WITH NEWLIBRARY MYLIB2%
WHERE REPORT TRANSFER
STOP
```

Verwandte Themen:

- *Direktkommandos*
- *Natural im Batch-Modus - Operations*-Dokumentation

Online-Modus

Das an den Object Handler in der Kommandozeile abgesetzte Kommando kann aus bis zu 20 Kommandoteilen bestehen.

Beispiel:

```
SYSOBJH UNLOAD * LIB EXAMPLE WHERE TRANSFER
```

Object Handler-Kommandos aus einem Natural-Programm heraus absetzen

Sie können aus einem Natural-Programm heraus Kommandos an den Object Handler absetzen. Dazu dient die Anwendungsprogrammierschnittstelle OBJHAPI, die als Subprogramm in der Natural System Library SYSOBJH enthalten ist. Die erforderlichen Parameter und Beispiele finden Sie in dem Natural-Programm `DOC-API`, das in der Library SYSOBJH verfügbar ist.

Text-Objekte für Report-, Restart- und Trace-Daten

Die vom Object Handler erzeugten Report-, Restart- und Trace-Daten werden als Natural Text Member (Natural-Objekte vom Typ Text) in der Workplan Library gespeichert. Der Object Handler generiert Namen für Textobjekte, die nicht explizit im Fenster Options angegeben wurden. Die generierten Namen sind eine Kombination aus dem Wochentag und der Zeit. Beispiel: Ein Objekt mit dem Namen 21415568 wurde am Dienstag (dem zweiten Tag der Woche) um 14:15:56,8 erstellt.

Die Workplan Library, in der die Textobjekte gespeichert werden, können Sie mit der Profiloption `Workplan Library` des Object Handlers festlegen (siehe ***Profileinstellungen***). Wenn Ihre Natural-Sitzung eine schreibgeschützte FNAT- oder FUSER-Systemdatei verwendet (wie mit dem Natural-Profilparameter `ROSY=ON` angegeben), werden die Textobjekte in einer Scratch-Pad-Datei (siehe *Operations*-Dokumentation) unter einer der folgenden Bedingungen gespeichert:

- Die für die Workplan Library angegebene Datenbankkennung und Dateinummer ist identisch mit der aktuellen FNAT- oder FUSER-Systemdatei.
- Es werden keine Datenbankkennung (DBID) und keine Dateinummer (FNR) für die Workplan Library angegeben.

Natural Security

Bei Verwendung des Object Handlers unter Natural Security muss ein entsprechendes **Utility Profile** in Natural Security definiert werden. Zumindest muss ein Standardprofil definiert sein. Informationen zu Utility-Profilen finden Sie im Abschnitt *Protecting Utilities* in der *Natural Security*-Dokumentation.

Wenn Natural Security installiert ist, prüft der Object Handler die SYSOBJH-Utility-Profile in Natural Security, um herauszufinden, ob die angeforderte Funktion und die Parametereinstellungen erlaubt sind.

Sollte während der Ladefunktion ein Natural Security-Fehler auftreten, gilt Folgendes:

- Wenn die Option **Write report** gesetzt ist, wird im Online-Modus die Fehlermeldung in die Reportdatei geschrieben und die Verarbeitung des aktuellen Ladekommandos fortgesetzt.
- Wenn die Option **Write report** gesetzt ist, wird im Batch-Modus die Fehlermeldung in die Reportdatei geschrieben und der Object Handler bricht ab, nachdem das Ladekommando, bei dem der Fehler aufgetreten ist, abgearbeitet wurde.
- Ist die Option **Write report** *nicht* gesetzt, wird eine Fehlermeldung ausgegeben und das Ladekommando abgebrochen.

Standard-PF-Tasten

Die folgenden PF-Tasten sind bei allen Vollbildschirmmasken vorhanden:

PF-Taste	Erläuterung
PF1	Aufruf der Hilfefunktion für das Feld, in dem sich der Cursor befindet.
PF3	Beenden der aktuellen Bildschirmanzeige und Rückkehr zum vorherigen Bildschirm.
PF6	Sprung an den Anfang einer Liste.
PF7	In einer Liste um eine Seite nach oben blättern. Auf Assistentenbildschirmen: einen Bildschirm/Schritt zurück.
PF8	In einer Liste um eine Seite nach unten blättern. Auf Assistentenbildschirmen: weiter zum nächsten Bildschirm/Schritt.
PF9	Sprung ans Ende einer Liste.
PF10	Aufrufen des Menüs **Commands** zum Auswählen von Kommandos für die Navigation und zum Zuweisen von speziellen Einstellungen. Siehe auch ***Kommandos für die Navigation und für spezielle Funktionen*** in *Direktkommandos*.
PF12	Abbruch der aktuellen Funktion.

PF-Taste	Erläuterung
PF20	Auflistung aller aktiven Programme des Object Handlers. Diese Liste kann beim Melden technischer Probleme an den Support hilfreiche Informationen liefern.

85 Funktionen

In diesem Kapitel werden die wichtigsten Funktionen des Object Handler beschrieben.

Sie können die Assistenten (Wizards) des Object Handler nutzen. Diese führen Sie durch die Schritte, die zur Ausführung der Funktionen Entladen (Unload), Laden (Load) und Durchsuchen (Scan) erforderlich sind. Die Assistenten sind standardmäßig aktiviert, sofern Sie die Standardeinstellungen nicht ändern. Wenn Sie stattdessen den Entlade-, Lade- oder Scan-Modus für erfahrene Benutzer bevorzugen, müssen Sie das Feld neben Advanced user im Hauptmenü markieren. Alternativ können Sie das interne Kommando `SET INPUT-MODE A` benutzen. Zusätzlich können Sie als erfahrener Benutzer den kompakten Eingabemodus für die Entlade-, Lade- oder Scan-Funktionen benutzen, indem Sie das interne Kommando `SET INPUT-MODE C` eingeben. Sie können auch den Standard-Eingabemodus (Wizard, Advanced oder Compact) einstellen, indem Sie die entsprechende Object Handler-Profiloption **Input-Mode** benutzen. Siehe auch *Profileinstellungen*.

Tipp: Sie können Standardverfahren erstellen, um wiederkehrende Einstellungen und Objektspezifikationen zu definieren, die die Verarbeitung der Entlade-, Lade- oder Scan-Funktion automatisieren, siehe *Arbeitspläne (Workplans)*.

In diesem Kapitel werden die folgenden Themen behandelt:

Assistenten (Wizards)
Advanced User
Kompaktmodus
Restart Load
View
Find
Administration
Arbeitsplan-Bibliothek ändern (Change Workplan Library)
Systemdatei auswählen
Bibliothek auswählen

Arbeitspläne auflisten und auswählen

Systemfehlermeldungen auswählen

Select Objects



Anmerkungen:

1. Das Thema *Arbeitsplan-Bibliothek ändern (Change Workplan Library)* wird im Kapitel *Administration* behandelt.
2. Das Thema *Arbeitspläne auflisten und auswählen* wird im Abschnitt *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* im Kapitel *Administration* behandelt.

86 Assistenten (Wizards)

- Schritt 1 - Vorgang starten 870
- Schritt 2 - Objekte in/aus Arbeitsdateien entladen/laden/durchsuchen 871
- Step 3 - Parameter festlegen 873
- Schritt 4 - Objekte auswählen 874
- Schritt 5 - Verarbeitung ausführen 876
- Schritt 6 - Verarbeitung fortsetzen 876

Der Object Handler stellt Assistenten (Wizards) zur Verfügung, die die Verarbeitungsreihenfolge für Folgendes festlegen:

- Entladen von Daten aus der Natural-Systemumgebung in Natural-Arbeitsdateien.
- Laden von Daten aus Arbeitsdateien in die Natural-Systemumgebung.
- Durchsuchen (Scannen) des Inhalts von Natural-Arbeitsdateien.

Um die Assistenten zu aktivieren:

- Markieren Sie im Hauptmenü (**Main Menu**) bei Bedarf das Feld **Advanced user** (das Feld ist standardmäßig nicht markiert).

In den Bildschirmen der Assistenten sind die Tasten PF8 und PF7 belegt, so dass Sie zwischen den Bildschirmen (Schritten) navigieren können. Mit PF12 können Sie Verarbeitungsschritte abbrechen.

Die in diesem Abschnitt beschriebenen Schritte behandeln die Verarbeitungsreihenfolge, die mit dem Entlade-, Lade- oder Scan-Assistenten ausgeführt wird.

Schritt 1 - Vorgang starten

Um den Entlade-, Lade- oder Scan-Vorgang zu starten:

1 Wählen Sie im Hauptmenü (**Main Menu**) die Option **Unload**, **Load** oder **Scan**, indem Sie ein beliebiges Zeichen neben der gewünschten Funktion eingeben oder die entsprechende PF-Taste drücken.

Es erscheint der erste **Wizard**-Bildschirm des Assistenten mit den folgenden Optionen:

- **Unload/Load/Scan objects into/from Natural work file(s)**.

  Entladen/Laden/Durchsuchen von Objekten in/aus Natural-Arbeitsdatei(en).
- **Start Object Handler command procedure**.

  Kommandoprozedur Object Handler starten.

2 Wenn Sie Objekte in eine Arbeitsdatei entladen, aus einer Arbeitsdatei laden oder in einer Arbeitsdatei durchsuchen möchten, fahren Sie mit *Schritt 2 - Objekte in/aus Arbeitsdateien entladen/laden/durchsuchen* fort.

Oder:

Wenn Sie eine Kommandoprozedur zum Entladen, Laden oder Durchsuchen (Scannen) von Objekten verwenden möchten, wählen Sie die Option **Start Object Handler command procedure** und gehen Sie wie folgt vor:

1. Wählen Sie im ersten **Wizard**-Bildschirm die Option **Start Object Handler command procedure**. Der Bildschirm **Procedure** erscheint.
2. Geben Sie im Feld **Name** den Namen eines Arbeitsplans vom Typ PROCEDURE ein, indem Sie eine der folgenden Optionen verwenden:
   - Geben Sie den Namen eines Arbeitsplans vom Typ PROCEDURE (siehe auch *Arbeitspläne (Workplans)*) ein, der für die Transaktion verwendet werden soll.
   - Wählen Sie **Select Workplan** oder wählen Sie PF5, um eine Liste der verfügbaren Arbeitspläne vom Typ PROCEDURE anzuzeigen.

     Geben Sie in der Zeile neben dem Arbeitsplan, den Sie auswählen möchten, das Kommando `S` oder `SE` ein.

     Wählen Sie ENTER, um das Kommando auszuführen und das Feld **Name** im Bildschirm **Procedure** zu füllen.
3. Wählen Sie **List Workplan** oder wählen Sie PF4, wenn Sie den angegebenen Arbeitsplan anzeigen wollen.

   Siehe auch *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* im Kapitel *Administration*.

3 Wählen Sie ENTER, um fortzufahren.

4 Fahren Sie fort mit *Schritt 5 - Verarbeitung ausführen*.

Schritt 2 - Objekte in/aus Arbeitsdateien entladen/laden/durchsuchen

Um Objekte in/aus Arbeitsdateien zu entladen/laden/durchsuchen:

1 Wählen Sie im ersten **Wizard**-Bildschirm die Option **Unload/Load/Scan objects into/from Natural work file(s)**.

2 Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren. Der Bildschirm **Options** des Assistenten erscheint mit den folgenden Feldern, Kommandos und alternativen PF-Tasten:

Feld	PF-Taste	Erläuterung
Transfer format		Übertragungsformat: Nur gültig, wenn **Use default options** (dies ist die Standardeinstellung) gewählt wurde. Falls ausgewählt, werden die zu verarbeitenden Daten im Transfer-Format in/aus der Arbeitsdatei geschrieben. Siehe auch ***Work File Format*** in *Arbeitsdateien (Work Files)*. Funktion **Unload**: Die zu entladenden Daten werden im Transfer-Format in die Arbeitsdatei geschrieben. Beachten Sie, dass Sie, wenn Sie die Einstellung dieses Feldes für ein späteres Entladen ändern wollen, zum Hauptmenü (**Main Menu**)zurückkehren oder das Kommando `GO UNLOAD END` eingeben müssen (siehe Kommandos für die Navigation und spezielle Funktionen in *Direktkommandos*) und die Entladefunktion erneut starten müssen. Funktionen **Load** und **Scan**: Die zu ladenden oder zu durchsuchenden Daten werden im Transfer-Format erwartet.
Unicode work file		Unicode-Arbeitsdatei: Gilt nur für die Entladefunktion und wenn **Transfer format** gewählt wurde. Wenn diese Option gewählt ist, werden alle Objekt-Quellcodes in Unicode/UTF-8 (Universal Transformation Format, 8-Bit-Form) konvertiert, bevor sie in die Arbeitsdatei geschrieben werden.
Use PC File		PC-Datei verwenden: Gilt nur, wenn Entire Connection installiert ist. Wenn diese Option gewählt ist, werden die zu verarbeitenden Daten aus einer Entire Connection-Arbeitsdatei gelesen oder in diese geschrieben.
PC File		PC-Datei: Gilt nur, wenn **Use PC File** ausgewählt wurde. Der Name des Pfads und der zu verwendenden Entire Connection-Arbeitsdatei.
Use default options		Standardoptionen verwenden: Die Standardoptionen sind aktiviert (dies ist die Standardeinstellung). Die verfügbaren Optionen finden Sie unter ***Profileinstellungen*** und ***Zusätzliche Optionen festlegen*** in *Einstellungen (Settings)*.
Set additional options	PF4	Zusätzliche Optionen festlegen: Nur gültig, wenn **Use default options** ausgewählt wurde.

Feld	PF-Taste	Erläuterung
		Ruft den Bildschirm **Options** des Assistenten auf. Dort können Sie die Standardeinstellungen ändern und zusätzliche Optionen für die Verarbeitungsreihenfolge eingeben. Siehe *Zusätzliche Optionen festlegen* in *Settings*.
Use Option Workplan		Arbeitsplan vom Typ OPTION verwenden: Wenn ausgewählt, wird ein Arbeitsplan vom Typ OPTION verwendet. Siehe *Arbeitspläne (Workplans)*.
Name		Nur gültig, wenn **Use Option Workplan** ausgewählt wurde. Der Name eines zu verwendenden Arbeitsplans vom Typ OPTION.
List Option Workplan	PF6	Nur gültig, wenn **Use Option Workplan** ausgewählt wurde. Zeigt den Inhalt des im Feld Name angegebenen Arbeitsplans an.
Select Option Workplan	PF5	Nur gültig, wenn **Use Option Workplan** ausgewählt wurde. Zeigt eine Auswahlliste der verfügbaren Arbeitspläne vom Typ OPTION an. Siehe auch *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* in *Administration*).

3 Wählen Sie eine der angebotenen Optionen aus und füllen Sie (falls erforderlich) die Felder aus, die für die Verarbeitungsfolge verwendet werden sollen.

4 Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren.

Der Bildschirm **Parameters** des Assistenten erscheint.

Step 3 - Parameter festlegen

Um Parameter für den Verarbeitungsvorgang festzulegen:

1 Wählen Sie im Bildschirm **Parameters** eine der folgenden Optionen und füllen Sie (falls nötig) die für den Verarbeitungsablauf verwendeten Felder aus:

Feld	PF-Taste	Erläuterung
Do not use parameters		Keine Parameter verwenden: Wenn markiert (Standardeinstellung), werden keine Parameter festgelegt.
Use global parameters		Wenn markiert, werden globale Parameter verwendet Siehe *Globale Parameter festlegen* in *Einstellungen (Settings)*.

Feld	PF-Taste	Erläuterung
Set global parameters	PF4	Globale Parameter festlegen: Nur gültig, wenn **Use global parameters** ausgewählt wurde. Wenn markiert, wird der Bildschirm **Parameters** aufgerufen. Beschreibung der Schlüsselwörter und gültige Werte siehe ***Globale Parameter festlegen*** in *Einstellungen (Settings)*.und die Beschreibung der Klausel `parameter-setting` in *Direktkommandos*.
Use Parameter Workplan		Arbeitsplan vom Typ PARAMETER verwenden: Wenn markiert, wird ein Arbeitsplan des Typs PARAMETER verwendet. Siehe ***Arbeitspläne (Workplans)***.
Name		Nur gültig, wenn das Feld **Use Parameter Workplan** markiert ist. Der Name eines Arbeitsplans vom Typ PARAMETER, der verwendet werden soll.
List Parameter Workplan	PF6	Nur gültig, wenn das Feld **Use Parameter Workplan** markiert ist. Falls ausgewählt, wird der Inhalt des im Feld **Name** angegebenen Workplans angezeigt.
Select Parameter Workplan	PF5	Nur gültig, wenn das Feld **Use Parameter Workplan** markiert ist. Wenn markiert, wird eine Auswahlliste der verfügbaren Arbeitspläne vom Typ PARAMETER angezeigt. Siehe ***Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek*** in *Administration*).

2 Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren.

Der Bildschirm **Select Unload/Load/Scan Type** erscheint.

Schritt 4 - Objekte auswählen

Um den Objekttyp auszuwählen, den Sie verarbeiten möchten:

1 Wählen Sie im Bildschirm **Select Unload/Load/Scan Type** eine der drei unten beschriebenen Optionen. Beachten Sie, dass die erste Option nur für die Lade- und Scan-Funktionen gilt.

Die Schlüsselwörter und gültigen Werte, die für jeden Objekttyp gelten, finden Sie in den entsprechenden Erläuterungen im Abschnitt ***Objekte angeben***.

1. Wählen Sie **Load/Scan all objects**, um alle Objekte aus der Arbeitsdatei zu verarbeiten.

2. Wählen Sie einen bestimmten Objekttyp aus:
   - **Natural-Library-Objekte**
   - **Natural-Systemfehlermeldungen**
   - **Natural-Kommandoprozessor-Quellcodes**
   - **Natural-zugehörige Objekte (Natural Related Objects)**
   - **DDMs**
   - **FDTs**

   Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren.

   Je nach ausgewähltem Objekttyp erscheint ein Bildschirm, auf dem Sie Auswahlkriterien für die zu verarbeitenden Objekte angeben können.

   Geben Sie die Auswahlkriterien ein und wählen Sie **Details** (falls vorhanden) für weitere Objektangaben, falls erforderlich. Informationen zu **Details** finden Sie in der entsprechenden Erläuterung im Abschnitt ***Objekte angeben***.
3. Wählen Sie **Use Selection or List Workplan**, um einen Arbeitsplan vom Typ SELECTION oder LIST zu verwenden. Siehe auch ***Arbeitspläne (Workplans)***.

   Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren.

   Der Bildschirm **Selection or List** erscheint. Geben Sie im Feld **Name** den Namen eines Arbeitsplans vom Typ SELECTION oder LIST ein, indem Sie eine der folgenden Möglichkeiten nutzen:

   Geben Sie den Namen eines Arbeitsplans ein.

   Oder:

   Wählen Sie Select Workplan oder PF5 (SelWP), um eine Liste aller verfügbaren Arbeitspläne anzuzeigen. Geben Sie in der Zeile neben dem Arbeitsplan, den Sie auswählen möchten, entweder das Kommando `S` oder `SE` ein.

   Wählen Sie ENTER, um das Kommando auszuführen und das Feld **Name** im Bildschirm **Select** oder **List** auszufüllen. Siehe auch ***Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek*** in Arbeitsplan-Bibliothek im Kapitel *Administration*.

   Wählen Sie **List Workplan** oder PF4 (Li-WP), wenn Sie den Inhalt des im Feld **Name** eingegebenen Arbeitsplans anzeigen möchten.

2 Wählen Sie ENTER oder wählen Sie PF8 (Next), um fortzufahren.

Der Assistent zeigt das aus den Eingabedaten generierte Verarbeitungskommando an.

Sie können das angezeigte Kommando als Arbeitsplan vom Typ PROCEDURE (siehe auch *Arbeitspläne (Workplans)*) speichern, indem Sie das Kommando SAVE eingeben oder PF5 (Save) wählen.

Schritt 5 - Verarbeitung ausführen

Um den Verarbeitungsvorgang auszuführen:

1 Wählen Sie im Bildschirm zur Befehlsausführung ENTER oder PF8 (Next), um die Einstellungen zu bestätigen und die angegebenen Objekte zu verarbeiten.

Falls erforderlich, wählen Sie PF7 (Back) und ändern Sie die Verarbeitungseinstellungen, bevor Sie die Kommandoausführung bestätigen.

Der Object Handler führt die Funktion aus und zeigt eine Bestätigungsmeldung an.

2 Wählen Sie ENTER, um fortzufahren.

Es erscheint ein Report-Bildschirm mit einer Liste der verarbeiteten Objekte.

3 Wählen Sie PF3 (Exit), um den Report zu verlassen, oder wählen Sie PF12 (Canc), um die Funktion zu beenden.

Es erscheint ein Fenster, in dem Sie wählen können, ob Sie die Verarbeitung der Daten fortsetzen wollen.

4 Wählen Sie **No** und dann ENTER, um die Funktion zu beenden.

Oder:

Wählen Sie PF12, um die Funktion abzubrechen.

Das Hauptmenü (**Main Menu**) erscheint.

Schritt 6 - Verarbeitung fortsetzen

Um die Verarbeitung fortzusetzen:

1 Wählen Sie auf dem Report-Bildschirm PF3 (Exit).

Es erscheint ein Fenster, in dem Sie wählen können, ob Sie mit dem nächsten Verarbeitungsschritt fortfahren möchten.

2 Wählen Sie **Yes**.

Es erscheint ein Bildschirm mit der Möglichkeit, vorherige Einstellungen wiederzuverwenden oder zu ändern.

87 Expertenmodus (Advanced User)

- Objekte verarbeiten 880

Im Folgenden wird beschrieben, wie Sie den Expertenmodus aufrufen und die Funktionen zum Entladen, Laden und Durchsuchen ausführen.

Anmerkung: Dieser Parameter wird nicht mehr verwendet und wird nur noch aus Kompatibilitätsgründen beibehalten.

Objekte verarbeiten

Um Objekte im Expertenmodus zu bearbeiten:

1 Markieren Sie im Hauptmenü (**Main Menu**) das Feld **Advanced user** (erfahrener Benutzer), um den Expertenmodus einzuschalten, und wählen Sie die gewünschte Funktion aus:

Unload, **Load** oder **Scan**.

2 Wählen Sie ENTER, um fortzufahren.

Es erscheint der Bildschirm **Unload Settings** bzw. **Load Settings** bzw. **Scan Settings**.

In den Abschnitten **Options** und **Parameters** können Sie je nach gewählter Funktion unterschiedliche Optionen markieren.

3 Markieren Sie gewünschten Optionen bzw. geben Sie Parameter ein. Weitere Informationen siehe Abschnitt *Einstellungen (Settings)*.

4 Wählen Sie ENTER, um fortzufahren.

Es erscheint der Bildschirm **Select Unload Type** bzw. **Select Load Type** bzw. **Select Scan Type**.

5 Wählen Sie die Objekte aus, die Sie verarbeiten wollen. Siehe *Objekte angeben*.

6 Wählen Sie **Details**, um zusätzliche Auswahlkriterien anzugeben. Siehe entsprechende Abschnitte in *Objekte angeben*.

7 Wählen Sie ENTER, um fortzufahren.

- Wenn der Parameter `Display-Cmd-in-Advanced-Mode` im Object Handler-Profil auf `N` (No) gesetzt ist (Standardeinstellung), oder wenn kein solches Profil existiert, wird das aus den Eingabedaten generierte Kommando sofort ausgeführt, nachdem Sie die Auswahldaten angegeben haben. Siehe auch *Profileinstellungen*.

  Es erscheint der Bildschirm **Display Unload Report** bzw. **Display Load Report** bzw. **Display Scan Report** mit einer Liste der bearbeiteten Objekte, wenn das Feld **Write report** markiert wurde (Standardeinstellung). Siehe dazu auch *Arbeitsdatei- und Report-Optionen* im Abschnitt *Einstellungen (Settings)*.

- Wenn der Parameter `Display-Cmd-in-Advanced-Mode` im Object Handler-Profil auf `Y` (Yes) gesetzt ist (siehe ***Profileinstellungen***), oder wenn das Kommando `SET ADVANCEDCMD ON` (siehe *Kommandos für die Navigation und spezielle Funktionen*) früher ausgeführt wurde, erscheint ein Bildschirm, der das aus den Eingabedaten generierte Kommando anzeigt.

  Sie können das angezeigte Kommando als Arbeitsplan vom Typ PROCEDURE (siehe auch ***Arbeitspläne (Workplans)***) speichern, indem Sie das Kommando `SAVE` eingeben oder PF5 (Save) wählen.

  Wählen Sie `ENTER`, um die Kommandoausführung zu bestätigen, oder wählen Sie PF3 (Exit), um die Verarbeitungseinstellungen zu ändern, bevor Sie die Kommandoausführung bestätigen.

  Der Bildschirm **Display Unload Report** bzw. **Display Load Report** bzw. **Display Scan Report** erscheint mit einer Liste der verarbeiteten Objekte, wenn das Feld **Write report** markiert wurde (Standardeinstellung). Siehe auch ***Arbeitsdatei- und Report-Optionen*** im Abschnitt *Einstellungen (Settings)*.

88 Kompaktmodus (Compact Mode)

- Wie Sie den Kompaktmodus wählen 884
- Wie Anweisungen im Kompaktmodus verarbeitet werden 884

Wenn Sie als erfahrener Benutzer (Advanced User) den Kompaktmodus (**Compact Mode**) wählen, benötigen Sie nur zwei Schritte, um Objektparameter für die Ausführung von Entlade-, Lade- oder Durchsuchungsfunktionen festzulegen.

Indem Sie ein entsprechendes Untermenü auswählen, können Sie auch andere Objektparameter angeben oder ändern. Außerdem können Sie im Kompaktmodus einzelne oder mehrere Verarbeitungsschritte ausführen, die im Assistentenmodus (siehe *Assistenten (Wizards)*) beschrieben sind, ohne die gesamte Abfolge der Verarbeitungsschritte durchlaufen zu müssen.

Damit ist der Kompaktmodus leistungsfähiger als der erweiterte Expertenmodus (Advanced Mode) und verlangt mehr Expertenwissen.

Dieses Kapitel behandelt die folgenden Themen:

Wie Sie den Kompaktmodus wählen

Es gibt zwei Möglichkeiten, den Kompaktmodus zu aktivieren:

- **Sie können das Object Handler-Profil anpassen**
  Siehe die Dokumentation zum Parameter `INPUT-MODE` in *Profileinstellungen*.
- **Durch das Kommando** `SET INPUT-MODE C` **oder** `SET IM C`
  Siehe die Dokumentation zum Kommando `SET INPUT` in *Kommandos für die Navigation und für spezielle Funktionen* in *Direktkommandos*.

Wie Anweisungen im Kompaktmodus verarbeitet werden

Im Kompaktmodus kann die Bearbeitung von Anweisungen in zwei Schritten erfolgen: Im ersten Schritt wählen Sie im Hauptmenü des Kompaktmodus (**Compact Mode Main Menu**) eine Funktion aus, die ausgeführt werden soll. Daraufhin wird ein zweites, für Ihre Wahl spezifisches Menü angezeigt. Hier können Sie die Parameter für die im ersten Schritt gewählte Funktion eingeben.

In diesem Abschnitt werden beide Menüs behandelt:

- Hauptmenü des Kompaktmodus

- Kompaktmodus-Folgemenü

Hauptmenü des Kompaktmodus

Im Hauptmenü des Kompaktmodus können Sie die folgenden Felder angeben:

Feld	PF-Taste	Erläuterung
Function		Wählen Sie eine Funktion: U Entladen (Unload, dies ist die Standardeinstellung) L Laden (Load) S Durchsuchen (Scan) **Vorsicht:** Nach dem Start der Entladefunktion sind weitere Änderungen bei **Function** nicht mehr möglich, es sei denn, Sie beenden die Entladefunktion durch Verlassen des Menüs und starten den Kompaktmodus neu.
Object		Typ des zu bearbeitenden Objekts: L Natural-Library-Objekt (Standardeinstellung) E Natural-System-Fehlermeldung C Natural-Kommandoprozessor-Quellcode R Natural-zugehöriges Objekt D DDM F FDT A Beliebiges Natural-Objekt. * * Nur verfügbar, wenn L oder S bei **Function.** gewählt wurde.
Work file format		Arbeitsdateiformat: Dateiformat für Lade- und Entladevorgänge, das auf die zu verarbeitenden Daten anzuwenden ist: I Internes Format (Standardeinstellung) T Transfer-Format Siehe ***Arbeitsdateiformat***.
Use PC file		PC-Datei verwenden: Gilt nur, wenn Entire Connection installiert ist. Wenn diese Option ausgewählt ist, werden die zu verarbeitenden Daten aus einer Entire Connection-Arbeitsdatei gelesen oder in eine solche geschrieben.
PC file		PC-Datei: Gilt nur, wenn Entire Connection installiert ist.

Feld	PF-Taste	Erläuterung
		Der vollständige Pfadname für die Entire Connection-Arbeitsdatei. Wenn Ihre Systemumgebung keinen Backslash (\) als Trennzeichen akzeptiert, verwenden Sie stattdessen einen Schrägstrich (/).
Write report		Report schreiben: Wenn diese Option auf `Y` gesetzt ist, wird ein Report über die verarbeiteten Objekte in das im Feld **Report text member** angegebene Report-Text-Member geschrieben. Die Option **Write report** ist standardmäßig auf `Y` gesetzt. Um den Report anzuzeigen, geben Sie das interne Kommando `SHOW REPORT FILE` ein (siehe *Kommandos für die Navigation und für spezielle Funktionen* in *Direktkommandos*).
Report text member		Nur gültig, wenn **Write report** ausgewählt wurde. Der Name des in der Arbeitsplan-Bibliothek (Workplan Library) gespeicherten Textbausteins, in den der Report geschrieben wird.
Set additional options	PF4	Zusätzliche Optionen setzen: Wenn diese Option auf `Y` gesetzt ist, wird das Fenster **Options** aufgerufen, in dem Sie die Standardeinstellungen ändern und zusätzliche Optionen für die Verarbeitungsabfolge eingeben können. Die verfügbaren Optionen finden Sie unter *Zusätzliche Optionen festlegen (Set Additional Options)*. Kann nicht verwendet werden / Optionsangaben werden ignoriert, wenn **Use Option Workplan** auf `Y` gesetzt ist. Hinweis: Wenn Optionen definiert wurden, wird der Text „(Options are defined)“ angezeigt.
Use Option Workplan		Option-Workplan verwenden: Wenn auf `Y` gesetzt, wird ein Arbeitsplan vom Typ OPTION verwendet. Siehe auch *Arbeitspläne (Workplans)*.
Option Workplan name		Option-Workplan-Name: Der Name eines Arbeitsplans vom Typ OPTION, der verwendet werden soll.
	PF5	Anzeige einer Auswahlliste der verfügbaren Arbeitspläne vom Typ OPTION. Siehe auch *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* in *Administration*.
	PF6	Zeigt den Inhalt des Arbeitsplans an, der im Feld **Option Workplan name** angegeben ist.
Set global parameters	PF7	Globale Parameter setzen: Ruft den Bildschirm **Parameters** auf. Siehe Abschnitt *Globale Parameter festlegen* und Klausel *parameter-setting* (*Direktkommandos*) bezüglich der Beschreibung der Schlüsselwörter und der gültigen Eingabewerte. Kann nicht verwendet werden / Parameterangaben werden ignoriert, wenn **Use Parameter Workplan** auf `Y` gesetzt ist.

Feld	PF-Taste	Erläuterung
		Hinweis: Wenn Parameter definiert wurden, wird der Text „Parameters are defined“ angezeigt.
Use Parameter Workplan		Parameter-Workplan verwenden: Wenn auf `Y` gesetzt, wird ein Arbeitsplan vom Typ PARAMETER verwendet. Siehe auch ***Arbeitspläne (Workplans)***.
Parameter Workplan name		Parameter-Workplan-Name: Der Name des zu verwendenden Arbeitsplans.
	PF8	Zeigt eine Auswahlliste der verfügbaren Arbeitspläne vom Typ PARAMETER an. Siehe auch ***Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek*** in *Administration*.
	PF9	Zeigt den Inhalt des in **Parameter Workplan name** angegebenen Arbeitsplans an.

Kompaktmodus-Folgemenü

Im Kompaktmodus-Folgemenü können Sie Informationen hinzufügen, die sich auf die Auswahl beziehen, die Sie im Kompaktmodus-Hauptmenü getroffen haben. Eine Dokumentation der zu spezifizierenden Attribute finden Sie unter ***Objekte angeben***.

89 Ladefunktion erneut starten (Restart Load)

Mit der Funktion **Restart Load** können Sie Ladefunktionen, die nicht ordnungsgemäß beendet wurden, wieder aufnehmen. Wenn die Ladefunktion abbricht, bevor die Arbeitsdatei vollständig abgearbeitet wurde, können Sie mit Restart Load an der Abbruchstelle fortfahren.

Der Neustart des Ladens setzt voraus, dass gemäß den für das Laden angegebenen Auswahlkriterien, Optionen und Parametereinstellungen eine Neustart-Information in ein Text-Member (Natural-Objekt vom Typ Text) geschrieben wird.

Um die Umgebung während des Ladevorgangs einzurichten:

1 Auf dem Bildschirm **Load Options**:

- Markieren Sie die Option **Write restart information**.
- Geben Sie in das Feld **Restart text member** den Namen des Text-Members ein, in das die Daten der Restart-Information geschrieben werden sollen. Das Text-Member muss in der Arbeitsplan-Bibliothek (Workplan Library) enthalten sein.

  Oder:

  Markieren Sie das Feld **Select text member** neben **Restart text member** und wählen Sie ein Text-Member aus der Liste der in der Arbeitsplan-Bibliothek enthaltenen Arbeitspläne aus.

Der Bildschirm **Load Options** ist unter *Arbeitsdatei- und Report-Optionen* im Abschnitt *Einstellungen (Settings)* beschrieben.

2 Führen Sie die Ladefunktion aus.

Um einen erneuten Start nach einem unterbrochenen Ladevorgang durchzuführen:

- Geben Sie in der Kommandozeile des Object Handler-Bildschirms das folgende Kommando ein:

```
GO RESTART
```

Der Bildschirm **Restart Options** erscheint. Dort können Sie ein Text-Member angeben, indem Sie einen Namen in das Feld **Restart text member** eingeben oder indem Sie **Select text member** und ein Text-Member aus einer Liste auswählen.

Oder:

Verwenden Sie das folgende Direktkommando:

```
RESTART
```

Die Syntax von `RESTART` ist im Abschnitt *Grundlegende Kommandosyntax* beschrieben.

Verwandte Themen:

Arbeitsplan-Bibliothek ändern (Change Workplan Library) im Kapitel *Administration*.
`GO RESTART` im Abschnitt *Kommandos für die Navigation und für spezielle Funktionen*.

90 Objekte in der Natural-Umgebung sichten (View)

- Natural-Library-Objekte 892
- Natural-Systemfehlermeldungen 893
- Natural-Kommandoprozessor-Quellcodes 894
- FDTs 895
- Natural-zugehörige Objekte (Natural Related-Objects) 896
- DDMs 897

Mit der Funktion **View** können Sie alle in Ihrer Natural-Systemumgebung enthaltenen Objekte anzeigen. Je nach ausgewähltem Objekttyp können Sie mit dieser Funktion bei Bedarf auch ein Objekt löschen.

Um die Funktion View aufzurufen:

- Wählen Sie im Hauptmenü (**Main Menu**) die Option **View**.

  Oder:

  Geben Sie in einem beliebigen anderen Object Handler-Bildschirm das folgende Direktkommando ein:

  ```
  GO VIEW
  ```

  (Siehe auch *Kommandos für die Navigation und spezielle Funktionen* unter *Direktkommandos*.)

  Es erscheint der Bildschirm **Select View Type** mit allen zur Auswahl stehenden Objekttypen.

In diesem Abschnitt wird beschrieben, wie Sie die auf dem Bildschirm **Select View Type** aufgeführten Objekttypen anzeigen können.

Natural-Library-Objekte

Bei Natural-Library-Objekten handelt es sich um Programmierobjekte und benutzerdefinierte Fehlermeldungen.

Um Natural-Library-Objekte anzuzeigen:

1. Wählen Sie auf dem Bildschirm **Select View Type** die Option **Natural library objects**.

   Der Bildschirm **View System Files** erscheint. Er zeigt die Liste aller in der aktuellen Natural-Umgebung verfügbaren Systemdateien.

   Erläuterungen zu den Spalten des Bildschirms finden Sie beim Bildschirm **Select System File** mit identischen Spalten, die in *Systemdatei auswählen (Select System File)* beschrieben sind.
2. Geben Sie in der Spalte **Cmd** ein beliebiges einzelnes Zeichen neben der Systemdatei ein, die Sie auswählen möchten. Die aktuelle FUSER- oder FNAT-Systemdatei ist standardmäßig ausgewählt.

   Es erscheint der Bildschirm **View Libraries** mit einer Liste aller in der angegebenen Systemdatei verfügbaren Libraries.

Sie können die Liste der Libraries von einer bestimmten Library aus starten oder Objekte filtern, indem Sie einen Library-Namen oder einen Bereich von Namen in das Feld **Library** eingeben. Gültige Namensbereiche finden Sie bei ***Name*** unter *Angabe von Name, Datum und Zeit*.

3 Geben Sie in der Spalte **Cmd** neben der Library, die Sie auswählen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

```
S
```

```
SE
```

Es erscheint der Bildschirm **View Library Objects** mit einer Liste aller in der angegebenen Library enthaltenen Objekte.

Erläuterungen zu diesem Bildschirm finden Sie in der Beschreibung des Bildschirms **List**, dessen **Spalten** mit den unter *Objekte auswählen* beschriebenen Spalten identisch sind.

4 Geben Sie in der Spalte **Cmd** neben dem Objekt, das Sie anzeigen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

Oder:

Falls erforderlich, geben Sie neben einem Objekt, das Sie löschen möchten, das folgende Zeilenkommando ein:

```
DE
```

Je nach eingegebenem Kommando wird entweder der Quellcode des ausgewählten Objekts auf dem Bildschirm angezeigt oder es erscheint ein Bestätigungsfenster, über das die Löschfunktion ausgeführt werden kann.

Natural-Systemfehlermeldungen

Um Natural-Systemfehlermeldungen anzuzeigen:

1 Wählen Sie auf dem Bildschirm **Select View Type** die Option **Natural system error messages**.

Der Bildschirm **View System Error Messages** (Systemfehlermeldungen anzeigen) erscheint.

2 Wenn die gewünschten Systemfehlermeldungen nicht in der aktuellen FNAT- oder FUSER-Systemdatei gespeichert sind, können Sie die Datenbankkennung in dem Feld **DBID** und die

Dateinummer in dem Feld **FNR** ersetzen. Geben Sie bei Bedarf das Adabas-Passwort (**Password**) und den Chiffrierschlüssel (**Cipher**) für die Systemdatei an.

Geben Sie bei Bedarf Auswahlkriterien in den Feldern **Error number range**, **Languages** und **Short/Long/All** an. Diese Felder entsprechen den **Spalten** **Number** (Nummer), **Language** (Sprache) und **S/L** (Kurz/Lang), die angezeigt werden, wenn die View-Funktion ausgeführt wird und die Liste der Fehlermeldungen auf dem Bildschirm **View System Error Messages** angezeigt wird.

Erläuterungen zu diesem **View System Error Messages**-Bildschirm finden Sie in der Beschreibung des Bildschirms **List System Error Messages**, der die gleichen **Spalten** enthält.

3 Wenn die Liste der ausgewählten Systemfehlermeldungen auf dem Bildschirm angezeigt wird, geben Sie in der Spalte **Cmd** neben der Fehlermeldung, die Sie anzeigen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

Oder:

Sie können eine Fehlermeldung löschen, indem Sie das folgende Zeilenkommando in der Spalte **Cmd** neben dem gewünschten Objekt eingeben:

```
DE
```

Je nach eingegebenem Kommando wird entweder der Quellcode der ausgewählten Fehlermeldung auf dem Bildschirm angezeigt oder es erscheint ein Bestätigungsfenster, über das die Löschfunktion ausgeführt werden kann.

Natural-Kommandoprozessor-Quellcodes

Um die in einer Adabas-Datei gespeicherten Natural-Kommandoprozessor-Quellcodes anzuzeigen:

1 Wählen Sie auf dem Bildschirm **Select View Type** die Option **Natural command processor sources**.

Der Bildschirm **View Natural Command Processors** erscheint.

2 Wenn die benötigten Natural-Kommandoprozessor-Quellcodes nicht in der aktuellen FUSER-Systemdatei gespeichert sind, geben Sie die erforderliche Datenbankkennung in das Feld **DBID** und die Dateinummer in das Feld **FNR** ein.

Geben Sie, falls erforderlich, ein Adabas-Passwort in das Feld **Passwort** und einen Chiffrierschlüssel in das Feld **Cipher** ein.

Der Bildschirm **View Libraries** erscheint mit einer Liste aller Libraries, in denen Natural-Kommandoprozessor-Quellcodes gespeichert sind.

3 Sie können die Liste der Libraries ab einer bestimmten Library beginnen lassen oder Natural-Kommandoprozessor-Quellcodeobjekte filtern, indem Sie einen Library-Namen oder einen Bereich von Namen in das Feld **Library** eingeben. Gültige Namensbereiche finden Sie unter ***Name*** im Abschnitt *Angabe von Name, Datum und Zeit*.

4 Geben Sie in der Spalte **Cmd** neben der Library, die Sie auswählen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

```
S
```

```
SE
```

Es erscheint der Bildschirm **View Command Processors** mit einer Liste aller Natural-Kommandoprozessor-Quellcodeobjekte, die in der angegebenen Library enthalten sind.

Erläuterungen zu diesem Bildschirm finden Sie in der Beschreibung des Bildschirms **List**, der die gleichen **Spalten** enthält.

5 Sie können ein Objekt löschen, indem Sie das folgende Zeilenkommando in der Spalte **Cmd** neben dem gewünschten Objekt eingeben:

```
DE
```

Es erscheint ein Bestätigungsfenster, in dem Sie die Löschfunktion ausführen können.

FDTs

Um die in einer Adabas-Datenbank verfügbaren FDTs anzuzeigen:

1 Wählen Sie auf dem Bildschirm **Select View Type** die Option **FDTs**.

Der Bildschirm **View FDTs** erscheint.

2 Wenn die gewünschten Objekte nicht in der aktuellen FNAT- oder FUSER-Systemdatei gespeichert sind, ersetzen Sie die Datenbankkennung im Feld **DBID** und, falls erforderlich, den in den Feldern **FNR from** und **FNR to** eingegebenen Bereich von Dateinummern.

Es erscheint der Bildschirm **View FDTs for DBID** mit einer Liste aller FDTs im angegebenen Dateibereich und für die angegebene Datenbank.

Natural-zugehörige Objekte (Natural Related-Objects)

Um Natural-zugehörige Objekte anzuzeigen:

1 Wählen Sie auf dem Bildschirm **Select View Type** die Option **Natural-related objects**.

Es erscheint der Bildschirm **Select Natural-Related Object Type**.

2 Wählen Sie den Natural-zugehörigen Objekttyp, den Sie anzeigen möchten: Profil oder Debug-Umgebung.

Es erscheint der Bildschirm **View**. Dort können Sie angeben, wo sich die Natural-zugehörigen Objekte befinden, die Sie anzeigen möchten.

3 Wenn die gewünschten Objekte nicht in der aktuellen Systemdatei gespeichert sind (FNAT für Profile, FUSER für Debug-Umgebungen), geben Sie die gewünschte Datenbankkennung in das Feld **DBID** und eine Dateinummer in das Feld **FNR** ein.

Geben Sie, falls erforderlich, ein Adabas-Passwort in das Feld **Password** und einen Chiffrierschlüssel in das Feld **Cipher** ein.

Auf dem Bildschirm wird eine Liste aller Natural-zugehörigen Objekte des angegebenen Typs und der angegebenen Systemdatei angezeigt.

Erläuterungen zu diesem Bildschirm finden Sie in der Beschreibung des Bildschirms **List**, der die gleichen **Spalten** hat.

4 Geben Sie in der Spalte **Cmd** neben dem Objekt, das Sie anzeigen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

Oder:

Sie können ein Objekt löschen, indem Sie das folgende Zeilenkommando in der Spalte **Cmd** neben dem gewünschten Objekt eingeben:

```
DE
```

Je nach eingegebenem Kommando wird entweder der Quellcode des ausgewählten Natural-zugehörigen Objekts auf dem Bildschirm angezeigt oder es erscheint ein Bestätigungsfenster, in dem Sie die Löschfunktion ausführen können.

DDMs

Um Natural-Datendefinitionsmodule (DDMs) anzuzeigen:

1 Wählen Sie auf dem Bildschirm **Select View Type** die Option **DDMs**.

Der Bildschirm **View DDMs** wird angezeigt.

2 Wenn die benötigten DDMs nicht in der aktuellen FDIC-Systemdatei gespeichert sind, geben Sie die erforderliche Datenbankkennung in das Feld **DBID** und die Dateinummer in das Feld **FNR** ein.

Geben Sie, falls erforderlich, ein Adabas-Passwort in das Feld **Password** und einen Chiffrierschlüssel in das Feld **Cipher** ein.

Auf dem Bildschirm wird eine alphabetisch sortierte Liste aller in den angegebenen Systemdateien enthaltenen DDMs angezeigt.

Erläuterungen zum Bildschirm **View DDMs** finden Sie in der Beschreibung des Bildschirms **List**, der die gleichen **Spalten** hat.

3 Geben Sie in der Spalte **Cmd** neben dem DDM, das Sie anzeigen möchten, eines der folgenden Zeilenkommandos ein:

```
L
```

```
LI
```

Oder:

Sie können ein DDM löschen, indem Sie das folgende Zeilenkommando in der Spalte **Cmd** neben dem gewünschten DDM eingeben:

```
DE
```

Je nach eingegebenem Zeilenkommando wird entweder der Quellcode des ausgewählten DDMs auf dem Bildschirm **List DDM** angezeigt oder es erscheint ein Bestätigungsfenster, in dem Sie die Löschfunktion ausführen können.

91 Objekte suchen (Find)

- Suchfunktion aufrufen 900

Mit der Funktion **Find** können Sie Objekte in Ihrer Natural-Umgebung suchen und eine Liste der gefundenen Objekte erstellen.

Suchfunktion aufrufen

Um die Suchfunktion aufzurufen:

- Geben Sie in einem beliebigen Object Handler-Bildschirm in der Kommandozeile folgendes Kommando ein:

```
GO FIND
```

Informationen zu den Spalten, die auf dem von der Suchfunktion erzeugten Report-Bildschirm erscheinen, finden Sie im Abschnitt ***Objekte angeben***. Die mit `GO FIND` verfügbaren Unterkommandos finden Sie in *Kommandos für die Navigation und spezielle Funktionen* unter *Direktkommandos*.

92 Administration (Arbeitspläne verwalten)

- Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek 903
- Neuen Arbeitsplan anlegen (Create a new Workplan) 905
- Arbeitsplan-Bibliothek ändern (Change Workplan Library) 907
- Report-Bibliothek ändern (Change Report Library) 908

Diese Funktion dient zur Verwaltung von Objekt Handler-Arbeitsplänen (Workplans).

Um die Funktion aufzurufen, müssen Sie im Object Handler Hauptmenü (**Main Menu**) den Menüeintrag **Administrate the Object Handler environment, process Workplans and direct commands** wählen.

Informationen zu Arbeitsplänen und der entsprechenden Syntax finden Sie in den Kapiteln *Arbeitspläne (Workplans)* und *Direktkommandos*.

Dieses Kapitel beschreibt die Optionen, die Sie auf dem Bildschirm **Administration** wählen können.

Beispiel:

```
13:29:26               ***** Natural Object Handler *****            2023-10-17
User XYZ                        - Administration -

         In this part of the Object Handler you can administrate and
         execute Workplans.

         This functionality is recommended for advanced users only.

         Select the desired function:

           _  List the available Workplans in the Workplan library

           _  Create a new Workplan

           _  Change the Workplan library

           _  Change the Report library




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  List  Creat Edit  Back  Next  ChLib Cmds        Canc   ↩
```

Anleitungen zum Ändern eines Arbeitsplans finden Sie unter *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek*.

Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek

Diese Funktion dient dazu, alle in der Arbeitsplan-Bibliothek enthaltenen Arbeitspläne aufzulisten und einen Arbeitsplan für die weitere Verarbeitung, z. B. das Bearbeiten oder Ausführen des Arbeitsplans, auszuwählen.

Um Arbeitspläne aufzulisten:

- Wählen Sie auf dem Bildschirm **Administration** die Option **Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek** oder wählen Sie PF4 (List).

  Es erscheint der Bildschirm **List Workplans** mit der Liste aller in der Arbeitsplan-Bibliothek enthaltenen Arbeitspläne.

  Handelt es sich bei dem Natural-Objekt vom Typ Text um einen Arbeitsplan, werden der Typ des Arbeitsplans und die ersten 50 Bytes der Arbeitsplanbeschreibung aufgelistet. Sie können PF5 wählen, um zusätzliche Informationen anzuzeigen.

Der Bildschirm **List Workplans** kann auch mit der Auswahlfunktion aufgerufen werden, die z.B. im Bildschirm **Unload Settings** oder **Load Settings** oder **Scan Settings** angeboten wird.

Die im Bildschirm **List Workplans** angezeigten Spalten und die Kommandos, die für einen Arbeitsplan ausgeführt werden können, werden im folgenden Abschnitt beschrieben.

- Spalten und Kommandos im Bildschirm List Workplans

Spalten und Kommandos im Bildschirm List Workplans

Die Spalten und Kommandos im Bildschirm **List Workplans** werden in der folgenden Tabelle erläutert.

Sie können die Eingabefelder unter den Spaltenüberschriften benutzen, um die Liste ab einem bestimmten Arbeitsplan beginnen zu lassen oder Arbeitspläne zu filtern. Gültige Eingabewerte sind in der folgenden Tabelle aufgeführt.

Spalte	PF-Taste	Erläuterung	
Cmd		Die folgenden Zeilenkommandos können in das Eingabefeld neben dem betreffenden Arbeitsplan eingegeben werden:	
		C oder CH	Prüft die Syntax. Gilt nur für Arbeitspläne der Typen PROCEDURE, SELECTION, PARAMETER und OPTION.
		DE	Löscht den Arbeitsplan.

Spalte	PF-Taste	Erläuterung	
		ED	Ermöglicht die Bearbeitung des Arbeitsplans. Sie können den Namen eines Arbeitsplans oder seine Beschreibung im Fenster **Save Workplan** ändern, das im Abschnitt *Arbeitsplan speichern* beschrieben ist.
		EX	Führt den Arbeitsplan aus. Gilt nur für Arbeitspläne des Typs PROCEDURE.
		L oder LI	Listet den Arbeitsplan auf.
		S oder SE	Wählt den Arbeitsplan aus, der für die aktuelle Funktion verwendet werden soll. Gilt nur, wenn der Bildschirm List Workplans mit der Auswahlfunktion aufgerufen wird, z.B. vom Bildschirm **Unload Settings** oder **Load Settings** oder **Scan Settings**.
Name		Der Name des Arbeitsplans. Sie können einen Namen oder einen Namensbereich eingeben, wie unter *Name* in *Angabe von Name, Datum und Zeit* beschrieben.	
Type		Der Typ des Arbeitsplans, z. B. PROCEDURE. Gültige Eingabewerte: PROCEDURE oder P SELECTION oder S LIST oder L PARAMETER oder A OPTION oder O TEXT oder T Sie können auch einen Stern (*) für alle Typen oder eine beliebige Kombination der Kurzformen eingeben, zum Beispiel SL.	
Description		Die Beschreibung des Arbeitsplans. Sie können eine Beschreibung oder einen Bereich von Beschreibung eingeben, wie unter *Name* in *Angabe von Name, Datum und Zeit* beschrieben.	
User ID		Wird nur mit PF5 angezeigt. Die Benutzerkennung des Benutzers, der den Arbeitsplan erstellt hat. Sie können eine Benutzerkennung oder einen Bereich von Benutzerkennungen eingeben, wie unter *Name* in *Angabe von Name, Datum und Zeit* beschrieben.	
Date		Wird nur mit PF5 angezeigt. Das Datum, an dem der Arbeitsplan erstellt wurde.	

Spalte	PF-Taste	Erläuterung
		Sie können ein Datum oder einen Datumsbereich eingeben, wie unter ***Datum*** in *Angabe von Name, Datum und Zeit* beschrieben.
Time		Wird nur mit PF5 angezeigt. Der Zeitpunkt, zu dem der Arbeitsplan erstellt wurde. Sie können eine Zeit oder einen Zeitbereich eingeben, wie unter ***Zeit*** in *Angabe von Name, Datum und Zeit* beschrieben.
	PF4	Schaltet von der Anzeige der Zusatzinformationen (PF5) auf die Standardanzeige um.
	PF5	Zeigt zusätzliche Informationen an: **Benutzerkennung**, **Datum** und **Zeit**.

Neuen Arbeitsplan anlegen (Create a new Workplan)

Diese Funktion ruft den Bildschirm **Create a new Workplan** auf, auf dem Sie den Typ des neuen Arbeitsplans und das zu verwendende Format für die Bearbeitung des Arbeitsplans angeben können.

Wenn Sie die Option **Free Format Editing** nicht wählen (Feld nicht markiert; dies ist die Standardeinstellung), erscheinen bei Arbeitsplänen der Typen OPTION, PARAMETER und SELECTION Bildschirme mit Eingabefeldern.

Wenn Sie die Option **Free Format Editing** nicht markieren (Feld nicht markiert; dies ist die Standardeinstellung), erscheinen bei Arbeitsplänen der Typen OPTION, PARAMETER und SELECTION Bildschirme mit Eingabefeldern.

Wählen Sie die Option **Free Format Editing** (Feld markiert) oder legen Sie einen Arbeitsplan eines anderen Typs an, so erhalten Sie eine Maske mit einem Editierbereich, in dem Sie den Inhalt des Arbeitsplans eingeben können. Siehe auch ***Inhalt von Arbeitsplänen*** im Abschnitt *Arbeitspläne*.

Ein alternatives Direktkommando, mit dem Sie die formatfreie Bearbeitung ein- und ausschalten können, finden Sie unter dem Kommando SET in *Kommandos für die Navigation und spezielle Funktionen* im Kapitel *Direktkommandos*.

Dieser Abschnitt behandelt die folgenden Themen:

- Arbeitsplan vom Typ PROCEDURE anlegen
- Arbeitsplan vom Typ LIST anlegen

- Arbeitsplan speichern

Arbeitsplan vom Typ PROCEDURE anlegen

Sie können einen Arbeitsplan vom Typ PROCEDURE aus dem für die aktuelle Object Handler-Funktion generierten Kommando erstellen.

Um einen Arbeitsplan vom Typ PROCEDURE aus einem generierten Kommando zu erstellen:

1 Führen Sie die Funktion, die Sie für den Arbeitsplan verwenden möchten, mit einem Objekt Handler-Assistenten (Wizard) aus, bis das für die auszuführende Funktion generierte Kommando auf dem Bildschirm angezeigt wird.

Oder:

Im Expertenmodus (Advanced) können Sie die Anzeige des generierten Kommandos aktivieren, indem Sie eine der folgenden Methoden wählen:

- Geben Sie das folgende Object Handler-Kommando ein:

```
SET ADVANCEDCMD ON
```

Oder: Setzen Sie im Object Handler-Profil den Parameter `Display-Cmd-in-Advanced-Mode` auf `Y` (Yes). Ausführliche Informationen finden Sie unter *Profileinstellungen*.

- Führen Sie die Funktion aus, die Sie für den Arbeitsplan verwenden wollen, bis das für die auszuführende Funktion generierte Kommando auf dem Bildschirm angezeigt wird.

2 Wählen Sie PF5 (Save).

Das Fenster **Save Workplan** erscheint.

3 Geben Sie den Namen und die Beschreibung des neuen Arbeitsplans ein und wählen Sie ENTER.

Der Arbeitsplan wird als PROCEDURE-Arbeitsplan in der Arbeitsplan-Bibliothek (Workplan Library) gesichert. Er enthält das für die aktuelle Funktion generierte Kommando.

Arbeitsplan vom Typ LIST anlegen

Ausführliche Informationen zum Anlegen eines Arbeitsplans vom Typ LIST finden Sie im Abschnitt *Objektliste - Arbeitsplantyp LIST*.

Arbeitsplan speichern

Um einen Arbeitsplan zu speichern:

1 Wenn Sie die Bearbeitung eines Arbeitsplans abgeschlossen haben, geben Sie in der Kommandozeile das folgende Kommando ein:

```
SAVE
```

Oder:

Wählen Sie PF5 (Save).

Das Fenster **Save Workplan** erscheint.

2 Geben Sie den Namen und die Beschreibung des Arbeitsplans ein oder ändern Sie sie und wählen Sie ENTER.

Der Arbeitsplan wird unter dem angegebenen Namen in der Arbeitsplan-Bibliothek (Workplan Library) gespeichert.

Arbeitsplan-Bibliothek ändern (Change Workplan Library)

Alle Arbeitspläne müssen in einer Arbeitsplan-Bibliothek gespeichert sein. Mit dieser Funktion können Sie die Arbeitsplan-Bibliothek (Workplan Library) ändern.

Anmerkung: Um die Standardbibliothek für Arbeitspläne festzulegen, können Sie auch den Parameter `Workplan-Library` in Ihrem Object Handler-Profil angeben (siehe *Profileinstellungen*).

Der Bildschirm **Change Workplan Library** enthält die folgenden Felder:

Feld	Erläuterung
Library	Bibliothek: Der Name der Arbeitsplan-Bibliothek. Standardeinstellung ist die Bibliothek `WORKPLAN`.
Select library	Bibliothek auswählen: Zeigt die Liste aller verfügbaren Arbeitsplan-Bibliotheken an. Siehe auch *Bibliothek auswählen (Select Library)*.

Feld	Erläuterung	
	Äquivalente PF-Taste: PF4 (SeLib)	
DBID/FNR	Gibt die Datenbankkennung (DBID) und die Nummer der Datei (FNR) an, in der sich die Arbeitsplan-Bibliothek befindet. Wenn keine Werte angegeben werden, wird die aktuelle FUSER- oder FNAT-Systemdatei verwendet.	
Passw./Ciph.	Passwort und Chiffrierschlüssel: Das Passwort und der Chiffrierschlüssel der Adabas-Datei, in der sich die Arbeitsplan-Bibliothek befindet.	
Store values in profile	Werte im Profil speichern: Legt fest, ob die für die Arbeitsplan-Bibliothek angegebenen Werte im Profil des Object Handlers gespeichert werden:	
	N	Die angegebenen Werte werden nicht gespeichert. Dies ist die Standardeinstellung.
	U	Die angegebenen Werte werden in den benutzerspezifischen Profileinstellungen gespeichert.
	G	Die angegebenen Werte werden in den allgemeinen Profileinstellungen gespeichert.
	Siehe auch ***Profileinstellungen***.	

Report-Bibliothek ändern (Change Report Library)

Mit dieser Funktion können Sie die Report-Bibliothek ändern. Alle Reports (Berichte), die erstellt werden, wenn die Option `REPORT` oder `NEWREPORT` in einem Object Handler-Kommando angegeben ist, werden in der Report-Bibliothek gespeichert. Wenn keine Report-Bibliothek angegeben ist, werden stattdessen die für die Arbeitsplan-Bibliothek angegebenen Werte verwendet.

Tipp: Sie können die Standardbibliothek für Reports festlegen, indem Sie den Parameter Report-Library in Ihrem Object Handler-Profil angeben (siehe ***SYSOBJH-Profilparameter*** im Abschnitt *Profileinstellungen*).

Die Funktion **Change the Report library** im Menü **Administration** ruft den Bildschirm **Change Report Library** auf, der die folgenden Felder enthält:

Feld	Erläuterung	
Library	Bibliothek: Der Name der Report-Bibliothek. Standard-Bibliothek ist die Library `WORKPLAN`.	
Select library	Bibliothek auswählen: Wenn markiert, wird eine Auswahlliste aller verfügbaren Bibliotheken angezeigt. Siehe auch ***Bibliothek auswählen***. Äquivalente PF-Taste: `PF4` (SeLib)	
DBID/FNR	Datenbankkennung/Dateinummer: Gibt die Datenbankkennung (DBID) und die Nummer der Datei (FNR) an, in der sich die Report-Bibliothek befindet. Wenn keine Werte angegeben werden, wird die aktuelle FUSER- oder FNAT-Systemdatei verwendet.	
Passw./Ciph.	Passwort und Chiffrierschlüssel: Das Passwort und der Chiffrierschlüssel der Adabas-Datei, in der sich die Report-Bibliothek befindet.	
Store values in profile	Werte im Profil speichern: Legt fest, ob die für die Report-Bibliothek angegebenen Werte im Profil des Object Handlers gespeichert werden:	
	`N`	Die angegebenen Werte werden nicht gespeichert. Dies ist die Standardeinstellung.
	`U`	Die angegebenen Werte werden in den benutzerspezifischen Profileinstellungen gespeichert.
	`G`	Die angegebenen Werte werden in den allgemeinen Profileinstellungen gespeichert.
	Siehe auch ***Profileinstellungen***.	

93 Systemdatei auswählen (Select System File)

Sie können die Systemdateien, die für die Entladefunktion verwendet werden sollen, aus einer Liste auswählen.

Sie können diese Auswahlliste aus einem Bildschirm zur Objektangabe der Entladefunktion erstellen, wenn diese im Expertenmodus (Advanced User) ausgeführt wird.

Die folgenden Anweisungen sind ein Beispiel für die Verwendung der Funktion beim Entladen von Natural-Library-Objekten.

Um eine Systemdatei aus einer Liste auszuwählen:

1 Wählen Sie auf dem Bildschirm **Unload Natural Library Objects** die Option **Select DBID/FNR** oder wählen Sie PF5 (DBIDs).

Es erscheint das Fenster **Select System File**, in dem die in der aktuellen Natural-Umgebung verfügbaren Systemdateien mit Namen (**System File**), Datenbankkennung (**DBID**) und Dateinummer (**FNR**). `User defined` (benutzerdefiniert) bezeichnet eine vom Benutzer angegebene Systemdatei.

2 Wählen Sie die Systemdatei, die Sie für die Funktionsverarbeitung verwenden möchten, indem Sie ein beliebiges Zeichen in die Spalte **Sel** neben der gewünschten Systemdatei eingeben. Die Systemdatei FUSER ist standardmäßig ausgewählt.

Oder: In den Feldern **DBID** und **FNR** neben `User-defined` können Sie die Datenbankkennung und die Dateinummer der Systemdatei eingeben, die Sie auswählen möchten. Geben Sie, falls erforderlich, in den Spalten **Password/VSAM name** und **Cipher Code** das Adabas-Passwort oder den VSAM-Namen und den Adabas-Chiffierschlüssel für die Systemdatei ein.

Die Datenbankkennung und die Dateinummer der ausgewählten Systemdatei werden in die Felder **DBID/FNR** des Bildschirms **Unload Natural Library Objects** eingetragen. Wenn Sie die Standardsystemdatei ausgewählt haben, bleiben diese Felder leer.

94 Bibliothek auswählen (Select Library)

Sie können die Library, die für die Entladefunktion verwendet werden soll, aus einer Liste auswählen.

Die Auswahlliste wird mit der Funktion **Select library** (oder **Select**) erstellt, die auf dem Bildschirm zur Objektangabe der Entladefunktion zur Verfügung steht, wenn diese im Expertenmodus (Advanced User) ausgeführt wird.

Die folgenden Anweisungen sind Beispiele für die Auswahl einzelner oder mehrerer Libraries beim Entladen von Natural-Library-Objekten.

Um eine einzelne Library aus einer Liste auszuwählen:

1 Wählen Sie auf dem Bildschirm **Unload Natural Library Objects** die Option **Select library** oder PF4 (SeLib).

Es erscheint das Fenster **Select Library** mit der Liste aller Libraries und den Datenbankkennungen (**DBID**) und Dateinummern (**FNR**) der Systemdatei, in der die Libraries gespeichert sind.

2 Geben Sie in der Spalte **Cmd** neben der gewünschten Library ein beliebiges einzelnes Zeichen ein.

3 Wählen Sie ENTER.

Das Feld **Library** und die Felder **DBID/FNR** des Bildschirms **Unload Natural Library Objects** werden mit den angegebenen Namen bzw. Nummern gefüllt. Wenn in den Feldern **DBID/FNR** keine Werte (oder 0) eingegeben werden, werden die aktuellen FUSER- und FNAT-Systemdateien ausgewählt.

Um mehrere Libraries aufzulisten und auszuwählen:

1 Wählen Sie im Bildschirm **Unload Natural Library Objects** die Option **Select library** oder PF4 (SeLib).

Es erscheint das Fenster **Select Library** mit einer Liste aller Libraries und den Datenbankkennungen (**DBID**) und Dateinummern (**FNR**) der Systemdatei, in der die Libraries gespeichert sind.

2 Geben Sie im Feld **Library** einen Namen oder einen Bereich von Namen ein, um die Libraries zu filtern, die Sie auswählen möchten. Wenn Sie einen einzelnen Library-Namen eingeben, wird die Liste mit dieser Library beginnen. Gültige Namensbereiche finden Sie unter ***Name*** in *Angabe von Name, Datum und Zeit*.

Oder:

Geben Sie in die Felder **DBID** und **FNR** die Datenbankkennung und die Dateinummer der Systemdatei ein, die die auszuwählenden Libraries enthält. Wenn keine Werte (oder 0) eingegeben werden, werden die aktuellen FUSER- und FNAT-Systemdateien verwendet.

Beachten Sie, dass die Felder **DBID** und **FNR** schreibgeschützt sind, wenn das Fenster **Select Library** von einem **Exceptions**-Bildschirm aus aufgerufen wurde.

3 Wählen Sie ENTER.

Im Fenster **Select Library** werden nun alle Libraries des angegebenen Bereichs aufgelistet.

4 Wählen Sie PF4 (Se Rng).

Das Feld **Library** und die **DBID/FNR**-Felder des Bildschirms **Unload Natural Library Objects** werden mit dem angegebenen Namen (oder Bereich) bzw. den Zahlen gefüllt. Wenn in den Feldern **DBID/FNR** keine Werte (oder **0**) eingegeben werden, werden die aktuellen FUSER- und FNAT-Systemdateien ausgewählt.

95 Systemfehlermeldungen auswählen (Select System Error Messages)

- Spalten und Kommandos 916

Sie können die zu entladenden Natural-Systemfehlermeldungen aus einer Liste auswählen.

Sie können diese Auswahlliste aus einem Bildschirm zur Objektangabe der Entladefunktion erstellen, wenn diese im Expertenmodus (Advanced User) ausgeführt wird.

Um Natural-Systemfehlermeldungen auszuwählen:

- Ändern Sie im Bildschirm **Unload Natural System Error Messages** bei Bedarf die Meldungsnummern in den Feldern **Error number from/to** (Standardeinstellung ist der gesamte Nummernbereich) und wählen Sie dann **Select system error messages**.

  Es erscheint der Bildschirm **List System Error Messages** mit einer Liste aller Systemfehlermeldungen, die in der angegebenen Systemdatei enthalten sind.

  Dieser Bildschirm wird im folgenden Abschnitt beschrieben.

Anmerkung: Die Auswahlfunktion für benutzerdefinierte Fehlermeldungen wird im Abschnitt *Objekte auswählen* beschrieben.

Spalten und Kommandos

Die Spalten und Kommandos, die im Bildschirm **List System Error Messages** zur Verfügung stehen, werden in der folgenden Tabelle erläutert.

Sie können die Eingabefelder unter den Spaltenüberschriften verwenden, um die Liste mit einer bestimmten Systemfehlermeldung beginnen zu lassen oder die Meldungen zu filtern. Gültige Eingabewerte sind in der nachstehenden Tabelle aufgeführt.

Spalte	PF-Taste	Erläuterung	
CMD		Eines der folgenden Zeilenkommandos kann in das Eingabefeld neben der gewünschten Systemfehlermeldung eingegeben werden:	
		`L` oder `LI`	Listet den Kurz- und Langtext der Meldung auf.
		`S` oder `SE` `UL` oder `U`	Markiert die Meldung zum späteren Entladen. **Achtung:** Jedes dieser Kommandos markiert nur die ausgewählte Meldung für die weitere Verarbeitung. Um die Entladefunktion auszuführen, müssen Sie PF2 (Unloa) wie unten beschrieben wählen.
		`DE`	Löscht die Meldung.
		`DL`	Löscht nur den Langtext der Meldung.

Spalte	PF-Taste	Erläuterung	
Number		Die Nummer der Systemfehlermeldung. Sie können eine Nummer oder einen Nummernbereich eingeben. Gültige Bereiche sind:	
		*value**	Alle Meldungen mit Nummern, die mit *value* beginnen.
		value>	Alle Meldungen mit Nummern, die größer oder gleich *value* sind. Beispiel: 10>
		value<	Alle Meldungen mit Nummern, die kleiner als oder gleich *value* sind. Beispiel: 100<
S/L		Die Art des Textes der Systemfehlermeldung:	
		S	Kurztext.
		L	Langtext.
		A	Kurz- und/oder Langtext.
Language		Der Sprachcode der Systemfehlermeldung. Sie können bis zu 8 gültige Sprachcodes (z. B. 1 für Englisch) für die zu wählenden Fehlermeldungen eingeben. Mit einem Stern (*) werden alle Sprachcodes ausgewählt.	
Error Message Text		Der Kurztext der Systemfehlermeldung.	
	PF2	Startet das Entladen der zur Verarbeitung ausgewählten Systemfehlermeldungen. Alternativ können Sie in der Kommandozeile eines der folgenden Direktkommandos eingeben: UNLOAD oder UNLD	
	PF11	Markiert alle aufgelisteten Systemfehlermeldungen zum späteren Entladen mit PF2. Alternativ können Sie in der Kommandozeile eines der folgenden Direktkommandos eingeben: SELECT ALL oder SEL ALL	

96 Objekte auswählen (Select Objects)

- Spalten und Kommandos in List-Bildschirmen 920

Sie können die zu entladenden Objekte aus einer Liste auswählen. Diese Auswahlliste kann auch für andere Zwecke verwendet werden, z.B. um die Quelle eines Objekts anzuzeigen oder um es zu löschen.

Die Auswahlliste wird mit der Funktion **Select objects** (oder **Select**) erstellt, die auf dem Bildschirm zur Objektangabe der Entladefunktion zur Verfügung steht, wenn diese im Expertenmodus (Advanced User) ausgeführt wird.

Die Auswahlliste wird in einem **List**-Bildschirm angezeigt, der im folgenden Abschnitt beschrieben wird.

Anmerkung: Die Auswahlfunktion für Natural-Systemfehlermeldungen ist im Abschnitt *Systemfehlermeldungen auswählen (Select System Error Messages)* beschrieben.

Spalten und Kommandos in List-Bildschirmen

Die Spalten und Kommandos, die in einem **List**-Bildschirm zur Verfügung stehen, werden in der folgenden Tabelle erläutert.

Die Anzeige der Spalten in einem **List**-Bildschirm hängt vom Typ des Objekts ab, das im Menü **Select Unload Type** ausgewählt wurde. Der Typ des bearbeiteten Objekts ist in der Bildschirmüberschrift enthalten, z.B. **List Library Objects** oder **List Command Processors**.

Sie können die Eingabefelder unter den Spaltenüberschriften verwenden, um die Liste ab einem bestimmten Objekt beginnen zu lassen oder Objekte zu filtern. Gültige Eingabewerte sind in der folgenden Tabelle aufgeführt.

Spalte	PF-Taste	Erläuterung	
Cmd		In das Eingabefeld neben dem gewünschten Objekt kann eines der folgenden Zeilenkommandos eingegeben werden:	
		`L` oder `LI`	Zeigt den Quellcode des Objekts an (gilt nicht für Natural-Kommandoprozessor-Quellcodes). Bei einer benutzerdefinierten Fehlermeldung: Listet den Kurz- und Langtext der Fehlermeldung auf.
		`S` oder `SE` `UL` oder `U`	Markiert das Objekt zum späteren Entladen. **Achtung:** Jedes dieser Kommandos markiert nur das ausgewählte Objekt für die Weiterverarbeitung. Um die Entladefunktion auszuführen, müssen Sie PF2 (Unloa) wählen (siehe unten).

<table>
<tr><th>Spalte</th><th>PF-Taste</th><th>Erläuterung</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td></tr>
<tr><td>DE</td><td>Löscht das Objekt.</td></tr>
</table>
</td></tr>
<tr><td>Name</td><td></td><td>Der Name des Objekts.

Sie können einen Namen oder einen Namensbereich eingeben. Siehe Name in Angabe von Name, Datum und Zeit.

Bei einer benutzerdefinierten Fehlermeldung wird die Meldungsnummer und der Sprachcode angezeigt. Beispiel: 10 (Lang =1) steht für die Meldung mit der Nummer 10 in der Sprache 1 (Englisch).</td></tr>
<tr><td>Type</td><td></td><td>Der Typ des Natural-Library-Objekts, z.B. Program.

Gültige Eingabewerte sind ein oder mehrere Objekttypcodes, z.B. P für Programm. Eine Liste der Codes finden Sie unter NATTYPE in der Beschreibung der select-clause-Klausel.

Für ein Natural-zugehöriges Objekt (Natural-related Object) vom Typ Profil enthält diese Spalte den Typ des Profils (z. B. Editor) .</td></tr>
<tr><td>S/C</td><td></td><td>Die Art des Natural-Library-Objekts:

Standardmäßig werden alle verfügbaren Quellcodeobjekte (S) und/oder katalogisierten Objekte (C) auf dem Bildschirm angezeigt.
<table>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Gültige Eingabewerte sind einer oder mehrere der folgenden Codes:</td></tr>
<tr><td></td><td></td></tr>
<tr><td>S</td><td>Nur Quellcodeobjekte.</td></tr>
<tr><td>C</td><td>Nur katalogisierte Objekte.</td></tr>
<tr><td>S/C</td><td>Sowohl Quellcode- als auch katalogisierte Objekte, wenn beide existieren.</td></tr>
<tr><td>W</td><td>Alle mit STOW gespeicherten Objekte: Quellcode- und katalogisierte Objekte mit identischem Datum und Zeit.</td></tr>
<tr><td>*</td><td>Alle Quellcodeobjekte und/oder katalogisierten Objekte.</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">Bei einer benutzerdefinierten Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.</td></tr>
</table>
</td></tr>
<tr><td>M</td><td></td><td>Der Programmiermodus des Natural-Library-Objekts. Standardmäßig wird „Beliebiger Modus“ angezeigt.

Gültige Eingabewerte sind einer oder mehrere der folgenden Codes:

S Nur Structured-Modus.</td></tr>
</table>

Spalte	PF-Taste	Erläuterung
		R Nur Reporting-Modus. * Beliebiger Modus, Structured und/oder Reporting. Bei einer benutzerdefinierten Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.
Version		Die Natural-Version unter der das Natural-Library-Objekt gespeichert und/oder katalogisiert wurde. Gültige Versionsnamensbereiche siehe ***Name*** in *Angabe von Name, Datum und Zeit*. Bei einer benutzerdefinierte Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.
User ID		Die Benutzerkennung des Benutzers, der das Natural-Library-Objekt oder das DDM gespeichert oder katalogisiert hat. Sie können eine einzelne Benutzerkennung oder einen Bereich von Benutzerkennungen eingeben. Siehe ***Name.*** Bei einer benutzerdefinierten Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.
Date		Das Datum, an dem das Natural-Library-Objekt oder das DDM gespeichert oder katalogisiert wurde. Sie können ein Datum oder einen Datumsbereich eingeben. Siehe ***Datum*** in *Angabe von Name, Datum und Zeit*. Bei einer benutzerdefinierten Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.
Time		Die Zeit, wann das Natural-Library-Objekt oder DDM gespeichert oder katalogisiert wurde. Sie können eine bestimmte Zeit oder einen Zeitbereich eingeben. Siehe ***Zeit*** in *Angabe von Name, Datum und Zeit*. Bei einer benutzerdefinierten Fehlermeldung enthält diese Spalte den Kurztext der Fehlermeldung.
DBID		Die Datenbankkennung der Systemdatei, in der DDMs oder Natural-zugehörige Objekte (Natural-related Objects) gespeichert sind.
FNR		Die Dateinummer der Systemdatei, in der DDMs oder Natural-zugehörige Objekte (Natural-related Objects) gespeichert sind.

	PF2	Startet das Entladen der zur Verarbeitung ausgewählten Objekte.
		
		Alternativ können Sie in der Kommandozeile auch eines der folgenden Direktkommandos eingeben:
		
		`UNLOAD`
		oder
		`UNLD`

Spalte	PF-Taste	Erläuterung

	PF11	Markiert alle aufgelisteten Objekte für ein späteres Entladen mit **PF2**.
		
		Alternativ können Sie in der Kommandozeile eines der folgenden Direktkommandos eingeben:
		
		`SELECT ALL`
		oder
		`SEL ALL`

97 Objekte angeben

Das Object Handler-Hauptmenü (**Main Menu**) enthält Einträge zur Auswahl der gewünschten Funktion (**Select the desired function**). Nach dem Aufruf einer Funktion erscheint der entsprechende Folgebildschirm **Unload Settings** oder **Load Settings** oder **Scan Settings**. Wenn Sie die Eingaben bestätigen, erscheint der entsprechende Folgebildschirm zur Auswahl des betreffenden Objekts: **Select Unload Type** bzw. **Select Load Type** bzw. **Select Scan Type**.

Im zugehörigen Bildschirm können Sie die zu bearbeitenden Objekttypen auswählen oder einen Arbeitsplan vom Typ SELECTION oder LIST angeben.

Für jeden ausgewählten Objekttyp werden Ihnen spezifische Bildschirme zur Objektangabe angeboten. Dort können Sie die Auswahlkriterien für die zu verarbeitenden Objekte festlegen.



Anmerkung: Als zeitsparende Alternative können erfahrene Benutzer den Kompaktmodus (**Compact Mode**) verwenden, siehe *Kompaktmodus*.

In diesem Kapitel werden die Optionen beschrieben, die auf jedem Bildschirm für die Objektangabe zur Verfügung stehen. Wenn ein Feld oder eine PF-Taste, das bzw. die in diesem Abschnitt beschrieben wird, nur bei einer bestimmten Funktion und/oder im Expertenmodus erscheint, ist dies durch einen entsprechenden Hinweis gekennzeichnet, zum Beispiel dass das Feld oder die PF-Taste nur bei der Funktion Entladen im Expertenmodus gilt.

Alle Objekte in der Arbeitsdatei laden/durchsuchen

Natural-Library-Objekte entladen/laden/durchsuchen

Natural Systemfehlermeldungen

Natural-Kommandoprozessoren

Natural-zugehörige Objekte (Natural Related-Objects)

Datendefinitionsmodule (DDMs)

Felddefinitionstabellen (FDTs)

Arbeitsplan-Typ SELECTION oder LIST verwenden

98 Objektangabe - Alle Objekte in der Arbeitsdatei laden/durchsuchen

Object Specification > Load/Scan all objects on the work file

(Gilt nur für die Funktion **Load** oder **Scan**.)

Die Option **Load/Scan All Objects on the Work File** (Alle Objekte in der Arbeitsdatei laden/durchsuchen) wird verwendet, um alle in der Arbeitsdatei verfügbaren Objekte zur Verarbeitung auszuwählen. Im Expertenmodus (Advanced User) können Sie vom Bildschirm **Load/Scan All Objects** aus den Bildschirm **Settings** aufrufen, in dem Sie Options- und Parametereinstellungen vornehmen können. Siehe den Abschnitt *Einstellungen (Settings)*.

99 Objektangabe - Natural-Library-Objekte entladen/laden/durchsuchen

- Natural-Library-Objekte 930
- Natural-Library-Objekte - weitere Auswahlkriterien angeben 931
- Natural-Library-Objekte - Eigenschaften angeben 934
- Natural-Library-Objekte - Ausnahmen angeben 935
- Natural-Library-Objekte - Eigenschaften der Ausnahmen für Natural-Library-Objekte 936

Object Specification > Unload/Load/Scan Natural Library Objects

Im Folgenden werden die Optionen beschrieben, die auf den Bildschirmen zur Objektangabe für die Verarbeitung von Natural-Library-Objekten zur Verfügung stehen. Natural-Library-Objekte sind Programmierobjekte und benutzerdefinierte Fehlermeldungen.

Beschreibungen der Schlüsselwörter und der gültigen Eingabewerte finden Sie außerdem bei der Auswahlklausel (**select-clause**) im Kapitel *Direktkommandos*.

Natural-Library-Objekte

Im Bildschirm **Unload/Load/Scan Natural Library Objects** stehen die folgenden Felder und PF-Tasten zur Verfügung

Feld	PF-Taste	Erläuterung
DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und die Dateinummer (FNR) der Systemdatei, in der die Natural Libraries gespeichert sind. Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle Systemdatei FUSER oder FNAT verwendet.
Select DBID/FNR	PF5 (nur im Expertenmodus)	Gilt nur für die Entladefunktion. Zeigt eine Auswahlliste der verfügbaren Systemdateien an.
Password/Cipher		Gilt nur für die Entladefunktion. Das Passwort und der Chiffrierschlüssel für die Adabas-Datei, in der die Natural Libraries gespeichert sind.
Library		Der Name einer Library oder ein Namensbereich. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select library	PF4	Zeigt eine Auswahlliste mit allen verfügbaren Libraries an. Siehe auch ***Library auswählen***.
Object name		Der Name eines Natural-Programmierobjekts oder ein Namensbereich. Siehe ***Name***.. Wird nur ausgewertet, wenn das Feld **Natural programming objects** (Standardeinstellung) im Bildschirm **Natural Library Objects, Details** markiert ist. Siehe auch ***Natural-Library-Objekte - weitere Auswahlkriterien angeben***.
Select objects		Gilt nur für die Entladefunktion im Expertenmodus. Wenn kein Library-Bereich angegeben ist, wird eine Auswahlliste mit allen verfügbaren Natural-Objekten angezeigt (siehe auch ***Objekte auswählen***).

Feld	PF-Taste	Erläuterung
Error number from/to		Ein gültiger Bereich (`1 - 9999`) von benutzerdefinierten Fehlermeldungen, begrenzt durch die erste und die letzte Meldungsnummer. Wird nur ausgewertet, wenn im Bildschirm **Natural Library Objects, Details** das Feld **Error messages** (Standardeinstellung) markiert ist (siehe auch *Natural-Library-Objekte - weitere Auswahlkriterien angeben*).
Details	PF6	**Ruft den Bildschirm Natural Library Objects, Details auf, in dem Sie detailliertere Objektangaben machen können. Siehe** *Natural-Library-Objekte - weitere Auswahlkriterien angeben*.
Settings	PF7	Gilt nur für Funktionen, die im Expertenmodus ausgeführt werden. Ruft den Bildschirm **Unload/Load/Scan Settings** auf, auf dem Sie Options- und Parametereinstellungen festlegen können. Siehe *Einstellungen (Settings)*.

Natural-Library-Objekte - weitere Auswahlkriterien angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural Library Objects** das Feld **Details** markiert haben, wird der Bildschirm **Unload/Load/Scan Natural Library Objects, Details** angezeigt. Er wird verwendet, um weitere Auswahlkriterien für Natural-Library-Objekte anzugeben.

Beschreibung der Schlüsselwörter und der gültigen Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Im Bildschirm **Unload/Load/Scan Natural Library Objects, Details** stehen die folgenden Felder und PF-Tasten zur Verfügung:

Feld	PF-Taste	Erläuterung
Library		Der Name einer Library oder ein Namensbereich. Siehe *Name* in *Angabe von Name, Datum und Zeit*. Die Angabe von Bereichen ist nicht zulässig, wenn die Option **Use Predict set** markiert ist.
Select (Library)	PF4	**Zeigt eine Auswahlliste mit allen verfügbaren Libraries an. Siehe auch** *Library auswählen*.
DBID/FNR		See **DBID/FNR** in *Natural Library Objects* above.
Passw./Ciph.		Gilt nur für die Entladefunktion. Das Passwort und der Chiffrierschlüssel für die Adabas-Datei, in der die Natural-Libraries gespeichert sind.

Feld	PF-Taste	Erläuterung
Object Types: **Natural programming objects**		Natural-Programmierobjekte.
Object Types: **Error messages**		Benutzerdefinierte Fehlermeldungen.
Object name		Siehe **Object name** in *Natural Library Objects* weiter oben.
Use Predict set		Gilt nur für die Funktionen Entladen und Suchen und wenn Predict installiert ist. Diese Option wird verwendet, um die Namen der zu verarbeitenden Objekte aus einem Retained Set zu lesen. Ein Retained Set wird mit der Option `Save Set` des Kommandos `LIST XREF` erstellt. Wenn die Option **Use Predict set** ausgewählt ist, gilt folgendes: ■ Das Feld **Object name** muss einen Stern (*) enthalten, der für alle Objekte steht. Dies ist die Standardeinstellung. ■ Das Feld **Library** muss den Namen einer einzelnen Library enthalten. Namensbereiche sind nicht zulässig. ■ Das Feld **Set number** muss ausgefüllt werden. Ausführliche Informationen zu Predict Sets finden Sie in der *Predict*-Dokumentation.
Set number		Gilt nur, wenn **Use Predict set** ausgewählt ist. Eine ein- oder zweistellige Nummer, die das zu verwendende Predict Set identifiziert.
Set library		Gilt nur, wenn **Use Predict set** ausgewählt ist. Der Name der Library, die nach einem Predict Set durchsucht werden soll. Wenn Sie keinen Namen angeben, wird standardmäßig die im Feld **Library** eingegebene Library verwendet.
Set user		Gilt nur, wenn **Use Predict set** ausgewählt ist. Die Benutzerkennung des Benutzers, der das Retained Set erstellt hat. Wird keine Benutzerkennung eingegeben, wird die mit der Systemvariablen `*USER` (siehe *Systemvariablen*-Dokumentation) angegebene Benutzerkennung verwendet.
Programmierobjekt-Optionen: **S/C-Kind**		Die Art des Natural-Programmierobjekts: `S` — Nur Quellcodeobjekte. `C` — Nur katalogisierte Objekte.

<table>
<tr><th>Feld</th><th>PF-Taste</th><th>Erläuterung</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>A oder *</td><td>Alle Quellcodeobjekte und/oder katalogisierten Objekte. Dies ist die Standardeinstellung.</td></tr>
<tr><td>W</td><td>Alle mit dem Systemkommando STOW behandelten Objekte: Quellcode- und katalogisierte Objekte mit identischen Werten für Datum und Zeit.</td></tr>
<tr><td>B</td><td>Sowohl Quellcode- als auch katalogisierte Objekte, wenn beide vorhanden sind.</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Anmerkung: W und B sind nur für die Entladefunktion gültig. W und B können zwar auch für die Lade- oder Scan-Funktion eingegeben werden, werden aber wie A behandelt.</td></tr>
</table>
</td></tr>
<tr><td>Programmierobjekt-Optionen:

Natural types</td><td></td><td>Ein Natural-Objekttypcode, zum Beispiel P für Programm. Eine Liste der gültigen Codes finden Sie unter NATTYPE im Abschnitt select-clause.</td></tr>
<tr><td>Select Natural types</td><td>PF6</td><td>Ruft ein Fenster auf, in dem Sie einen oder mehrere Typen von Natural-Objekten auswählen können.</td></tr>
<tr><td>Properties</td><td>PF7</td><td>Ruft ein zusätzliches Fenster auf, in dem Sie zusätzliche Eigenschaften von Natural-Programmierobjekten angeben können: siehe Eigenschaften von Natural-Library-Objekten.</td></tr>
<tr><td>Fehlermeldungen:

Error number from/to</td><td></td><td>Ein Bereich von benutzerdefinierten Fehlermeldungen, wie sie in den Feldern Error number from/to eingegeben wurden (siehe Natural Library Objects weiter oben).</td></tr>
<tr><td>Fehlermeldungen:

Language codes</td><td></td><td>Bis zu 8 gültige Sprachcodes (z. B. Code 1 für Englisch) für die angegebenen Fehlermeldungen. Ein Stern (*) wählt alle Sprachcodes aus.</td></tr>
<tr><td>Fehlermeldungen:

S/L-Kind</td><td></td><td>
Die Art des Fehlermeldungstextes:
<table>
<tr><td></td><td></td></tr>
<tr><td>S</td><td>Kurztext.</td></tr>
<tr><td>L</td><td>Langtext.</td></tr>
<tr><td>A</td><td>Kurz- und/oder Langtext. Dies ist die Standardeinstellung.</td></tr>
<tr><td>B</td><td>Kurz- und Langtext, wenn beide vorhanden sind (nur bei der Entladefunktion).</td></tr>
</table>
</td></tr>
<tr><td>Exceptions</td><td>PF8</td><td>Ruft ein zusätzliches Fenster auf, in dem Sie Ausnahmen für die Auswahl von Natural-Programmierobjekten angeben können. Siehe Natural Library Object Exceptions - Ausnahmen für Natural-Library-Objekte.</td></tr>
</table>

Natural-Library-Objekte - Eigenschaften angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural Library Objects, Details** das Feld **Properties** markiert haben, wird der Bildschirm **Unload/Load/Scan Natural Library Objects, Properties** angezeigt. Auf diesem Bildschirm können Sie Ausnahmen zu den Eigenschaften der für die Bearbeitung ausgewählten Natural-Library-Objekte angeben.

Beschreibung der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan Library Objects, Properties** enthält die folgenden Felder:

Feld	Erläuterung	
User ID	Die Benutzerkennung des Benutzers, der ein Natural-Programmierobjekt gespeichert oder katalogisiert hat. Geben Sie eine einzelne Benutzerkennung oder einen Bereich von Benutzerkennungen an. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.	
Programming mode	Der Programmiermodus der Natural-Programmierobjekte:	
	R	Nur Reporting Mode.
	S	Nur Structured Mode.
	A	Es erfolgt keine Prüfung des Programmiermodus. Dies ist die Standardeinstellung.
Natural version	Die Natural-Version des Natural-Programmierobjekts. Sie können auch einen Bereich von Natural-Versionen angeben. Siehe auch ***Name***.	
Objektdatum: **Select all objects (no date check)**	Auswahl aller Objekte (keine Datumsprüfung).	
Objektdatum: **Select objects modified between/and**	Auswahl aller Objekte mit Speicher- oder Katalogisierungsdatum und/oder -Zeit, die mit dem in diesen Feldern angegebenen Datum/Zeit übereinstimmen, indem Sie ein genaues Datum und/oder eine genaue Zeit für den Anfang und/oder das Ende der Auswahl eingeben. Gültige Eingabewerte siehe ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*. Zulässige spezielle Datumsangaben: `TODAY`, `YESTERDAY`, `MONTH` und `YEAR`.	
Objektdatum: **Select objects modified on**	Auswahl aller Objekte mit Speicher- oder Katalogisierungsdatum und/oder -Zeit, die mit dem in diesen Feldern angegebenen Datum/Zeit übereinstimmen, indem Sie ein genaues Datum und/oder eine genaue Zeit eingeben. Gültige Eingabewerte siehe ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*. Zulässige spezielle Datumsangaben: `TODAY` und `YESTERDAY`.	

Feld	Erläuterung
Objektdatum: **Check date for object kind**	Art des Objekts (Quellcode oder katalogisiertes Objekt), das für das definierte Datum oder den definierten Datumsbereich geprüft werden soll. Gültige Eingabewerte: `S` Prüfung des Datums von Quellcodeobjekten und (falls vorhanden) des Datums von Objekten, die nur als katalogisierte Objekte existieren. `C` Prüfung des Datums von katalogisierten Objekten und (falls vorhanden) des Datums von Objekten, die nur als Quellcodeobjekte vorhanden sind. leer Die Objektart wird nicht geprüft. Dies ist die Standardeinstellung. Das Suchergebnis der Objektprüfung hängt auch von der Einstellung des Feldes **S/C-Kind** ab.
Objektgröße: **Select all objects (no size check)**	Alle Objekte auswählen (keine Größenprüfung).
Objektgröße: **Select objects with size between/and**	Wählt alle Objekte mit einer Größe innerhalb des in diesen Feldern angegebenen Bereichs aus, indem eine Startgröße und/oder eine Endgröße eingegeben wird.
Objektgröße: **Select objects with size**	Wählt alle Objekte mit einer Größe aus, die der in diesem Feld angegebenen Größe entspricht.

Natural-Library-Objekte - Ausnahmen angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural Library Objects, Details** das Feld **Exceptions** markiert haben, wird der Bildschirm **Unload/Load/Scan Natural Library Objects, Exceptions** angezeigt. Dieser Bildschirm ermöglicht die Angabe von Ausnahmen bei der Auswahl von Natural-Library-Objekten.

Alle Objekte, die den in *Natural Library Objects*, *Natural-Library-Objekte - weitere Auswahlkriterien angeben* und *Natural Library Properties* angegebenen Auswahlkriterien entsprechen, werden gegen die auf dem Bildschirm **Unload/Load/Scan Library Objects, Exceptions** gemachten Angaben geprüft. Objekte, die *allen* als Ausnahmen definierten Angaben entsprechen, werden von der Verarbeitung ausgenommen.

Zur Beschreibung der Schlüsselwörter und der gültigen Eingabewerte siehe auch `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan Library Objects, Exceptions** ist im Wesentlichen identisch mit dem Bildschirm **Unload/Load/Scan Natural Library Objects, Details**. Erläuterungen zu den in

der folgenden Tabelle aufgeführten Feldern, Kommandos und alternativen PF-Tasten finden Sie in der Tabelle weiter unten. Das Feld **Add/change properties for selection** dient zur Angabe zusätzlicher Eigenschaften für Natural-Programmierobjekt-Ausnahmen: siehe *Natural Library Object Exception Properties*.

Feld	PF-Taste
Library	PF4
Select (Library)	
Objekttypen: **Natural programming objects** **Error messages**	
Object name	
S/C-Kind	
Natural types	
Select Natural types	PF6
Properties	PF7
Error number	
S/L-Kind	
Languages	

Natural-Library-Objekte - Eigenschaften der Ausnahmen für Natural-Library-Objekte

Wenn Sie im Bildschirm **Unload/Load/Scan Library Objects, Exceptions** die Taste PF7 (Prope) wählen, erscheint der Bildschirm **Unload/Load/Scan Library Objects, Exceptions, Properties**. Hier können Sie Ausnahmen zu den Eigenschaften der für die Verarbeitung ausgewählten Natural-Library-Objekte angeben.

Der Bildschirm enthält die folgenden Felder:

Feld	Erläuterung
User ID	Siehe **User ID** in *Eigenschaften von Natural-Library-Objekten*.
Programming mode	Siehe **Programming mode** in *Eigenschaften von Natural-Library-Objekten*.
Natural version	Siehe **Natural version** in *Eigenschaften von Natural-Library-Objekten*.
Objektdatum: **Ignore object date**	Es erfolgt keine Datumsprüfung. Die Objekte werden unabhängig von ihrem Datum verarbeitet. Objektdatum:

Feld	Erläuterung
Objektdatum: **Exclude objects modified between/and**	Schließt alle Objekte von der Verarbeitung aus, deren Speicher- oder Katalogisierungsdatum und/oder -zeitpunkt innerhalb des in diesen Feldern angegebenen Bereichs liegt, indem ein genaues Startdatum und/oder eine genaue Startzeit und/oder ein Enddatum und/oder eine Endzeit eingegeben werden. Gültige Eingabewerte siehe ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*. Zulässige spezielle Datumsangaben: `TODAY`, `YESTERDAY`, `MONTH` und `YEAR`.
Objektdatum: **Exclude objects modified on**	Schließt alle Objekte von der Verarbeitung aus, deren Speicher- oder Katalogisierungsdatum und/oder -zeit mit dem in diesen Feldern angegebenen Datum/Zeit übereinstimmt, indem ein genaues Datum und/oder eine genaue Zeit eingegeben wird. Gültige Eingabewerte siehe ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*. Zulässige spezielle Datumsangaben: `TODAY` und `YESTERDAY`.
Objektgröße: **Ignore object size**	Es wird keine Prüfung der Größe durchgeführt. Die Objekte werden unabhängig von ihrer Größe verarbeitet.
Objektgröße: **Exclude objects with size between/and**	Schließt alle Objekte von der Verarbeitung aus, deren Größe innerhalb des in diesen Feldern angegebenen Bereichs liegt, indem eine Startgröße und/oder eine Endgröße eingegeben wird.
Objektgröße: **Exclude objects with size**	Schließt alle Objekte von der Verarbeitung aus, deren Größe mit der in diesem Feld angegebenen Größe übereinstimmt.

100 Objektangabe - Natural Systemfehlermeldungen

- Natural-Systemfehlermeldungen 940
- Natural-Systemfehlermeldung, weitere Auswahlkriterien angeben 940
- Natural-Systemfehlermeldungen, Ausnahmen angeben 941

Dieses Kapitel beschreibt die Optionen, die auf den Bildschirmen zur Objektangabe für die Verarbeitung von Natural-Systemfehlermeldungen aus der angegebenen Systemdatei zur Verfügung stehen.

Beschreibungen der Schlüsselwörter und der gültigen Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Natural-Systemfehlermeldungen

Der Bildschirm **Unload/Load/Scan Natural System Error Messages** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
Error number from/to		Ein Bereich von Natural-Systemfehlermeldungen, der durch die erste und die letzte Meldungsnummer begrenzt ist. Wählen Sie **Select system error messages**, um eine Liste aller verfügbaren Systemfehlermeldungen zu erhalten.
Details	PF6	Ruft den Bildschirm **Unload/Load/Scan Natural Library Objects, Details** auf, in dem Sie ausführlichere Objektangaben machen können. Siehe *Natural-Systemfehlermeldung, weitere Auswahlkriterien angeben*.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen angeben können. Siehe *Einstellungen (Settings)*.

Natural-Systemfehlermeldung, weitere Auswahlkriterien angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural System Error Messages** das Feld **Details** markiert haben, wird der Bildschirm **Unload/Load/Scan System Error Messages, Details** angezeigt. Er dient der Angabe weiterer Auswahlkriterien für Natural-Systemfehlermeldungen.

Beschreibung der Schlüsselwörter und der gültigen Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan System Error Messages, Details** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und die Nummer der Adabas-Datei (FNR), in der die Natural-Fehlermeldungen gespeichert sind.
Password/Cipher		Gilt nur für die Entladefunktion. Das Kennwort und der Chiffrierschlüssel für die Adabas-Datei, in der die Quellcodes der Natural-Fehlermeldungen gespeichert sind.
Error number from/to		Siehe **Error number** in *Natural-Systemfehlermeldungen* weiter oben.
Language codes		Siehe **Language codes** in *Natural-Library-Objekte - weitere Auswahlkriterien angeben*.
S/L-Kind		Siehe **S/L-Kind** in *Natural-Library-Objekte - weitere Auswahlkriterien angeben*.
Exceptions	PF8	Ruft einen zusätzlichen Bildschirm auf, in dem Sie Ausnahmen von der Auswahl der Natural-Systemfehlermeldungen angeben können. Siehe ***Natural-Systemfehlermeldungen, Ausnahmen angeben***.

Natural-Systemfehlermeldungen, Ausnahmen angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural System Error Messages, Details** das Feld **Exceptions** markiert haben oder PF8 (Excep) wählen, wird der Bildschirm **Unload/Load/Scan System Error Messages, Exceptions** angezeigt. Er dient dazu, Ausnahmen von der Auswahl der Natural-Systemfehlermeldungen anzugeben.

Alle Fehlermeldungen des Natural-Systems, die den in ***Natural System Error Messages*** und ***Natural System Error Message, Details*** angegebenen Auswahlkriterien entsprechen, werden gegen die im Bildschirm **Unload/Load/Scan System Error Messages, Exceptions** gemachten Angaben geprüft. Fehlermeldungen, die *allen* als Ausnahmen definierten Angaben entsprechen, werden von der Verarbeitung ausgenommen.

Erläuterungen zu den Feldern des Bildschirms Ausnahmen siehe `select-clause` im Kapitel *Direktkommandos*.

101 Objektangabe - Natural-Kommandoprozessoren

- Natural-Kommandoprozessoren entladen/laden/durchsuchen 944
- Natural-Kommandoprozessor-Quellcodes, Ausnahmen angeben 945

Dieser Abschnitt beschreibt die Optionen, die auf den Bildschirmen zur Objektangabe für die Verarbeitung der Quellcodes von Natural-Kommandoprozessoren zur Verfügung stehen.

Beschreibungen von Schlüsselwörtern und gültigen Eingabewerten siehe `select-clause` im Kapitel *Direktkommandos*.

Natural-Kommandoprozessoren entladen/laden/durchsuchen

Der Bildschirm **Unload/Load/Scan Natural Command Processors** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
Library		Der Name einer Natural-Kommandoprozessor-Library oder einen Namensbereich: siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select library	PF4	Ruft eine Auswahlliste der verfügbaren Natural-Kommandoprozessor-Libraries auf. Siehe auch ***Library auswählen***.
DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und die Dateinummer der Adabas-Datei (FNR), in der die Natural-Kommandoprozessor-Quellcodes gespeichert sind.
Password/Cipher		Gilt nur für die Entladefunktion. Das Passwort und der Chiffrierschlüssel für die Adabas-Datei, in der die Natural-Kommandoprozessor-Quellcodes gespeichert sind.
Object name		Der Name eines Natural-Kommandoprozessor-Quellcodes oder ein Namensbereich. Siehe ***Name***.
Select objects		Gilt nur für die Entladefunktion. Wenn kein Library-Bereich angegeben wurde und dieses Feld markiert ist, wird eine Auswahlliste der verfügbaren Natural-Kommandoprozessor-Quellcodes angezeigt (siehe auch ***Objekte auswählen***).
Exceptions	PF8	Ruft einen zusätzliches Bildschirm auf, in dem Sie Ausnahmen von der Auswahl der Natural-Kommandoprozessor-Quellcodes angeben können. Siehe ***Natural-Kommandoprozessor-Quellcodes, Ausnahmen angeben***.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf. Dort können Sie die Einstellungen für Optionen und Parameter angeben. Siehe ***Einstellungen (Settings)***.

Natural-Kommandoprozessor-Quellcodes, Ausnahmen angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural Command Processors** die Taste PF8 (Excep) gewählt haben, wird der Bildschirm **Unload/Load/Scan Natural Command Processors, Exceptions** angezeigt. Hier können Sie Ausnahmen von der Auswahl der Natural-Kommandoprozessor-Quellcodes angeben.

Alle Objekte, die den unter ***Natural Command Processor Sources*** angegebenen Auswahlkriterien entsprechen, werden gegen die im Bildschirm **Unload/Load/Scan Natural Command Processors, Exceptions** gemachten Angaben geprüft. Natural-Kommandoprozessor-Quellcodes, die alle als Ausnahmen definierten Angaben erfüllen, werden von der Verarbeitung ausgenommen.

Erläuterungen zu den Feldern im Ausnahmen-Fenster siehe unter `select-clause` im Kapitel *Direktkommandos*.

102 Objektangabe - Natural-zugehörige Objekte

- Natural-Profile 948
- Natural-Debug-Umgebungen 949

In diesem Kapitel werden die Optionen beschrieben, die auf den Bildschirmen zur Objektangabe für die Bearbeitung von Natural-zugehörigen Objekten zur Verfügung stehen. Natural-zugehörige Objekte (Natural Related-Objects) sind Profile und Debug-Umgebungen.

Beschreibung der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Wenn Sie Natural-zugehörige Objekte (Natural Related-Objects) auf dem Bildschirm **Unload/Load Type** auswählen, erscheint der Bildschirm **Select Related Type**, auf dem Sie den Typ des Natural-zugehörigen Objekts angeben können: Natural-Profile oder Debug-Umgebungen.

Natural-Profile

Der Bildschirm **Unload/Load/Scan Natural Profiles** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und Dateinummer (FNR) der Adabas-Datei, in der die Natural-Profile gespeichert sind. Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle FNAT-Systemdatei verwendet.
Password/Cipher		Gilt nur für die Entladefunktion. Das Passwort und der Chiffrierschlüssel für die Adabas-Datei, in der die Natural-Profile gespeichert sind.
Select (DBID/FNR)	PF5	Gilt nur für die Entladefunktion. Ruft das Fenster **Select System File** mit einer Liste aller in Ihrer Natural-Umgebung verfügbaren Systemdateien auf. Siehe auch ***Systemdatei auswählen***.
Profile Types		Die Art(en) des zu bearbeitenden Profils: Parameter, Editor, Maske und/oder Gerät.
Object name		Der Name eines Profils oder ein Namensbereich: siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select (Objektname)		Gilt nur für die Entladefunktion. Ruft den Bildschirm **List Profiles** mit einer Auswahlliste der verfügbaren Profile auf (siehe auch ***Objekte auswählen***).
Exceptions	PF8	Ruft einen zusätzlichen Bildschirm auf, auf dem Sie Ausnahmen für die Auswahl von Profilen angeben können. Siehe ***Natural-Profile, Ausnahmen angeben***.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen festlegen können. Siehe ***Einstellungen (Settings)***.

Beschreibung der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

- Natural-Profile, Ausnahmen angeben

Natural-Profile, Ausnahmen angeben

Wenn Sie im Bildschirm **Unload/Load/Scan Natural Profiles** die Taste PF8 (Excep) gewählt haben, wird der Bildschirm **Unload/Load/Scan Natural Profiles, Exceptions** angezeigt. Hier können Sie Ausnahmen von der Auswahl von Natural-Profilen angeben.

Alle Objekte, die den im Abschnitt ***Natural Profiles*** angegebenen Auswahlkriterien entsprechen, werden gegen die Angaben im Bildschirm **Unload/Load/Scan Natural Profiles, Exceptions** geprüft. Objekte, die *allen* als Ausnahmen definierten Angaben entsprechen, werden von der Verarbeitung ausgenommen.

Beschreibung der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan Natural Profiles, Exceptions** enthält die folgenden Felder:

Feld	Erläuterung
Object name	Der Name eines Profils oder ein Namensbereich. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Profile Types	Die Art(en) des zu bearbeitenden Profils: Parameter, Editor, Maske und/oder Gerät.

Natural-Debug-Umgebungen

Der Bildschirm **Unload/Load/Scan Natural Debug Environments** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
Library		Der Name einer Library oder ein Namensbereich. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select library	PF4	Zeigt eine Auswahlliste mit allen verfügbaren Libraries an. Siehe auch ***Library auswählen***.
DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und Dateinummer (FNR) der Adabas-Datei, in der die Natural-Debug-Umgebungen gespeichert sind.

Feld	PF-Taste	Erläuterung
		Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle FNAT-Systemdatei verwendet.
Password/Cipher		Gilt nur für die Entladefunktion. Passwort und Chiffrierschlüssel für die Adabas-Datei, in der die Natural-Debug-Umgebungen gespeichert sind.
Object name		Der Name einer Debug-Umgebung oder ein Namensbereich. Siehe ***Name***.
Select objects		Gilt nur für die Entladefunktion. Zeigt eine Auswahlliste der verfügbaren Debug-Umgebungen an (siehe auch ***Objekte auswählen***).
Exceptions	PF8	Ruft einen zusätzlichen Bildschirm auf, auf dem Sie Ausnahmen von der Auswahl der Profile angeben können. Siehe ***Natural-Debug-Umgebungen, Ausnahmen angeben***.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen festlegen können. Siehe ***Einstellungen (Settings)***.

Beschreibungen von Schlüsselwörtern und gültigen Eingabewerten siehe `select-clause` im Kapitel *Direktkommandos*.

- Natural-Debug-Umgebungen, Ausnahmen angeben

Natural-Debug-Umgebungen, Ausnahmen angeben

Wenn Sie im Bildschirm **Unload Natural Debug Environments** die Taste PF8 (Excep) gewählt haben, wird der Bildschirm **Unload Debug Environments, Exceptions** angezeigt. Hier können Sie Ausnahmen von der Auswahl der Natural-Debug-Umgebungen anzugeben.

Alle Objekte, die den im Bildschirm ***Natural Debug Environments*** angegebenen Auswahlkriterien entsprechen, werden gegen die im Bildschirm **Unload/Load/Scan Debug Environments, Exceptions** gemachten Angaben geprüft. Objekte, die allen als Ausnahmen definierten Spezifikationen entsprechen, werden von der Verarbeitung ausgenommen.

Beschreibung der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan Debug Environments, Exceptions** enthält die folgenden Felder:

Feld	Erläuterung
Library	Der Name einer Library oder ein Namensbereich siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select (Library)	Zeigt eine Auswahlliste mit allen verfügbaren Libraries an. Siehe auch ***Library auswählen***.
Object name	Der Name einer Debug-Umgebung oder ein Namensbereich. Siehe auch ***Name***.

103 Objektangabe - Datendefinitionsmodule (DDMs)

- DDMs entladen/laden/durchsuchen 954
- DDM-Eigenschaften angeben 955
- DDM-Ausnahmen angeben 956

Dieses Kapitel beschreibt die Optionen, die auf den Bildschirmen zur Objektangabe für die Verarbeitung von Natural-Datendefinitionsmodulen (DDMs) zur Verfügung stehen.

Beschreibungen der Schlüsselwörter und gültigen Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

DDMs entladen/laden/durchsuchen

Wenn Sie im Bildschirm **Select Unload/Load/Scan Type** die Option **DDMs** gewählt haben, wird der Bildschirm **Unload/Load/Scan DDMs** angezeigt. Er enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
FDIC DBID/FNR		Gilt nur für die Entladefunktion. Die Datenbankkennung (DBID) und die Dateinummer der Adabas-Datei, in der die DDMs gespeichert sind. Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle FDIC-Systemdatei verwendet.
FDIC Password/Cipher		Gilt nur für die Entladefunktion. Das Passwort und der Chiffrierschlüssel für die Adabas-Datei, in der die DDMs gespeichert sind.
DDM name		Gilt nur für die Entladefunktion. Der Name eines DDMs oder ein Namensbereich. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
Select objects		Gilt nur für die Entladefunktion. Zeigt eine Auswahlliste der verfügbaren DDMs an (siehe auch ***Objekte auswählen***).
Properties	PF7	Ruft einen zusätzlichen Bildschirm auf, in dem Sie zusätzliche Eigenschaften von DDMs angeben können. Siehe ***DDM-Eigenschaften angeben***.
Exceptions	PF8	Ruft einen zusätzlichen Bildschirm auf, auf dem Sie Ausnahmen von der Auswahl von DDMs angeben können. Siehe ***DDM-Ausnahmen angeben***.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen festlegen können. Siehe ***Einstellungen (Settings)***.

DDM-Eigenschaften angeben

Wenn Sie im Bildschirm **Unload/Load/Scan DDMs** die Taste PF7 (Prope) gewählt haben, wird der Bildschirm **Unload/Load/Scan DDMs, Properties** angezeigt. Hier können Sie Eigenschaften für die zur Verarbeitung ausgewählten DDMs angeben.

Beschreibungen der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan DDMs, Properties** enthält die folgenden Felder:

Feld	Erläuterung
User ID	Die Benutzerkennung des Benutzers, der ein DDM gespeichert oder katalogisiert hat. Geben Sie eine einzelne Benutzerkennung oder einen Bereich von Benutzerkennungen an. Siehe ***Name*** in *Angabe von Name, Datum und Zeit*.
DDM DBID	Die Datenbankkennung (DBID) des DDMs. Gültige Einträge sind: `1` bis `65535` oder `0` (alle DBIDs).
DDM FNR	Die Dateinummer (FNR) der DDMs. Gültige Einträge sind: `1` bis `65535` oder `0` (alle FNRs).
Objektdatum: **Select all objects (no date check)**	Wählt alle DDMs aus, unabhängig von ihrem Datum.
Objektdatum: **Select objects modified between/and**	Siehe **Object Date** in *Eigenschaften von Natural-Library-Objekten*.
Objektdatum: **Select objects modified on**	Siehe **Object Date** in *Eigenschaften von Natural-Library-Objekten*.
Objektgröße: **Select all objects (no size check)**	Wählt alle DDMs aus, unabhängig von ihrer Größe.
Objektgröße: **Select objects with size between/and**	Wählt alle DDMs mit einer Größe innerhalb des in diesen Feldern angegebenen Bereichs aus, wenn eine Startgröße und/oder eine Endgröße eingegeben wird.
Objektgröße: **Select objects with size**	Wählt alle DDMs mit einer Größe aus, die der in diesem Feld angegebenen Größe entspricht.

DDM-Ausnahmen angeben

Wenn Sie im Bildschirm **Unload/Load/Scan DDMs** die Taste PF8 (Excep) gewählt haben, wird der Bildschirm **Unload/Load/Scan DDMs, Exceptions** angezeigt. Hier können Sie Ausnahmen von der Auswahl von DDMs angeben.

Alle Objekte, die den in *DDMs* und *DDM Properties* angegebenen Auswahlkriterien entsprechen, werden gegen die Angaben auf dem Bildschirm **Unload/Load/Scan DDM, Exceptions** geprüft. Objekte, die *allen* als Ausnahmen definierten Angaben entsprechen, werden von der Verarbeitung ausgenommen. Beschreibungen der Schlüsselwörter und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Der Bildschirm **Unload/Load/Scan DDMs, Exceptions** enthält die folgenden Felder:

Feld	Erläuterung
DDM name	Der Name eines DDM oder ein Bereich von Namen. Siehe *Name* in *Angabe von Name, Datum und Zeit*.
DDM DBID	Siehe **DDM DBID** in *DDM Properties*.
DDM FNR	Siehe **DDM FNR** in *DDM Properties*.
User ID	Siehe **User ID** in *DDM Properties*.
Objektdatum: **Ignore object date**	Es erfolgt keine Datumsprüfung. DDMs werden unabhängig von ihrem Datum verarbeitet.
Objektdatum: **Exclude objects modified between/and**	Siehe **Object Date** in *Natural Library Object Exception Properties*.
Objektdatum: **Exclude objects modified on**	Siehe **Object Date** in *Natural Library Object Exception Properties*.
Objektgröße: **Ignore object size**	Es erfolgt keine Größenprüfung. DDMs werden unabhängig von ihrer Größe verarbeitet.
Objektgröße: **Exclude objects with size between/and**	Schließt alle DDMs von der Verarbeitung aus, deren Größe innerhalb des in diesen Feldern angegebenen Bereichs liegt, wenn eine Startgröße und/oder eine Endgröße eingegeben wird.
Objektgröße: **Exclude objects with size**	Schließt alle DDMs von der Verarbeitung aus, deren Größe mit der in diesem Feld angegebenen Größe übereinstimmt.

104 Objektangabe - Felddefinitionstabellen (FDTs)

In diesem Abschnitt werden die Optionen beschrieben, die auf dem Bildschirm zur Objektangabe für die Verarbeitung von Adabas FDTs (Field Definition Tables) zur Verfügung stehen. Ausführliche Informationen zu den Adabas FDTs finden Sie in der *Adabas for Mainframes*-Dokumentation unter *Concepts and Facilities* > *Adabas Design* > *Record and Field Definitions*.

Beschreibungen von Schlüsselwörtern und gültige Eingabewerte siehe `select-clause` im Kapitel *Direktkommandos*.

Wenn Sie im Bildschirm **Select Unload/Load/Scan Type** die Option **FDTs** gewählt haben, wird der Bildschirm **Unload/Load/Scan FDTs** angezeigt. Er enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
DBID		Die Datenbankkennung, in der sich die FDT befindet. Laden und Scannen: Eine gültige Datenbankkennung (DBID) oder 0 für alle DBIDs.
FNR		Die Nummer der Datei, in der sich die FDT befindet. Laden und Scannen: Eine gültige Dateinummer (FNR) oder 0 für alle FDTs.
Password/Cipher		Gilt nur für die Funktionen Entladen und Laden. Das Adabas-Passwort und der Chiffrierschlüssel der Adabas-Datei, in der sich die FDT befindet.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen festlegen können. Siehe ***Einstellungen (Settings)***.

105 Arbeitsplan-Typ SELECTION oder LIST verwenden

Die Option **Use Selection or List Workplan** wird verwendet, um einen Arbeitsplan vom Typ SELECTION oder LIST anzugeben. Diese Arbeitspläne geben Auswahlkriterien für die zu bearbeitenden Objekte an. Siehe auch den Abschnitt *Arbeitspläne (Workplans)*.

Wenn Sie im Bildschirm **Select Unload/Load/Scan Type** die Option **Use Selection or List Workplan** gewählt haben, wird der Bildschirm **Unload/Load/Scan Selection or List** angezeigt. Er enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
Name	PF4	Der Name des zu verarbeitenden Arbeitsplans.
List Workplan		Zeigt den Inhalt des im Feld **Name** angegebenen Arbeitsplans an.
Select Workplan	PF5	Zeigt eine Liste aller verfügbaren Arbeitspläne an. Siehe auch *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* im Kapitel *Administration*.
Settings	PF7	Ruft den Bildschirm **Unload/Load/Scan Settings** auf, in dem Sie Options- und Parametereinstellungen festlegen können. Siehe *Einstellungen (Settings)*.

106 Einstellungen (Settings)

- Felder auf dem Settings-Bildschirm 962
- Zusätzliche Optionen festlegen (Set Additional Options) 964
- Globale Parameter festlegen 974

Die Option **Settings** (Einstellungen) wird verwendet, um Optionseinstellungen für die Funktion **Unload**, **Load**, **Find** oder **Scan** und Parametereinstellungen für die Funktion **Unload** oder **Load** festzulegen.

Um den Bildschirm Unload/Load/Scan Settings aufzurufen:

- Geben Sie in einem der Bildschirme für die Funktion **Unload**, **Load**, **Find** oder **Scan** das folgende interne Kommando ein:

  ```
  SETTINGS
  ```

  Siehe auch *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos*.

  Oder:

  Aktivieren Sie den Expertenmodus (Advanced User), wählen Sie eine Funktion und drücken Sie ENTER, um den Verarbeitungsvorgang zu starten.

  Oder:

  Wählen Sie auf Bildschirmen für den Expertenmodus PF7 (Setti).

Um die Optionen auf dem unten beschriebenen Bildschirm **Unload/Load/Scan Settings** zu aktivieren, müssen Sie das entsprechende Eingabefeld mit einem beliebigen Zeichen markieren, sofern es nicht standardmäßig markiert ist.

Felder auf dem Settings-Bildschirm

Der Bildschirm **Unload/Load/Scan Settings** enthält die folgenden Felder und PF-Tasten:

Feld	PF-Taste	Erläuterung
Transfer format		Gilt nur, wenn **Use default options** markiert wurde (dies ist die Standardeinstellung). Wenn diese Option markiert ist, werden die zu verarbeitenden Daten im Transferformat in die/aus der Arbeitsdatei geschrieben/gelesen. Siehe auch *Arbeitsdateiformat* in *Arbeitsdateien (Work Files)*. Entlade-Funktion: Die zu entladenden Daten werden im Übertragungsformat in die Arbeitsdatei geschrieben. Wenn Sie die Einstellung dieses Feldes für ein anschließendes Entladen ändern wollen, müssen Sie zum Hauptmenü (**Main Menu**) zurückkehren oder das Kommando `GO UNLOAD END` eingeben (siehe *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos*) und die Entladefunktion erneut starten.

Feld	PF-Taste	Erläuterung
		Lade- und Scan-Funktionen: Die zu ladenden oder zu scannenden Daten werden im Transfer-Format erwartet.
Unicode work file		Gilt nur für die Entladefunktion und wenn **Transfer format** markiert wurde. Wenn diese Option markiert ist, werden alle Objektquellen in Unicode/UTF-8 (Universal Transformation Format, 8-Bit-Form) konvertiert, bevor sie in die Arbeitsdatei geschrieben werden. Wenn eine Unicode-Arbeitsdatei angegeben ist, können Sie die Übertragungsoptionen **Use conversion table**, **Substitute line references** und **Incorporate free rules** nicht verwenden.
Use PC File		Gilt nur, wenn Entire Connection installiert ist. Wenn aktiviert, werden die zu verarbeitenden Daten aus einer Entire Connection-Arbeitsdatei gelesen oder in diese geschrieben.
PC File		Gilt nur, wenn Entire Connection installiert ist. Der vollständige Pfadname zur Arbeitsdatei von Entire Connection. Wenn Ihre Systemumgebung keinen Backslash (\) als Trennzeichen akzeptiert, können Sie auch einen Schrägstrich (/) verwenden.
Use default options		Es werden die Standardoptionen verwendet (dies ist die Standardeinstellung). Siehe auch *Profileinstellungen* und *Zusätzliche Optionen festlegen (Set Additional Options)*.
Set additional options	PF4	Gilt nur, wenn **Use default options** markiert wurde (dies ist die Standardeinstellung). Ruft den Bildschirm **Options** auf. Dort können Sie die Standardeinstellungen ändern und zusätzliche Optionen für die Verarbeitungsreihenfolge eingeben. Die verfügbaren Optionen finden Sie unter *Zusätzliche Optionen festlegen (Set Additional Options)*.
Use Option Workplan		Es wird ein Arbeitsplan vom Typ OPTION verwendet. Siehe auch *Arbeitspläne (Workplans)*.
Name (neben **Use Option Workplan**)		Gilt nur, wenn **Use Option Workplan** markiert wurde. Der Name eines zu verwendenden Arbeitsplans vom Typ OPTION.
List Option Workplan	PF6	Gilt nur, wenn **Use Option Workplan** markiert wurde. Zeigt den Inhalt des Arbeitsplans an, der im Feld **Name** neben **Use Option Workplan** angegeben ist.
Select Option Workplan	PF5	Gilt nur, wenn **Use Option Workplan** markiert wurde. Zeigt eine Auswahlliste der verfügbaren Arbeitspläne vom Typ OPTION an (siehe auch *Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek* in *Administration*).

Feld	PF-Taste	Erläuterung
Do not use parameters		Keine Parameter verwenden. Wenn dieses Feld markiert ist (Standardeinstellung), werden keine Parameter gesetzt.
Use global parameters		Globale Parameter verwenden. Es werden globale Parameter verwendet. Siehe ***Globale Parameter festlegen***.
Set global parameters	PF7	Gilt nur, wenn **Use global parameters** markiert wurde. Ruft den Bildschirm **Parameters** auf. Beschreibungen der Schlüsselwörter und gültigen Eingabewerte siehe ***Globale Parameter festlegen*** und `parameter-setting` (im Kapitel *Direktkommandos*).
Use Parameter Workplan		Es wird ein Arbeitsplan vom Typ PARAMETER verwendet. Siehe auch ***Arbeitspläne (Workplans)***.
Name (neben **Use Parameter Workplan**)		Gilt nur, wenn **Use Parameter Workplan** markiert wurde. Der Name eines zu verwendenden Arbeitsplans vom Typ PARAMETER.
List Parameter Workplan	PF9	Gilt nur, wenn **Use Parameter Workplan** markiert wurde. Zeigt den Inhalt des Arbeitsplans an, der im Feld **Name** neben **Use Parameter Workplan** angegeben ist.
Select Parameter Workplan		Gilt nur, wenn **Use Parameter Workplan** markiert wurde. Zeigt eine Auswahlliste der verfügbaren Arbeitspläne vom Typ PARAMETER an. Siehe auch ***Liste der verfügbaren Arbeitspläne in der Arbeitsplan-Bibliothek*** im Kapitel *Administration*.

Zusätzliche Optionen festlegen (Set Additional Options)

Im Folgenden werden spezielle Abschnitte des Bildschirms **Options** beschrieben, die in Abhängigkeit von der verwendeten Funktion, den gewählten Einstellungen und den installierten Produkten angezeigt werden.

Beschreibung der Schlüsselwörter und gültigen Eingabewerte siehe `option-setting` im Kapitel *Direktkommandos*.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Arbeitsdatei- und Report-Optionen
- XREF-Optionen
- Überlegungen bezüglich XRef
- Versionsprüfung

- Übertragungsoptionen
- Ersetzungsoptionen (Replace options)
- Zu verarbeitende Anzahl (Number to process)
- FDIC-Einstellungen (FDIC settings)
- FSEC settings: FSEC-Einstellungen

Arbeitsdatei- und Report-Optionen

In der folgenden Tabelle werden die Optionen erläutert, die für Arbeitsdateien und Reports zur Verfügung stehen.

Feld	Erläuterung
Use PC File	PC-Datei verwenden. Gilt nur, wenn Entire Connection installiert ist. Wenn diese Option ausgewählt ist, werden die zu verarbeitenden Daten aus einer Entire Connection-Arbeitsdatei gelesen oder in eine solche geschrieben.
PC File	PC-Datei. Gilt nur, wenn Entire Connection installiert ist. Der vollständige Pfadname, der der Entire Connection-Arbeitsdatei zugewiesen ist. Wenn Ihre Systemumgebung keinen Backslash (\) als Trennzeichen akzeptiert, können Sie auch einen Schrägstrich (/) verwenden.
Unicode work file	Unicode-Arbeitsdatei. Gilt nur für die Entlade-Funktion und wenn **Transfer format** markiert wurde. Wenn diese Option markiert ist, werden alle Objekt-Quellcodes in Unicode/UTF-8 (Universal Transformation Format, 8-Bit-Form) konvertiert, bevor sie in die Arbeitsdatei geschrieben werden. Wenn eine Unicode-Arbeitsdatei angegeben ist, können Sie die Übertragungsoptionen **Use conversion table**, **Substitute line references** und **Incorporate free rules** nicht verwenden.
Write report	Report schreiben. Schreibt einen Report über die verarbeiteten Objekte in das im Feld **Report text member** angegebene Report-Text-Member. Die Option **Write report** ist standardmäßig markiert. Um den Report anzuzeigen, geben Sie das interne Kommando `SHOW REPORT FILE` ein (siehe ***Kommandos für die Navigation und für spezielle Funktionen*** im Kapitel *Direktkommandos*).

Feld	Erläuterung
Start new report	Gilt nur, wenn **Write report** markiert wurde. Löscht den Inhalt des Report-Text-Member, bevor ein neuer Report geschrieben wird. Andernfalls wird ein neuer Report an den bestehenden Report angehängt.
Error report only	Gilt nur, wenn **Write report** markiert wurde. Es werden nur Fehlermeldungen in den Report geschrieben. Dazu gehören Meldungen von Natural Security und Meldungen, die bei der Ausführung eines `LOAD`-Kommandos aufgetreten sind, z.B. „not replaced“ (nicht ersetzt). Siehe auch `REPORT-OPTION-1` im Kapitel *Direktkommandos*, *option-setting*.
Change report library	Gilt nur, wenn **Write report** markiert wurde. Öffnet ein neues Fenster, in dem Sie die Report Library ändern können. Siehe auch ***Report Library ändern*** (im Kapitel *Administration*) und `REPORT-LIBRARY` (im Kapitel *Direktkommandos*, *option-settings*).
Report text member	Gilt nur, wenn **Write report** markiert wurde. Der Name des in der Arbeitsplan-Bibliothek gespeicherten Text-Members, in das der Report geschrieben wird.
Select text member	Zeigt eine Liste aller in der Arbeitsplan-Bibliothek gespeicherten Text-Members an. Aus dieser Liste können Sie ein Report-Text-Member auswählen.
Write restart information	Gilt nur für die Ladefunktion. Wenn diese Option markiert ist, werden Neustartinformationen für die Funktion **Restart load** bereitgestellt. Ausführliche Informationen finden Sie unter ***Ladefunktion erneut starten (Restart Load)*** im Kapitel *Funktionen*.
Restart text member	Gilt nur für die Ladefunktion und wenn die Option **Write restart information** markiert wurde. Der Name des Text-Members in der Arbeitsplan-Bibliothek, in die die Restart-Information geschrieben wird. Wenn Sie keinen Namen angeben, generiert der Object Handler einen Namen und weist ihn dem Text-Member zu.
Select text member	Text-Member auswählen. Zeigt eine Liste aller in der Arbeitsplan-Bibliothek gespeicherten Text-Member an. Aus dieser Liste können Sie ein **Restart text member** für den Neustart auswählen.
Delete allowed	Löschen erlaubt. Gilt nur für die Ladefunktion.

Feld	Erläuterung
	Verarbeitet Löschanweisungen aus Arbeitsdateien beim Laden von Objekten im internen Format.

XREF-Optionen

XREF-Optionen stehen nur beim Entladen oder Laden von Daten im internen Format zur Verfügung, d.h. wenn das Feld **Transfer format** nicht ausgewählt wurde. Um XRef-Daten zu verarbeiten, muss Predict muss installiert sein.

Die angebotenen XREF-Optionen und die Funktionen, für die diese Optionen gelten, werden im folgenden Abschnitt beschrieben.

Feld	Erläuterung	Funktion
Yes (unload XRef data) oder **Yes (load XRef data)**	Entlädt katalogisierte Objekte und deren Cross-Referenzdaten, falls vorhanden. Lädt katalogisierte Objekte und ihre Cross-Referenzdaten, wenn Cross-Referenzen in der Arbeitsdatei vorhanden sind.	Unload Load
No (ignore XRef data)	Es werden keine XRef-Daten verarbeitet.	Unload Load
Force	Lädt katalogisierte Objekte und deren Cross-Referenzdaten nur dann, wenn Cross-Referenzen in der Arbeitsdatei vorhanden sind und wenn Predict-Einträge für die Objekte in der FDIC-Systemdatei vorhanden sind.	Load
Doc	Lädt katalogisierte Objekte und deren Cross-Referenzdaten (falls vorhanden) nur, wenn für die Objekte Predict-Einträge in der FDIC-Systemdatei vorhanden sind.	Load
Special	Lädt katalogisierte Objekte und deren Cross-Referenzdaten (falls vorhanden).	Load

Überlegungen bezüglich XRef

Alle Cross-Referenzdaten (XRef), die in der Predict-Systemdatei gespeichert sind, können mit dem Object Handler verarbeitet werden. Die Option XREF gibt an, ob der Object Handler XRef-Daten verarbeiten soll. XRef-Daten werden immer gelöscht, wenn die Funktion **Delete** (Löschen) oder **Replace** (Ersetzen) für ein katalogisiertes Objekt ausgeführt wird.

Wenn Predict nicht installiert wurde, sollten Sie die Option XREF auf `N` setzen, so dass keine Validierung von Predict-Dateien durchgeführt wird. Wenn die XREF-Option auf `Y` gesetzt ist und die verwendete FDIC-Datei keine gültige Predict-Datei ist, wird eine Fehlermeldung zurückgegeben.

Die Regeln für die Einstellung der Option XREF sind die gleichen wie die von Natural Security auferlegten. In einer Nicht-Security-Umgebung gibt es keine Einschränkungen, siehe die ersten fünf unten beschriebenen Fälle. Wenn jedoch Natural Security aktiv ist, wie im letzten Fall, hängt

die Einstellung der XREF-Option im Object Handler vom Wert der XREF-Option in den Utility-Profilen von Natural Security ab.

Es gilt folgende Einstellungen für XREF zu berücksichtigen.

- XREF auf OFF oder No gesetzt
- XREF auf ON oder Yes oder Force gesetzt
- XREF auf Force gesetzt
- XREF auf Doc gesetzt
- XREF auf Special gesetzt
- XREF-Option mit Natural Security

XREF auf OFF oder No gesetzt

Wenn die Option XREF auf **OFF** oder **No** gesetzt ist, werden keine XRef-Daten verarbeitet. In Situationen, in denen ein katalogisiertes Objekt gelöscht oder ersetzt wird, löscht der Object Handler die XRef-Daten jedoch. Die Ziel-Predict-Systemdatei wird entsprechend den aktuellen Einstellungen der FDIC-Option bestimmt. Die Standardeinstellung ist der Wert, der dem Profilparameter `FDIC` (siehe *FDIC - Predict-Systemdatei* in der *Parameter-Referenz*-Dokumentation) zu Beginn der Natural-Sitzung zugewiesen wurde.

XREF auf ON oder Yes oder Force gesetzt

Wenn die Option XREF auf **Yes** oder **Force** gesetzt ist, werden während der Verarbeitung die folgenden Aktionen angewendet:

- **Unload**
  Entlädt katalogisierte Objekte und deren Cross-Referenzdaten (falls vorhanden).
- **Load**
  Lädt katalogisierte Objekte und deren Cross-Referenzdaten, wenn Cross-Referenzen in der Arbeitsdatei vorhanden sind.

XREF auf Force gesetzt

Gilt nur für LOAD.

Lädt katalogisierte Objekte und ihre Cross-Referenzdaten nur, wenn Cross-Referenzen in der Arbeitsdatei vorhanden sind und wenn Predict-Einträge für die Objekte in der FDIC-Systemdatei vorhanden sind.

Wenn die Option XREF auf Force gesetzt ist, prüft der Object Handler zusätzlich, ob für das katalogisierte Objekt ein Predict-Programmeintrag in der Ziel-Predict-Systemdatei definiert ist. Wenn nicht, wird die Verarbeitung des Objekts abgebrochen.

XREF auf Doc gesetzt

Gilt nur für LOAD.

Wenn die Option XREF auf **Doc** gesetzt ist, prüft der Object Handler, ob für das katalogisierte Objekt ein Predict-Programmeintrag in der Ziel-Predict-Systemdatei definiert ist. Wenn nicht, wird die Verarbeitung des Objekts abgebrochen. Die katalogisierten Objekte, die Cross-Referenzdaten haben, werden mit ihren Cross-Referenzdaten verarbeitet, und die katalogisierten Objekte, die keine haben, werden ebenfalls verarbeitet.

XREF auf Special gesetzt

Gilt nur für LOAD.

Wenn die Option XREF auf **Special** gesetzt ist, gilt der Sonderfall, dass ein Bereich von angegebenen katalogisierten Objekten mit entsprechenden XRef-Daten verarbeitet wird, unabhängig davon, ob alle katalogisierten Objekte Cross-Referenzdaten haben oder nicht: Die katalogisierten Objekte, die Cross-Referenzdaten haben, werden mit ihren Cross-Referenzdaten verarbeitet, und die katalogisierten Objekte, die keine haben, werden ebenfalls verarbeitet.

XREF-Option mit Natural Security

Wenn Natural Security aktiv ist, gilt Folgendes:

- Wenn der Wert der Option XREF in den Utility Profiles von Natural Security `N` oder `S` ist, können Sie für die Option XREF im Object Handler jeden Wert (**OFF**/**No**, **ON**/**Yes**, **Doc**, **Force** oder **Special**) angeben.
- Wenn der Wert der Option XREF in den Utility Profiles von Natural Security `D` ist, können Sie für die Option XREF im Object Handler nur die Werte **Doc** oder **Force** angeben. Wenn Sie **ON**/**Yes** angeben, wird intern der Wert **Force** verwendet. Wenn Sie **OFF**/**No** oder **Special** angeben, wird intern der Wert **Doc** verwendet.
- Wenn der Wert der Option XREF in den Utility-Profilen von Natural Security `Y` ist, können Sie für die Option XREF im Object Handler nur die Werte **ON**/**Yes** oder **Force** angeben. Wenn Sie **Doc** angeben, wird intern der Wert **Force** verwendet, wenn Sie **OFF**/**No** oder **Special** angeben, wird intern der Wert **ON**/**Yes** verwendet.
- Wenn der Wert der XREF-Option in den Utility-Profilen von Natural Security `F` ist, können Sie für die XREF-Option im Object Handler nur den Wert **Force** angeben. Wenn Sie einen anderen Wert angeben, wird intern der Wert **Force** verwendet.

Versionsprüfung

Die Option **Version check** steht nur beim Laden von Daten im internen Format zur Verfügung, d.h. wenn das Feld **Transfer format** *nicht* ausgewählt wurde.

Wenn **Version check** markiert ist, wird die Natural-Version, unter der die Objekte katalogisiert und in die Arbeitsdatei geschrieben wurden, mit der aktuellen Natural-Version verglichen. Objekte, die unter einer höheren Natural-Version als der aktuellen katalogisiert sind, werden abgelehnt.

Übertragungsoptionen

Die Übertragungsoptionen (Transfer Options) stehen nur bei der Verarbeitung von Daten im Transfer-Format zur Verfügung, d.h. wenn das Feld **Transfer format** markiert wurde.

Die zur Verfügung stehenden Übertragungsoptionen und die Funktionen, auf die sie sich beziehen, werden in der folgenden Tabelle erläutert.

Option	Erläuterung	Funktion
Substitute line references	**Zeilenreferenzen ersetzen** Gilt nur, wenn im Quellcode Zeilennummern für Statement-Referenzen verwendet werden. Wenn Zeilennummern als Referenzen im Quellcode verwendet werden, werden die Zeilennummern der referenzierten Zeilen und die Zeilennummernreferenzen durch Labels ersetzt. Die Quellcodes in der Datenbank werden nicht geändert.	Unload
Include line numbers	**Zeilennummern übernehmen** Wenn Sie diese Option wählen, werden die Zeilennummern übertragen. (Standardmäßig werden Zeilennummern in Natural-Objekten nicht übertragen.)	Unload
Incorporate free rules	**Freie Regeln einbinden** Wenn Predict installiert ist, werden Predict-Regeln, die mit einer Map (Maske) verbunden sind, in den Map-Quellcode eingebunden.	Unload
Use conversion table	**Umsetzungstabelle verwenden** **Vorsicht:** Verwenden Sie diese Option nur in besonderen Fällen, z. B. bei einer Nicht-FTP-Übertragung zwischen Umgebungen mit unterschiedlichen Zeichensätzen, bei denen das Übertragungstool keine Umsetzung vornimmt. Unload: Wandelt Daten in das ASCII-Format um, indem die Natural-interne Umsetzungstabelle (**System table**) oder eine vom Benutzer definierten Umsetzungstabelle (**User table**) verwendet wird. Load:	Unload Load

Option	Erläuterung	Funktion
	Wandelt die Daten in das EBCDIC-Format um, indem die Natural-interne Umwandlungstabelle (**System table**) oder eine vom Benutzer definierte Umwandlungstabelle (**User table**) verwendet wird. Beachten Sie, dass dies nur gilt, wenn die Daten in der Arbeitsdatei im ASCII-Format vorliegen oder wenn ein Umwandlungsprogramm angegeben ist (siehe **User table**).	
System table	**Systemtabelle** Gilt nur, wenn **Use conversion table** markiert wurde. Unload: Wandelt Daten in das ASCII-Format um, indem die Natural-interne Umsetzungstabelle verwendet wird. Load: Wandelt Daten in das ASCII-Format um, indem die Natural-interne Umsetzungstabelle verwendet wird.	Unload Load
User table	**Benutzertabelle** Gilt nur, wenn **Use conversion table** markiert wurde. Wurde der Name eines Umsetzungsprogramms in das Feld eingegeben, werden die Daten unter Verwendung des definierten Umsetzungsprogramms in das EBCDIC- oder ASCII-Format umgesetzt. Damit Sie ein individuelles Umsetzungsprogramm angeben können, muss sich das Programm in der Library SYSOBJH oder einer ihrer Steplibs befinden. Siehe die Beispiel-Subprogramme `OTNCONAE` und `OTNCONEA` in der Library SYSOBJH. Wenn kein Umsetzungsprogramm angegeben ist, wird standardmäßig die entsprechende Umsetzungstabelle in der Natural-Datei *NATCONV.INI* für die Funktionen Entladen ([ISO8859_1->EBCDIC]) und Laden ([EBCDIC->ISO8859_1]) verwendet.	Unload Load
Use load code page	**Codepage zum Laden verwenden** Wenn Sie diese Option wählen, erscheint ein Fenster, in dem Sie den Namen der Codepage eingeben können, die für die Ladefunktion verwendet werden soll. Wenn Sie diese Option wählen, werden alle Objektquellcodes, die in UTF-8 in eine Arbeitsdatei entladen werden, mit der angegebenen Codepage umgesetzt, wenn sie in eine Arbeitsdatei geladen werden. Siehe auch **Unicode work file**. Wenn Sie `*CODEPAGE` als Codepage-Name angeben, wird der der Systemvariablen `*CODEPAGE` zugewiesene Wert verwendet (siehe *Systemvariablen*-Dokumentation). Wenn kein Codepage-Name angegeben wird, werden die Quellcodeobjekte mit der beim Entladen verwendeten Codepage umgesetzt. Wenn **Use load code page** angegeben ist, können Sie die Optionen **Use conversion table** und **Translate to upper case** *nicht* verwenden.	Load

<table>
<tr><th>Option</th><th>Erläuterung</th><th>Funktion</th></tr>
<tr><td>Translate to upper case</td><td>In Großbuchstaben umsetzen

Setzt jeden zu ladenden Quellcode in Großbuchstaben um.</td><td>Load</td></tr>
<tr><td>Data area format</td><td>Gilt nur für Datenbereiche.

Gibt das Format an, in dem die Quellcodes des Datenbereichs entladen oder geladen werden sollen. Mögliche Eingabewerte sind:

N — Wandelt Datenbereiche in das neue interne Datenbereichsformat um.

O — Wandelt Datenbereiche in das alte interne Datenbereichsformat um. Falls ein oder mehrere Datenbereichsquellcodes nicht in das alte interne Datenbereichsformat umgewandelt werden können, gibt der Object Handler eine entsprechende Meldung aus, nachdem das Entladen abgeschlossen ist. Außerdem erscheint in der Spalte Status des von der Entladefunktion erzeugten Entladereports ein entsprechender Hinweis neben den Namen der betroffenen Datenbereichsquellcodes.

D — Wandelt Datenbereiche in das <code>DEFINE DATA</code>-Format um. Der einzige Unterschied bei dieser Art der Datenübertragung besteht darin, dass die in der Arbeitsdatei gespeicherten Datendefinitionen in einer besser lesbaren Form vorliegen.

* — Keine Umwandlung von Datenbereichen. Dies ist die Standardeinstellung.

Ausführliche Informationen finden Sie unter Datenbereich-Editor (Data Area Editor) in der Editoren-Dokumentation</td><td>Unload
Load</td></tr>
<tr><td>Unshape Arabic characters</td><td>Arabische Zeichen von geformt in ungeformt umwandeln

Diese Option wird verwendet, um geformte arabische Zeichen (Codepage IBM420) durch die entsprechenden ungeformten Zeichen in den folgenden Quellcodeobjekten zu ersetzen:

Natural-Programme, DDMs, Benutzer-Fehlermeldungen und Systemfehlermeldungen.</td><td>Unload
Load</td></tr>
</table>

Ersetzungsoptionen (Replace options)

Die im Folgenden beschriebenen Ersetzungsoptionen gelten nur für die Ladefunktion:

Do not replace	Es werden keine Objekte ersetzt. Dies ist die Standardeinstellung.
Replace all	Alle Objekte werden ersetzt.
Replace obsolete	Alle Objekte mit einem Datum, das älter ist als das Datum der Objekte in der Ladedatei, werden ersetzt.
Replace except newer	Alle Objekte mit Ausnahme derjenigen, deren Datum neuer ist als das Datum der Objekte in der Ladedatei, werden ersetzt.

Zu verarbeitende Anzahl (Number to process)

Number to process gilt nur für die Funktionen Laden und Scannen.

Geben Sie in das Feld **Number to process** einen Wert mit maximal 5 Ziffern ein. Wenn ein Wert größer als `0` angegeben wird, stoppt die Lade- oder Scan-Funktion, nachdem die angegebene Anzahl von Objekten verarbeitet wurde.



Anmerkung: Wenn ein katalogisiertes Natural-Objekt direkt nach dem gleichnamigen Quellcodeobjekt verarbeitet wird, werden beide als ein Objekt betrachtet.

FDIC-Einstellungen (FDIC settings)

FDIC-Einstellungen werden verwendet, um die Predict-Datei (FDIC) anzugeben, die für die Verarbeitung von XRef-Daten benutzt werden soll (gilt nur, wenn Predict installiert ist) oder die für das Laden von DDMs benutzt werden soll:

DBID	Die Kennung der Datenbank, in der sich die FDIC-Datei befindet.
FNR	Die Dateinummer der Datei, in der sich die FDIC-Datei befindet.
Password	Optional. Das Adabas-Passwort für die Adabas-Datei, in der sich die FDIC-Datei befindet.
Cipher	Optional. Der Chiffrierschlüssel für die Adabas-Datei, in der sich die FDIC-Datei befindet.

FSEC settings: FSEC-Einstellungen

Die FSEC-Einstellungen gelten nur, wenn Natural Security installiert ist.

Mit den FSEC-Einstellungen wird die für die Sicherheitsprüfungen zu verwendende Natural Security-Datendatei (FSEC) festgelegt:

DBID	Die Kennung der Datenbank, in der sich die FSEC-Datei befindet.
FNR	Die Dateinummer der Datei, in der sich die FSEC-Datei befindet.
Password	Optional. Das Adabas-Passwort für die Adabas-Datei, in der sich die FSEC-Datei befindet. Chiffre Optional.
Cipher	Optional. Der Chiffrierschlüssel für die Adabas-Datei, in der sich die FSEC-Datei befindet.

Globale Parameter festlegen

Gilt nur für die Funktion Laden oder Entladen.

Wenn Sie im Bildschirm **Unload/Load Settings** die Taste PF7 (PaSet) wählen, wird der Bildschirm **Unload/Load Parameters** angezeigt.

Die Felder auf dem Bildschirm **Unload/Load Parameters** können Sie benutzen, um globale Parametereinstellungen für die mit der Lade- oder Entladefunktion zu bearbeitenden Objekte und die Zielumgebung für die Ladefunktion zu ändern. Sie können z.B. neue Namen (oder Namensbereiche) angeben, unter denen die markierten Objekte in die Arbeitsdatei entladen werden, oder Sie können eine andere Library angeben, in die die markierten Objekte aus der Arbeitsdatei geladen werden.

Wenn während der Ausführung der Entladefunktion globale Parameter angegeben werden, wirken sich die Parametereinstellungen auf die Objekte aus, bevor diese in die Arbeitsdatei geschrieben werden. Wenn sie während der Ausführung der Ladefunktion angegeben werden, wirken sich die Parametereinstellungen auf die Objekte aus, bevor sie in die Zielumgebung geschrieben werden.

Die Werte, die zum Ändern von Parametereinstellungen angegeben werden können, können Sie in den Feldern neben dem betreffenden **Parameter** in der Spalte **Check Value** bzw. **New Value** eingegeben.

Wenn in **Check Value** kein Wert eingegeben wurde, wirkt sich der in **New Value** eingegebene Wert auf alle Objekte aus, für die die spezifische Parametereinstellung gilt. Wurde in **Check Value** ein Wert eingegeben, wirkt sich der in **New Value** eingegebene Wert nur auf die Objekte aus, für die die spezifische Parametereinstellung und der in **Check Value** eingegebene Wert gelten. Wenn ein **Check Value** oder **New Value** für den Typ des zu bearbeitenden Objekts nicht relevant ist, wird jeder in eines der beiden Felder eingegebene Wert ignoriert. Zum Beispiel: Natural-Systemfehlermeldungen haben keinen Library-Namen. Daher wird bei der Verarbeitung von Fehlermeldungen des Natural-Systems ein in das Feld **Check Value** oder **New Value** eingegebener Wert für das Feld **Library** ignoriert.

Check Value und **New Value** gelten nicht für den Parameter **Error number difference** und die im Abschnitt **System files for load** des Bildschirms **Parameters** enthaltenen Parameter.

Gültige Parametereinstellungen siehe *parameter-setting* im Kapitel *Direktkommandos*.

Die folgenden Felder sind im Bildschirm **Unload/Load Parameters** vorhanden:

Feld/Abschnitt	Erläuterung
Object name	**Check Value/New Value:** Prüfwert/Neuer Wert: Ein einzelner Objektname oder ein Namensbereich. Siehe ***Name*** in *Angabe von Name, Datum und Zeit* und ***Regeln für neue Werte***. **Anmerkung:** Gilt nicht für DDMs auf Großrechner-Plattformen.
Library	**Check Value/New Value:** Ein einzelner Library-Name oder ein Namensbereich. Siehe ***Name*** und ***Regeln für neue Werte***.
Date	**Check Value/New Value:** Ein einzelnes Datum oder ein Datumsbereich. Siehe ***Datum*** in *Angabe von Name, Datum und Zeit* und ***Regeln für neue Werte***.
Time	**Check Value/New Value:** Eine Zeit oder ein Zeitbereich. Siehe ***Zeit*** in *Angabe von Name, Datum und Zeit* und ***Regeln für neue Werte***.
User ID	**Check Value/New Value:** Eine einzelne Benutzerkennung oder ein Bereich von Benutzerkennungen. Siehe ***Name*** und ***Regeln für neue Werte***.
Terminal ID	**Check Value/New Value:** Eine einzelne Terminalkennung oder ein Bereich von Terminalkennungen. Siehe ***Name*** und ***Regeln für neue Werte***.
Lang. codes	**Sprachcodes:** Gilt nur für die Verarbeitung von Natural-Systemfehlermeldungen oder benutzerdefinierten Fehlermeldungen. **Check Value/New Value:** Bis zu 8 gültige Sprachcodes, z. B. Code 4 für Spanisch. Wenn mehr als ein Sprachcode angegeben wird, muss **Check Value** die gleiche Anzahl an Sprachcodes enthalten. In diesem Fall wird der Sprachcode in **Check Value** durch den Sprachcode im entsprechenden **New Value** ersetzt. New Value gilt nicht für die Langtexte von Natural-Systemfehlermeldungen, für die Englisch (Code 1) die einzige gültige Sprache ist.
Error number difference	**Fehlernummern-Differenz:** Gilt nur bei der Verarbeitung von Natural-Systemfehlermeldungen oder benutzerdefinierten Fehlermeldungen.

Feld/Abschnitt	Erläuterung
	Ein vierstelliger positiver oder negativer Wert (+/-*nnnn*), der als neuer Nummernkreis für Fehlermeldungen verwendet werden soll. Start- und Endwerte müssen in den Feldern **Error number from/to** (siehe *Natural-Library-Objekte*) angegeben werden, um zu validieren, ob der neue Bereich auf die ausgewählten Fehlermeldungen angewendet werden kann. Beispiel: Wenn in **Error number from/to** die Meldungsnummern 1 bis 10 ausgewählt werden und **Error number difference** auf `2000` gesetzt ist, werden die Meldungen von 2001 bis 2010 neu durchnummeriert. Ein Wert von `-1000` in **Error number difference** würde einen Validierungsfehler verursachen.
FDT DBID/FNR	**Check Value/New Value:** Eine gültige Datenbankkennung (DBID) und/oder Dateinummer (FNR) für Adabas-FDTs.
Systemdateien zum Laden: FNAT laden **DBID** **FNR** **Password** **Cipher**	Gilt nur für die Ladefunktion. Die Datenbankkennung (**DBID**) und Dateinummer (**FNR**) der FNAT-Systemdatei des Ziels. Diese Systemdatei wird für alle Library-Objekte verwendet, deren Library-Name mit `SYS`, aber nicht mit `SYSTEM` beginnt. Geben Sie, falls erforderlich, das Adabas-Passwort (**Password**) und den Chiffrierschlüssel (**Cipher**) für die Systemdatei an.
Systemdateien zum Laden: FUSER laden **DBID** **FNR** **Password** **Cipher**	Gilt nur für die Ladefunktion. Die Datenbankkennung (**DBID**) und die Dateinummer (**FNR**) der FUSER-Systemdatei des Ziels. Diese Systemdatei wird für alle Library-Objekte verwendet, deren Library-Name nicht mit `SYS` beginnt, sowie für die Library `SYSTEM`. Geben Sie, falls erforderlich, das Adabas-Passwort (**Password**) und den Chiffrierschlüssel (**Cipher**) für die Systemdatei an.
Systemdateien zum Laden: FNAT/FUSER laden **Select**	Gilt nur für die Ladefunktion. Ruft das Fenster **Select System File** mit einer Liste aller in Ihrer Natural-Umgebung verfügbaren Systemdateien auf. Siehe ***Systemdatei auswählen***.
Systemdateien zum Laden: NCP laden **DBID** **FNR** **Password** **Cipher**	Gilt nur für die Ladefunktion. Die Datenbankkennung (**DBID**) und Dateinummer (**FNR**) der Adabas-Zieldatei, in die die Natural-Kommandoprozessor-Quellcodes geladen werden sollen. Geben Sie, falls erforderlich, das Adabas-Passwort (**Password**) und den Chiffrierschlüssel (**Cipher**) für die Systemdatei an.

Regeln für neue Werte

Das Folgende gilt für **New Value** für **Object name**, **Library**, **Date/Time**, **User ID** und **Terminal ID**.

Wenn **New Value** einen Bereich mit einem Stern (*) enthält, z. B. `ABC*`, bestimmt die Anzahl der Zeichen vor dem Stern (*) die Anzahl der in **Check Value** zu ersetzenden Zeichen. Dies gilt auch, wenn **Check Value** kürzer ist als der in **New Value** angegebene Bereich (siehe das zweite Beispiel in *Beispiele* unten).

Beispiele:

1. Wenn **Object name** `ABCDEFG` ist und **New Value** auf `ZYX*` gesetzt ist, lautet der resultierende Objektname ZYXDEFG.
2. Wenn **Object name** `AB` ist und **New Value** auf `ZYX*` gesetzt wird, ist der resultierende Objektname ZYX.
3. Wenn **Date/Time** 2005-03-26 ist und **New Value** auf `2006*` gesetzt wird, ist das resultierende Objektdatum 2006-03-26.

107 Arbeitspläne (Workplans)

- Arbeitspläne erstellen, auswählen und ändern 980
- Inhalt von Arbeitsplänen 980
- Beispiele für Arbeitspläne 981
- Arbeitspläne referenzieren 982

Arbeitspläne können verwendet werden, um individuelle Standardprozeduren für die Ausführung von Kommandos, die Auswahl von Objekten und die Einstellung von Parametern oder Optionen zu definieren und so die Funktionsverarbeitung noch weiter zu automatisieren.

Arbeitspläne sind Natural-Objekte vom Typ Text. Arbeitspläne werden standardmäßig in der Library WORKPLAN („Arbeitsplan-Bibliothek“) gespeichert, die sich in der aktuellen Systemdatei FUSER befindet.

Arbeitspläne erstellen, auswählen und ändern

Sie können die Funktion **Administration** benutzen (siehe Kapitel *Administration (Arbeitspläne verwalten)*), um einen Arbeitsplan zu erstellen, einen Arbeitsplan aus einer Liste auszuwählen, einen Arbeitsplan zu ändern oder die Standard-Library für Arbeitspläne zu ändern. Die Standard-Library kann außerdem durch Angabe des Parameters `Workplan-Library` in Ihrem Object Handler Profil geändert werden (siehe *Profileinstellungen*).

Inhalt von Arbeitsplänen

Ein Arbeitsplan besteht aus einer Kopfzeile (die vom Object Handler generiert wird) und einem zugehörigen Anweisungs- oder Textteil. Anweisungsteile enthalten Kommandos des Object Handlers und Parameter- und/oder Optionseinstellungen. Textteile enthalten nur reinen Text. Kopf- und Anweisungs- bzw. Textteil können Kommentare enthalten (z. B. die Kurzbeschreibung des Arbeitsplans), die mit den Begrenzungszeichen `/*` beginnen müssen und auf eine Zeile beschränkt sind.

Es gibt sechs Typen von Arbeitsplänen: PROCEDURE, SELECTION, LIST, PARAMETER, OPTION und TEXT.

Die folgende Tabelle enthält die gültigen Kopfzeilen (die beim Erstellen eines Arbeitsplans außerhalb des Object Handlers eingegeben werden müssen) für die entsprechenden Arbeitsplantypen und erläutert den Inhalt des Anweisungs- oder Textteils. Außerdem enthält sie Referenzen zu den Klauseln, die bei der Angabe von Object Handler-Direktkommandos gelten. Die zur Verfügung stehenden Object Handler-Direktkommandos werden im Kapitel *Direktkommandos* beschrieben.

Gültige Kopfzeilen	Inhalt	Zugehöriges Thema in *Direktkommandos*
`TYPE PROCEDURE`	Eine Object Handler-Kommandoprozedur. Dieser Arbeitsplantyp kann jede Kombination von Object Handler-Kommandos enthalten, die für PROCEDURE verfügbar sind. Geben Sie eine Folge von Kommandos ein, die durch Semikolon (;) getrennt sein müssen.	*Grundlegende Kommandosyntax*
`TYPE SELECTION`	Auswahlkriterien für Objekte. Dieser Arbeitsplantyp kann in Object Handler-Arbeitsplankommandos verwendet werden.	`select-clause`
`TYPE LIST`	Eine Liste mit Objekten. Dieser Arbeitsplantyp kann in Object Handler-Arbeitsplankommandos verwendet werden.	`select-clause` *Objektliste - Arbeitsplantyp LIST*
`TYPE PARAMETER`	Parameter für die Entlade- oder Ladefunktion. Dieser Arbeitsplantyp kann verwendet werden, um Attribute für die zu bearbeitenden Objekte zu ändern, wie z.B. den Namen einer neuen Ziel-Library, in die Objekte geladen werden. `TYPE PARAMETER` kann in Arbeitsplankommandos des Object Handlers verwendet werden.	`parameter-setting`
`TYPE OPTION`	Optionen für die Entlade- oder Ladefunktion, z.B. Reporteinstellungen. Dieser Arbeitsplantyp kann in Arbeitsplankommandos des Object Handlers verwendet werden.	`option-setting`
`TYPE TEXT`	Kommentare oder sonstiger Text, der zu Dokumentationszwecken dienen kann.	Nicht zutreffend.

Beispiele für Arbeitspläne

In der folgenden Tabelle sind einige Beispiele für die in einem Arbeitsplan enthaltenen Anweisungsteile aufgeführt.

Arbeitsplantyp	Anweisung	Erläuterung
PROCEDURE	`FINDLIB * LIB TEST`	Prüfe, ob die Library TEST existiert.
PROCEDURE	`UNLOAD A* LIB TEST`	Entlade aus der Library TEST in die Arbeitsdatei 1 alle Natural-Programmierobjekte, die mit `A` beginnen, und alle benutzerdefinierten Fehlermeldungen.
SELECTION	`* LIB TEST`	Verarbeite alle Objekte aus der Library TEST.
TEXT	`This is a Workplan comment.`	Beliebiger Text.

Beispiele für Arbeitsplaninhalte

Das folgende Beispiel zeigt die Auflistung eines Arbeitsplans vom Typ PROCEDURE, in dem das Kommando `UNLOAD` ausgeführt wird.

```
TYPE PROCEDURE /* VERSION=03.01 NATURAL VERSION=08.02.01 PL=0 AUTHOR=SAG ↩
DATE=2010-07-20 09:40:12
/* unload from library TEST with target library PROD01
UNLOAD * LIB TEST OBJTYPE N
WITH NEWLIBRARY PROD01
WHERE REPORT MYREP01
```

Arbeitspläne referenzieren

Sie können einen Arbeitsplan mit Hilfe von Object Handler-Menüfunktionen oder Direktkommandos referenzieren (siehe auch ***Direktkommandos***).

Die folgende Syntax gilt, wenn auf einen Arbeitsplan mit den im Kapitel ***Direktkommandos*** beschriebenen Object Handler-Direktkommandos verwiesen wird.

```
( workplan-name
  [ LIBRARY library-name ]
  [DBID dbid [FNR fnr ]][NAME vsam-name ]
  [CIPHER cipher ]
  [ { PASSWORD } password ]
  [ { PSW      }          ]
)
```

Die syntaktischen Optionen werden im folgenden Abschnitt erläutert

Bedeutung der Schlüsselwörter

In der folgenden Tabelle werden die Schlüsselwörter und Werte beschrieben, die in der Syntax zur Referenzierung von Arbeitsplänen gelten.

Schlüsselwort	Werte	Standardwert
workplan-name	Der Name des Natural Text Member in der Arbeitsplan-Library, das als Arbeitsplan verwendet werden soll.	Kein Standardwert.
LIBRARY	Der Name der Library, in der sich der Arbeitsplan befindet.	WORKPLAN
DBID	Die Kennung der Adabas-Datenbank, in der sich die Arbeitsplan-Library befindet.	0 (aktuelle FNAT/FUSER)
FNR	Die Nummer der Adabas-Datei, in der sich die Arbeitsplan-Library befindet.	0 (aktuelle FNAT/FUSER)
NAME	Gilt nur für Objekte auf Großrechnern. Der Name einer gültigen VSAM-Datei, in der sich die Arbeitsplan-Library befindet.	leer (aktuelle FNAT/FUSER)
CIPHER	Gilt nur für Objekte auf Großrechnern. Ein 8-stelliger Chiffrierschlüssel.	leer (aktuelle FNAT/FUSER)
PASSWORD	Gilt nur für Objekte auf Großrechnern. Ein 8-stelliges Adabas-Passwort.	leer (aktuelle FNAT/FUSER)

108 Angabe von Name, Datum und Zeit

- Name 986
- Datum 987
- Zeit 989

Um Natural-Library-Objekte, Natural-Kommandoprozessor-Quellcodes, Natural-zugehörige Objekte (Natural Related-Objects) oder Natural-Datendefinitionsmodule (DDMs) auszuwählen, können Sie einen Namen, ein Datum, eine Zeit oder einen Namensbereich, Datumsbereich und Zeitbereich verwenden

Name

Sie können einen Namen oder einen Namensbereich angeben.

In der nachstehenden Liste der Optionen ist *value* (*Wert*) eine beliebige Kombination aus einem oder mehreren Zeichen:

	Eingabe	Ausgewählte Elemente
	value	Alle Elemente mit Namen, die gleich *value* sind.
	* >	Alle Elemente.
	?	Alle Elemente mit einem beliebigen Zeichen für jedes eingegebene Fragezeichen (?).
Führende Zeichen	*value**	Alle Einträge, deren Namen mit *value* beginnen. Beispiel: AB* Ausgewählt: AB, AB1, ABC, ABEZ Nicht ausgewählt: AA1, ACB
Platzhalterwert (Wildcard)	*value*?	Alle Elemente, deren Namen mit *value* beginnen und mit einem beliebigen einzelnen Zeichen für jedes eingegebene Fragezeichen (?) enden. Beispiel: ABC? Ausgewählt: ABCA, ABCZ Nicht ausgewählt: AXC, ABCAA
	value?*value*?	Alle Elemente, die mit *value* in Kombination mit Stern (*) und Fragezeichen (?) in beliebiger Reihenfolge übereinstimmen. Beispiel: A?C*Z Ausgewählt: ABCZ, AXCBBBZ, ANCZ Nicht ausgewählt: ACBZ, ABDEZ, AXCBBBZA
	*value**value*?	
	value*?*value	
Startwert	*value*>	Alle Elemente mit Namen größer oder gleich *value*. Beispiel: AB> Ausgewählt: AB, AB1, BBB, ZZZZZZZ Nicht ausgewählt: AA1, AAB
Endwert	*value*<	Alle Elemente mit Namen kleiner oder gleich *value*. Beispiel: AX<

	Eingabe	Ausgewählte Elemente
		Ausgewählt: AB, AWW, AX Nicht ausgewählt: AXA, AY

Anmerkung: Die Parameterangabe-Option **New Value** erlaubt nur führende Zeichen (Stern-Notation (*)). Siehe *Regeln für neue Werte* in *Globale Parameter festlegen* unter *Einstellungen (Settings)*.

Datum

Innerhalb des Object Handlers werden alle Datumswerte im internationalen Datumsformat angegeben.

Sie können ein Datum, einen Datumsbereich, einen speziellen Datumswert oder einen speziellen Datumswertbereich angeben. Ein Datum muss im Format *YYYY-MM-DD* (*YYYY* = Year (Jahr), *MM* = Month (Monat), *DD* = Day (Tag) angegeben werden.

In der nachstehenden Liste der Optionen steht der unterstrichene Teil eines Schlüsselworts für dessen gültige Abkürzung, und *value* (*Wert*) ist eine beliebige Kombination aus einer oder mehreren Ziffern:

	Eingabewert	Ausgewählte Elemente
Datum	*YYYY-MM-DD*	Alle Elemente mit Datum gleich *YYYY-MM-DD*. Beispiel: 2003-02-15
Führende Zeichen	*value**	Alle Elemente mit Datum, das mit *value* beginnt. Beispiel: 2002* Ausgewählt: 2002-01-01, 2002-12-31 Nicht ausgewählt: 2001-12-31, 2003-01-01
Startwert	*value*>	Alle Elemente mit Datum größer als *value*. Beispiel: 2002-05> Ausgewählt: 2002-05-01, 2002-12-31, 2003-01-01, 2003-12-31 Nicht ausgewählt: 2002-04-31, 2001-12-31 **Spezielle Datumswerte** können als *value* (*Wert*) verwendet werden (siehe unten).
Endwert	*value*<	Alle Elemente mit Datum kleiner als *value*. Beispiel: 2003-02< Ausgewählt: 2002-05-01, 2002-12-31, 2003-01-01, 2003-01-31 Nicht ausgewählt: 2003-02-01, 2003-05-18

	Eingabewert	Ausgewählte Elemente
		Spezielle Datumsangaben können als *value* (*Wert*) verwendet werden (siehe unten).
Spezielle Datumswerte		
TODAY (+/-*nnnn*)		Alle Einträge mit dem Datum des aktuellen Tages. Der Tag kann von +*nnnn* oder -*nnnn* gefolgt werden, wobei *nnnn* maximal 4 Ziffern hat. Das resultierende Datum wird als das Datum des heutigen Tages plus oder minus *nnnn* Tage berechnet. Beispiel: Wenn das aktuelle Datum der 2003-03-01 ist, ergibt TODAY +5 das Datum 2003-03-06.
YESTERDAY		Alle Elemente mit dem Datum des Tages vor dem aktuellen Tag.
MONTH		Alle Elemente mit dem Datumsbereich des aktuellen Monats. Beispiel: Der aktuelle Monat ist 2003-02. Ausgewählt: 2003-02-01, 2003-02-30 Nicht ausgewählt: 2003-03-01
		FMDATE: Beginnt mit dem ersten Tag des aktuellen Monats. TODATE: Endet mit dem letzten Tag des aktuellen Monats. Wenn die Werte von FMDATE und TODATE identisch sind, ist die Auswahl auf einen Tag beschränkt.
YEAR		Alle Elemente mit dem Datumsbereich des aktuellen Jahres. Beispiel: Das aktuelle Jahr ist 2003. Ausgewählt: 2003-01-01, 2002-12-31 Nicht ausgewählt: 2002-31-12
		FMDATE: Beginnt mit dem ersten Tag des aktuellen Jahres. TODATE: Endet mit dem letzten Tag des aktuellen Jahres. Wenn die Werte von FMDATE und TODATE identisch sind, ist die Auswahl auf ein Jahr beschränkt.

Anmerkung: Die Parameterangabe-Option **New Value** erlaubt nur führende Zeichen (Stern-Notation (*)). Siehe ***Regeln für neue Werte*** in *Globale Parameter festlegen* unter *Einstellungen (Settings)*.

Zeit

Sie können eine Zeit oder einen Zeitbereich angeben. Die Zeit muss im Format *HH*:*II*:*SS* (*HH* = Hours (Stunden), *II* = Minutes (Minuten), *SS* = Seconds (Sekunden)) angegeben werden.

In der nachstehenden Liste der Optionen ist *value* (*Wert*) eine beliebige Kombination aus einer oder mehreren Ziffern:

	Eingabewert	Ausgewählte Elemente
Zeit	*HH*:*II*:*SS*	Alle Elemente mit einer Zeit gleich *HH*:*II*:*SS*. Beispiel: `14:15:16`
Führende Zeichen	*value**	Alle Elemente mit einer Zeit, die mit *value* beginnt. Beispiel: `13:*` Ausgewählt: 13:00:00, 13:10:53, 13:59:59 Nicht ausgewählt: 12:59:59, 14:00:00

109 Arbeitsdateien (Work Files)

- Zuweisung von Arbeitsdateien 992
- Arbeitsdateiformat 992

Dieses Kapitel behandelt Arbeitsdateien (Work Files) und gültige Formate, die für die Entlade-, Lade- und Scan-Funktionen des Object Handlers gelten.

Siehe auch ***Arbeitsdatei- und Report-Optionen*** im Kapitel *Einstellungen (Settings)*.

Zuweisung von Arbeitsdateien

In der folgenden Tabelle sind die vom Object Handler verwendeten Arbeitsdateien aufgeführt.

Arbeitsdatei	Erläuterung
Work File 1	Wird für die Funktionen **Unload** (Entladen), **Load** (Laden) und **Scan** (Durchsuchen) verwendet. Enthält die entladenen Daten.
Work File 7	Wird nur verwendet, wenn Entire Connection installiert und **Use PC File** im Bildschirm **Options** markiert ist (siehe auch ***Zusätzliche Optionen festlegen***). Die Arbeitsdatei 7 muss als Entire Connection-Arbeitsdatei definiert sein, um für die Entlade-, Lade- und Scan-Funktionen verwendet werden zu können. Enthält die entladenen Daten.
Work File 8	Wird nur verwendet, wenn Entire Connection installiert und **Use PC File** im Bildschirm **Options** markiert ist (siehe auch ***Zusätzliche Optionen festlegen*** Die Arbeitsdatei 8 muss als Entire Connection-Arbeitsdatei definiert sein, um für die Entlade-, Lade- und Scan-Funktionen verwendet werden zu können. Wird als interne Datei für die Verarbeitung von Entire Connection-Kommandos verwendet. **Anmerkung:** Die Nummer der Arbeitsdatei kann mit der User Exit Routine **OBJHEX03** (siehe ***Condition Codes und User Exit Routinen***) oder mit den Optionen `PCCOMMANDFILENUMBER`, `PCCOM` und `PCCFN` (siehe *`option-setting`*) bei Verwendung eines Arbeitsplans vom Typ PROCEDURE geändert werden.

Arbeitsdateiformat

Es gibt zwei Dateiformate zum Entladen von Objekten aus der Quellumgebung in Arbeitsdateien und zum Laden von Objekten aus Arbeitsdateien in die Zielumgebung: ein internes Format und das Transfer-Format. Für das Übertragen von Binärdaten müssen Arbeitsdateien das interne Format haben. Für das Übertragen von Textdaten müssen die Arbeitsdateien das Transfer-Format haben.

In diesem Abschnitt werden die Formate beschrieben.

- Internes Format

- Transfer-Format

Internes Format

Das interne Format ist ein internes Satzlayout für Arbeitsdateien, die dazu dienen, Natural-Quellcodeobjekte und katalogisierte Objekte, Fehlermeldungen, Natural-Kommandoprozessoren, Adabas Field Definition Tables (FDTs), Natural Data Definition Modules (DDMs) und Natural-zugehörige Objekte (Natural Related Objects) von einer Umgebung in eine andere zu übertragen.

Arbeitsdateien im internen Format können zur Übertragung von Objekten zwischen identischen Plattformen verwendet werden.

Wenn das interne Format aktiviert ist, werden Natural-Objekte aus der Quellumgebung gelesen und mit der Entladefunktion des Object Handlers in eine Natural-Arbeitsdatei geschrieben. Diese Arbeitsdatei kann unter Verwendung von Standard-Dateitransferdiensten in eine andere Umgebung transportiert werden. In der Zielumgebung können die Objekte dann aus der Arbeitsdatei gelesen und mit der Ladefunktion des Object Handlers in das lokale Datei- oder Datenbanksystem geladen werden.

Es wird dringend empfohlen, die Arbeitsdatei 1 mit `RECFM=VB` (variable blocked) zu definieren, weil der Object Handler Sätze mit variabler Länge in die Arbeitsdatei schreibt, wodurch sich die Größe der Arbeitsdatei verringert.

Wenn Sie Arbeitsdatei 1 mit `RECFM=FB` (fixed blocked) definieren, müssen Sie für die Entladefunktion die Option `FIXEDLENGTH` angeben. Andernfalls werden die in die Arbeitsdatei geschriebenen Sätze mit binären Nullen aufgefüllt.



Anmerkung: Arbeitsdateien, die mit dem Dienstprogramm NATUNLD auf dem Server erstellt wurden, müssen im internen Format verarbeitet werden. Die Arbeitsdateien müssen auf einem Server der gleichen Plattform erstellt werden, auf der NATUNLD angewendet wurde.

Transfer-Format

Siehe auch **Transfer format** unter *Einstellungen (Settings)*.

Das Transfer-Format ist ein generelles Satzlayout für Arbeitsdateien, die Lade- oder Entladedaten enthalten. Dieses Format ist plattformunabhängig und kann verwendet werden, um die Quellen von Natural-Objekten, Natural-Kommandoprozessorquellen, Fehlermeldungen, DDMs und Adabas-FDTs von einer Hardware-Plattform auf eine andere und zwischen UNIX-, Mainframe- und Windows-Plattformen zu übertragen.

Wenn die Option **Transfer format** gesetzt ist, liest die Entladefunktion des Object Handlers Natural-Objekte von einer Hardware-Plattform und strukturiert sie anschließend neu.

Die formatierten Datensätze werden in eine Natural-Arbeitsdatei geschrieben, die mit Standard-Dateitransferdiensten auf eine andere Plattform transportiert werden kann. Auf der Zielplattform liest die Ladefunktion des Object Handlers dann die Objekte aus der Arbeitsdatei und lädt sie in das lokale Datei- oder Datenbanksystem. Die aus der Arbeitsdatei gelesenen Objekte werden entsprechend der Struktur der neuen Hardware-Plattform restrukturiert.

Behandlung von Quellcode in Unicode/UTF-8

Das Transfer-Format wird auch verwendet, um Quellcode von Natural-Objekten in Unicode/UTF-8 (Universal Transformation Format, 8-Bit-Form) zu entladen oder zu laden. Wenn Sie die entsprechende Entladeoption angeben (`WORKFILETYPE` auf `UTF-8` im Kommandomodus oder **Unicode work file** im Menümodus), werden alle Objektquellen in eine Arbeitsdatei in UTF-8 entladen. Wenn Sie die entsprechende Ladeoption angeben (`LOAD-CODE-PAGE` im Kommandomodus oder **Use load code page** im Menümodus), werden alle Objektquellcodes in UTF-8 mit der angegebenen Codepage konvertiert, wenn sie in eine Natural-Systemdatei geladen werden.

Arbeitsdateien aus SYSTRANS

Verwenden Sie das Transfer-Format, um Arbeitsdateien zu verarbeiten, die mit dem Dienstprogramm SYSTRANS erstellt wurden. Arbeitsdateien, die in UTF-8 kodierte Objektquellcodes enthalten, können nicht mit SYSTRANS verarbeitet werden.

110 Direktkommandos

Der Object Handler bietet Direktkommandos für die folgenden Zwecke:

- Zum Ausführen einer Object Handler-Funktion wie z.B. das Entladen oder Laden von Objekten im Batch-Modus oder im Direktkommando-Online-Modus ohne Verwendung der Object Handler-Menüs (siehe auch *Batch- oder Direktkommandoaufrufe*).
- Zum Ausführen oder Referenzieren eines Arbeitsplans (siehe auch *Arbeitspläne (Workplans)*).
- Zum Verwenden als Anweisung in einem Arbeitsplan.
- Zum Navigieren zwischen Bildschirmen.
- Zum Ausführen spezieller Funktionen.

Dieses Kapitel beschreibt die grundlegende Kommandosyntax und die einzelnen Klauseln, Parameter- und Optionseinstellungen, die zur Ausführung dieser Aufgaben zur Verfügung stehen. Darüber hinaus werden Beispiele gezeigt, die die Verwendung von Direktkommandos veranschaulichen.

Die in den Syntaxdiagrammen dieses Kapitels verwendeten Symbole sind im Kapitel *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation beschrieben.

In diesem Kapitel werden die folgenden Themen behandelt:

Grundlegende Kommandosyntax

`select-clause`

Objektliste - Arbeitsplantyp LIST

`parameter-setting`

`option-setting`

Beispiele für die Verwendung von Object Handler-Direktkommandos

Kommandos für die Navigation und für spezielle Funktionen

111 Grundlegende Kommandosyntax

Dieses Kapitel beschreibt die Direktkommandos des Object Handlers, die für die Ausführung von Object Handler-Funktionen und Arbeitsplänen vom Typ PROCEDURE vorgesehen sind. Außerdem werden die Kommandos beschrieben, die für die Migration von den alten Utilities NATUNLD/NATLOAD und SYSTRANS zum Object Handler verwendet werden.

Erläuterungen zu den Variablenwerten, die in den Syntaxdiagrammen dieses Abschnitts enthalten sind, finden Sie in den entsprechenden Kapiteln der vorliegenden Object Handler-Dokumentation. Erläuterungen zu den in den Syntaxdiagrammen verwendeten Symbolen finden Sie unter *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation.

```
EXECUTE (procedure-workplan)
```

Führt einen Arbeitsplan vom Typ PROCEDURE aus. Siehe auch ***Arbeitspläne (Workplans)***.

```
UNLOAD select-clause [parameter-setting] [option-setting]
```

Entlädt die in *select-clause* definierten Objekte mit den in *parameter-setting* definierten Parametern mit den in *option-setting* definierten Optionen.

```
LOAD select-clause [parameter-setting] [option-setting]
```

Lädt die in *select-clause* definierten Objekte mit den in der *parameter-setting*-Klausel definierten Parametern mit den in der *option-setting*-Klausel definierten Optionen.

```
LOADALL [parameter-setting] [option-setting]
```

Lädt alle Objekte aus einer Arbeitsdatei mit den in der *parameter-setting*-Klausel definierten Parametern mit den in der *option-setting*-Klausel definierten Optionen.

```
SCAN select-clause [option-setting]
```

Durchsucht eine Arbeitsdatei nach den in *select-clause* definierten Objekten mit den in der *option-setting*-Klausel definierten Optionen.

```
SCANALL [option-setting]
```

Durchsucht eine Arbeitsdatei nach allen Objekten mit den in der *option-setting*-Klausel definierten Optionen.

```
FIND select-clause [option-setting]
```

Sucht die in der *select-clause* definierten Objekte mit den in der *option-setting*-Klausel definierten Optionen und schreibt einen Report über die gefundenen Objekte in ein Natural Text Member, das in der Arbeitsplan-Library gespeichert wird. Zusätzlich kann ein Report über die gefundenen Objekte in eine angegebene Reportdatei als Natural Text Member in der Arbeitsplan-Bibliothek geschrieben werden.

```
FINDLIB select-clause [option-setting]
```

Sucht die in der *select-clause*-Klausel definierten Libraries für Natural-Objekte oder Natural-Kommandoprozessor-Quellcodes mit den in der *option-setting*-Klausel definierten Optionen und schreibt einen Report über die gefundenen Objekte in ein Natural Text Member, das in der Arbeitsplan-Library gespeichert wird. Zusätzlich kann ein Report über die gefundenen Objekte in eine angegebene Report-Datei als ein Natural Text Member in der Arbeitsplan-Library geschrieben werden.

```
DELETE select-clause [option-setting]
```

Löscht die in der *select-clause*-Klausel definierten Objekte mit den in der *option-setting*-Klausel definierten Optionen. **Einschränkung:** Es ist nicht möglich, eine FDT zu löschen.

```
UNDELI  select-clause [option-setting]
```

Entlädt die Löschanweisungen für die in der *select-clause*-Klausel definierten Objekte mit den in der *option-setting*-Klausel definierten Optionen.

```
RESTART  [restart-text-member]
```

Bewirkt die Fortsetzung einer unterbrochenen Ladefunktion. Dies ist nur möglich, wenn während des abgebrochenen Ladevorgangs Informationen in ein Natural Text Member in der Arbeitsplan-Library geschrieben wurden. Siehe auch RESTART unter *option-setting* (im Kapitel *Direktkommandos*) und ***Ladefunktion erneut starten (Restart Load)***.

```
DISPLAY STATISTICS
```

Zeigt Statistikinformationen über die verarbeiteten Objekte an.

```
NATUNLD natunld-direct-command
```

Führt ein Object Handler-Kommando in der Syntax des alten Dienstprogramms NATUNLD aus. Siehe auch ***Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler***.

```
NATLOAD natload-direct-command
```

Führt ein Object Handler-Kommando in der Syntax des alten Dienstprogramms NATUNLD aus. Siehe auch ***Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler***.

```
SYSTRANS systrans-direct-command
```

Führt ein Direktkommando des Object Handlers in der Syntax des alten Dienstprogramms SYSTRANS aus. Siehe auch ***Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler***.

112 select-clause

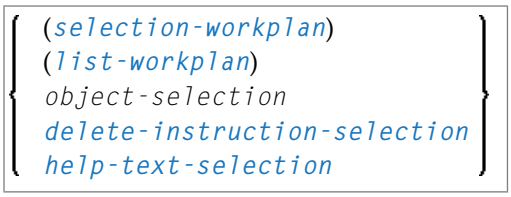
- Syntax der select-clause 1002
- Arbeitsplan vom Typ SELECTION oder LIST 1002
- Auswahl von Natural-Library-Objekten und DDMs 1003
- Auswahl von Natural-Related-Debug-Umgebungen 1010
- Auswahl von Natural-Related-Profilen 1012
- Auswahl von Natural-Systemfehlermeldungen 1013
- Auswahl von Natural-Kommandoprozessoren 1015
- Auswahl von FDTs 1017
- Auswahl von Anwendungen (Applications) 1018
- Auswahl von Löschanweisungen für Objekte 1021
- Auswahl der Hilfetextverarbeitung 1023

Die *select-clause* besteht entweder aus einem Arbeitsplan vom Typ SELECTION oder LIST oder aus Auswahlangaben für die zu verarbeitenden Objekte, FDTs oder Anwendungen.

In diesem Kapitel wird die für die *select-clause* geltende Syntax beschrieben. Die in den Syntaxdiagrammen enthaltenen Schlüsselwörter und Variablenwerte stellen die Parameter dar, die zur Angabe von Objektauswahlkriterien verwendet werden können. Wird ein Variablenwert angegeben, so muss er zusammen mit einem Schlüsselwort eingegeben werden.

Syntax der select-clause

Die *select-clause* besteht aus einer der folgenden Optionen:



Die Optionen *selection-workplan* und *list-workplan* werden unter ***Arbeitsplan vom Typ SELECTION oder LIST*** erläutert.

Die Benutzung der *object-selection*-Klausel zur Objektauswahl ist abhängig von dem Objekttyp, dem DDM, der FDT oder der Anwendung, die Sie verarbeiten möchten. Für jeden dieser Objekttypen werden die entsprechende Syntax und die Schlüsselwörter im weiteren Verlauf dieses Kapitels erläutert.

Die Optionen zur *delete-instruction-selection*-Klausel werden unter ***Auswahl von Löschanweisungen für Objekte*** erläutert.

Die Option *help-text-selection* wird unter ***Auswahl der Hilfetextverarbeitung*** erläutert.

Arbeitsplan vom Typ SELECTION oder LIST

Ein Arbeitsplan vom Typ SELECTION enthält eine Kopfzeile (`TYPE SELECTION`) und eine Auswahl eines der folgenden Objekt- oder Dateitypen: Natural-Library-Objekte, Natural-zugehörige Objekte (Natural Related Objects), Natural-Systemfehlermeldungen, Natural-Kommandoprozessor-Quellcodes, Datendefinitionsmodule (DDMs) oder Adabas Field Definition Tables (FDTs).

Ein Arbeitsplan vom Typ LIST enthält eine Kopfzeile (`TYPE LIST`) und eine Auswahlliste von Objekten, wie im Abschnitt ***Objektliste - Arbeitsplantyp LIST*** beschrieben. Eine solche Objektliste kann nur für die Kommandos `UNLOAD`, `LOAD` oder `FIND` verwendet werden.

Weitere Informationen zur Verwendung von Arbeitsplänen finden Sie im Kapitel ***Arbeitspläne (Workplans)***.

Auswahl von Natural-Library-Objekten und DDMs

Diese Auswahl wird verwendet, um Natural-Objekte zur Verarbeitung auszuwählen, einschließlich Natural-DDMs (Datendefinitionsmodule) und benutzerdefinierte Fehlermeldungen.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax zur Auswahl von Natural-Library-Objekten und DDMs

Syntax zur Auswahl von Natural-Library-Objekten und DDMs

```
object-name
LIBRARY library-name
[   DBID dbid FNR fnr                                                    ]
[   [NAME vsam-name]      [                {PASSWORD} password]      ]
[   [CIPHER cipher]                        {PSW     }                 ]
[OBJTYPE group-type]
[   SETNO set-number [SETUSER set-user] [SETLIBRARY set-library]   ]
[NATTYPE object-type]
[SCKIND object-kind]
[MODE object-mode]
[FMNUM error-number-from]
[TONUM error-number-to]
[SLKIND message-type]
[LANGUAGE languages]
[DDMDBID ddm-dbid] [DDMFNR ddm-fnr]
[NATVERS natural-version]
[   DATE date [DATECHECK date-check]                                     ]
[   [FMDATE date-from] [TODATE date-to]                                  ]
[   [DATECHECK date-check]                                               ]
[   [SIZE size]                                                          ]
[   [FMSIZE size-from] [TOSIZE size-to]                                  ]
[USERID user-id]
[TID terminal-id]
[except-clause]
```

except-clause

```
EXCEPT
( object-name
[LIBRARY library-name]
[OBJTYPE group-type]
[SCKIND object-kind]
[NATTYPE object-type]
[MODE object-mode]
[SLKIND message-type]
[FMNUM error-number-from] [TONUM error-number-to]
[LANGUAGE languages]
[DDMDBID ddm-dbid] [DDMFNR ddm-fnr]
[NATVERS natural-version]
[ DATE date                                          ]
[ [FMDATE date-from] [TODATE date-to]                ]
[ SIZE size                                          ]
[ [FMSIZE size-from] [TOSIZE size-to]                ]
[USERID user-id]
[TID terminal-id]
)
```

Anmerkungen:

1. Bei dem Kommando `FINDLIB` werden nur die folgenden Schlüsselwörter verarbeitet: `LIBRARY`, `DBID`, `FNR`, `NAME`, `CIPHER` und `PASSWORD` oder `PSW`.
2. Bei der Verarbeitung von Natural-DDMs müssen Sie `OBJTYPE` auf `D` setzen. Außerdem gelten einige Schlüsselwörter nicht für DDMs, was im Folgenden beschrieben wird.

Erläuterung der Schlüsselwörter zur Auswahl von Natural-Library-Objekten und DDMs

Die Schlüsselwörter und gültigen Werte für die zu verarbeitenden Objekte werden in der folgenden Tabelle erläutert.

Schlüsselwort	Gültige Werte	Standardwerte
object-name	Ein gültiger Objektname oder ein Namensbereich. Wenn *object-name* Leerzeichen enthält, muss er in doppelte Hochkommas (" ") eingeschlossen werden. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	keine
LIBRARY	Ein gültiger Library-Name oder ein Namensbereich. Wenn OBJTYPE (siehe unten) auf D gesetzt ist, wird der Library-Name ignoriert. Wenn SETNO angegeben ist, ist ein Namensbereich nicht zulässig. Siehe auch ***Name***.	keine
DBID	Nicht gültig bei DDMs auf Großrechnern (OBJTYPE auf D gesetzt; siehe unten). Eine gültige Datenbankkennung.	0 (aktuelle FNAT/FUSER)
FNR	Nicht gültig bei DDMs auf Großrechnern (OBJTYPE auf D gesetzt; siehe unten). Eine gültige Dateinummer.	0 (aktuelle FNAT/FUSER)
NAME	Gilt nur für Objekte auf Großrechnern. Nicht gültig bei DDMs auf Großrechnern (OBJTYPE auf D gesetzt; siehe unten). Ein gültiger VSAM-Name.	leer (aktuelle FNAT/FUSER)
CIPHER	Gilt nur für Objekte auf Großrechnern. Nicht gültig bei DDMs auf Großrechnern (OBJTYPE auf D gesetzt; siehe unten). Der 8-stellige Chiffrierschlüssel der Adabas-Datei, in der die Objekte gespeichert sind.	leer (aktuelle FNAT/FUSER)
PASSWORD oder PSW	Gilt nur für Objekte auf Großrechnern. Nicht gültig bei DDMs auf Großrechnern (OBJTYPE auf D gesetzt; siehe unten). Ein 8-stelliges Adabas-Passwort.	leer (aktuelle FNAT/FUSER)
OBJTYPE	Objekttypen sind: E Benutzerdefinierte Fehlermeldungen N Natural-Programmierobjekte * Stern (alle) oder eine gültige Kombination. **Ausnahme:** Der Objekttyp D kann mit keinem anderen Typ kombiniert werden.	*

Schlüsselwort	Gültige Werte	Standardwerte
SETNO	Gilt nur für die Funktionen **Unload** (Entladen) und **Find** (Suchen) und wenn Predict installiert ist. Gilt nicht für Anwendungsobjekte . Siehe auch *Auswahl von Anwendungsobjekten*. Eine ein- oder zweistellige Zahl, die das Retained Set identifiziert, das für die Namen der zu verarbeitenden Objekte verwendet werden soll. Ein Retained Set wird mit der Option save set des Kommandos LIST XREF erzeugt. Wenn SETNO angegeben wird, wird der für *object-name* angegebene Wert ignoriert. Ausführliche Informationen zu Predict Sets finden Sie in der *Predict*-Dokumentation.	keine
SETUSER	Gilt nur für die Funktionen **Unload** (Entladen) und **Find** (Suchen) und wenn Predict installiert ist. Gilt nicht für Anwendungsobjekte (siehe auch *Auswahl von Anwendungsobjekten*). Die Benutzerkennung des Benutzers, der das Predict Set erstellt hat. Wenn keine Benutzerkennung angegeben wird, wird der Wert der Systemvariablen *USER verwendet. Siehe *Systemvariablen*-Dokumentation.	*USER
SETLIBRARY	Gilt nur für die Funktionen **Unload** (Entladen) und **Find** (Suchen) und wenn Predict installiert ist. Gilt nicht für Anwendungsobjekte (siehe auch *Auswahl von Anwendungsobjekten*). Der Name der Library, in der nach einem Predict Set gesucht werden soll. Wenn Sie SETLIBRARY nicht angeben, wird stattdessen die mit LIBRARY angegebene Library verwendet.	
NATTYPE	Gilt nicht, wenn OBJTYPE auf D gesetzt ist. Ein oder mehrere aus einem Zeichen bestehende Codes für Natural-Objekttypen: P Program N Subprogram S Subroutine C Copycode H Helproutine T Text 7 Function 8 Adapter G Global Datenbereich L Local Datenbereich A Parameter Datenbereich	*

Schlüsselwort	Gültige Werte	Standardwerte
	M Map 4 Class 3 Dialog 5 Natural Kommandoprozessor 9 Resource * Alle Objekttypen	
SCKIND	Gilt nicht, wenn OBJTYPE auf D gesetzt ist. Die Art der Natural-Programmierobjekte. Gültige Eingabewerte: S Quellcodeobjekte: Objekte, die nur in Quellcodeform gespeichert sind. C Katalogisierte Objekte: Objekte, die nur in katalogisierter Form gespeichert sind. A Alle Quellcodeobjekte und katalogisierten Objekte. W Alle mit STOW gespeicherten Objekte: Quellcodeobjekte und katalogisierte Objekte mit identischem Datum und Zeit. B Quellcodeobjekte und katalogisierte Objekte, wenn beide existieren. **Anmerkung:** W und B sind nur für die Kommandos UNLOAD und FIND gültig. Für LOAD und SCAN sind W und B gültige Einträge, aber sie werden wie A (alle Objekte) behandelt. Wenn Daten im Transfer-Format verarbeitet werden, gilt nur S (Quellcodeobjekte) oder A. Das Suchergebnis einer Objektprüfung hängt außerdem von den Einstellungen DATE (oder FMDATE und TODATE) und DATECHECK ab.	A
MODE	Gilt nicht, wenn OBJTYPE auf D gesetzt ist. Der Programmiermodus der Natural-Programmierobjekte. Gültige Eingabewerte: A Any. R Alle Objekte im Reporting-Modus. S Alle Objekte im Structured-Modus.	
FMNUM	Eine Startnummer für Natural-Fehlermeldungen. Gültiger Bereich: 1 bis 9999.	1
TONUM	Eine Endnummer für natürliche Fehlermeldungen. Gültiger Bereich: 1 bis 9999.	9999 oder Wert von FMNUM

Schlüsselwort	Gültige Werte	Standardwerte
	Der Wert muss größer oder gleich dem Wert von FMNUM sein, falls angegeben.	(falls angegeben)
SLKIND	Die Art des Textes der Natural-Fehlermeldung. Gültige Eingabewerte: S Kurztext. Kann nicht beim Kommando DELETE angewendet werden. Siehe ***Grundlegende Kommandosyntax***. L Langtext. A Kurz- und/oder Langtext. B Kurz- und Langtext, wenn beide existieren.	A
LANGUAGE	Bis zu 8 gültige Sprachcodes (z. B. Code 1 für Englisch) für benutzerdefinierte Fehlermeldungen. Mit einem Stern (*) werden alle Sprachcodes ausgewählt.	*
DDMDBID	Die gültige Datenbankkennung (1 bis 65535) eines DDM. UNLOAD, LOAD und SCAN: 0 bedeutet, dass keine Prüfung durchgeführt wird. DDMs werden unabhängig von ihrer Datenbankkennung (DBID) verarbeitet.	0
DDMFNR	Die gültige Dateinummer (1 bis 65535) eines DDM. UNLOAD, LOAD und SCAN: 0 bedeutet, dass keine Prüfung durchgeführt wird. DDMs werden unabhängig von ihrer Dateinummer (FNR) verarbeitet.	0
NATVERS	Die Natural-Version der Natural-Programmierobjekte. Sie können auch einen Versionsbereich angeben. Siehe ***Name***.	leer (keine Prüfung)
DATE	Das Speicherungs- oder Katalogisierungsdatum von Natural-Programmierobjekten. Sie können eine Zeit hinzufügen, indem Sie ein Leerzeichen zwischen Datum und Zeit einfügen. Das Format und die zulässigen Bereiche finden Sie unter ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*. Zulässige spezielle Angaben sind YESTERDAY und TODAY. Siehe ***Spezielle Datumswerte*** in *Datum*. Eine Datumsprüfung kann auch für eine bestimmte Objektart (Quellcode- oder katalogisiertes Objekt) definiert werden. Siehe DATECHECK.	leer (keine Prüfung)
DATECHECK	Art des Objekts (Quellcode- oder katalogisiertes Objekt), das für das Datum oder den Datumsbereich geprüft werden soll, die mit DATE bzw. FMDATE und TODATE definiert wurden. Gültige Eingabewerte: S Prüfung des Datums von Quellcodeobjekten und (falls vorhanden) des Datums von Objekten, die nur als katalogisierte Objekte existieren.	leer (keine Prüfung)

Schlüsselwort	Gültige Werte	Standardwerte
	C Prüfung des Datums von katalogisierten Objekten und (falls vorhanden) des Datums von Objekten, die nur als Quellcodeobjekte existieren. Das Suchergebnis der Objektprüfung hängt auch von der Einstellung von SCKIND ab.	
FMDATE	Ein Startwert: Das Datum, an dem oder nach dem die Natural-Programmierobjekte katalogisiert oder gespeichert wurden. Das Format ist identisch mit DATE. Siehe ***Date***. Zulässige spezielle Datumswerte sind: YEAR, MONTH, YESTERDAY und TODAY. See ***Spezielle Datumswerte*** in *Datum*. Eine Datumsprüfung kann auch für eine bestimmte Objektart (Quellcodeobjekt oder katalogisiertes Objekt) definiert werden. Siehe DATECHECK.	leer (keine Prüfung)
TODATE	Ein Endwert: Das Datum, an dem oder vor dem die Natural-Programmierobjekte katalogisiert oder gespeichert wurden. Das Format ist identisch mit DATE. Siehe ***Datum***. Zulässige spezielle Datumswerte sind: YEAR, MONTH, YESTERDAY und TODAY. Siehe ***Spezielle Datumswerte*** in *Datum*. Eine Datumsprüfung kann auch für eine bestimmte Objektart (Quellcodeobjekt oder katalogisiertes Objekt) definiert werden. Siehe DATECHECK.	leer (keine Prüfung) oder hoher Wert (wenn FMDATE angegeben)
SIZE	Die Größe von Natural-Programmierobjekten (bis zu 7 Stellen).	leer (keine Prüfung)
FMSIZE	Ein Startwert: Die Mindestgröße von Natural-Programmierobjekten (bis zu 7 Stellen).	leer (keine Prüfung)
TOSIZE	Ein Endwert: Die maximale Größe von Natural-Programmierobjekten (bis zu 7 Stellen). .	leer (keine Prüfung) oder hoher Wert (wenn FMDATE angegeben)
USERID	Die Benutzerkennung des Benutzers, der die Natural-Programmierobjekte gespeichert oder katalogisiert hat. Sie können auch einen Bereich von Benutzerkennungen angeben. Siehe auch ***Name***.	leer (keine Prüfung)

Schlüsselwort	Gültige Werte	Standardwerte
TID	Gilt nicht, wenn OBJTYPE auf D gesetzt ist. Die Kennung des Terminals, auf dem die Natural-Programmierobjekte gespeichert oder katalogisiert wurden (wird durch die Natural-Systemvariable *INIT-ID bereitgestellt). Sie können auch einen Bereich von Terminalkennungen angeben. Siehe auch *Name*.	leer (keine Prüfung)
EXCEPT	Alle Objekte, die den vor der Eingabe in EXCEPT eingegebenen Auswahlkriterien entsprechen, werden auf alle Parameter geprüft, die in den Klammern nach dem Schlüsselwort EXCEPT stehen. Wenn sie ebenfalls allen diesen Parametern entsprechen, werden sie nicht verarbeitet.	nicht zutreffend

Anmerkungen:

1. Parameter, die für OBJTYPE irrelevant sind, werden ignoriert. Zum Beispiel: DATE, SIZE und USERID haben keine Bedeutung bei Natural-Fehlermeldungen.
2. DBID, FNR, NAME, CIPHER und PASSWORD oder PSW werden vom Kommando LOAD oder SCAN ignoriert. Diese Parameter müssen stattdessen in der *parameter-setting*-Klausel angegeben werden, wie bei LOADFNAT... und LOADFUSER... in *Erläuterung der Schlüsselwörter für parameter-clause* beschrieben.

Auswahl von Natural-Related-Debug-Umgebungen

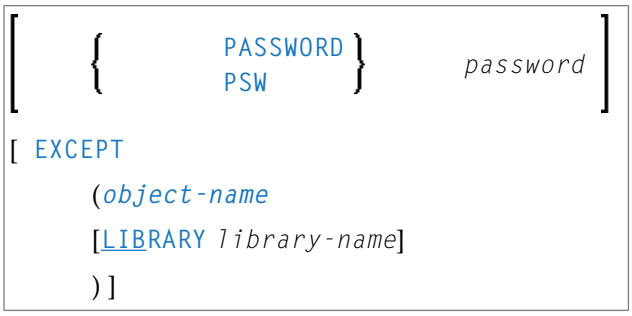
Diese Auswahl wird verwendet, um Natural-Related-Debug-Umgebungen für die Verarbeitung auszuwählen.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax der Natural-Related-Debug-Umgebungsauswahl

Syntax der Natural-Related-Debug-Umgebungsauswahl

```
object-name
NATPATH DEBUG
[LIBRARY library-name]
[     DBID dbid[FNR fnr] ]
[NAME vsam-name]
[CIPHER cipher]
```



Erläuterung der Schlüsselwörter zur Natural-Related-Debug-Umgebungsauswahl

Die Schlüsselwörter und gültigen Eingabewerte für die zu verarbeitenden Debug-Umgebungen werden im folgenden Abschnitt beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
object-name	Ein gültiger Debug-Umgebungsname oder ein Namensbereich. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	keine
LIBRARY	Ein gültiger Library-Name oder ein Namensbereich. Siehe auch ***Name***.	keine
DBID	Eine gültige Datenbankkennung.	0 (aktuelle FUSER)
FNR	Eine gültige Dateinummer.	0 (aktuelle FUSER)
NAME	Ein gültiger VSAM-Name.	leer (aktuelle FUSER)
CIPHER	Der 8-stellige Chiffrierschlüssel der Adabas-Datei, in der die Debug-Umgebungen gespeichert sind.	leer (aktuelle FUSER)
PASSWORD oder PSW	Ein 8-stelliges Adabas-Passwort.	leer (aktuelle FUSER)
EXCEPT	Siehe EXCEPT in *Auswahl von Natural-Library-Objekten und DDMs*.	nicht zutreffend

Anmerkung: DBID, FNR, NAME, CIPHER und PASSWORD oder PSW werden vom Kommando LOAD oder SCAN ignoriert. Diese Parameter müssen stattdessen in der *parameter-setting*-Klausel angegeben werden, wie für LOADFNAT... und LOADFUSER... in ***Erläuterung der Schlüsselwörter für parameter-clause*** beschrieben.

Auswahl von Natural-Related-Profilen

Diese Auswahl wird verwendet, um Natural-Related-Profile für die Verarbeitung auszuwählen.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax zur Auswahl von Natural-Related-Profilen

Syntax zur Auswahl von Natural-Related-Profilen

```
object-name
NATPATH PROFILE
[OBJTYPE profile-type]
[    DBID dbid[FNR fnr] ]
[NAME vsam-name]
[CIPHER cipher]
[   { PASSWORD }     password   ]
    { PSW      }
[ EXCEPT
    (object-name
    [OBJTYPE profile-type]
    ) ]
```

Erläuterung der Schlüsselwörter zur Auswahl von Natural-Related-Profilen

Die Schlüsselwörter und gültigen Eingabewerte für die zu verarbeitenden Profile sind im folgenden Abschnitt beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
object-name	Ein gültiger Profilname oder ein Namensbereich. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	keine
OBJTYPE	Der Typ des Profils: D — Geräteprofil (Device profile) E — Editor-Profil M — Maskenprofil (Map profile) P — Parameterprofil	*

Schlüsselwort	Gültige Werte	Standardwerte
	* Stern (alle Profiltypen) oder eine beliebige Kombination.	
DBID	Eine gültige Datenbankkennung.	0 (aktuelle FNAT)
FNR	Eine gültige Dateinummer.	0 (aktuelle FNAT)
NAME	Ein gültiger VSAM-Name.	leer (aktuelle FNAT)
CIPHER	Der 8-stellige Chiffrierschlüssel der Adabas-Datei, in der die Profile gespeichert sind.	leer (aktuelle FNAT)
PASSWORD oder PSW	Ein 8-stelliges Adabas-Passwort.	leer (aktuelle FNAT)
EXCEPT	Siehe EXCEPT in *Auswahl von Natural-Library-Objekten und DDMs*.	nicht zutreffend

Anmerkung: DBID, FNR, NAME, CIPHER und PASSWORD oder PSW werden vom Kommando LOAD oder SCAN ignoriert. Diese Parameter müssen stattdessen in der *parameter-setting*-Klausel angegeben werden, wie für LOADFNAT... und LOADFUSER... in ***Erläuterung der Schlüsselwörter für*** *parameter-clause* beschrieben.

Auswahl von Natural-Systemfehlermeldungen

Diese Auswahl wird verwendet, um Natural-Systemfehlermeldungen zur Verarbeitung auszuwählen.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Schlüsselwörter zur Auswahl von Natural-Systemfehlermeldungen

Schlüsselwörter zur Auswahl von Natural-Systemfehlermeldungen

```
ERROR NATERROR
   DBID dbid FNR
[  fnr [NAME
   vsam-name]   [      {      PASSWORD }      password      ]      ]
   [CIPHER
   cipher]
[FMNUM error-number-from] [TONUM error-number-to]
[SLKIND message-type]
[LANGUAGE languages]
[ EXCEPT
   (
   [FMNUM error-number-from] [TONUM error-number-to]
   [SLKIND message-type]
   [LANGUAGE languages]
   ) ]
```

Erläuterung der Schlüsselwörter zur Auswahl von Natural-Systemfehlermeldungen

Im folgenden Abschnitt werden die Schlüsselwörter und die gültigen Eingabewerte für die zu verarbeitenden Natural-Systemfehlermeldungen beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
DBID	Gilt nur für Systemfehlermeldungen auf Großrechnern. Eine gültige Datenbankkennung.	0 (aktuelle FNAT)
FNR	Gilt nur für Systemfehlermeldungen auf Großrechnern. Eine gültige Dateinummer.	0 (aktuelle FNAT)
NAME	Gilt nur für Systemfehlermeldungen auf Großrechnern. Ein gültiger VSAM-Name.	leer (current FNAT)
CIPHER	Gilt nur für Systemfehlermeldungen auf Großrechnern. Der 8-stellige Chiffrierschlüssel der Adabas-Datei, in der die Systemfehlermeldungen gespeichert sind.	leer (aktuelle FNAT)
PASSWORD oder PSW	Gilt nur für Systemfehlermeldungen auf Großrechnern. Ein 8-stelliges Adabas-Passwort.	leer (aktuelle FNAT)

Schlüsselwort	Gültige Werte	Standardwerte
FMNUM	Eine Startnummer für Systemfehlermeldungen. Gültiger Bereich: 1 bis 9999.	1
TONUM	Eine Endnummer für Systemfehlermeldungen. Gültiger Bereich: 1 bis 9999. Der Wert muss größer oder gleich dem Wert von FMNUM sein, falls angegeben.	9999 oder Wert von FMNUM (falls angegeben)
SLKIND	Siehe SLKIND in *Auswahl von Natural-Library-Objekten und DDMs*.	A
LANGUAGE	Bis zu 8 gültige Sprachcodes (z. B. Code 1 für Englisch) für Systemfehlermeldungen. Ein Stern (*) wählt alle Sprachcodes aus.	*
EXCEPT	Siehe EXCEPT in *Auswahl von Natural-Library-Objekten und DDMs*.	

Anmerkung: DBID, FNR, NAME, CIPHER und PASSWORD oder PSW werden vom Kommando LOAD oder SCAN ignoriert. Diese Parameter müssen stattdessen in der *parameter-setting*-Klausel angegeben werden, wie für LOADFNAT... und LOADFUSER... in ***Erläuterung der Schlüsselwörter für*** *parameter-clause* beschrieben.

Auswahl von Natural-Kommandoprozessoren

Diese Auswahl wird verwendet, um die Quellcodeobjekte von Natural-Kommandoprozessoren für die Verarbeitung auszuwählen.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax zur Auswahl der Quellcodeobjekte von Natural-Kommandoprozessoren

Syntax zur Auswahl der Quellcodeobjekte von Natural-Kommandoprozessoren

```
object-name  PROCESSOR ncp-library-name
[            DBID ncp-dbid FNR ncp-fnr [file-options] ]
[EXCEPT
             (object-name
             [LIBRARY ncp-library-name]
             ) ]
```

file-options

```
[NAME ncp-vsam-name]
[CIPHER ncp-cipher]
[ { PASSWORD } ncp-password ]
  { PSW      }
```

Anmerkung: Beim Kommando FINDLIB werden nur die folgenden Schlüsselwörter verarbeitet: PROCESSOR, DBID, FNR, NAME, CIPHER und PASSWORD oder PSW.

Erläuterung der Schlüsselwörter zur Auswahl der Quellcodeobjekte von Natural-Kommandoprozessoren

Im folgenden Abschnitt werden die Schlüsselwörter und die gültigen Eingabewerte zur Auswahl der zu verarbeitenden Quellcodeobjekte von Natural-Kommandoprozessoren beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
object-name	Der Name eines gültigen Natural-Kommandoprozessor-Quellcodeobjekts oder ein Namensbereich. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	keine
PROCESSOR	Ein gültiger Library-Name oder ein Namensbereich. Siehe auch ***Name***.	keine
DBID	Die gültige Datenbankkennung der Adabas-Datei, in der die Natural-Kommandoprozessor-Quellcodeobjekte gespeichert sind.	0 (aktuelle FNAT/FUSER) aktuelle
FNR	Die gültige Dateinummer der Adabas-Datei, in der die Natural-Kommandoprozessor-Quellcodeobjekte gespeichert sind.	0 (aktuelle FNAT/FUSER)
NAME	Gilt nur für Natural-Kommandoprozessor-Quellcodeobjekte auf Großrechnern. Ein gültiger VSAM-Name.	leer
CIPHER	Der 8-stellige Chiffrierschlüssel der Adabas-Datei, in der die Natural-Kommandoprozessor-Quellcodeobjekte gespeichert sind.	leer
PASSWORD oder PSW	Das 8-stellige Adabas-Passwort für die Adabas-Datei, in der die Natural-Kommandoprozessor-Quellcodeobjekte gespeichert sind.	leer
EXCEPT	Siehe EXCEPT in *Auswahl von Natural-Library-Objekten und DDMs*.	

Anmerkung: DBID, FNR, NAME, CIPHER und PASSWORD oder PSW werden vom Kommando LOAD oder SCAN ignoriert. Diese Parameter müssen stattdessen in der *parameter-setting*-Klausel angegeben werden, wie für LOADNCP . . . in ***Erläuterung der Schlüsselwörter für*** *parameter-clause* beschrieben.

Auswahl von FDTs

Diese Auswahl wird verwendet, um Adabas Field Definition Tables (FDTs) für die Verarbeitung auszuwählen.

Ausführliche Informationen zu Adabas FDTs sind in der *Adabas for Mainframes*-Dokumentation enthalten: *Concepts and Facilities* > *Adabas Design* > *Record and Field Definitions*.

Bezüglich Informationen zum Laden von FDTs siehe auch ***Objektangabe für Felddefinitionstabellen (FDTs)*** im Kapitel *Objektangabe*.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax der FDT-Auswahl

Syntax der FDT-Auswahl

```
FDT
DBID dbid
{ FNR fnr [CIPHER cipher] [ { PASSWORD } password ] }
{                         { PSW      }          ]   }
{ FMFNR fnr-start TOFNR fnr-end                    }
```

Erläuterung der Schlüsselwörter zur FDT-Auswahl

Die Schlüsselwörter und gültigen Eingabewerte für die zu verarbeitenden FDTs werden im folgenden Abschnitt beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
DBID	Die Datenbankkennung der FDT.	keine
FNR	Die Dateinummer der FDT.	keine
CIPHER	Der 8-stellige Adabas-Chiffrierschlüssel der FDT.	keine
PASSWORD oder PSW	Das 8-stellige Adabas-Passwort für die FDT.	keine

Schlüsselwort	Gültige Werte	Standardwerte
FMFNR	Gilt nur für die Kommandos FIND oder UNLOAD. Ein Startwert: Die Dateinummer (FNR) einer FDT.	keine
TOFNR	Gilt nur für die Kommandos FIND oder UNLOAD Ein Endwert: Die Dateinummer (FNR) einer FDT.	keine

Auswahl von Anwendungen (Applications)

Diese Auswahl gilt für Anwendungen, die im Application Workspace von Natural Studio erstellt und gepflegt werden, und für die Libraries oder Objekte, die zu diesen Anwendungen gehören.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Auswahl von Base und Compound Applications
- Auswahl von Application Libraries
- Auswahl von Anwendungsobjekten

Auswahl von Base und Compound Applications

Diese Auswahl gilt nur für die Funktion **Find**.

Syntax

```
APPLICATION APNAME application-name
[APTYPE application-type]
[COMPAPPLICATION compound-application-name]
[                       EXCEPT
                        (APNAME application-name
                        [APTYPE application-type]
                        ) ]
```

Auswahl von Application Libraries

Diese Auswahl gilt nur für die Funktion **Find**.

Syntax

```
APPLICATION  APLIBRARY application-library-name
[BASEAPPLICATION base-application-name]
[ COMPAPPLICATION compound-application-name]
[            DBID dbid [FNR fnr] ]
[            EXCEPT
             (APLIBRARY application-library-name
             [BASEAPPLICATION base-application-name]
             ) ]
```

Auswahl von Anwendungsobjekten

Diese Auswahl gilt nur für die Funktionen **Find** und **Unload**.

Syntax

```
APPLICATION  APOBJECTS application-object-name
[BASEAPPLICATION base-application-name]
[COMPAPPLICATION compound-application-name]
[LIBRARY library-name]
[object-specification]
[                        EXCEPT
                         (APOBJECT application-object-name
                         [LIBRARY library-name]
                         [BASEAPPLICATION base-application-name]
                         [object-specification]
                         ) ]
```

Erläuterung der Schlüsselwörter zur Auswahl von Anwendungsobjekten

Im folgenden Abschnitt werden die Schlüsselwörter und die gültigen Eingabewerte für die zu verarbeitenden Anwendungen, Application Libraries oder Anwendungsobjekte beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
APNAME	Ein gültiger Name einer Natural-Anwendung oder ein Namensbereich. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	*
APTYPE	Ein gültiger Anwendungstyp: B — Base Application O — Compound Application * — Alle: Base und/oder Compound Applications	*
COMPAPPLICATION	Gilt nur, wenn APTYPE auf * oder B gesetzt ist. Der Name einer Compound Application, zu der die angegebene Base Application gehört, oder ein Namensbereich. Es werden nur Base Applications ausgewählt, die zu der (den) angegebenen Compound Application(s) gehören. Base Applications, die nicht zu einer Compound Application gehören, werden nicht ausgewählt.	keine
EXCEPT	Siehe EXCEPT in *Auswahl von Natural-Library-Objekten und DDMs*.	nicht zutreffend
APLIBRARY	Gültiger Name einer Library, die zu einer Natural Base oder Compound Application gehört, oder ein Namensbereich. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	*
BASEAPPLICATION	Gültiger Name einer Natural Base Application, zu der eine Application Library oder ein Anwendungsobjekt gehört. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	*
DBID	Gültige Datenbankkennung einer Application Library.	0 (keine Prüfung)
FNR	Gültige Dateinummer einer Application Library.	0 (keine Prüfung)
APOBJECT	Gültiger Name eines Anwendungsobjekts, das zu einer Base oder Compound Application gehört. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	*
LIBRARY	Ein gültiger Library-Name oder ein Namensbereich. Wenn OBJTYPE auf D gesetzt ist (siehe *Auswahl von Natural-Library-Objekten und DDMs*), wird der Library-Name ignoriert. Siehe auch ***Name*** in *Angabe von Name, Datum und Zeit*.	*

Schlüsselwort	Gültige Werte	Standardwerte
object-specification	Gibt an, dass zusätzliche Auswahlkriterien für Anwendungsobjekte angegeben werden können, so wie es in ***Syntax zur Auswahl von Natural Library-Objekten und DDMs*** gezeigt wird: Alle unter LIBRARY *library-name* aufgelisteten Elemente können auch bei Anwendungsobjekten angewendet werden, während *object-name* in der EXCEPT-Klausel bei Anwendungsobjekten nicht relevant ist.	nicht zutreffend

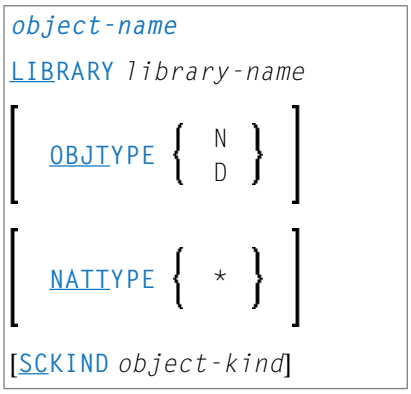
Auswahl von Löschanweisungen für Objekte

Diese Auswahl wird verwendet, um Löschanweisungen für Natural-Library-Objekte, DDMs, benutzerdefinierte Fehlermeldungen und Natural-Systemfehlermeldungen anzugeben. Die Löschanweisungen werden ausgeführt, wenn bei angegebener Option DELETEALLOWED eine Arbeitsdatei mit internem Format in die Zielungebung geladen wird.

Die entsprechende Syntax wird im folgenden Abschnitt dargestellt und erläutert.

- Syntax zur Auswahl von Löschanweisungen für Natural-Library-Objekte und DDMs
- Syntax zur Auswahl von Löschanweisungen für benutzerdefinierte Fehlermeldungen
- Syntax zur Auswahl von Löschanweisungen für Natural-Systemfehlermeldungen

Syntax zur Auswahl von Löschanweisungen für Natural-Library-Objekte und DDMs



Erläuterung der Schlüsselwörter zur Auswahl von Löschanweisungen für Natural-Library-Objekte und DDMs

Im folgenden Abschnitt werden die Schlüsselwörter und die gültigen Eingabewerte für die zu verarbeitenden Objekte beschrieben.

Schlüsselwort	Gültige Werte	Standardwerte
object-name	Ein gültiger Objektname oder ein Startwert (*value*) für einen Namensbereich wie z.B. ABC*	keine
LIBRARY	Ein gültiger Library-Name. Eine Bereichsangabe ist *nicht* erlaubt. Wenn OBJTYPE (siehe unten) auf D gesetzt ist, wird der Library-Name ignoriert.	keine
OBJTYPE	Ein gültiger Objekttyp-Code: D DDMs N Natural-Programmierobjekte Der Objekttyp D kann nicht mit dem Objekttyp N kombiniert werden.	*
NATTYPE	Nicht anwendbar, wenn OBJTYPE auf D gesetzt ist. Ein Natural-Objekttyp. Gültige Eingabewerte sind: * Alle Objekttypen	*
SCKIND	Nicht anwendbar, wenn OBJTYPE auf D gesetzt ist. Die Art der Natural-Programmierobjekte. Gültige Eingabewerte sind: S Quellcodobjekte. Wenn in der *except-clause* verwendet (siehe *Syntax zur Auswahl von Natural-Library-Objekten und DDMs*): Objekte, die nur in Quellcodeform gespeichert sind. C Katalogisierte Objekte. Wenn in der *except-clause* verwendet: Objekte, die nur in katalogisierter Form gespeichert sind. A Alle Quellcodeobjekte und katalogisierten Objekte.	A

Syntax zur Auswahl von Löschanweisungen für benutzerdefinierte Fehlermeldungen

```
*
LIBRARY library-name
OBJTYPE E
FMNUM error-number-from
[TONUM error-number-to]
[SLKIND message-type]
[LANGUAGE languages]
```

library-name bezeichnet den Namen einer einzelnen Library; eine Bereichsangabe ist nicht zulässig.

Erläuterungen zu den in dieser Syntax verwendeten Elementen finden Sie unter *Erläuterung der Schlüsselwörter zur Auswahl von Natural-Library-Objekten und DDMs*.

Syntax zur Auswahl von Löschanweisungen für Natural-Systemfehlermeldungen

```
ERROR NATERROR
FMNUM error-number-from
[TONUM error-number-to]
[SLKIND message-type]
[LANGUAGE languages]
```

Erläuterungen zu den in dieser Syntax verwendeten Elementen finden Sie unter *Erläuterung der Schlüsselwörter zur Auswahl von Natural-Systemfehlermeldungen*.

Auswahl der Hilfetextverarbeitung

Mit dieser Auswahl legen Sie fest, dass Natural-Hilfetexte während der Funktionen **Unload**, **Load**, **Scan** und **Find** verarbeitet werden, wenn das interne Format für die Arbeitsdateien verwendet wird. Diese Hilfetexte sind identisch mit den Hilfeinformationen, die das Dienstprogramm Natural Help liefert, das mit dem Systemkommando `HELP` aufgerufen wird (siehe *Systemkommandos*-Dokumentation).

Syntax zur Auswahl der Hilfetextverarbeitung

```
ERROR NATURAL HELP
[ DBID dbid FNR fnr [NAME vsam-name]  [ { PASSWORD } password ] ]
  [CIPHER cipher]                     [ { PSW      }          ]
```

Erläuterungen zu den in dieser Syntax verwendeten Elementen finden Sie unter *Erläuterung der Schlüsselwörter zur Auswahl von Natural-Library-Objekten und DDMs*.

113 Objektliste - Arbeitsplantyp LIST

- Syntax von object-type-and-location 1026
- Syntax von object-name-description 1027
- Beispiel für eine Objektliste 1029

Eine Objektliste ist ein Arbeitsplan vom **Typ LIST**, der Objektauswahlkriterien für die zu bearbeitenden Objekte im Kommando `UNLOAD`, `LOAD`, `FIND` oder `DELETE` festlegt. Eine Objektliste kann als Alternative zur *`select-clause`* und zum Arbeitsplan vom **Typ SELECTION** verwendet werden.

Die folgende Syntax gilt für eine Objektliste:

```
TYPE LIST
{ object-type-and-location (object-name-description ...) } ...
```

Die Syntaxoptionen werden im folgenden Abschnitt erläutert. Die in den Syntaxdiagrammen dieses Abschnitts enthaltenen Schlüsselwörter und Variablenwerte stellen Parameter dar, die zur Angabe von Objektauswahlkriterien verwendet werden. Falls angegeben, muss ein Variablenwert zusammen mit einem Schlüsselwort angegeben werden. Jedes Syntaxelement (mit Ausnahme der in Klammern eingeschlossenen Elemente) muss in einer neuen Zeile beginnen und in derselben Zeile enden.

Erläuterungen zu den in den Syntaxdiagrammen enthaltenen Schlüsselwörtern finden Sie im Kapitel *`select-clause`*.

Syntax von object-type-and-location

Im folgenden Abschnitt sind die Syntaxdiagramme dargestellt, die für *`object-type-and-location`* gelten.

- Natural-Objekte und DDMs
- Debug-Umgebungen (Natural-Related)
- Profile (Natural-Related)
- FDTs

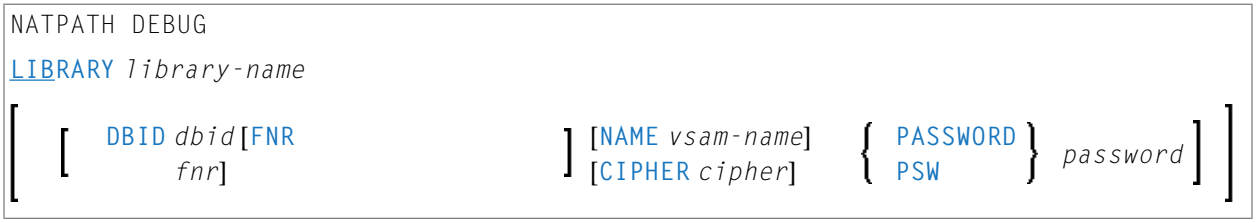
Natural-Objekte und DDMs

```
LIBRARY library-name
[ DBID dbid FNR fnr [NAME vsam-name] [CIPHER cipher]  [ { PASSWORD | PSW } password ] ]
[ OBJTYPE group-type]
```

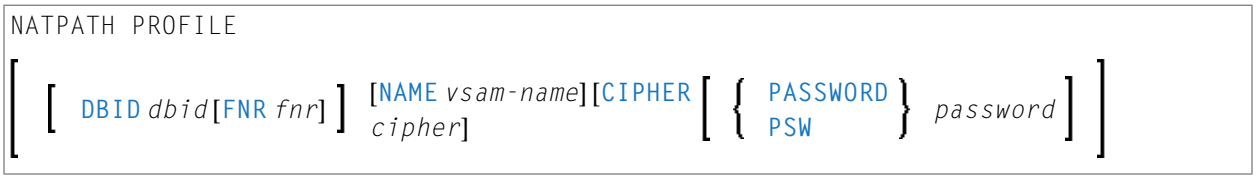
Anmerkungen:

1. Für *`library-name`* sind keine Bereiche erlaubt..
2. Für DDMs muss `OBJTYPE` auf `D` gesetzt werden.

Debug-Umgebungen (Natural-Related)



Profile (Natural-Related)



FDTs



Syntax von object-name-description

Im folgenden Abschnitt sind die Syntaxdiagramme dargestellt, die für *object-name-description* gelten

- Natural-Objekte
- DDMs
- Natural-Systemfehlermeldungen
- Natural-Kommandoprozessor-Quellcodes
- Debug-Umgebungen (Natural-Related)
- Profile (Natural-Related)

- FDTs

Natural-Objekte

```
{ object-name [SCKIND object-kind]
  error-number [ SLKIND message-type] [ LANGUAGE languages]
  FMNUM error-number-from TONUM error-number-to [SLKIND message-type] [LANGUAGE
  languages] }
```

DDMs

```
object-name
```

Natural-Systemfehlermeldungen

```
{ error-number [ SLKIND message-type] [ LANGUAGE languages]
  FMNUM error-number-from TONUM error-number-to [ SLKIND message-type] [ LANGUAGE
  languages] }
```

Natural-Kommandoprozessor-Quellcodes

```
object-name
```

Debug-Umgebungen (Natural-Related)

```
object-name
```

Profile (Natural-Related)

```
object-name [ OBJTYPE profile-type]
```

FDTs

```
DBID dbid FNR fnr [CIPHER cipher] [ { PASSWORD } password ]
                                   { PSW      }
```

Beispiel für eine Objektliste

Nachfolgend ein Beispiel für einen Arbeitsplan vom **Typ LIST**:

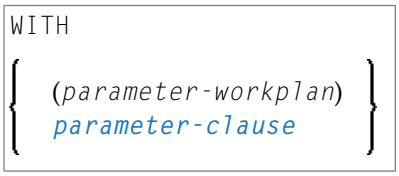
```
TYPE LIST
  LIBRARY LIB-1 OBJTYPE N    /* process Natural objects from library 'LIB-1'
   ( A* SCKIND S             /* all sources objects whose names start with 'A'
  B1                         /* source and/or cataloged object of  'B1'
  CDE> SCKIND C )           /* all cataloged objects with names greater than/equal ↩
to 'CDE'
  /*                         /* comment line
  LIBRARY LIB-2              /* process Natural objects from library 'LIB-2'
                             /* including error messages and shared resources
  ( *                        /* all source and/or cataloged objects
                             /* including shared resources
  FMNUM 1 TONUM 100          /* error messages from 1 to 100
  )
```

114 parameter-setting

- Syntax von parameter-clause 1032
- Erläuterung der Schlüsselwörter für parameter-clause 1033

Die *parameter-setting*-Klausel wird verwendet, um Attribute für das Kommando LOAD oder UNLOAD für die zu bearbeitenden Objekte zu ändern und um Ziel-Destinationen für das LOAD-Kommando zu definieren (z.B. FNAT).

Die folgende Syntax gilt für die *parameter-setting*-Klausel:

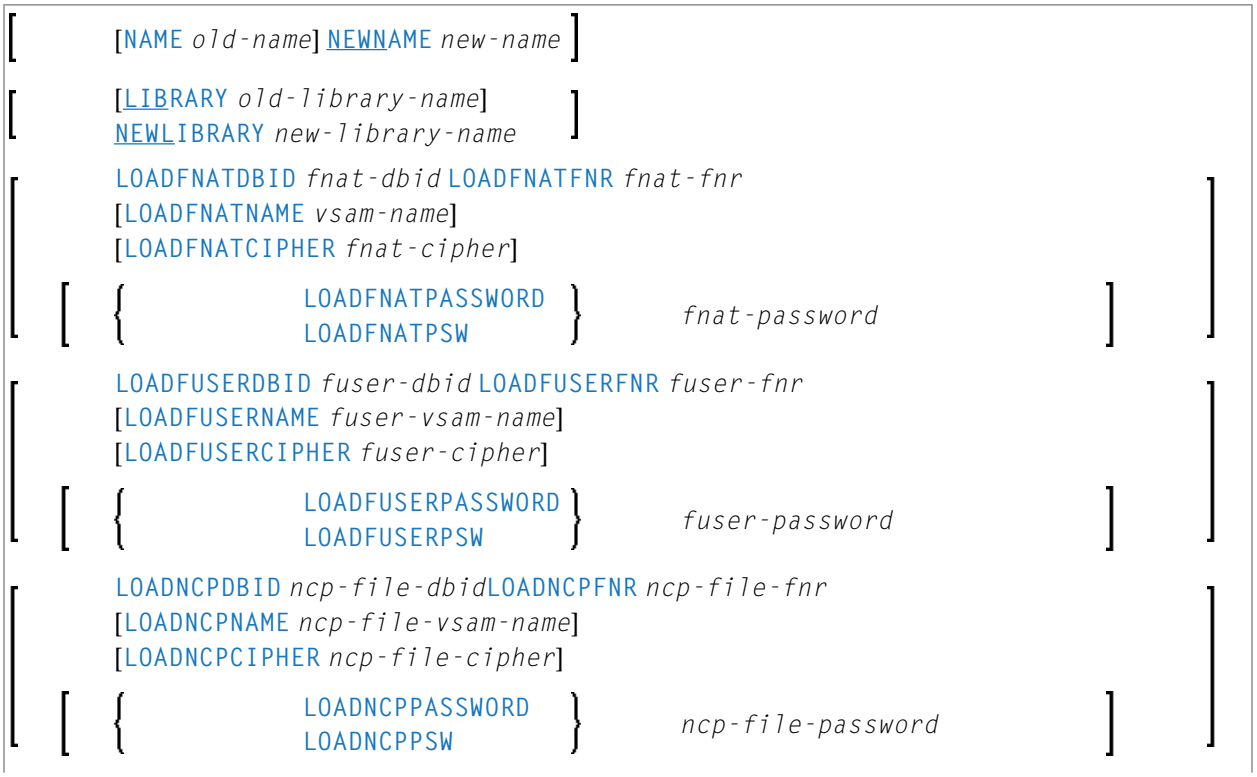


Eine Erläuterung der Syntax, die für *parameter-workplan* gilt, finden Sie unter ***Arbeitspläne referenzieren*** im Kapitel *Arbeitspläne (Workplans)*.

In diesem Abschnitt werden die folgenden Themen behandelt:

Syntax von parameter-clause

Die Syntax der *parameter-clause* ist in dem folgenden Diagramm dargestellt. Ein Variablenwert (falls angegeben) muss mit einem Schlüsselwort angegeben werden.



```
[       [FDTDBID old-fdt-dbid FDTFNR old-fdt-fnr] NEWFDTDBID new-fdt-dbid NEWFDTFNR ]
        new-fdt-fnr
[ERRNUMDIFF modification-of-error-message-range]
[       [LANGUAGE old-language]                                                      ]
        NEWLANGUAGE new-language
[       [DATE old-date] NEWDATE new-date                                             ]
[       [USERID old-userid]  NEWUSERID                                               ]
        new-userid
[       [TID old-terminal-id] NEWTID                                                 ]
        new-terminal-id
[       [PATH old-external-path-name]                                                ]
        NEWPATH new-external-path-name
```

Erläuterung der Schlüsselwörter für parameter-clause

Im folgenden Abschnitt werden die Schlüsselwörter und Variablenwerte (sofern relevant) der *parameter-clause* erläutert.

Keyword	Values	Restricted to Command
NAME	Der Objektname, der geprüft werden soll, wenn NEWNAME angegeben ist.	
NEWNAME	Ein neuer Objektname. **Anmerkung:** Nicht anwendbar bei DDMs auf Großrechner-Plattformen.	
LIBRARY	Der Name der Library, die geprüft werden soll, wenn NEWLIBRARY angegeben wird.	
NEWLIBRARY	Ein neuer Library-Name. **Hinweis für die Funktion LOAD:** NEWLIBRARY hat *keine* Auswirkungen auf den Library-Namen, der in der Löschanweisung einer Arbeitsdatei verwendet wird, die mit der Option DELETEALLOWED verarbeitet wird.	
LOADFNATDBID	Die Datenbankkennung (DBID) von FNAT-Libraries.	LOAD
LOADFNATFNR	Die Dateinummer (FNR) von FNAT-Libraries.	LOAD
LOADFNATNAME	Gilt nur für Objekte auf Großrechnern. Ein FNAT-VSAM-Dateiname.	LOAD
LOADFNATCIPHER	Gilt nur für Objekte auf Großrechnern. Ein FNAT-Chiffrierschlüssel.	LOAD

Keyword	Values	Restricted to Command
LOADFNATPASSWORD oder LOADFNATPSW	Gilt nur für Objekte auf Großrechnern. Ein FNAT-Adabas-Passwort.	LOAD
LOADFUSERDBID	Die Datenbankkennung (DBID) der FUSER-Libraries.	LOAD
LOADFUSERFNR	Die Dateinummer (FNR) der FUSER-Libraries.	LOAD
LOADFUSERNAME	Gilt nur für Objekte auf Großrechnern. Ein FUSER-VSAM-Dateiname.	LOAD
LOADFUSERCIPHER	Gilt nur für Objekte auf Großrechnern. Ein FUSER-Chiffrierschlüssel.	LOAD
LOADFUSERPASSWORD oder LOADFUSERPSW	Gilt nur für Objekte auf Großrechnern. Ein FUSER-Adabas-Passwort.	LOAD
LOADNCPDBID	Die Datenbankkennung (DBID) der Adabas-Datei für Natural-Kommandoprozessor-Quellcodes.	LOAD
LOADNCPFNR	Die Dateinummer (FNR) der Adabas-Datei für Natural-Kommandoprozessor-Quellcodes.	LOAD
LOADNCPNAME	Gilt nur für Objekte auf Großrechnern. Der VSAM-Name der Adabas-Datei für Natural-Kommandoprozessor-Quellcodes.	LOAD
LOADNCPCIPHER	Der Chiffrierschlüssel der Adabas-Datei für Natural-Kommandoprozessor-Quellcodes.	LOAD
LOADNCPPASSWORD oder LOADNCPPSW	Gilt nur für Objekte auf Großrechnern. Das Adabas-Passwort der Adabas-Datei für Natural Kommandoprozessor-Quellcodes.	LOAD
FDTDBID	Die Datenbankkennung (DBID) der Adabas Field Definition Table (FDT), die geprüft werden soll, wenn NEWFDTDBID angegeben ist.	
NEWFDTDBID	Eine neue DBID der FDT.	
FDTFNR	Die DBID der zu prüfenden FDT, wenn NEWFDTFNR angegeben ist.	
NEWFDTFNR	Eine neue Dateinummer (FNR) der FDT.	
ERRNUMDIFF	Eine Zahl (positiv oder negativ), die zu den Natural-Fehlermeldungen beim UNLOAD- oder LOAD-Kommando addiert werden soll. ERRNUMDIFF kann nur angegeben werden, wenn FMNUM und TONUM (siehe *select-clause*) als Auswahlkriterien angegeben wurden. Andernfalls ist es nicht möglich, auf gültige Ergebnisse zu prüfen.	

Keyword	Values	Restricted to Command
LANGUAGE	Bis zu 8 gültige Sprachcodes (z.B. Code 1 für Englisch) von Natural-Fehlermeldungen, die geprüft werden sollen, wenn NEWLANGUAGE (siehe unten) angegeben ist. Enthält *language* mehr als einen Sprachcode, muss *new-language* die gleiche Anzahl von Sprachcodes enthalten. Jeder Sprachcode einer Sprache wird durch den Sprachcode an der entsprechenden Stelle von *new-language* ersetzt. Wird *language* nicht angegeben, darf *new-language* nicht mehr als einen Sprachcode enthalten.	
NEWLANGUAGE	Bis zu 8 gültige Sprachcodes (z. B. Code 4 für Spanisch) für neue benutzerdefinierte Fehlermeldungen. Diese Option gilt nicht für die Langtexte der Natural-Systemfehlermeldungen. Für diese ist Englisch (Sprachcode 1) die einzige gültige Sprache. Siehe auch LANGUAGE oben.	
DATE	Ein Objektdatum. Sie können eine Zeit hinzufügen, indem Sie ein Leerzeichen zwischen Datum und Zeit einfügen. Das Format und die zulässigen Bereiche finden Sie unter ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*.	
NEWDATE	Ein neues Objektdatum. NEWDATE kann ein Datum gefolgt von einem Zeitwert sein. Sie können eine Zeit hinzufügen, indem Sie ein Leerzeichen zwischen Datum und Zeit einfügen. Siehe auch ***Datum*** und ***Zeit*** in *Angabe von Name, Datum und Zeit*.	
USERID	Die zu prüfende Benutzerkennung, wenn NEWUSERID angegeben ist.	
NEWUSERID	Eine neue Benutzerkennung.	
TID	Gilt nur für Objekte auf Großrechnern. Die Terminalkennung, die geprüft wird, wenn NEWTID angegeben ist.	
NEWTID	Gilt nur für Objekte auf Großrechnern. Eine neue Terminalkennung.	
PATH	Der Pfadname, der geprüft werden soll, wenn NEWPATH angegeben ist.	
NEWPATH	Ein neuer Pfadname.	

Anmerkungen:

1. Parameter, die nicht für das verarbeitete Auswahlkriterium zutreffen, werden ignoriert.
2. LOADFNAT..., LOADFUSER... und LOADNCP... werden nur für das Kommando LOAD verwendet und ansonsten ignoriert.
3. LOADFNAT... wird für Libraries verwendet, deren Namen mit SYS beginnen (außer SYSTEM)..

4. LOADFUSER... wird für Libraries verwendet, deren Namen nicht mit SYS beginnen (aber einschließlich SYSTEM).
5. LOADNCP... wird für Natural-Kommandoprozessor-Quellcodes verwendet.

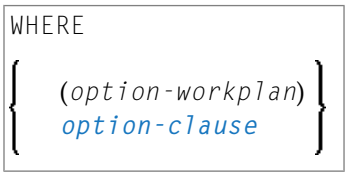
115 option-setting

- Syntax der option-setting-Klausel 1038
- Erläuterung der Schlüsselwörter für option-setting 1040

Die *option-setting*-Klausel wird verwendet, um die Standardwerte der Object Handler-Kommando-Optionen zu ändern.

Die Syntax, die für die *option-setting*-Klausel gilt, wird im folgenden Abschnitt dargestellt und erläutert. Die in den Syntaxdiagrammen enthaltenen Schlüsselwörter und Variablenwerte stellen die Parameter dar, die zur Angabe der Standardwerte verwendet werden. Ein Variablenwert (wenn angegeben) muss zusammen mit einem Schlüsselwort angegeben werden.

Syntax der option-setting-Klausel

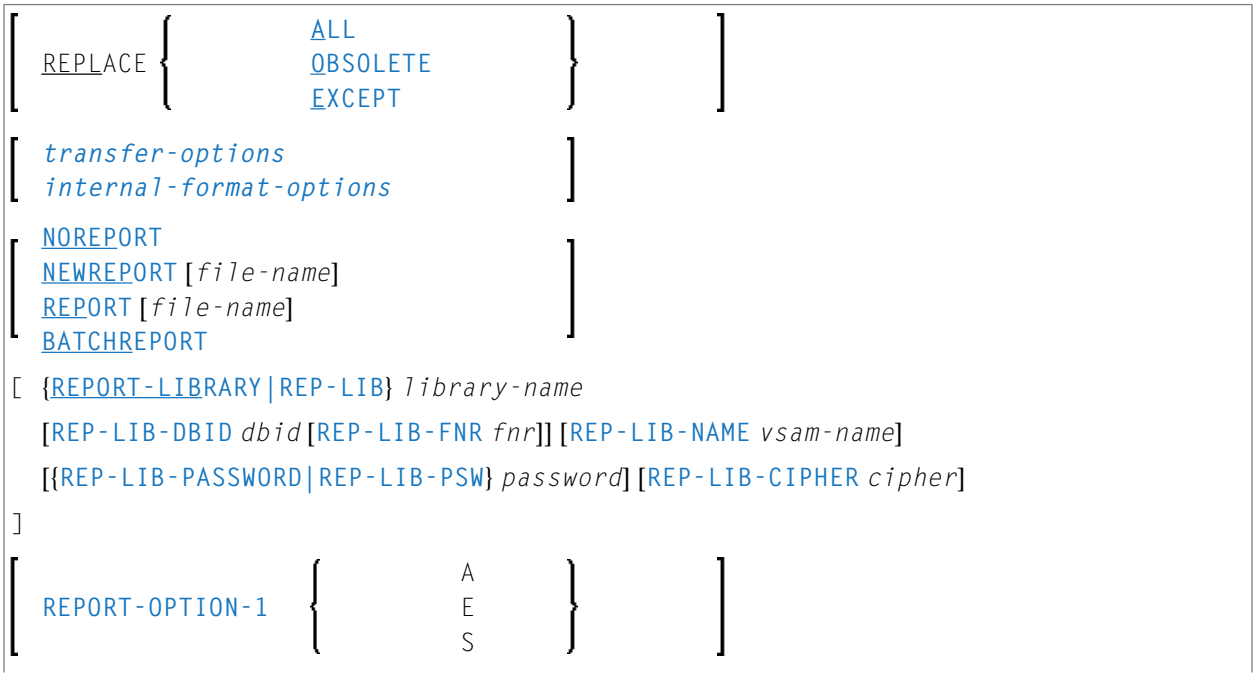


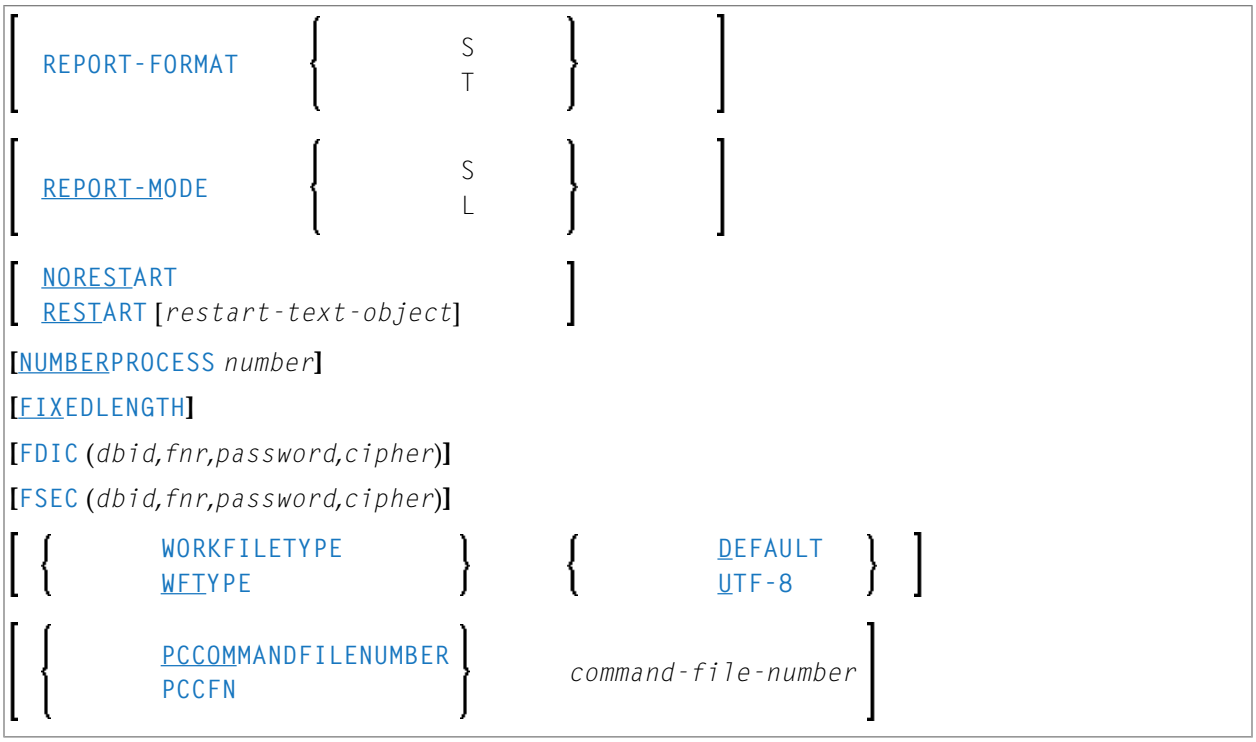
Das Syntaxdiagramm, das für *option-workplan* gilt, ist unter ***Arbeitspläne referenzieren*** im Kapitel *Arbeitspläne (Workplans)* dargestellt und beschrieben.

Die Syntax der *option-clause* ist im folgenden Abschnitt dargestellt.

- Syntax der option-clause

Syntax der option-clause



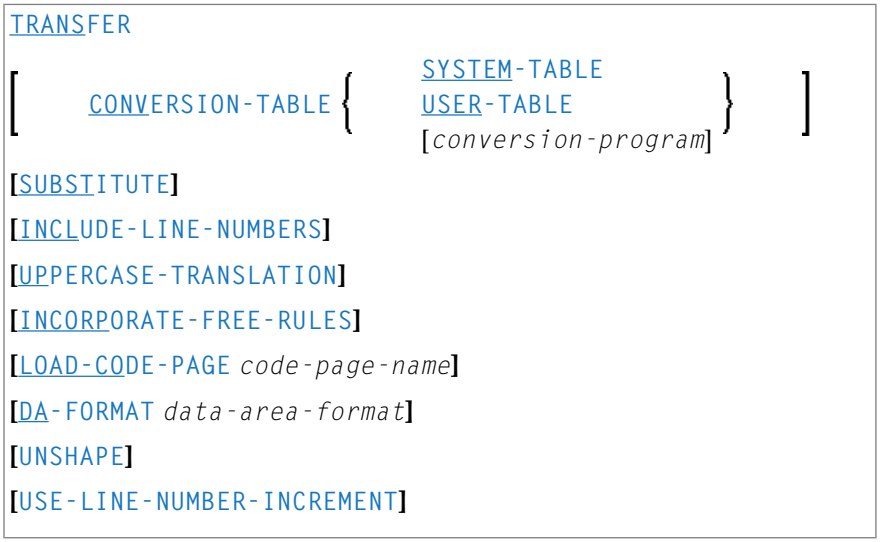


Trennzeichen

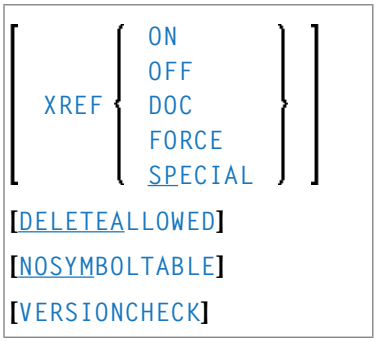
Kommas müssen als Trennzeichen zwischen den Werten nach den Schlüsselwörtern `FDIC` und `FSEC` verwendet werden, oder wenn ein Wert weggeleassen wird. Zum Beispiel: `FDIC (10,21,,2a)`.

Wenn der Session-Parameter `ID` (siehe *ID - Input-Begrenzungszeichen* in der *Parameter-Referenz*-Dokumentation) auf ein Komma gesetzt wurde, müssen Sie einen Schrägstrich (/) als Trennzeichen zwischen den Werten verwenden.

transfer-options



internal-format-options



Erläuterung der Schlüsselwörter für option-setting

Im folgenden Abschnitt werden die Schlüsselwörter und die Variablenwerte (sofern relevant) für *option-setting* erläutert:

Option	Erläuterung	Nur für Kommando
REPLACE	Ersetzt vorhandene Objekte entsprechend der angegebenen Option: ALL — Alle Objekte (Standardeinstellung). OBSOLETE — Alle Objekte mit einem Datum, das älter ist als das Datum des Objekts in der Ladedatei. EXCEPT — Alle Objekte mit Ausnahme derjenigen, deren Datum neuer ist als das Datum des Objekts in der Ladedatei.	LOAD LOADALL
TRANSFER	Setzt den Übertragungsmodus. Die Daten werden im Transfer-Format gelesen und geschrieben. Gültige Optionen finden Sie unter ***Erläuterung der Schlüsselwörter für transfer-options***.	UNLOAD LOAD SCAN
NOREPORT	Gibt die Einstellung für die Reportdatei an: Es werden keine Daten in einer Reportdatei aufgezeichnet. Dies ist die Standardeinstellung für die KommandosFIND und FINDLIB.	
NEWREPORT	Gibt die Einstellung für die Reportdatei an: Die Reportdaten werden aufgezeichnet und in ein Natural Text Member geschrieben, das in der Arbeitsplan-Bibliothek gespeichert ist. Eine vorhandene Datei wird überschrieben.	
REPORT	Gibt die Einstellung für die Reportdatei an: Die Reportdaten werden aufgezeichnet und in ein Natural Text Member geschrieben, das in der Arbeitsplan-Bibliothek gespeichert	

Option	Erläuterung	Nur für Kommando
	ist. Dies ist die Standardeinstellung für die die Kommandos UNLOAD, LOAD, LOADALL, SCAN, SCANALL und DELETE. Zum Ändern der Standardeinstellungen siehe Profilparameter DELETE-TEMPORARY-REPORT-TEXT.	
BATCHREPORT	Gibt die Reporteinstellung für die Batch-Verarbeitung oder bei Verwendung der Anwendungsprogrammierschnittstelle OBJHAPI an: Die Reportdaten werden entweder nach SYSOUT geschrieben oder auf dem Bildschirm ausgegeben (die Reportdaten werden *nicht* in eine Datei geschrieben).	
REPORT-LIBRARY oder REPORT-LIB oder REP-LIB	Gibt die Library an, in der das Reporttextobjekt gespeichert wird, wenn die Option REPORT oder NEWREPORT angegeben ist. Wenn diese Option nicht angegeben ist, wird die Library WORKPLAN oder die im Object-Handler-Profilparameter Workplan-Library angegebene Library verwendet. Siehe auch *SYSOBJH-Profilparameter*.	
REP-LIB-DBID	Die Datenbankkennung der Systemdatei, in der sich die REPORT-LIBRARY befindet. (0 ist die aktuelle FNAT/FUSER.)	
REP-LIB-FNR	Die Dateinummer der Systemdatei, in der sich die REPORT-LIBRARY befindet. (0 ist die aktuelle FNAT/FUSER.)	
REP-LIB-NAME	Der VSAM-Name der Systemdatei, in der sich die REPORT-LIBRARY befindet.	
REP-LIB-PASSWORD oder REP-LIB-PSW	Das Adabas-Passwort der Systemdatei, in der sich die REPORT-LIBRARY befindet.	
REP-LIB-CIPHER	Der Adabas-Chiffrierschlüssel für die Systemdatei, in der sich die REPORT-LIBRARY befindet.	
REPORT-OPTION-1 oder REPOPT1 oder REP-OPT-1	Gibt die Reportoption an, die verwendet werden soll, wenn ein Direktkommando ausgeführt wird und ein Report geschrieben werden soll: A Alle Reportelemente anzeigen (Standardeinstellung). E Nur Fehlermeldungen anzeigen. Dazu gehören Meldungen von Natural Security und Meldungen, die bei der Ausführung eines LOAD-Kommandos aufgetreten sind, z.B. „not replaced“ (nicht ersetzt).	UNLOAD LOAD SCAN DELETE

Option	Erläuterung	Nur für Kommando
	S Nur für den Batch-Modus. Der Fehlerreport wird in zwei Teile aufgeteilt: Reportelemente außer Fehlermeldungen werden auf das Standard-Report-Gerät (REPORT(0)/CMPRINT) geschrieben, Fehlermeldungen (einschließlich Meldungen von Natural Security und Meldungen, die während der Ausführung eines LOAD-Kommandos aufgetreten sind) werden auf das zweite Report-Gerät (REPORT(1)/CMPRT01) geschrieben. Beachten Sie, dass S im Online-Modus die gleiche Wirkung wie A hat.	
REPORT-FORMAT oder REPFRMT	Gibt das Reportformat an, das verwendet werden soll, wenn ein Direktkommando ausgeführt wird und ein Report geschrieben werden soll: S Anzeige der Quelldaten im Entladereport, d.h. der Daten des entladenen Objekts, bevor Parameter (z.B. der Library-Name) geändert werden (Standardeinstellung). T Anzeige der Zieldaten im Entladereport, d. h. der Daten nach Änderung von Parametern (z. B. des Library-Namens).	UNLOAD
REPORT-MODE oder REPM	Gibt den Reportmodus an, der verwendet werden soll, wenn ein Direktkommando ausgeführt wird und ein Report geschrieben werden soll: S Einen Kurzbericht schreiben, d.h. die wichtigsten Daten werden in den ersten 80 Spalten der Reportzeile mit kurzen Begrenzungszeichen angezeigt. L Einen langen Bericht schreiben, d.h. die Daten werden in der ursprünglichen Reihenfolge mit langen Begrenzungszeichen angezeigt (Standardeinstellung).	
NORESTART	Es werden keine Restart-Informationen in eine Datei geschrieben.	LOAD
RESTART	Die Restart-Informationen werden in ein Natural Text Member geschrieben, das in der Arbeitsplan-Bibliothek gespeichert ist.	LOAD
NUMBERPROCESS	Gibt die Anzahl der zu verarbeitenden Objekte an. Das Kommando LOAD oder SCAN stoppt die Ausführung nach der angegebenen Anzahl.	LOAD SCAN
FIXEDLENGTH	Legt das Format der Entladearbeitsdatei auf eine maximale Satzlänge fester Größe fest. Jeder Datensatz enthält 256 Bytes, wenn er im internen Format geschrieben wird, oder 100 Bytes im Transferformat.	UNLOAD

Option	Erläuterung	Nur für Kommando
FDIC	Gibt die Systemdatei FDIC an, die für die Verarbeitung verwendet werden soll: Datenbankkennung (*dbid*), file number (*fnr*), Passwort (*password*) und Chiffrierschlüssel (*cipher*) der Adabas-Datei. Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle FDIC-Systemdatei verwendet.	UNLOAD LOAD DELETE
FSEC	Gibt die Systemdatei FSEC an, die für die Verarbeitung verwendet werden soll: Datenbankkennung (*dbid*), Dateinummer (*fnr*), Passwort (*password*) und Chiffrierschlüssel (*cipher*) der Adabas-Datei. Wenn keine Werte (oder 0) angegeben werden, wird die aktuelle FSEC verwendet.	UNLOAD LOAD DELETE
WORKFILETYPE oder WFTYPE	Der Arbeitsdateityp von Natural Work File 1, wenn Daten gelesen und in die Arbeitsdatei geschrieben werden: DEFAULT Standardmäßige binäre Arbeitsdatei. UTF-8 Unicode/UTF-F8-kodierte binäre Arbeitsdatei. UTF-8 gilt nur für die Entladefunktion und wenn TRANSFER angegeben ist. Wenn UTF-8 angegeben ist, können Sie die Optionen CONVERSION-TABLE, SUBSTITUTE und INCORPORATE-FREE-RULES nicht verwenden. Wenn WORKFILETYPE nicht angegeben wurde, wird der aktuelle Typ verwendet.	UNLOAD
PC NEWPC	Gilt nur, wenn Entire Connection installiert ist. Schreibt Daten in eine Entire-Connection-Arbeitsdatei oder liest Daten daraus. *file-name* bezeichnet den vollständigen Pfadnamen, der der Entire-Connection-Arbeitsdatei zugeordnet ist. Wenn Ihre Systemumgebung keinen Backslash (\) als Trennzeichen akzeptiert, verwenden Sie stattdessen einen Schrägstrich (/). Wenn Sie *file-name* nicht angeben, werden Sie von Entire Connection aufgefordert, den Namen einer Arbeitsdatei einzugeben. Wenn NEWPC angegeben ist, überschreiben die entladenen Daten den Inhalt der vorhandenen Arbeitsdatei oder füllen eine neue Arbeitsdatei ab der obersten Zeile. Andernfalls werden die Daten angehängt. Siehe auch ***Zuweisung von Arbeitsdateien*** in *Arbeitsdateien (Work Files)*.	UNLOAD LOAD SCAN

Option	Erläuterung	Nur für Kommando
PCCOMMANDFILENUMBER oder PCCOM oder PCCFN	Gilt nur, wenn Entire Connection installiert ist. Gibt die Nummer der Arbeitsdatei an, die für die Verarbeitung von Entire Connection-Kommandos verwendet wird. Der Standardwert ist 8 für Work File 8, das als Entire Connection-Arbeitsdatei definiert sein muss. Siehe auch ***Zuweisung von Arbeitsdateien*** in *Arbeitsdateien (Work Files)*.	UNLOAD LOAD SCAN

Die Schlüsselwörter und die variablen Werte (falls relevant) von *transfer-options* und *internal-format-options* werden im folgenden Abschnitt erläutert.

- Erläuterung der Schlüsselwörter für transfer-options
- Erläuterung der Schlüsselwörter für internal-format-options

Erläuterung der Schlüsselwörter für transfer-options

Wenn Sie das Schlüsselwort TRANSFER verwenden, können Sie die folgenden Optionen angeben:

Option	Erläuterung	Nur für Kommando
CONVERSION-TABLE	Konvertiert Daten, die im Transferformat verarbeitet werden, unter Verwendung einer der folgenden Konvertierungstabellen: SYSTEM-TABLE: Die Natural-interne Konvertierungstabelle. USER-TABLE: Eine benutzerdefinierte Konvertierungstabelle, wenn ein Konvertierungsprogramm (*conversion-program*) angegeben wurde. Dieses Programm muss in der Library SYSOBJH oder einer ihrer Steplibs abgelegt sein. Siehe die Beispielprogramme OTNCONAE und OTNCONEA in der Library SYSOBJH.	UNLOAD LOAD SCAN
SUBSTITUTE	Ersetzt beim Entladen im Transferformat die Zeilenreferenzen durch Labels. Diese Option gilt nur, wenn Ihre Quellcode-Zeilennummern für Statement-Referenzen verwendet werden. Ist dies der Fall, werden die Zeilennummern der referenzierten Zeilen und die Zeilennummernreferenzen durch Labels ersetzt. Die Quellcodes werden in der Datenbank nicht verändert.	UNLOAD

Option	Erläuterung	Nur für Kommando
INCLUDE-LINE-NUMBERS	Überträgt Zeilennummern während des Entladens im Transferformat. Standardmäßig werden Zeilennummern in Natural-Objekten *nicht* entladen.	UNLOAD
USE-LINE-NUMBER-INCREMENT oder USE-LNI	UNLOAD Wenn die Option INCLUDE-LINE-NUMBERS nicht angegeben ist, wird das Zeilennummerninkrement von Natural-Quellcodeobjekten entladen. Standardmäßig wird das Zeilennummerninkrement in Natural-Quellobjekten *nicht* entladen. LOAD Wenn das Zeilennummerninkrement übertragen wurde, wird es zum Neuaufbau der Zeilennummern der Natural-Quellcodeobjekte verwendet.	UNLOAD LOAD
UPPERCASE-TRANSLATION	Setzt jeden Quellcode beim Laden im Transferformat in Großbuchstaben um. Standardmäßig wird der Quellcode in Natural-Objekten *nicht* umgesetzt.	LOAD
INCORPORATE-FREE-RULES	Übernimmt beim Entladen im Transferformat den Quellcode von Predict Free Rules, die mit einer Map verbunden sind, in den Map-Quellcode, wenn Predict installiert ist.	UNLOAD
LOAD-CODE-PAGE	Gibt die Codepage an, die für die Konvertierung von in Unicode/UTF-8 (Universal Transformation Format, 8-Bit-Form) kodierten Objektquellcodes verwendet werden soll. Wenn Sie diese Option verwenden, werden alle Objektquellcodes, die in eine Arbeitsdatei in UTF-8 entladen werden, mit der angegebenen Codepage konvertiert, wenn sie in eine Arbeitsdatei geladen werden. Wenn Sie *CODEPAGE als *code-page-name* angeben oder wenn *code-page-name* nicht angegeben ist, wird der der Systemvariablen *CODEPAGE zugewiesene Wert verwendet (siehe *Systemvariablen*-Dokumentation). Wenn LOAD-CODE-PAGE angegeben ist, können Sie die Optionen CONVERSION-TABLE und UPPERCASE-TRANSLATION nicht verwenden.	LOAD LOADALL
DA-FORMAT	Legt die Formatkonvertierung von Datenbereichsquellcodes fest. Mögliche Werte:	UNLOAD LOAD

Option	Erläuterung	Nur für Kommando
	N Wandelt Datenbereiche in das neue interne Datenbereichsformat um. O Wandelt Datenbereiche in das alte interne Datenbereichsformat um. D Wandelt Datenbereiche in das DEFINE DATA-Format um. Die in der Arbeitsdatei gespeicherten Datendefinitionen liegen in einer besser lesbaren Form vor. Ansonsten keine Auswirkung auf die Datenübertragung. * Keine Umwandlung von Datenbereichen. Dies ist die Standardeinstellung. Siehe **Data area format** in *Übertragungsoptionen* in *Einstellungen (Settings)*.	
UNSHAPE	Ersetzt geformte arabische Zeichen (Codepage IBM420) durch die entsprechenden ungeformten Zeichen in den folgenden Quellcodeobjekten: Natural-Programme, benutzerdefinierte Fehlermeldungen und Systemfehlermeldungen.	UNLOAD LOAD

Erläuterung der Schlüsselwörter für internal-format-options

Wenn Sie *internal-format-options* verwenden, können Sie die folgenden Optionen angeben:

Option	Erläuterung		Nur für Kommando
XREF	Gilt nur, wenn Predict installiert ist. Lädt oder entlädt XRef-Daten von katalogisierten Natural-Objekten. Sie können einen der folgenden Werte angeben:		LOAD UNLOAD
	ON	UNLOAD: Entlädt katalogisierte Objekte und ihre Cross-Referenzdaten (falls vorhanden). LOAD: Lädt katalogisierte Objekte und deren Cross-Referenzdaten, wenn Cross-Referenzen in der Arbeitsdatei vorhanden sind.	
	OFF	Es werden keine Cross-Referenzdaten verarbeitet. Dies ist die Standardeinstellung.	

Option	Erläuterung		Nur für Kommando
	DOC	Gilt nur für LOAD. Lädt katalogisierte Objekte und ihre Cross-Referenzdaten (falls vorhanden) nur, wenn Predict-Einträge für die Objekte in der FDIC-Systemdatei vorhanden sind.	
	FORCE	Gilt nur für LOAD. Lädt katalogisierte Objekte und ihre Cross-Referenzdaten nur, wenn Cross-Referenzen in der Arbeitsdatei existieren und wenn Predict-Einträge für die Objekte in der FDIC-Systemdatei vorhanden sind.	
	SPECIAL	Gilt nur für LOAD. Lädt katalogisierte Objekte und deren Cross-Referenzdaten (falls vorhanden).	
DELETEALLOWED	Verarbeitet Löschanweisungen aus Arbeitsdateien beim Laden von Objekten im internen Format.		LOAD
NOSYMBOLTABLE	Entlädt katalogisierte Natural-Library-Objekte ohne ihre entsprechenden internen Natural-Symboltabellen. Dadurch wird der benötigte Plattenspeicherplatz reduziert. Dies ist jedoch nur für eine Produktionsumgebung sinnvoll, da dann einige Anwendungsentwicklungsfunktionen, die die Symboltabellen benötigen, nicht zur Verfügung stehen. Außerdem ist der Profilparameter RECAT=ON (siehe *Parameter-Referenz*-Dokumentation) nicht anwendbar.		UNLOAD
VERSIONCHECK	Prüft die Natural-Version des zu ladenden katalogisierten Objekts. Dabei wird die Natural-Version, unter der die Objekte katalogisiert und in die Arbeitsdatei geschrieben wurden, mit der aktuellen Natural-Version verglichen. Objekte, die unter einer höheren Natural-Version als der aktuellen katalogisiert wurden, werden abgelehnt. VERSIONCHECK kann nur verwendet werden, wenn die Daten im internen Format geladen werden, d.h. wenn die Option TRANSFER *nicht* angegeben ist.		LOAD

116 Beispiele für die Verwendung von Object

Handler-Direktkommandos

- Entladen von Objekten für dieselbe Plattform 1050
- Entladen von Objekten für verschiedene Plattformen 1051
- Laden von Objekten im internen Format 1052
- Laden von Objekten im Transferformat 1052

Dieses Kapitel enthält einige Beispiele für die Verwendung von Object Handler-Direktkommandos.

Tipp: Für weitere Beispiele können Sie sich das für eine Object Handler-Funktion generierte Kommando ansehen. Dieses Kommando wird automatisch angezeigt, wenn Sie einen Assistenten verwenden. Im Expertenmodus können Sie die Anzeige des Kommandos aktivieren, indem Sie entweder das Object Handler-Kommando `SET ADVANCEDCMD ON` eingeben oder den Parameter `Display-Cmd-in-Advanced-Mode` im Object Handler-Profil auf `Y` (Yes) setzen (siehe auch *Profileinstellungen*).

Entladen von Objekten für dieselbe Plattform

In diesem Abschnitt finden Sie Beispiele, wie Sie Objekte im internen Format in eine Arbeitsdatei entladen können, um sie auf derselben Plattform, entweder in einer lokalen Großrechner-, UNIX- oder Windows-Umgebung, zu laden:

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) aus der Library `ABC`:

```
UNLOAD * LIB ABC OBJTYPE N SCKIND S
```

- Entlade alle Natural-Programmierobjekte (nur katalogisierte Objekte) aus der Library `ABC`:

```
UNLOAD * LIB ABC OBJTYPE N SCKIND C
```

- Entlade alle Natural-Programmierobjekte (katalogisierte Objekte und Quellcodeobjekte) aus der Library `ABC`:

```
UNLOAD * LIB ABC OBJTYPE N SCKIND A
```

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) aus der Library `ABC` mit dem Datum 2019-10-01 als Katalogierungsdatum (wenn sowohl Quellcodeobjekte als auch katalogisierte Objekte existieren) und als Quellcode-Datum (wenn nur ein Quellcodeobjekt existiert):

```
UNLOAD * LIB ABC OBJTYPE N SCKIND S DATE 2019-10-01 DATECHECK C
```

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) aus der Library `ABC`, um sie in die Library `ABCNEW` zu laden:

```
UNLOAD * LIB ABC OBJTYPE N SCKIND S WITH NEWLIBRARY ABCNEW
```

- Auf einem Großrechner: Entlade alle DDMs, deren Namen mit `EMP` beginnen und die auf die Datenbank `88` zeigen:

```
UNLOAD EMP* LIB * OBJTYPE D DDMDBID 88
```

- Unter UNIX oder Windows: Entlade alle DDMs, deren Namen mit `EMP` beginnen und die auf die Datenbank `88` zeigen:

```
UNLOAD EMP* LIB * OBJTYPE N NATTYPE V DDMDBID 88
```

- Unter UNIX oder Windows: Entlade alle DDMs, deren Namen mit `EMP` beginnen, aus der Library `VLIB` und lade sie in die Library `VLIBNEW`:

```
UNLOAD EMP* LIB VLIB OBJTYPE N NATTYPE V WITH NEWLIBRARY VLIBNEW
```

- Entlade alle benutzerdefinierten Fehlermeldungen aus der Library `ERRLIB`, um sie in die Library `NEWERR` zu laden:

```
UNLOAD * LIB ERRLIB OBJTYPE E SLKIND A WITH NEWLIBRARY NEWERR
```

- Unter Windows: Entlade alle Natural-Programmierobjekte (katalogisierte Objekte und Quellcodeobjekte) aus der Library `ABC` in eine portable Arbeitsdatei auf einem PC:

```
UNLOAD * LIB ABC OBJTYPE N WHERE WORKFILE C:\WF1.SAG WORKFILETYPE PORTABLE
```

  oder

```
UNLOAD * LIB ABC OBJTYPE N WHERE WORK C:\WF1.SAG WFT P
```

Entladen von Objekten für verschiedene Plattformen

Dieser Abschnitt enthält Kommandobeispiele für das Entladen von Objekten im Transferformat in eine Arbeitsdatei, um sie auf einer anderen Plattform zu laden, z. B. Entladen auf einem Großrechner und Laden in einer UNIX- oder Windows-Umgebung.

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) mit Objekttyp N (Subprogramm) aus der Library `ABC`:

```
UNLOAD * LIB ABC OBJTYPE N WHERE TRANSFER
```

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) und benutzerdefinierte Fehlermeldungen aus der Library `ABC`:

```
UNLOAD * LIB ABC WHERE TRANSFER
```

- Entlade alle Natural-Programmierobjekte (nur Quellcodeobjekte) mit fester Satzlänge aus der Library `ABC`:

```
UNLOAD * LIB ABC OBJTYPE N WHERE TRANSFER FIXEDLENGTH
```

Laden von Objekten im internen Format

Dieser Abschnitt enthält Kommandobeispiele für das Laden von Objekten aus einer Arbeitsdatei im internen Format.

- Lade alle Objekte in die Library `LIBNEW` und ersetze alle, die bereits existieren:

```
LOADALL WITH NEWL LIBNEW WHERE REPLACE ALL
```

- Lade alle Objekte mit Ziel-Library `TGTLIB` in die neue Ziel-Library `NEWTGT`.

```
LOAD * LIB TGTLIB WITH NEWLIBRARY NEWTGT
```

- Lade nur die benutzerdefinierten Fehlermeldungen `1000` bis `1500` aus der Library `ERRLIB`:

```
LOAD * LIB ERRLIB OBJTYPE E FMNUM 1000 TONUM 1500
```

Laden von Objekten im Transferformat

Dieser Abschnitt enthält Kommandobeispiele für das Laden von Objekten aus einer Arbeitsdatei im Transferformat.

- Lade alle Objekte in die Library `LIBNEW` und ersetze alle, die bereits existieren:

```
LOADALL WITH NEWL LIBNEW WHERE TRANSFER REPLACE ALL
```

- Lade alle Objekte mit der Ziel-Library `TGTLIB` in die neue Ziel-Library `NEWTGT`:

```
LOAD * LIB TGTLIB WITH NEWLIBRARY NEWTGT WHERE TRANSFER
```

117 Kommandos für die Navigation und für spezielle Funktionen

Object Handler-Kommandos in CUI-Umgebungen (Character User Interface, zeichenorientierte Benutzeroberfläche) dienen hauptsächlich der Navigation und der Einstellung spezieller Funktionen, z.B. der Angabe von Trace-Dateien.

Ein Object Handler-Kommando können Sie in der Kommandozeile eines beliebigen Object Handler-Bildschirms eingeben.



Anmerkung: Wenn Sie ein Natural-Systemkommando von einem Object Handler-Bildschirm aus ausführen wollen, müssen Sie vor dem Systemkommando zwei Schrägstriche (//) eingeben. Beachten Sie, dass bei der Eingabe eines Natural-Systemkommandos der Object Handler beendet wird.

Um das Object Handler Commands-Menü aufzurufen:

- Wählen Sie PF10 (Cmds).

  Oder:

  Geben Sie in einem beliebigen Object Handler-Bildschirm in der Kommandozeile das folgendes Kommando ein:

  ```
  CMDS
  ```

Die Kommandos des Object Handlers sind unten aufgeführt. Ein unterstrichener Teil eines Schlüsselworts stellt eine zulässige Abkürzung dar, Sub *n* bedeutet ein Unterkommando.

Kommando	Sub 1	Sub 2	Erläuterung
CANCEL			Abbruch der aktuellen Funktion und Anzeige des Object Handler-Hauptmenüs (**Main Menu**).
CHANGE	REPORT	LIBRARY	Aufruf der Funktion **Administration** und Anzeige des Bildschirms, in dem Sie die Report-Bibliothek (Report Library) ändern können.
CHANGE	WORKPLAN	LIBRARY	Aufruf der Funktion **Administration** und Anzeige des Bildschirms, in dem Sie die Arbeitsplan-Bibliothek (Workplan Library) ändern können.
CLEAR			Zurücksetzen der aktuellen Inhalte der Eingabefelder der Maske auf die Standardwerte.
CMDS oder COMMANDS			Aufruf des Bildschirms **Commands**.
BYE			Beendigung des Object Handler.
EXIT			
QUIT			
.			
FIN			Beendigung des Object Handler und der Natural-Sitzung.
GO	HOME		Anzeige des Object Handler-Hauptmenüs (**Main Menu**).
GO	UNLOAD		Aufruf der Entladefunktion.
GO	UNLOAD	END	Beendigung der aktuellen Entladefunktion.
		ERROR	Aufruf der Entladefunktion für Natural-Systemfehlermeldungen.
		DDM	Aufruf der Entladefunktion für DDMs.
		FDT	Aufruf der Entladefunktion für FDTs.
		LIBRARY	Aufruf der Entladefunktion für Natural-Library-Objekte.
		NCP	Aufruf der Entladefunktion für Natural-Kommandoprozessor-Quellcodes.
		RELATED	Aufruf der Entladefunktion für Natural-zugehörige Objekte (Natural Related-Objects).
		SELECTION oder LIST	Anzeige eines Bildschirms, in dem Sie den für die Entladefunktion zu verwendenden Arbeitsplan des Typs SELECTION oder LIST auswählen können.
GO	LOAD		Aufruf der Ladefunktion.
GO	LOAD	ALL	Aufruf der Ladefunktion für alle in der Arbeitsdatei enthaltenen Objekte.
		END	Beendigung der Ladefunktion.
		ERROR	Aufruf der Ladefunktion für Natural-Systemfehlermeldungen.

Kommando	Sub 1	Sub 2	Erläuterung
		DDM	Aufruf der Ladefunktion für DDMs.
		FDT	Aufruf der Ladefunktion für FDTs.
		LIBRARY	Aufruf der Ladefunktion für Natural-Library-Objekte.
		NCP	Aufruf der Ladefunktion für Natural-Kommandoprozessor-Quellcodes.
		RELATED	Aufruf der Ladefunktion für Natural-zugehörige Objekte (Natural Related-Objects).
		SELECTION oder LIST	Anzeige eines Bildschirms, in dem Sie den für die Ladefunktion zu verwendenden Arbeitsplan des Typs SELECTION oder LIST auswählen können.
GO	RESTART		Anzeige eines Bildschirms, in dem Sie das für die Restart Load-Funktion zu verwendende Textobjekt angeben können.
GO	SCAN		Aufruf der Funktion **Scan**.
GO	SCAN	ALL	Aufruf der Funktion **Scan** für alle in der Arbeitsdatei enthaltenen Objekte.
		END	Beendigung der Funktion **Scan**.
		ERROR	Aufruf der Funktion **Scan** für Natural-Systemfehlermeldungen.
		DDM	Aufruf der Funktion **Scan** für DDMs.
		FDT	Aufruf der Funktion **Scan** für FDTs.
		LIBRARY	Aufruf der Funktion **Scan** für Natural-Library-Objekte.
		NCP	Aufruf der Funktion **Scan** für Natural-Kommandoprozessor-Quellcodes.
		RELATED	Aufruf der Funktion **Scan** für Natural-zugehörige Objekte (Natural Related-Objects).
		SELECTION oder LIST	Anzeige eines Bildschirms, in dem Sie den für die Funktion **Scan** zu verwendenden Arbeitsplan des Typs SELECTION oder LIST eingeben oder auswählen können.
GO	ADMIN		Aufruf der Funktion **Administration**
GO	ADMIN	CHANGE	Anzeige eines Bildschirms, in dem Sie die Arbeitsplan-Bibliothek ändern können.
		CREATE	Aufruf eines Menüs, in dem Sie einen Arbeitsplan erstellen können.
		LIST	Generierung einer Liste der in der Arbeitsplan-Bibliothek vorhandenen Arbeitspläne.
		REPORT	Anzeige eines Bildschirms, in dem Sie die Report-Library ändern können.
GO	VIEW		Anzeige der Funktion **View**.

Kommando	Sub 1	Sub 2	Erläuterung
GO	VIEW	DDM	Aufruf der Funktion **View** für DDMs.
		ERROR	Aufruf der Funktion **View** für Natural-Systemfehlermeldungen
		FDT	Aufruf der Funktion **View** für FDTs.
		LIBRARY	Aufruf der Funktion **View** für Natural-Library-Objekte.
		NCP	Aufruf der Funktion **View** für Natural-Kommandoprozessor-Quellcodes
		RELATED	Aufruf der Funktion **View** für Natural-zugehörige Objekte (Natural Related-Objects).
GO	FIND		Aufruf der Funktion **Find**.
GO	FIND	DDM	Aufruf der Funktion **Find** für DDMs.
		ERROR	Aufruf der Funktion **Find** für Natural-Systemfehlermeldungen.
		FDT	Aufruf der Funktion **Find** für FDTs.
		LIBRARY	Aufruf der Funktion **Find** für Natural-Library-Objekte.
		NCP	Aufruf der Funktion **Find** für Natural-Kommandoprozessor-Quellcodes.
		RELATED	Aufruf der Funktion **Find** für Natural-zugehörige Objekte (Natural Related Objects).
		SELECTION oder LIST	Anzeige eines Bildschirms, in dem Sie den für die Funktion **Find** zu verwendenden Arbeitsplan des Typs SELECTION oder LIST eingeben oder auswählen können.
HELP			Aufruf der Object Handler-Hilfe-Funktion.
INIT			Reinitialisierung der Object Handler Utility.
READ	PROFILE		Aktualisierung der im Textobjekt PROFILE definierten Einstellungen für den Object Handler (siehe auch *Profileinstellungen*).
SET	ADVANCEDCMD	ON	Aktivierung der Anzeige von Kommandos, die vom Object Handler im Expertenmodus (Advanced User) und Compact-Eingabemodus generiert werden.
		OFF	Dektivierung der Anzeige von Kommandos, die vom Object Handler im Expertenmodus (Advanced User) und Compact-Eingabemodus generiert werden.
	EXECUTIONMSG	ON	Aktivierung eines Fensters zur Anzeige des Verarbeitungsstatus.
		OFF	Dektivierung eines Fensters zur Anzeige des Verarbeitungsstatus.
	FREE	ON	Aktivierung der Bearbeitung im Format FREE.
		OFF	Dektivierung der Bearbeitung im Format FREE.
	Input-Mode, IM	W	Auswahl des **Assistentenmodus (Wizard)**.
		A	Auswahl des **Expertenmodus (Advanced User)**.
		C	Auswahl des **Kompaktmodus (Compact Mode)**.

Kommando	Sub 1	Sub 2	Erläuterung
	TRACE	ON	Aktivierung des Trace-Modus: Die Ausgabe der Protokollierung einer jeden Aktion des Object Handler erfolgt auf dem Bildschirm.
		OFF	Deaktivierung des Trace-Modus.
		WORKFILE	Aktivierung des Trace-Modus: Die Ausgabe der Protokollierung einer jeden Aktion des Object Handler erfolgt in dem Natural-Textobjekt, das in der Arbeitsplan-Bibliothek gespeichert ist.
	TRACEFILE		Anzeige eines Bildschirms, in dem Sie das Natural-Textobjekt in der Arbeitsplan-Bibliothek angeben können, das für die Trace-Aufzeichnung verwendet werden soll.
SETTINGS			Anzeige eines Bildschirms, in dem Sie die Einstellungen für das Entladen, Laden oder Scannen angeben können.
SHOW oder DISPLAY	LAST	MESSAGE	Anzeige von letztem Interface-Rückmeldecodes und letzter Meldung von der Verarbeitungsschnittstelle des Object Handler.
		RESULT	Anzeige des letzten von der Verarbeitungsschnittstelle des Object Handler ausgegebenen Ergebnisses.
	PROFILE		Anzeigen oder Ändern des Object Handler-Profils
	REPORT		Anzeige des zuletzt erstellten Reports.
	STATISTICS		Anzeige von Statistikinformationen über die verarbeiteten Objekte.
	STATUS		Anzeige des aktuellen Status des Object Handler (Inhalte von globalen Variablen).
	TRACE	FILE	Anzeige des Natural-Textobjekts in der Arbeitsplan-Bibliothek, die die Trace-Aufzeichnung enthält.

118 Batch Condition Codes und User Exit Routinen

- Im Batch-Modus zurückgegebene Condition Codes 1060
- User Exit Routinen anwenden 1060
- Verfügbare User Exit Routinen 1061

Dieses Kapitel beschreibt die Condition Codes (Bedingungscodes), die für Object Handler-Funktionen im Batch-Modus zurückgegeben werden, und die für die Funktionsverarbeitung verfügbaren User Exit Routinen.

Im Batch-Modus zurückgegebene Condition Codes

Die Object Handler-Verarbeitung im Batch-Modus wird mit einem der folgenden Condition Codes beendet:

Condition Code	Erläuterung
0	Object Handler-Prozess erfolgreich beendet.
30	Ein interner Object Handler-Fehler ist aufgetreten.
40	Ein Fehler wurde im Object Handler-Kommando entdeckt.
50	Ein Fehler ist während der Object Handler-Verarbeitung aufgetreten.
60	Ein Natural Security-Fehler ist während der Object Handler-Verarbeitung aufgetreten.
99	Ein Natural-Fehler ist während der Object Handler-Verarbeitung aufgetreten.

User Exit Routinen anwenden

Die User Exit Routinen des Object Handlers werden als Quellcodeobjekte in der Natural System Library SYSOBJH ausgeliefert. Diese Quellcodeobjekte tragen den Namen `SRC-EX`*nn*, wobei *nn* die Nummer der User Exit Routine angibt.

Um eine User Exit Routine zu aktivieren:

- Benutzen Sie das Systemkommando `CATALOG` oder `STOW`, um das Quellcodeobjekt `SRC-EX`*nn* unter dem Namen `OBJHEX`*nn* in der Natural System Library SYSOBJH zu aktivieren.

  Es werden unterschiedliche Namen verwendet, um zu gewährleisten, dass das (eventuell nach Ihren Anforderungen modifizierte) Quellcodeobjekt und das katalogisierte Objekt der User Exit Routine nicht durch eine Update-Installation überschrieben werden.

  Detaillierte Beschreibungen der User Exit Routinen finden Sie in den Quellcodeobjekten von `SRC-EX`*nn* in der Library SYSOBJH.

Verfügbare User Exit Routinen

Folgende User Exit Routinen stehen zur Verfügung:

- OBJHEX01 bei fehlgeschlagener Verarbeitung
- OBJHEX02 für abgelehnte Objekte
- OBJHEX03 für die Standardwerte der Optionen
- OBJHEX04 für Natural-Objekttyp-Statistiken

OBJHEX01 bei fehlgeschlagener Verarbeitung

Immer wenn im Batch-Modus ein Condition Code auf einen Wert größer als `0` (Null) gesetzt wird, wird die User Exit Routine `OBJHEX01` (falls vorhanden) aufgerufen, bevor der Object Handler die Verarbeitung beendet. Mit dieser User Exit Routine können Sie festlegen, ob die Verarbeitung des Object Handler fortgesetzt oder abgebrochen werden soll. Im Falle eines Abbruchs können Sie den Condition Code ändern. Weitere Einzelheiten finden Sie im Quellcode der User Exit Routine `SRC-EX01` in der Natural System Library SYSOBJH.

OBJHEX02 für abgelehnte Objekte

Wenn die Ladefunktion des Object Handlers im Batch-Modus (mit Condition Code `0`) oder im Online-Kommandomodus erfolgreich ausgeführt wurde, aber ein oder mehrere Objekte während des Ladens abgelehnt wurden (z.B. nicht ersetzt), bevor der Object Handler die Verarbeitung beendet, wird die User Exit Routine `OBJHEX02` (falls vorhanden) aufgerufen. Mit `OBJHEX02` können Sie festlegen, ob die Verarbeitung des Object Handler fortgesetzt oder abgebrochen werden soll. Im Falle eines Abbruchs können Sie einen Condition Code setzen. Weitere Einzelheiten finden Sie im Quellcode der User Exit Routine `SRC-EX02` in der Natural System Library SYSOBJH.

OBJHEX03 für die Standardwerte der Optionen

Sie können die User Exit Routine `OBJHEX03` anwenden, um die Standardeinstellungen der Optionen für die Verarbeitung von Object Handler-Kommandos festzulegen. Dieser User-Exit wird aufgerufen, bevor ein Object Handler-Kommando verarbeitet wird. Weitere Einzelheiten finden Sie im Quellcodeobjekt der User Exit Routine `SRC-EX03` in der Natural System Library SYSOBJH.

OBJHEX04 für Natural-Objekttyp-Statistiken

Wenn die Object Handler-Funktionen **Unload**, **Load** oder **Scan** im Kommandomodus erfolgreich ausgeführt wurden, wird die User Exit Routine `OBJHEX04` (falls vorhanden) aufgerufen. Sie liefert für jeden Natural-Objekttyp Statistiken über die verarbeiteten, ersetzten, nicht ersetzten oder zurückgewiesenen Quellcodeobjekte und katalogisierten Objekte. Mit `OBJHEX04` können Sie festlegen, ob die Object Handler-Verarbeitung im Batch-Modus fortgesetzt oder abgebrochen werden soll. Im Falle eines Abbruchs können Sie einen Condition Code setzen. Weitere Einzelheiten finden Sie im Quellcode der User Exit Routine `SRC-EX04` in der Natural System Library SYSOBJH.

119 Profileinstellungen (Profile Settings)

- PF-Tasten 1065
- Zeilenkommandos 1065
- Profilparameter 1066

Natural bietet die Möglichkeit, die Standardeinstellungen Ihrer aktuellen Object Handler-Utility-Umgebung anzupassen. Zu diesem Zweck stellt Natural das Textmember OBJHPROF in der Natural-Systembibliothek SYSOBJH zur Verfügung. OBJHPROF wird verwendet, um umgebungsspezifische Standardwerte für Flags und Optionen anzugeben, die beim Aufrufen der entsprechenden Object Handler-Bildschirme erscheinen.

Um einzelne Profileinstellungen zu aktivieren:

- Geben Sie im Online-Modus das interne Object Handler-Kommando PROFILE (oder SHOW PROFILE) ein.

  Dieses Kommando ruft das Profile Maintenance Tool auf, das folgende Funktionen bietet:

  - Anzeige einer Maske mit den allgemeinen benutzerspezifischen Profilparametern und ihren aktuellen Werten.
  - Erstellen eines neunen Object Handler-Profils mit Standardwerten, die intern verwendet werden, falls kein Object Handler-Profil existiert.
  - Änderungen an den allgemeinen Einstellungen der Profilparameter im Object Handler-Profil (unter Kontrolle von Natural Security).
  - Änderungen an benutzerspezifischen Einstellungen der Profilparameter im Object Handler-Profil (unter Kontrolle von Natural Security).
  - Anzeige einer Beschreibung und Hilfeinformationen zu jedem Profilparameter.

Durch Drücken von PF4 können Sie zwischen Benutzerprofileinstellungen umschalten. Sie können ein *Zeilenkommando* verwenden, um im benutzerspezifischen Profil Parameter hinzuzufügen, zu ändern oder zu löschen.

Anmerkungen:

1. Wenn neue Parameter hinzugefügt oder Parameter verbessert werden, können Sie mit dem Profile Maintenance Tool-internen Kommando UPDATE die Änderungen im Object Handler-Profil aktualisieren. Weitere Informationen erhalten Sie nach Aufrufen von **Help**.
2. Das Object Handler-Profil selbst hat den Namen OBJHPROF und befindet sich in der Library SYSOBJH. Das für Aktualisierungen des Object Handler-Profils verwendete Standardprofil hat den Namen OBJHDEFP und befindet sich ebenfalls in der Library SYSOBJH.

Um einzelne Profileinstellungen zu deaktivieren:

- Löschen Sie in der Library SYSOBJH das Textobjekt OBJHPROF.

PF-Tasten

Folgende PF-Tasten stehen zur Verfügung:

PF-Taste	Beschreibung
PF1 (**Help**)	Die allgemeine oder oder kontextabhängige Hilfe aufrufen.
PF2 (**Print**)	Die aktuellen Profileinstellungen ausdrucken.
PF3 (**Exit**)	Die Profildaten speichern und das Profile Maintenance Tool beenden.
PF4 (**Gener**/**User**)	Zwischen allgemeinen (**Gener**) und benutzerspezifischen (**User**) Profileinstellungen umschalten.
PF5 (**Modif**)	Den (mittels Cursorauswahl markierten) Parameter in einem separaten Menü ändern.
PF6 (**--**)	An den Anfang der Parameterliste zurückblättern.
PF7 (**-**)	Eine Seite zurück blättern.
PF8 (**+**)	Eine Seite vor blättern.
PF9 (**++**)	An das Ende der Parameterliste blättern.
PF10 (**ALLDU**/**ALLAU**)	Bei jedem der Parameter das Zeilenkommando `DU` oder `AU` einfügen.
PF12 (**Canc**)	Das Profile Maintenance Tool beenden.

Zeilenkommandos

Folgende Zeilenkommandos stehen zur Verfügung:

Zeilenkommando	Beschreibung
MO	Parameterwert im erweiterten Modus ändern.
AU	Eintrag im Benutzerprofil hinzufügen.
DU	Eintrag im Benutzerprofil löschen.
DI	Beschreibung des Parameters anzeigen.
HE	Hilfe-Informationen zum Parameter anzeigen.

Profilparameter

Die folgende Tabelle zeigt die im Object Handler-Profil OBJHPROF enthaltenen Parameter, die möglichen Werte, die eingegeben werden können, und die Object Handler-Funktionen, für die die Parameter gelten. Darüber hinaus enthält die Tabelle eine kurze Beschreibung der Parameter oder einen Verweis auf den entsprechenden Abschnitt der Object Handler-Dokumentation. Parameter-Standardwerte sind unterstrichen.

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
`Input-Mode`	`W`	Unload Load Scan	***Assistenten (Wizards)***
	`A`		***Expertenmodus (Advanced User)***
	`C`		***Kompaktmodus (Compact Mode)***

`Display-Cmd-in-Advanced-Mode`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Zeigt das Object Handler- Kommando an, das für eine Funktion generiert wurde, die im Expertenmodus (Advanced User) und im Kompaktmodus ausgeführt wurde.

`Display-ExecutionMsg`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Aktiviert ein Fenster, das den Bearbeitungsstatus anzeigt.

`Display-Statistics`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Zeigt Statistiken über Objekte an, die nach der Ausführung der Funktion verarbeitet wurden (entspricht dem Direktkommando `SHOW STATISTICS`).

`Workplan-Library`	`WORKPLAN` oder eine beliebige andere Arbeitsplan-Bibliothek	Unload Load Scan Administration	***Arbeitspläne (Workplans)*** und ***Arbeitsplan-Bibliothek ändern (Change Workplan Library)*** in *Administration*
`Workplan-Library-DBID`	`0` (current FNAT/FUSER) oder eine beliebige andere Adabas-Datenbankkennung (DBID)	Unload Load Scan Administration	***Arbeitsplan-Bibliothek ändern (Change Workplan Library)***
`Workplan-Library-FNR`	`0` (current FNAT/FUSER) oder eine beliebige andere Adabas-Dateinummer (FNR)	Unload Load Scan Administration	***Arbeitsplan-Bibliothek ändern (Change Workplan Library)***
`Report-Library`	`WORKPLAN` oder eine beliebige andere Arbeitsplan-Bibliothek	Unload Load Scan	`REPORT-LIBRARY` in *Direktkommandos, option-setting*

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
`Report-Library-DBID`	`0` (current FNAT/FUSER) oder eine beliebige andere Adabas-Datenbankkennung (DBID)	Unload Load Scan	`REP-LIB-DBID` in *Direktkomma option-setting*
`Report-Library-FNR`	`0` (current FNAT/FUSER) oder eine beliebige andere Adabas-Dateinummer (FNR)	Unload Load Scan	`REP-LIB-FNR` in *Direktkomman option-setting*

`Trace`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Aufzeichnungen (Traces) in *Tool*

`Trace-Target`	`S` (Screen) oder `W` (Work file)	Unload Load Scan	***Aufzeichnungen (Traces)***

`Option-Replace`	`N`
	oder
	`Y`
	oder
	`O` (Obsolet)
	oder
	`E` (Except)
		Load	***Ersetzungsoptionen (Replace op*** in *Einstellungen (Settings)*

`Option-Transfer-Format`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Arbeitsdateiformat in *Arbeitsda (Work Files)*

`Option-Use-PC-Work-File`	`N`
	oder
	`Y`
		Unload
		Load
		Scan
			Siehe
			Use PC File in *Zusätzliche Optio festlegen (Set Additional Options)*

`Option-TR-Include-Line-Numbers`	`N`
	oder
	`Y`
		Unload	**Include line numbers** in
			Übertragungsoptionen (*Einstell (Settings)*)
			oder
			Übertragungsoptionen
			(Direktkommandos)

`Option-TR-Line-Number-Increment`	`N`
	oder
	`Y`
		Unload
		Load
			USE-LINE-NUMBER-INCREM in *Übertragungsoptionen* (Direktkommandos)

`Option-TR-Substitute`	`N`
	oder
	`Y`
		Unload	**Substitute line references** in *Übertragungsoptionen* (*Einstellur (Settings)*)
			oder

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
			SUBSTITUTE in *Übertragungsoptionen (Direktkommandos)*

`Option-TR-Translate-to-Upper`	`N`
	oder
	`Y`
		Load	**Translate to upper case** in *Übertragungsoptionen(Settings)*
			oder
			UPPERCASE-TRANSLATION in *Übertragungsoptionen (Direktkommandos)*

`Option-TR-Use-Conversion-Table`	`N` oder `S` (System Table) oder `U` (User Table)	Unload Load	**Use conversion table** in *Übertragungsoptionen (Einstellungen (Settings))* oder **CONVERSION-TABLE** in *Übertragungsoptionen (Direktkommandos)*
`Option-TR-Conv-Table-Name-LOAD`	`OTNCONEA` oder ein von einem Benutzer geschriebenen Subprogramm	Load	**Use conversion table** in *Übertragungsoptionen (Einstellungen (Settings))* oder **CONVERSION-TABLE** in *Übertragungsoptionen (Direktkommandos)*
`Option-TR-Conv-Table-Name-UNLD`	`OTNCONAE` oder ein von einem Benutzer geschriebenen Subprogramm	Unload	**Use conversion table** in *Übertragungsoptionen (Einstellungen (Settings))* oder **CONVERSION-TABLE** in *Übertragungsoptionen (Direktkommandos)*

`Option-TR-DA-Format`	`N`
	oder
	`O`
	oder
	`D`
	oder
	`*`
		Unload
		Load
			Data area format in *Übertragungsoptionen (Einstellungen (Settings))*
			oder
			DA-FORMAT in *Übertragungsoptionen (Direktkommandos)*

`Option-TR-Unicode-Work-File`	`N`
	oder
	`Y`
		UNLOAD	**Unicode work file** in *Felder auf dem Settings-Bildschirm*

`Option-TR-Load-Code-Page`	`Code page name`
	oder
	`*CODEPAGE`
		LOAD	**Use load code page** in *Übertragungsoptionen (Einstellungen (Settings))*

`Option-Write-Report`	`N`	Unload	***Reports*** in *Werkzeuge (Tools)*

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
	oder Y	Load Scan	**Write report** in *Arbeitsdatei- und Report-Optionen*
Option-Fixed-Length	N oder Y	Unload	Setzt das Format der Entlade-Arbeitsdatei auf eine maximale Satzlänge fester Größ Mögliche Werte siehe Option FIXEDLENGTH.
Default-Report-Direct-Command	B oder N oder Y	Unload Load Scan	Definiert die Art des Reports, d geschrieben werden soll, wenn Direktkommando ausgeführt w und keine Reportoption im Obj Handler- Kommando angegebe B Einen Batch-Report schreiben Reportdaten werden direkt a Ausgabegerät geschrieben. Entspricht der Option BATCHREPORT. Y Report oder bei der Arbeit a Großrechner ein Text-Memb schreiben. Entspricht der Op REPORT. N Keinen Report schreiben. Entspricht der Option NOREP
Default-Report-Option-1	A oder E oder S	Unload Load Scan Delete	Bestimmt die Option für den zu schreibenden Report, wenn ein Direktkommando ausgeführt w und die Option REPORT-OPTIO im Object Handler-Kommando angegeben ist. Mögliche Werte REPORT-OPTION-1
Default-Report-Format	S oder T	Unload	Bestimmt das Format des zu schreibenden Reports, wenn ei Direktkommando ausgeführt w und die Option REPORT-FORMA Object Handler-Kommando nic angegeben ist. Mögliche Werte REPORT-FORMAT
Default-Report-Mode	S oder L	Unload Load Scan	Bestimmt den Reportmodus, de verwendet werden soll, wenn e Direktkommando ausgeführt w und die Option REPORT-MODE i Object Handler-Kommando nic

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
			angegeben ist. Mögliche Werte siehe REPORT-MODE.
Default-Type-Natural-Objects	NER oder * oder eine beliebige Kombination von N, E und R		Legt den Standardwert für die Typen von Natural Library Objects (OBJTYPE) fest. Jede Kombination der gültigen Werte ist zulässig. Der Wert wird ausgewertet, wenn der Object Handler gestartet wird.
Option-Write-Restart-Info	Y oder N	Load	**Write restart information** in *Arbeitsdatei- und Report-Optionen*
Use-Option-Workplan	N oder Y	Unload Load Scan	***Arbeitspläne (Workplans)***
Option-Workplan-Name	OPTIONWP oder ein beliebiger anderer Arbeitsplan des Typs OPTION	Unload Load Scan	***Arbeitspläne (Workplans)***
Use-Parameter-Workplan	N oder Y	Unload Load	***Arbeitspläne (Workplans)***
Parameter-Workplan-Name	PARAWPLN oder ein beliebiger anderer Arbeitsplan des Typs PARAMETER	Unload Load	***Arbeitspläne (Workplans)***
Delete-Temporary-Report-Text	N oder Y	Unload Load Scan	Y = temporäre Textmember für den Report werden gelöscht, wenn der Object Handler beendet wird (Standardwert). N = temporäre Textmember für den Report werden nicht gelöscht, wenn der Object Handler beendet wird.
Work-File-1-Name	Die für die PC-Datei verwendete Arbeitsdatei (Work File 7).	Unload Load Scan	***Arbeitsdateien (Work Files)*** Gilt nur, wenn Entire Connection installiert ist.
Report-File-Name	Der Name des Textmembers in der Arbeitsplan-Bibliothek, der für den Report verwendet werden soll.	Unload Load Scan	**Write report** in *Arbeitsdatei- und Report-Optionen*

Parameter	Mögliche Werte	Funktion	Beschreibung bzw. Verweis auf Dokumentation
`Restart-File-Name`	Der Name des Textbausteins in der Arbeitsplan-Bibliothek, der für die Neustartinformationen verwendet werden soll.	Load	**Write restart information** in *Arbeitsdatei- und Report-Optione*
`Trace-File-Name`	Der Name des Textmembers in der Arbeitsplan-Bibliothek, der für das Trace verwendet werden soll.	Alle Funktionen	***Aufzeichnungen (Traces)*** in *Wer (Tools)*

120 Werkzeuge (Tools)

- Status 1074
- Letztes Ergebnis (Last Result) 1074
- Aufzeichnungen (Traces) 1074
- Reports 1075

Der Object Handler bietet spezielle Funktionen zum Anzeigen von Statusinformationen und Reports sowie zum Überprüfen und Ändern von Trace-Einstellungen.

Status

Zeigt die aktuell verwendeten Object Handler-Funktionen, die Benutzerumgebung, die Arbeitsplan-Bibliothek (Workplan Library) und die Einstellung der unten beschriebenen Trace-Option an.

Um den Status anzuzeigen:

- Geben Sie in der Kommandozeile eines beliebigen Object Handler-Bildschirms Folgendes ein:

  ```
  SHOW STATUS
  ```

  Siehe auch das Kommando SHOW, das in *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos* beschrieben ist.

Letztes Ergebnis (Last Result)

Zeigt das letzte interne Kommando an, das von der Verarbeitungsschnittstelle des Object Handlers ausgegeben wurde, sowie mögliche Rückgabecodes und Meldungen.

Um das letzte Ergebnis anzuzeigen:

- Geben Sie in der Kommandozeile eines beliebigen Object Handler-Bildschirms Folgendes ein:

  ```
  SHOW LAST RESULT
  ```

  Siehe auch das Kommando SHOW, das in *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos* beschrieben ist.

Aufzeichnungen (Traces)

Aktiviert oder deaktiviert die Trace-Funktion. **Traces**, zeichnet interne Object Handler-Programmabläufe auf, um Kontrollinformationen für die Fehlerdiagnose bereitzustellen. Die Trace-Option ist standardmäßig ausgeschaltet.

Um die Einstellung zu ändern:

- Verwenden Sie das Kommando `SET TRACE`, das in *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos* beschrieben ist.

Reports

Listet die geladenen, entladenen oder gescannten Objekte auf und zeichnet Fehler auf, die die Verarbeitung unterbrechen können. Siehe auch ***Arbeitsdatei- und Report-Optionen*** im Kapitel *Einstellungen (Settings)*. Die Option **Report** ist standardmäßig eingeschaltet und wird nach dem Entladen, Laden oder Scannen angezeigt.

Um den Inhalt der letzten Report-Datei anzuzeigen:

- Geben Sie in der Kommandozeile eines beliebigen Object Handler-Bildschirms Folgendes ein:

```
SHOW REPORT
```

  Siehe auch das Kommando `SHOW`, das in *Kommandos für die Navigation und für spezielle Funktionen* im Kapitel *Direktkommandos* beschrieben ist.

121 Migration von NATUNLD/NATLOAD und SYSTRANS zum Object Handler

- Konvertierung einzelner Kommandos 1078
- Verarbeitung von Kommandos mit einer User Exit Routine 1080
- Verarbeitung von SYSTRANS-Kommandos mit OBJHAPI 1080
- Nicht unterstützte SYSTRANS-Optionen 1081

Sie können von den alten Dienstprogrammen NATUNLD/NATLOAD und SYSTRANS zum Object Handler migrieren, indem Sie eine der beiden in diesem Kapitel beschriebenen Methoden anwenden.

Konvertierung einzelner Kommandos

Sie können die Direktkommandos `NATUNLD`/`NATLOAD` oder `SYSTRANS` in die entsprechenden Object Handler-Kommandos konvertieren, indem Sie die für die Migration vorgesehenen Object Handler-Kommandos verwenden. Diese Migrationskommandos konvertieren automatisch die von den alten Dienstprogrammen verwendete Kommandosyntax in die vom Object Handler verwendete Kommandosyntax.

Um ein einzelnes Kommando zu konvertieren:

1 Verwenden Sie eines der folgenden Direktkommandos des Object Handlers:

```
NATUNLD
```

gefolgt von einem `NATUNLD`-Direktkommando.

Oder:

```
NATLOAD
```

gefolgt von einem `NATLOAD`-Direktkommando.

Oder:

```
SYSTRANS
```

gefolgt von einem `SYSTRANS`-Direktkommando.

Das angegebene alte Dienstprogramm-Kommando wird in das entsprechende Object Handler-Kommando umgewandelt.

2 Geben Sie jedes nachfolgende Kommando für den Object Handler in der Syntax an, die für das Dienstprogramm NATUNLD, NATLOAD bzw. SYSTRANS gilt.

Die Syntax dieses Dienstprogramms bleibt für die Dauer der Object Handler-Sitzung gültig.

Beispiel für ein NATUNLD-Kommando:

Das folgende Beispiel zeigt zwei aufeinanderfolgende NATUNLD-Kommandos und die entsprechenden Object Handler-Kommandos.

Alte NATUNLD-Kommandos:	`NATUNLD ALL * FM LIB1 TO LIB2`
	`ALL PG* FM LIB2`
Neues Object Handler Kommando:	`SYSOBJH NATUNLD ALL * FM LIB1 TO LIB2`
Daraus folgendes Object Handler-Kommando in NATUNLD-Syntax:	`ALL PG* FM LIB2`

Beispiel für ein SYSTRANS-Kommando:

Das folgende Beispiel zeigt zwei aufeinanderfolgende SYSTRANS-Dienstprogramm-Kommandos und die entsprechenden Object Handler-Kommandos.

Alte SYSTRANS-Kommandos:	`TRANSCMD EXECUTE UNLOAD N FROM LIB1 NAME ETID`
	`END`
Neues Object Handler-Kommando:	`SYSOBJH SYSTRANS EXECUTE UNLOAD N FROM LIB1 NAME ETID END`
Daraus folgendes Object Handler-Kommando in SYSTRANS-Syntax:	`END`

Beispiel für die Batch-Verarbeitung in SYSTRANS:

Im Folgenden finden Sie ein Beispiel für die Batch-Verarbeitung eines SYSTRANS-Dienstprogramm-Kommandos unter Verwendung von Using Map-Eingabedaten sowie des entsprechenden Object Handler-Kommandos und der Eingabedaten.

Alte SYSTRANS-Batch-Sequenz:

```
SYSTRANS
U
N,N,N,Y,N,N,N,N
N
SRCLIB1,PGM1,*,TGTLIB1
```

Neue Object Handler-Batch-Sequenz:

```
SYSOBJH SYSTRANS
U
N,N,N,Y,N,N,N,N
N
SRCLIB1,PGM1,*,TGTLIB1
```

Verarbeitung von Kommandos mit einer User Exit Routine

Alternativ zur Neudefinition jedes einzelnen Dienstprogramm-Kommandos können Sie in einer User Exit Routine festlegen, dass Dienstprogramm-Kommandos zur Funktionsausführung das Dienstprogramm SYSOBJH aufrufen. Diese Routine bestimmt, ob ein an NATUNLD/NATLOD oder SYSTRANS gerichtetes Direktkommando als SYSOBJH-Kommando an den Object Handler weitergeleitet wird oder nicht.

Um die User Exit Routine zu aktivieren:

1 Für NATUNLD/NATLOAD:

Speichern Sie mit `SAVE` das Quellcodeobjekt `U-S-EX03` unter dem Namen `UNLDEX03` in der Natural System Library SYSUNLD. Das Quellcodeobjekt wird in SYSUNLD bereitgestellt.

Für SYSTRANS:

Speichern Sie mit `SAVE` das Quellcodeobjekt `TRA-E2-S` unter dem Namen `TRA-EX-2` in der Natural System Library SYSTRANS. Das Quellcodeobjekt wird in SYSTRANS bereitgestellt.

2 Öffnen Sie den Quellcode von `UNLDEX03` bzw. `TRA-EX-2` und setzen Sie `USE-SYSOBJH` auf `Y` (dies ist der eingegebene Standardwert). Mit `N` wird die User Exit Routine deaktiviert.

Sie können Bedingungen für die Verwendung festlegen, z.B. wann der Object Handler anstelle eines alten Dienstprogramms verwendet werden soll oder wer berechtigt ist, den Object Handler zu verwenden.

3 Speichern Sie das Quellcodeobjekt mit `CATALOG` oder `STOW`.

Verarbeitung von SYSTRANS-Kommandos mit OBJHAPI

Sie können die Anwendungsprogrammierschnittstelle `OBJHAPI` (die in der Natural System Library SYSOBJH enthalten ist) verwenden, um ein Object Handler-Kommando in der Syntax des SYSTRANS-Dienstprogramms auszuführen.

Wenn Sie `OBJHAPI` zu diesem Zweck verwenden, müssen Sie den Parameter `P-EXTENSIONS-EXEC-SYSTRANS-CMD` in dem Programm angeben, das `OBJHAPI` aufruft. Einzelheiten hierzu finden Sie in dem Beispielprogramm `DOC-API`, das in der Library SYSOBJH enthalten ist.

Nicht unterstützte SYSTRANS-Optionen

Der Object Handler unterstützt die folgenden Optionen des Direktkommandos `SYSTRANS` nicht:

`WORK-FILE-INPUT`, `SPECIAL-CONVERSION`, `RULE-LOAD` und `UNLOAD-RULES`.

XXI SYSPARM Utility

122 SYSPARM Utility

- Utility SYSPARM aufrufen 1086
- Profile auflisten - Funktion List Profiles 1087
- Profil anzeigen - Funktion Display Profile 1089
- Neues Profil anlegen - Funktion Add New Profile 1090
- Profil ändern - Funktion Modify Profile 1090
- Profile bearbeiten - Editier-Funktion 1091
- Profil kopieren - Funktion Copy Profile 1095
- Profil löschen - Funktion Delete Profile 1095
- Direktkommandos und Batch-Modus 1096
- Parameterprofile in unterschiedlichen Umgebungen verwalten 1106
- Hilfe zu Parametern aus der Kommandozeile aufrufen 1106

Mit dem Dienstprogramm („Utility“) SYSPARM können Sie einen Satz Natural-Profilparameter als individuelles Parameterprofil anlegen und pflegen. Ein solches Parameterprofil können Sie für jede Natural-Sitzung verwenden.

Wenn Sie Natural mit dynamischen Parametern aufrufen, können Sie jedes Mal einzelne Parameter angeben. Komfortabler ist es jedoch, wenn Sie mit `SYSPARM` einen Satz Parameter angeben, diesen Satz unter einem Profilnamen speichern und anschließend Natural mit einem dynamischen Profilparameter aufrufen:

```
PROFILE=profile-name
```

Die in diesem Parameterprofil definierten Parameter werden dann als dynamische Parameter an Natural übergeben. Sie müssen mit der Syntax des Profilparameters `PROFILE` übereinstimmen.

Siehe auch *Verwendung des Profilparameters PROFILE* weiter unten.

Mit dem Profilparameter `USER` können Sie die Benutzung eines Profils einschränken.

Utility SYSPARM aufrufen

Um die Utility SYSPARM aufzurufen:

- Setzen Sie das folgende Natural-Systemkommando ab:

  ```
  SYSPARM
  ```

  Das Menü der Utility SYSPARM wird angezeigt. Es enthält folgende Funktionen und Felder:

Feld/Funktion	Erklärung
List Profiles	Zeigt eine Liste aller Parameterprofile. In dieser Liste können Sie ein oder mehrere Profile zum Anzeigen, Ändern oder Löschen markieren.
Display Profile	Anzeigen eines bestimmten Parameterprofils.
Add New Profile	Anlegen eines neuen Parameterprofils.
Modify Profile	Ändern eines vorhandenen Parameterprofils.
Copy Profile	Anlegen eines neuen Parameterprofils durch Kopieren eines vorhandenen Parameterprofils.
Delete Profile	Löschen eines vorhandenen Parameterprofils.
Profile	Auswahl des angegebenen Parameterprofils. Geben Sie einen gültigen Profilnamen ein oder verwenden Sie Stern-Notation (*), um einen Bereich von Parameterprofilen auszuwählen. Um alle vorhandenen Parameterprofile auszuwählen, können Sie das Feld leer lassen oder einen Stern (*) eingeben.
Copy to	Kopieren des angegebenen Parameterprofils.

Feld/Funktion	Erklärung
DBID	Auswahl der Datenbankkennung (DBID) der Natural-Systemdatei, in der das Parameterprofil gespeichert ist. Standardmäßig ist das die aktuelle Systemdatei FPROF (falls gesetzt), ansonsten die aktuelle Systemdatei FNAT.
FNR	Auswahl der Dateinummer (FNR) der Natural-Systemdatei, in der das Parameterprofil gespeichert ist. Standardmäßig ist das die aktuelle Systemdatei FPROF (falls gesetzt), ansonsten die aktuelle Systemdatei FNAT.
Password	Angabe des Passworts (8 Zeichen) der Adabas-Datei, in der das Parameterprofil gespeichert ist.
Cipher	Angabe des Chiffrierschlüssels (8 Zeichen) der Adabas-Datei, in der das Parameterprofil gespeichert ist.

Die oben aufgeführten Funktionen werden in den folgenden Abschnitten beschrieben.

Profile auflisten - Funktion List Profiles

Diese Funktion dient dazu, alle in der angegebenen Natural-Systemdatei enthaltenen Parameterprofile aufzulisten. In der Liste der Parameterprofile können Sie ein oder mehrere Parameterprofile anzeigen, ändern oder löschen. Außerdem können Sie eine Eigenschaft eines Parameterprofils anzeigen, z.B. die Codepage eines Parameterprofils oder wann oder durch wen ein Parameterprofil zuletzt geändert worden ist.

Um Parameterprofile aufzulisten:

- Geben Sie im SYSPARM-Menü den Funktionscode `L` ein. Geben Sie Feld **Profile** den Namen eines Parameterprofils an oder geben Sie einen Namensbereich an:
  - Lassen Sie das Feld leer oder geben Sie einen Stern (*) ein, um alle vorhandenen Parameterprofile auszuwählen.
  - Verwenden Sie Stern-Notation (*), um alle Parameterprofile auszuwählen, deren Namen mit einem angegebenen Wert beginnen. Dabei kein der Wert ein einzelnes oder eine beliebige Kombination von Zeichen sein, z.B. `ABC*`.
  - Verwenden Sie das Größer-als-Zeichen (>), um alle Parameterprofile aufzulisten, deren Namen größer als oder gleich einem angegebenen Wert sind, z.B. `ABC>`.
  - Verwenden Sie das Kleiner-als-Zeichen (<), um alle Parameterprofile aufzulisten, deren Namen kleiner als ein angegebener Wert sind, z.B. `ABC<`.

  Der Bildschirm **List Profiles** wird angezeigt. Er enthält eine Liste aller Parameterprofile, die dem angegebenen Namensbereich entsprechen.

  Sie können die PF-Taste PF7 bzw. PF8 drücken, um in der Liste eine Seite nach unten bzw. nach oben zu blättern. Sie können die PF-Taste PF11 (> bzw. <) drücken, um in der Anzeige hin und

her zu schalten, um Folgendes zu sehen: Wann (**Date**, **Time**) und von wem (**User ID**) das Profil zuletzt geändert worden ist, welches die zum Zeitpunkt der Änderung gültige Natural-Version (**Version**) war, und die Codepage (**Code Page**) des Profils.

Anmerkung: Falls ein Eintrag für **Code Page** mehr als 17 Zeichen enthält, wird er abgeschnitten. Einen abgeschnittenen Eintrag erkennen Sie daran, dass er am Ende ein Größer-als-Zeichen (>) hat. Der vollständige Name der Codepage wird angezeigt, wenn Sie PF11 (>) noch einmal drücken.

Um ein einzelnes Parameterprofil anzuzeigen, zu ändern oder zu löschen:

- Stellen Sie den Cursor in die Spalte **Sel.** oder **Profile** des Parameterprofils, und drücken Sie eine der folgenden PF-Tasten:

PF-Taste	Zeilenkommando	Funktion
PF4	D	Ruft die Funktion **Display Profile** auf.
PF5	M	Ruft die Funktion **Modify Profile** auf.
PF6	X	Ruft die Funktion **Delete Profile** auf.

Oder:

Geben Sie in der Spalte **Sel.** neben dem Parameterprofil eines der oben aufgelisteten Zeilenkommandos ein, und drücken Sie Enter.

Je nachdem, welche PF-Taste Sie gedrückt oder welches Zeilenkommando Sie eingegeben haben, wird das gewählte Parameterprofil am Bildschirm angezeigt oder ein Bestätigungsfenster erscheint für dieses Profil.

Um mehrere Parameterprofile anzuzeigen, zu ändern oder zu löschen:

1. Geben Sie in der Spalte **Sel.** neben den Parameterprofilen, die Sie bearbeiten möchten, eines der **Tabelle** aufgelisteten Zeilenkommandos ein, und drücken Sie Enter.

   Sie können die PF-Taste PF7 bzw. PF8 drücken, um in der Liste nach unten bzw. nach oben zu blättern und zu den gewünschten Einträgen zu gelangen.
2. Drücken Sie Enter, wenn Sie alle Zeilenkommandos für alle Parameterprofile eingegeben habe.

   Je nach eingegebenem Zeilenkommando wird das markierte Parameterprofil entweder am Bildschirm angezeigt oder ein Bestätigungsfenster erscheint für dieses Profil.
3. Wenn Sie mit der Bearbeitung des ersten Parameterprofils fertig sind, drücken Sie PF3 oder PF12, um das nächste Parameterprofil zu bearbeiten. Mit PF3 wird außerdem das aktuelle Parameterprofil gespeichert, falls es geändert worden ist.

Profil anzeigen - Funktion Display Profile

Mit dieser Funktion können Sie sich ein Parameterprofil ansehen.

Um ein Parameterprofil anzuzeigen:

- Geben Sie im SYSPARM **Menu** den Funktionscode `D` und den Namen eines vorhandenen Parameterprofils ein:

  Oder:

  Markieren Sie im Bildschirm **List Profiles** ein oder mehrere Parameterprofile, siehe entsprechenden Abschnitt.

  Der Bildschirm **Display Profile** mit den Parameterdefinitionen des angegebenen Parameterprofils wird angezeigt (Beispiel):

```
14:30:54             ***** NATURAL SYSPARM UTILITY *****            2009-02-11
                     - Display Profile TESTPROF (10,2410) -
 >  /* This is a test profile. */                                            <
 >  AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(1099,1100,PASSWORD,12345678)    <
 >  RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON,DFS=(SRV2,NODE1,,ACI))          <
 >  PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)           <
 >  PRINT=((1,3,6-11,15),AM=NAF)                                             <
 >  ESIZE=90                                                                 <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <

Command ===>                                                               ↩
                                                                           ↩

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

  In der Bildschirmüberschrift wird der Name des Parameterprofils (hier: `TESTPROF`) und die Nummer der Datei (hier: `10,2410`) in der das Parameterprofil gespeichert ist, angezeigt.

Neues Profil anlegen - Funktion Add New Profile

Mit dieser Funktion können Sie ein neues Parameterprofil anlegen.

Um ein neues Parameterprofil anzulegen:

- Geben Sie im SYSPARM **Menu** den Funktionscode `A` und den Namen des Parameterprofils, das Sie anlegen wollen, an. Drücken Sie Enter.

  Es erscheint ein leerer **Bearbeitungsbildschirm**. Eine Anleitung, wie Sie die Parameter eingeben, die Sie im neuen Parameterprofil angeben wollen, finden Sie im Abschnitt *Profile bearbeiten - Editing Profiles*.

Profil ändern - Funktion Modify Profile

Mit dieser Funktion können Sie die Parameterangaben eines Parameterprofils ändern.

Um ein Parameterprofil zu ändern:

- Geben Sie im SYSPARM **Menu** den Funktionscode `M` und den Namen des Parameterprofils an. Drücken Sie Enter.

  Oder:

  Rufen Sie den Bildschirm **Display Profile** für ein vorhandenes Parameterprofil auf und drücken Sie PF5.

  Oder:

  Rufen Sie den Bildschirm **List Profiles** auf und markieren Sie ein oder mehrere Parameterprofile, siehe entsprechenden Abschnitt.

  Es erscheint ein **Bearbeitungsbildschirm** mit den Parameterdefinitionen des angegebenen Parameterprofils. Eine Anleitung, wie Sie den Bildschirm benutzen, um Parameterdefinitionen hinzuzufügen oder zu ändern, finden Sie im Abschnitt *Profile bearbeiten - Editing Profiles*.

Profile bearbeiten - Editier-Funktion

Der Bearbeitungsbildschirm der Utility SYSPARM wird angezeigt, wenn Sie die Funktion **Add New Profile** oder die Funktion **Modify Profile** ausführen.

Dieser Bildschirm dient dazu, die Parameterangaben einzugeben, die Sie in ein Parameterprofil aufnehmen wollen. Informationen zu den verfügbaren Profilparametern können Sie der *Parameter-Referenz*-Dokumentation entnehmen. Alternativ können Sie das Systemkommando `HELP *` benutzen, siehe ***Hilfe zu Parametern aus der Kommandozeile aufrufen***.

Beispiel für einen Bearbeitungsbildschirm:

```
15:13:44            ***** NATURAL SYSPARM UTILITY *****           2009-02-11
 >  /* This is a test profile. */                                          <
 >  AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(1099,1100,PASSWORD,12345678)    <
 >  RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON,DFS=(SRV2,NODE1,,ACI))          <
 >  PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)           <
 >  PRINT=((1,3,6-11,15),AM=NAF)                                             <
 >  ESIZE=90                                                                 <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
 >                                                                           <
Help with parameters .. _______             Lower=ON   (Profile name: TESTPROF)

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

Der Name des aktuellen Profils wird im Feld `Profile name` angezeigt (hier: `TESTPROF`). Benutzung des Feldes `Help with parameters` siehe Abschnitt ***Hilfe zu Parametern***.

Der Bearbeitungsbildschirm enthält 18 Eingabezeilen (markiert mit einem `>`), die maximal jeweils 72 Zeichen aufnehmen können. Die eingegebenen Einzelparameter müssen voneinander durch (ein oder mehrere) Leerzeichen oder Kommas abgetrennt werden. Sie können die Parameterangaben auf so viele Zeilen verteilen, wie Sie möchten.

Sie können in jeder Zeile einen Kommentartext eingeben. Sie können einen Kommentar aber auch auf so viele Zeilen verteilen, wie Sie möchten. Kommentare müssen durch vorangestellte `/*`-Zeichen

eingeleitet und durch abschließende */-Zeichen gekennzeichnet werden, siehe im obigen Beispiel: `/* This is a test profile. */`.

Wenn die Umsetzung in Großbuchstaben ausgeschaltet worden ist, dann wird `Lower=ON` links neben dem Feld `Profile name` angezeigt. Siehe auch Schlüsselwort `LOWER` beim `SYSPARM`-Kommando.

Dieser Abschnitt behandelt folgende Themen:

- Verwendung des Profilparameters PROFILE
- PF-Tasten
- Hilfe zu Parametern - Help with parameters

Verwendung des Profilparameters PROFILE

Falls Sie zusätzlichen Platz für die Bearbeitung benötigen oder wenn Sie verschiedene Parameterprofile kombinieren oder Parameterprofile nach Kategorien in Gruppen ordnen wollen, können Sie mehrere Parameterprofile miteinander verketten, indem Sie den Profilparameter `PROFILE` als letzten Eintrag bzw. letzte Einträge in dem bzw. den betreffenden Profil bzw. Profilen eingeben. Das erste Parameterprofil wertet dann die im zweiten Parameterprofil angegebenen Parameterzeichenketten als Teil der Zeichenkette des ersten Parameterprofils aus.

Beispiel:

`PROFILE=P2` eingegeben am Ende im Parameterprofil P1, und `PROFILE=P3` eingegeben am Ende im Parameterprofil P2 bewirkt, dass zuerst 1 und dann P2 und anschließend P3 aufgerufen wird.

Ausführliche Informationen siehe Profilparameter `PROFILE` in der *Parameter-Referenz*-Dokumentation.

PF-Tasten

Auf dem **Bearbeitungsbildschirm** stehen folgende PF-Tasten zur Verfügung:

PF-Taste	Funktion
PF4	Prüft die Parameterangaben in dem Parameterprofil auf syntaktische Korrektheit.
PF5	Speichert das Parameterprofil.
PF9	Fügt unterhalb der Zeile, in der sich der Cursor befindet, eine Leerzeile ein.
PF10	Löscht die Zeile, in der sich der Cursor befindet.
PF11	Kopiert die Zeile, in der sich der Cursor befindet.

Hilfe zu Parametern - Help with parameters

Sie können das Feld **Help with parameters** auf dem **Bearbeitungsbildschirm** benutzen, um Informationen über gültige Parameterangaben zu erhalten oder um neue Angaben in das aktuelle Parameterprofil zu übernehmen.

Um die Online-Hilfe anzuzeigen und neue Parameter zu übernehmen:

1 Geben Sie im Feld **Help with parameters** wahlweise Folgendes ein:

- Name des benötigten Parameters.
- Ein oder mehrere Zeichen mit Stern-Notation (*), z.B. `PR*` für alle Parameter, deren Namen mit dem bzw. den angegebenen Zeichen beginnen.
- Ein Stern (*) für alle verfügbaren Parameter.

2 Drücken Sie jetzt *noch nicht* Enter, wenn Sie eine neue Parameterangabe in ein vorhandenes Profil *einfügen* wollen, sondern machen Sie mit dem nächsten Schritt weiter.

Markieren Sie die Eingabezeile, unter der Sie die neue(n) Parameterangabe(n) einfügen wollen, indem Sie den Cursor in diese Zeile platzieren und dann Enter drücken.

Oder:

Drücken Sie Enter, wenn Sie eine neue Parameterangabe am Ende des Parameterprofils *anhängen* wollen.

3 Nach Beendigung des vorigen Schritts wird, je nach zuvor eingegebenem Parameterwert, Folgendes angezeigt:

- Wenn Sie einen vollständigen Namen eingegeben haben, wird ein Hilfe-Bildschirm angezeigt (s. unten).
- Wenn Sie einen Namensbereich mit Stern (*) angegeben haben, wird eine Liste aller Parameter des angegebenen Bereichs angezeigt.

  Sie können aus dieser Liste den benötigten Parameter auswählen, indem Sie ein beliebiges Zeichen in das Eingabefeld neben dem benötigten Parameter eingeben und Enter drücken.

Beispiel eines Hilfe-Bildschirms:

```
13:08:21            ***** NATURAL SYSPARM UTILITY *****           2009-02-11

  The parameter AUTO is used to cause an automatic logon at the start
  of the Natural session.

  AUTO=ON     An automatic logon is executed at the start of the Natural
              session. The value contained in the Natural system variable
              *INIT-USER is used as the user ID for the logon.
  AUTO=OFF    No automatic logon is performed.




   Enter your parameter specification:                More Help:   (+,-)
 >
 >
 >
 >
 >

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

Der obere Abschnitt des Bildschirms enthält einen Hilfe-Text zu dem angegebenen Parameter (im obigen Beispiel ist das der Parameter `AUTO`). Wenn der Text mehr als eine Seite umfasst, können Sie ein Pluszeichen (+) vor dem Feld **More Help** eingeben, um die nächste Seite anzuzeigen. Geben Sie ein Minuszeichen (-) ein, um zur vorherigen Seite zurückzukehren. Geben Sie einen Punkt (.) ein, um die Hilfe-Funktion zu beenden und zum **Bearbeitungsbildschirm** zurückzukehren.

Der untere Abschnitt des Bildschirms enthält fünf Eingabezeilen, die mit > markiert sind.

4 Drücken Sie Enter.

5 Geben Sie in die Eingabezeilen die benötigten Parameterangaben ein.

Sie können PF4 drücken, um die Angaben zu prüfen.

6 Drücken Sie PF3, um die Hilfe-Funktion zu beenden

Der **Bearbeitungsbildschirm** wird angezeigt. Er enthält die neuen Parameterangaben. Diese sind entweder am Ende des Parameterprofils angehängt oder unterhalb der Eingabezeile eingefügt, die mit dem Cursor wie in Schritt beschrieben markiert wurde.

Anmerkung: Wie Sie Hilfe zu Parametern und Subparametern aus der Kommandozeile aufrufen können, erfahren Sie im Abschnitt *Hilfe zu Parametern aus der Kommandozeile aufrufen.*

Profil kopieren - Funktion Copy Profile

Mit dieser Funktion können Sie ein Parameterprofil kopieren.

Um ein Parameterprofil zu kopieren:

1 Geben Sie im SYSPARM **Menu** Folgendes ein:

- Funktionscode `C`.
- Im Feld **Profile** den Namen des Parameterprofils, das Sie als Vorlage für das neue Parameterprofil benutzen wollen.
- Im Feld **Copy to** den Namen des neuen Parameterprofils

2 Drücken Sie Enter.

Nach erfolgreicher Beendigung des Kopiervorgangs erscheint eine entsprechende Meldung.

Profil löschen - Funktion Delete Profile

Mit dieser Funktion können Sie ein Parameterprofil löschen.

Um ein Parameterprofil zu löschen:

1 Geben Sie im SYSPARM **Menu** den Funktionscode `X` und den Namen des Parameterprofils ein, das Sie löschen wollen, und drücken Sie Enter.

Oder:

Rufen Sie den Bildschirm **List Profiles** auf und wählen Sie ein oder mehrere Parameterprofile aus.

Das Fenster **Delete a Profile** wird angezeigt. Es enthält den Namen des Parameterprofils, das Sie löschen wollen

2 Bestätigen Sie die Löschung, indem Sie den Namen des Parameterprofils in das Eingabefeld eingeben und Enter drücken.

Um den Löschvorgang abzubrechen, können Sie PF3 drücken oder das Eingabefeld leer lassen und Enter drücken.

Es erscheint eine Meldung, die entweder die erfolgreiche Löschung oder den Abbruch des Löschvorgangs anzeigt.

Direktkommandos und Batch-Modus

Sie können die zuvor beschriebenen Funktionen der Utility SYSPARM auch ausführen, indem Sie entsprechende SYSPARM-Kommandos im Batch-Modus oder Online-Modus benutzen.

Zusätzlich zu den auf den Bildschirmen der Utility SYSPARM vorhandenen Funktionen, stehen im Batch-Modus Optionen zur Verfügung, mit denen Sie Kommentartexte, das Eingabebegrenzungszeichen und das Input-Zuweisungszeichen angeben können.

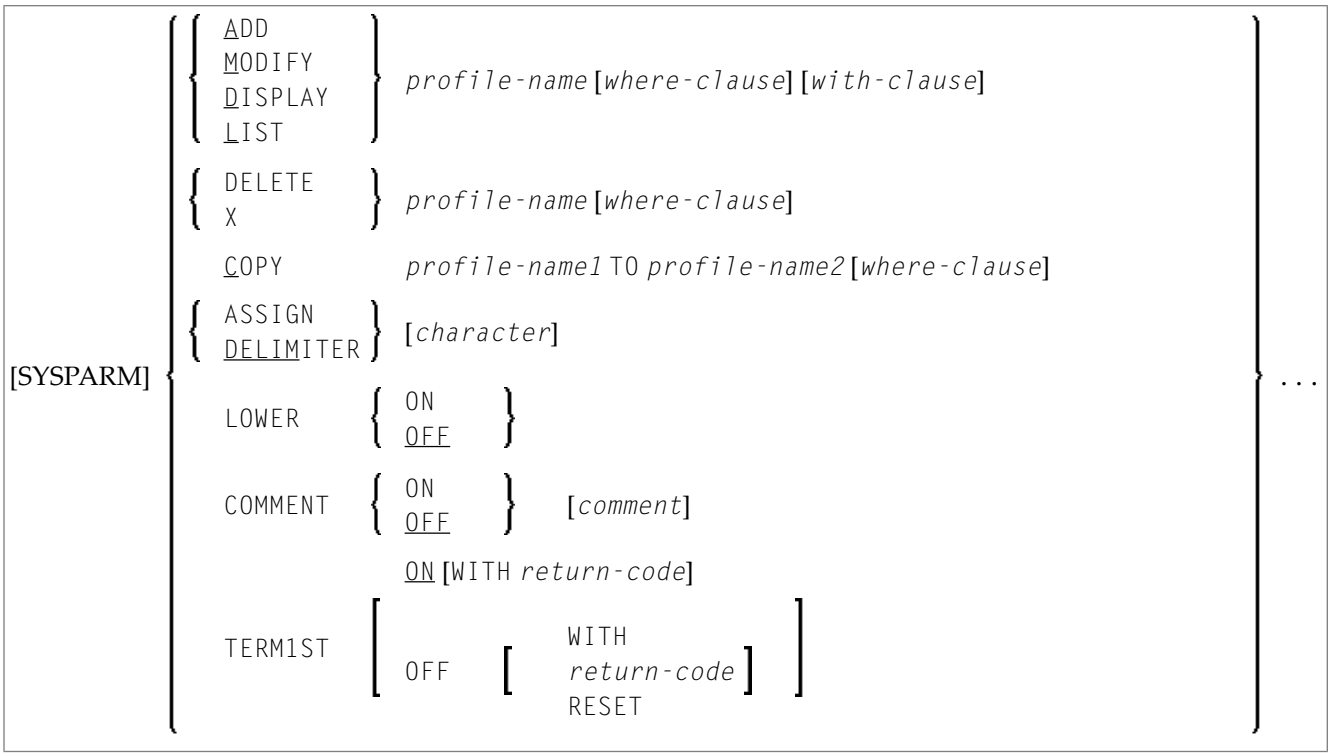
Die in den Syntax-Diagrammen verwendeten Symbole werden im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erklärt.

Dieser Abschnitt behandelt folgende Themen:

- Grundlegende Kommandosyntax
- where-clause
- with-clause
- Batch-Modus
- Beispiel für SYSPARM im Batch-Modus

Grundlegende Kommandosyntax

Das folgende Syntax-Diagramm zeigt die grundlegende Kommandosyntax, die für die Verarbeitung von SYSPARM-Kommandos gilt.



Die folgende Tabelle enthält die Werte der Variablen, die mit den Schlüsselwörtern mitgeliefert werden müssen, und die optionalen Klauseln, die in der grundlegenden SYSPARM-Kommandosyntax angezeigt werden.

Schlüsselwort	Erklärung
SYSPARM	Ruft die Utility SYSPARM auf. Dieses Schlüsselwort ist nur erforderlich, wenn Sie SYSPARM-Kommandos am Eingabeaufforderungszeichen NEXT oder MORE absetzen.
ADD	Führt die Funktion **Add New Profile** aus. Zum Beenden des Kommandos ADD geben Sie einen Punkt (.) in einer separaten Zeile ein.
MODIFY	Führt die Funktion **Modify Profile** aus. Zum Beenden des Kommandos MODIFY geben Sie einen Punkt (.) in einer separaten Zeile ein. Um einen Kommentartext in ein Parameterprofil einzufügen, platzieren Sie den Kommentar zwischen die Zeichen /* und */. Siehe auch ***Beispiel-Input***.
DISPLAY	Führt die Funktion **Display Profile** aus.
LIST	Führt die Funktion **List Profiles** aus.
DELETE	Führt die Funktion **Delete Profile** aus.

Schlüsselwort	Erklärung	
oder X		
COPY	Führt die Funktion **Copy Profile** aus.	
ASSIGN	Dient zur Angabe oder Anzeige des Input-Zuweisungszeichens. Sie müssen das Input-Zuweisungszeichen ändern, bevor Sie mit der Verarbeitung von Daten beginnen, die das Input-Zuweisungszeichen enthalten, das für Ihre aktuelle Sitzung definiert worden ist. Sonst erhalten Sie eine entsprechende Natural-Systemfehlermeldung. Wird kein Wert angegeben, dann enthält die Ausgabe das Zeichen, das für Ihre aktuelle Sitzung definiert ist.	
DELIMITER	Dient zur Angabe oder Anzeige des Input-Begrenzungszeichens. Sie müssen das Input-Begrenzungszeichen ändern, bevor Sie mit der Verarbeitung von Daten beginnen, die das Input-Begrenzungszeichen enthalten, das für Ihre aktuelle Sitzung definiert worden ist. Sonst erhalten Sie eine entsprechende Natural-Systemfehlermeldung. Wird kein Wert angegeben, dann enthält die Ausgabe das Zeichen, das für Ihre aktuelle Sitzung definiert ist.	
LOWER	Bestimmt, ob beim Anlegen oder Ändern eines Profils der Inhalt des Profils in Großbuchstaben umgesetzt wird. Mögliche Werte:	
	OFF	Buchstaben werden in Großbuchstaben umgesetzt.
	ON	Buchstaben werden nicht in Großbuchstaben umgesetzt. **Anmerkung:** Wenn ON gesetzt ist, dann wird Lower=ON im **Bearbeitungsbildschirm** der SYSPARM-Utility angezeigt (siehe ***Profile bearbeiten - Editier-Funktion***.
	Wenn das Schlüsselwort LOWER in der with-clause eines SYSPARM MODIFY-Kommandos mitgegeben wird, dann wird der in der with-clause angegebene Wert bei der entsprechenden Ändern-Operation benutzt.	
COMMENT	Gilt nicht beim Kommando REPLACE. ON: Aktiviert die Kommentar-Option und schreibt einen Kommentar nach jedem Parameter, der geändert wird. OFF: Deaktiviert die Kommentar-Option. Wenn die Kommentar-Option aktiviert ist und wenn kein individueller Kommentar (*comment*, s. unten) angegeben wird, enthält der bei jedem Parameter eingefügte Kommentar folgende Daten:	

Schlüsselwort	Erklärung	
	■ Die Benutzerkennung (erzeugt durch die Natural-Systemvariable *USER) des Benutzers, der das Parameterprofil zuletzt geändert hat, und ■ das Datum und die Uhrzeit (erzeugt durch die Natural-Systemvariablen *DATV und *TIMX), der letzten Änderung des Parameterprofils. Weitere Informationen siehe *Systemvariablen*-Dokumentation. Mit dem Kommando REPLACE können Sie einen Kommentar ändern (siehe *with-clause*).	
character	Ein beliebiges Zeichen: Siehe Natural-Session-Parameter ID (Input-Begrenzungszeichen) und IA (Input-Zuweisungszeichen).	
comment	Ein Kommentartext, der zwischen die Zeichen /* und */ zu setzen ist..	
profile-name	Der Name des Parameterprofils oder ein Bereich von Profilen. Im Falle eines Bereichs können einen der folgenden Werte angeben. Dabei ist *value* eine beliebige Kombination aus einem oder mehreren Zeichen.	
	*value**	Auswahl aller Parameterprofile, deren Namen mit dem angegebenen Wert (*value*) beginnen, z.B. AB* wählt AB und AB1, aber nicht AA1 aus.
	value>	Auswahl aller Parameterprofile, deren Namen größer oder gleich dem angegebenen Wert (*value*) sind, z.B. AB> wählt AB und AB1, aber nicht AA1 aus.
	value<	Alle Parameterprofile, deren Namen kleiner oder gleich dem angegebenen Wert (*value*) sind, z.B. AX< wählt AB und AWW, aber nicht AXA aus.
	Namensbereiche sind nur beim Kommando LIST oder im Batch-Modus bei den Kommandos LIST und DISPLAY zulässig.	
profile-name1 *profile-name2*	Gilt nur beim Kommando COPY. Das Quell-Parameterprofil (*profile-name1*), von dem ein neues Parameterprofil erstellt werden soll, und das neue Ziel-Parameterprofil, (*profile-name2*), in das die Daten kopiert werden sollen.	
where-clause	Signalisiert eine *where-clause*, s. entsprechenden Abschnitt.	
with-clause	Signalisiert eine *with-clause*, s. entsprechenden Abschnitt.	
TERM1ST	Gilt nur im Batch-Modus. Legt fest, ob die Natural-Sitzung beendet wird, wenn ein SYSPARM-Fehler oder eine Warnung auftritt.	
	OFF	Die Natural-Sitzung wird fortgesetzt. Jeder Fehler, der auftritt, zählt als Warnung.
	ON	Die Natural-Sitzung wird beendet. Dies ist die Standardeinstellung.

Schlüsselwort	Erklärung	
	WITH *return-code*	Gibt an, welcher Code zurückgegeben werden soll, wenn Natural durch SYSPARM beendet wird. Der gültige Code-Bereich für den *return-code* ist 0 bis 255. Wird die Klausel WITH *return-code* weggelassen, wird standardmäßig der Rückgabe-Code 4 verwendet. Wenn TERM1ST auf OFF gesetzt ist, wird der Code 127 abgesetzt, wenn mehr als eine Warnung aufgetreten ist. Dieser Rückgabe-Code wird zu dem in der WITH-Klausel angegebenen *return-code* hinzugefügt. Beispiel: Ein Wert von 4 (Standardwert) in der WITH-Klausel ergibt dann einen Rückgabe-Code von 131.
	Wenn TERM1ST auf OFF gesetzt ist und wenn mehr als eine Warnung aufgetreten ist, fügt SYSPARM einen Wert von 127 zu dem in der WITH-Klausel angegebenen *return-code* hinzu, und zwar bis zum Maximalwert von 255. Beispiel: Ein Wert von 4 (Standardwert) in der WITH-Klausel ergibt dann einen endgültigen Rückgabe-Code von 131.	
. . .	Gilt nur im Batch-Modus. Zeigt an, dass Sie mehr als eine SYSPARM-Funktion angeben können, indem Sie jede einzelne Funktion in eine separate Zeile platzieren.	

where-clause

Die Klausel *where-clause* ist optional und gilt bei den Kommandos ADD, MODIFY, DISPLAY, LIST, DELETE und COPY. Sie hat folgende Syntax:

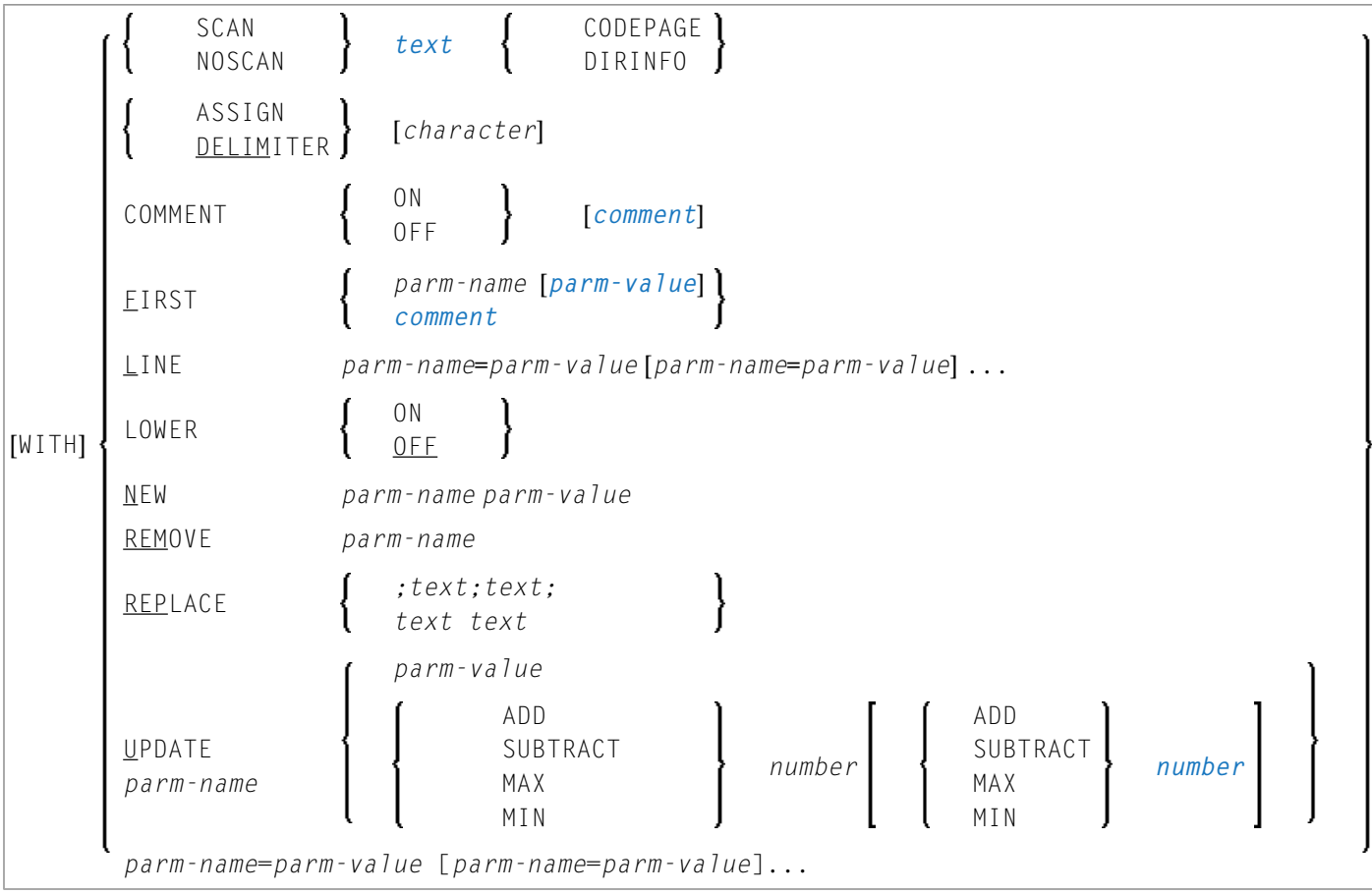
```
[WHERE] [DBID dbid] [FNR fnr] [PASSWORD password] [CIPHER cipher]
```

In der folgenden Tabelle werden die Schlüsselwörter der *where-clause* und der Wert der Variablen erklärt, die zu jedem Schlüsselwort mitgeliefert werden muss.

Schlüsselwort	Gültige Werte
WHERE	Optionales Schlüsselwort, das den Anfang einer *where-clause* anzeigt.
DBID	Die Datenbankkennung (*dbid*) der Adabas-Datei, in der das Parameterprofil gespeichert ist.
FNR	Die Dateinummer (*fnr*) der Adabas-Datei, in der das Parameterprofil gespeichert ist.
PASSWORD	Das 8 Zeichen lange Adabas-Passwort (*password*) der Adabas-Datei, in der das Parameterprofil gespeichert ist.
CIPHER	Der 8 Zeichen lange Chiffrierschlüssel (*cipher*) der Adabas-Datei, in der das Parameterprofil gespeichert ist.

with-clause

Die Klausel *with-clause* ist optional und gilt bei den Kommandos ADD, MODIFY, DISPLAY und LIST. Sie hat folgende Syntax:



Anmerkung: Im obigen Syntax-Diagramm gilt Folgendes nur beim Kommando ADD: [WITH] *parm-name=parm-value [parm-name=parm-value]...*

In der folgenden Tabelle werden die Schlüsselwörter der *with-clause* und der Wert der Variablen erklärt, die zu jedem Schlüsselwort mitgeliefert werden muss.

Schlüsselwort	Gültige Werte
WITH	Optionales Schlüsselwort, das den Anfang einer *with-clause* anzeigt. Ausnahme: Beim Kommando ADD müssen Sie das Schlüsselwort WITH benutzen, um das Kommando eindeutig vom Inhalt des Parameterprofils abzutrennen.
SCAN	Gilt nur bei den Kommandos LIST und DISPLAY. Durchsucht ein oder mehrere, in der angegebenen Natural-Systemdatei vorhandene Parameterprofile nach einer *text*-Zeichenkette (z.B. ein Parameter oder Parameterwert) und liefert eine Liste der Parameterprofile, die die angegebene *text*-Zeichenkette enthalten.

Schlüsselwort	Gültige Werte
NOSCAN	Gilt nur bei den Kommandos LIST und DISPLAY. Schließt eine *text*-Zeichenkette von der Suche in einem oder mehreren, in der angegebenen Natural-Systemdatei vorhandenen Parameterprofile aus und liefert eine Liste der Parameterprofile, die die angegebene *text*-Zeichenkette *nicht* enthalten.
CODEPAGE	Gilt nur beim Kommando LIST. Der Bildschirm **List Profiles** enthält die Spalte **Code Page**.
DIRINFO	Gilt nur beim Kommando LIST. Der Bildschirm **List Profiles** enthält die Spalten **UserID**, **Date**, **Time**, **Version** und **Code Page**. **Anmerkung:** Codepage-Namen, die aus mehr als 17 Zeichen bestehen, werden abgeschnitten, siehe **Anmerkung weiter oben**.
ASSIGN	Siehe ASSIGN in der grundlegenden Kommandosyntax.
DELIMITER	Siehe DELIMITER in der grundlegenden Kommandosyntax.
LOWER	Siehe LOWER in der grundlegenden Kommandosyntax.
COMMENT	Siehe COMMENT in der grundlegenden Kommandosyntax.
FIRST	Verschiebt einen Parameter und seinen Wert oder einen Kommentar an die erste Stelle in einem Parameterprofil. Dies kann erforderlich sein, wenn Sie z.B. den Profilparameter PARM angeben wollen. Kommentare, die sich an der ersten Stelle eines Parameterprofils befinden, verbleiben an dieser Stelle. Ein Parameter und sein Wert werden nur dann an die erste Stelle in einem Parameterprofil verschoben, wenn diese Stelle nicht mit einem Kommentar belegt ist. Andernfalls wird der Parameter nach dem (den) Kommentar(en) angehängt. Ein Kommentar, der mit FIRST eingefügt wird, wird immer an die erste Stelle gesetzt, und zwar vor einem oder mehreren ursprünglich an dieser Stelle vorhandenen Kommentaren.
LINE	Stellt den bzw. die hinter LINE angegebenen Parameter in eine neue Zeile am Ende eines Parameterprofils.
NEW	Hängt einen Parameter am Ende eines Parameterprofils an, jedoch nicht notwendigerweise in einer neuen Zeile. Verwenden Sie den Parameter LINE, um einen Parameter in eine neue Zeile zu stellen.
REMOVE	Entfernt einen Parameter und den zugewiesenen Wert (einschließlich Klammern) aus dem Parameterprofil. Der zu entfernende Parameter kann ein Subparameter sein, z.B. der Parameter DFS, der ein Subparameter des Profilparameters RPC ist.
REPLACE	Ersetzt eine alte *text*-Zeichenkette durch eine neue, in einem Parameterprofil enthaltene *text*-Zeichenkette. Siehe auch Option COMMENT in der grundlegenden Kommandosyntax.
UPDATE	Aktualisiert den Wert, der einem Parameter zugewiesen worden ist. Sollte der angegebene Parameter noch nicht existieren, wird er an das Parameterprofil angehängt.

Schlüsselwort	Gültige Werte
ADD	Erhöht die Größe eines Parameterwerts durch Addition der angegebenen Zahl. Die Addition kann durch Angabe von MAX bzw. MIN auf ein Maximum bzw. Minimum begrenzt werden (siehe unten).
SUBTRACT	Verringert die Größe eines Parameterwerts durch Subtraktion der angegebenen Zahl. Die Subtraktion kann durch Angabe von MAX bzw. MIN auf ein Maximum bzw. Minimum begrenzt werden (siehe unten).
MAX	Gibt das Maximum für einen Parameterwert an.
MIN	Gibt das Minimum für einen Parameterwert an.
text	Eine beliebige Text-Zeichenkette, die in einem Parameterprofil enthalten ist. Das Folgende gilt für die Kommandos SCAN und NOSCAN: Die Text-Zeichenkette darf *keine* Leerzeichen enthalten. Das Folgende gilt für das Kommando REPLACE: Wenn weder der alte Text noch der neue Text Leerzeichen enthält, setzen Sie ein Leerzeichen zwischen den alten und den neuen Text. Wenn der alte Text oder der neue Text Leerzeichen enthält, setzen Sie das angegebene Input-Begrenzungszeichen zwischen den alten und den neuen Text und umschließen Sie die gesamte alte/neue Text-Zeichenkette mit Input-Begrenzungszeichen. Beispiel: `;This comment is old;This comment is new;` Alternativ zum angegebenen Input-Begrenzungszeichen können Sie eines der folgenden Zeichen wählen: ' Hochkomma , Komma . Punkt ; Semikolon / Schrägstrich \ Rückwärtsschrägstrich | senkrechter Strich
character	Siehe *character* in der grundlegenden Kommandosyntax.
parm-name	Der vollständige Name eines Parameters.
parm-value	Der einem Parameter zugewiesene Wert.
comment	Siehe *comment* in der grundlegenden Kommandosyntax.
number	Ein numerischer Wert.

Batch-Modus

Bei der SYSPARM-Verarbeitung im Batch-Modus ist Folgendes zu berücksichtigen:

- Um SYSPARM zu beenden, können Sie in einer separaten Zeile einen Punkt (.) oder `FIN` eingeben, wobei mit `FIN` auch die Natural-Sitzung beendet wird.
- Um mehr als eine SYSPARM-Funktion auszuführen, geben Sie jede einzelne Funktion in einer separaten Zeile ein.
- Um eine Funktion über zwei Zeilen zu erweitern, geben Sie an beliebiger Stelle in der ersten Zeile das Zeichen ein, das beim Session-Parameter `CF` definiert ist. Standardmäßig ist das das Prozentzeichen (`%`). Damit wird die Fortsetzung in der nächsten Zeile angezeigt.

Siehe auch ***Beispiel für SYSPARM im Batch-Modus***.

Verwandte Themen:

Natural in Batch Mode (*Operations*-Dokumentation)
INPUT-Statement unter Nicht-Screen-Modi (*Statements*-Dokumentation)
INPUT-Statement im Batch-Betrieb (*Statements*-Dokumentation)

Beispiel für SYSPARM im Batch-Modus

Die folgenden Beispiele veranschaulichen die Benutzung der SYSPARM-Kommandos und die Folgen von Parameterprofiländerungen im Batch-Modus.

- Beispiel-Profil TESTPROF - Vor dem Jobstart
- Beispiel-Input
- Beispiel-Parameterprofil TESTPROF - Nach der Jobausführung
- Beispiel-Parameterprofil TESTPRO1 - Nach der Jobausführung

Beispiel-Profil TESTPROF - Vor dem Jobstart

```
/* This is a test profile. */
AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(1099,1100,PASSWORD,12345678)
RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON,DFS=(SRV2,NODE1,,ACI))
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
ESIZE=90
```

Anmerkung: Falls Ihr aktuelles Input-Zuweisungs- oder Input-Begrenzungszeichen Teil der Eingabe ist, müssen Sie dieses Input-Zeichen ändern (siehe `ASSIGN` bzw. `DELIMITER` in *Direktkommandos und Batch-Modus*). Dazu können Sie eine entsprechende `ASSIGN`- bzw. `DELIMITER`-Zuweisung vor dem Statement hinzufügen, das andernfalls einen Laufzeitfehler verursachen würde. Siehe ***Beispiel-Input*** .

Beispiel-Input

```
/*JCL
.
.
.
*/
SYSPARM
COPY TEST1 TO TESTPROF WHERE DBID 10 FNR 32
DISP   TESTPROF
COMMENT ON /* TESTUSER 29.Jan.2009 */
MODIFY TESTPROF with REM DFS
   REPLACE ;test profile;test profile for SYSPARM in batch;
   NEW BPSIZE 4096
   UPDATE ESIZE  ADD  20  MAX 100
   FIRST PARM INHOUSE
   DELIMITER $
   UPDATE FUSER (,6)
   ASSIGN :
   LINE IM=D, INTENS=1, AUTO=T, MT=0, MADIO=0
   .
DISP   TESTPROF
ADD TESTPRO1 WITH /* 106,210 */
    FNAT=(106,210,PASSWORD),FUSER=(,211)
    .
DISPLAY TESTPRO1
DELETE  TESTPRO2 WHERE DBID 10 FNR 32
X TESTPRO3   WHERE DBID 10 FNR 32
.
FIN
```

Beispiel-Parameterprofil TESTPROF - Nach der Jobausführung

```
/* This is a test profile for SYSPARM in batch. */
PARM=INHOUSE /* TESTUSER 29.Jan.2009 */
AUTO=ON FNAT = (102,110,PASSWORD) FUSER=(,6) /* TESTUSER 29.Jan.2009 */
RPC=(RPCSIZE=80,SRVNAME=MYSERV,SERVER=ON, )
PRINT=((2,12,18),AM=STD,DEST='PRINT**',OPEN=INITOBJ,CLOSE=CMD)
PRINT=((1,3,6-11,15),AM=NAF)
ESIZE=100 /* TESTUSER 29.Jan.2009 */ BPSIZE=4096 /* TESTUSER 29.Jan.2009
 */
IM=D, INTENS=1, AUTO=T, MT=0, MADIO=0 /* TESTUSER 29.Jan.2009 */
```

Beispiel-Parameterprofil TESTPRO1 - Nach der Jobausführung

```
/* 106,210 */
FNAT=(106,210,PASSWORD),FUSER=(,211)
```

Parameterprofile in unterschiedlichen Umgebungen verwalten

Mit der Utility SYSPARM können Sie Parameterprofile innerhalb derselben Systemdatei FNAT oder FUSER verwalten.

Um Parameterprofile zwischen verschiedenen FNAT- oder FUSER-Systemdateien zu übertragen (z.B. kopieren oder verschieben) und um ein Parameterprofil in einer anderen Umgebung zu bearbeiten (z.B. löschen oder suchen), können Sie die Utility SYSMAIN benutzen. Weitere Informationen siehe ***Profile bearbeiten*** in der *SYSMAIN Utility*-Dokumentation.

Hilfe zu Parametern aus der Kommandozeile aufrufen

Mit dem Systemkommando `HELP` können Sie die Online-Hilfe zu einem Parameter oder Subparameter aufrufen. Dazu können Sie folgende Kommandos absetzen:

`HELP <name>`	Anzeige des Hilfe-Bildschirms für den Parameter *name*. Falls ein Parameter dieses Namens nicht existiert, wird die Suche auf Subparameter ausgedehnt.
`HELP SUB <name>`	Sucht nur nach Subparametern. Dieses Kommando ist nützlich, wenn *name* nicht eindeutig ist und sich sowohl auf einen Parameter als auch auf einen Subparameter beziehen kann.
`HELP <prefix>*`	Anzeige einer Auswahlliste, die nur passende Parameter enthält. Wird die Zeichenkette <*prefix*> leer gelassen, dann werden alle Parameter angezeigt.
`HELP HELP`	Anzeige der Kategorien `ADD-ON`, `DRIVER` und `MAIN` in einem separaten Menü. Wählen Sie eine Kategorie aus, um eine Liste der zugehörigen Parameter anzuzeigen. Aus dieser Liste können Sie dann einen Parameter auswählen, um den entsprechenden Hilfe-Bildschirm zu sehen.
`HELP <category>`	Auflistung aller Parameter, die in Bezug zu einer der erwähnten Kategorien stehen.

Informationen zur Benutzung der mit diesen Kommandos aufgerufenen Hilfe-Bildschirme siehe ***Hilfe zu Parametern - Help with parameters***.

XXII SYSRDC Utility

123 SYSRDC Utility

- Funktionsbausteine von SYSRDC 1110
- Datenerfassung auslösende Ereignisse 1110
- Gesammelte Daten 1112
- Natural Data Collector aktivieren und konfigurieren 1114
- Trace Recording-Funktion 1115
- User Exits für externes Monitoring/Accounting 1116
- Aufruf des CMRDC-Interface 1118
- Beispiel-Programme 1122

Mit dem Dienstprogramm („Utility“) SYSRDC können Sie Monitoring- und Abrechnungsdaten über den internen Prozessablauf innerhalb einer Natural-Anwendung aufzeichnen und für die Auswertung von Natural-Session-Aktivitäten in externen oder Natural-Programmen verwenden. Die Daten werden bei innerhalb von Natural auftretenden Ereignissen gesammelt

Verwandte Themen:

- *Profiler Utility*, ein menügesteuertes Werkzeug zum Erzeugen und Auflisten von Trace-Datensätzen
- *RDC - Konfiguration des Natural Data Collector* in der *Profilparameter*-Dokumentation.

Funktionsbausteine von SYSRDC

Im Folgenden werden die Funktionsbausteine der SYSRDC Utility beschrieben, mit denen Daten gesammelt und zur Weiterverarbeitung zur Verfügung gestellt werden können.

- **Natural Data Collector als Teil des Natural-Nukleus:**
  Sammelt Daten und steuert die Datenaufzeichnung im **Natural Data Collector Buffer**.
- **User Exits:**
  Übergibt Daten an externe Monitoring- und Abrechnungsprogramme.
- **CMRDC Interface:**
  Wertet Daten in einem Natural-Programm für die aktuelle Natural-Session aus.

Datenerfassung auslösende Ereignisse

Ereignisse (**Events**) geben Aktivitäten innerhalb von Natural an. Die folgende Tabelle enthält eine Liste der Ereignisse, bei denen Daten innerhalb von Natural gesammelt werden, und der zur Verfügung stehenden Ereignistypen.

Jedem Ereignistyp ist ein ein- oder zweistelliger Code zugeordnet, wobei der erste Buchstabe die Gruppe des Ereignisses und der zweite Buchstabe den Untertyp darstellt, zum Beispiel:

Ereignistyp **PL**

P steht für die Ereignisgruppe *program* und **L** für den Untertyp *load*.

Ereignis	Ereignistyp
Session initialization (Initialisierung der Session)	SI
Session termination (Beendigung der Session)	ST
Program load (Laden eines Programms)	PL
Program start (Starten eines Programms)	PS
Program termination (Beendigen eines Programms)	PT
Before database call (vor einem Datenbankaufruf)	DB
After database call (nach einem Datenbankaufruf)	DA
Before terminal I/O (vor einer Terminal-Ein-/Ausgabe)	IB
After a terminal I/O (nach einer Terminal-Ein-/Ausgabe)	IA
Before non-Natural program call (vor dem Aufrufen eines Nicht-Natural-Programms)	CB
After non-Natural program call (nach dem Aufrufen eines Nicht-Natural-Programms)	CA
Runtime error (Laufzeitfehler)	E
Internal trace call (interner Trace-Aufruf) **Anmerkung:** Die von internen Traces gelieferten Informationen sind nur für Debugging-Zwecke durch den Support bestimmt. Siehe **unten**.	N
Natural Statement **Anmerkung:** Siehe **unten**.	NS
Outbound RPC message (ausgehende RPC-Meldung)	RO
Inbound RPC message (eingehende RPC-Meldung)	RI
Start of RPC request execution (Start der Ausführung der RPC-Anforderung)	RS
Wait for RPC client request (Warten auf RPC-Client-Anforderung)	RW
User-defined event (Benutzerdefiniertes Ereignis)	U

Um ein internes Trace-Aufruf-Ereignis zu aktivieren:

1 Setzen Sie den Natural-Profilparameter `ITRACE=ON`.

2 Definieren Sie die Natural-Komponenten, die interne Trace-Aufrufe absetzen sollen:

- Benutzen Sie dazu das Makro `NTTRACE` im Natural-Parametermodul.
- Oder setzen Sie den dynamischen Natural-Profilparameter `TRACE`.

Um ein Natural-Statement-Ereignis zu aktivieren:

- Setzen Sie die Natural-Profilparameter `ITRACE=ON` und `TRACE=NATPROX`.

Informationen zu den oben genannten Profilparametern und dem Parameter-Makro siehe *Parameter-Referenz*-Dokumentation.

Gesammelte Daten

Die Daten, die der Natural Data Collector sammelt, werden in den folgenden Abschnitten beschrieben:

Informationen zum Layout der Daten können Sie dem Natural Source Dataset (Datei) NAMRDC oder dem lokalen Datenbereich RDCDATA entnehmen, der sich in der Natural-System-Library SYSRDC befindet.

Die gesammelten Daten können in zwei Kategorien unterteilt werden: allgemeine und ereignisspezifische Daten:

- Allgemeine Daten
- Ereignisspezifische Daten

Allgemeine Daten

Die folgenden allgemeinen Daten werden bei jedem Ereignis gesammelt:

- Produktname
- Produktversion
- Betriebssystem
- TP-Monitor
- Laufmodi, z.B. Adressierungs- und Residenzmodus des Natural-Nukleus und des Natural Buffer Pool
- TP-Benutzername oder Batch-Jobname
- TP-Terminalkennung (ID)
- Aktuelle Natural-Benutzerkennung (ID)
- Aktuelle Natural Security-Benutzergruppenkennung (ID)
- Aktuelle Natural-Library
- Aktuelles Natural-Programm
- Aktuelle Programmebene
- Zeilennummer des zurzeit ausgeführten Programm-Statements
- Session CPU Time (gleiches Format wie `*CPU-TIME`)

Ereignisspezifische Daten

Die folgenden Daten werden nur bei den folgenden Ereignissen gesammelt:

Ereignis	Datenelemente
Session initialization (Initialisierung der Session)	keine
Session termination (Beendigung der Session)	Beendigungs-Rückgabecode Natural-Beendigungsmeldungscode NAT99*nn* Name des Back-end-Programms
Program load (Laden eines Programms)	Name des zu ladenden Programms Name der Load Library Aufruftyp
Program start/termination (Starten/Beenden eines Programms)	Programmtyp Programmname Programm-Library-Name Datenbankkennung (ID) der Programm-Library Dateinummer (FNR) der Programm-Library
Database call (Datenbankaufruf)	Datenbanktyp Kommandocode Kommandokennung (ID) Datenbankkennung (ID) Dateinummer (FNR) Zeiger, der Parameteradressen auflistet (nur bei User Exits nützlich) Rückgabecode (nur beim Ereignistyp DA) Fehlersubcode (nur beim Ereignistyp DA) Adabas-Kommandozeit (nur beim Ereignistyp DA)
Terminal I/O (Terminal-Ein-/Ausgabe)	Anzahl der gesendeten Bytes Anzahl der gelesenen Bytes Zugewiesener Gesamt-Session-Speicher Länge des komprimierten Session-Speichers
Non-Natural program call (Aufrufen eines Nicht-Natural-Programms)	Name des aufgerufenen Programms Aufrufmodus, z.B. dynamischer oder statischer Modus Program-Link-Speicherort Parametertyp Parameteradresse Programmeingabeadresse Rückgabecode (nur beim Ereignistyp CA)
Runtime error (Laufzeitfehler)	Natural-Systemfehlermeldungscode Externer Abbruchcode Name des Fehlerbehandlungsprogramms gemäß Angabe im Natural-Profilparameter ETA (siehe *Parameter-Referenz*-Dokumentation).
Internal trace call (interner Trace-Aufruf)	Bis zu 250 Bytes Informationen zu Natural-Nukleus-Komponenten Die gelieferten Informationen sollen nur für Debugging-Zwecke durch Support-Mitarbeiter dienen.

Ereignis	Datenelemente
Natural Statement	Bis zu 250 Bytes Informationen zu ausgeführten Natural-Statements
RPC request (RPC-Anforderung)	RPC-spezifische Informationen: Umgebung: C = Client, S = Server RDC Subtyp S: Subprogrammname Adabas-Benutzerkennung (ETID) Status der Konversation Logon-Indikator (Y = Anmeldung erfolgt) Impersonationsanzeige der RPC-Anforderung (Y = Impersonation erfolgt). RDC-Subtypen O, I und W: Transportprotokoll RPC-Funktion Typ der Client-Benutzerkennung Länge der Meldung Rückgabecode der Tansportschicht Externe Konversationskennung Client-Benutzerkennung Serverknoten (outbound message of client only), Servername (nur ausgehende Meldung des Client).
User-defined event (Benutzerdefiniertes Ereignis)	Bis zu 250 Bytes benutzerdefinierte Informationen.

Natural Data Collector aktivieren und konfigurieren

Der Natural Data Collector wird mit Hilfe des Natural-Profilparameters `RDC` aktiviert und konfiguriert (siehe *RDC - Konfiguration des Natural Data Collector* in der *Parameter-Referenz*-Dokumentation).

Die Standardeinstellung ist `RDC=OFF`. Das bedeutet, der Natural Data Collector ist deaktiviert und die Trace Recording-Funktion muss separat aktiviert werden. Um die Einstellungen zu ändern, können Sie im Natural-Parametermodul das Makro *NTRDC* des Profilparameters `RDC` benutzen (siehe *RDC - Konfiguration des Natural Data Collector* in der *Parameter-Referenz*-Dokumentation).

Folgende Werte und Schlüsselwort-Parameter stehen zur Verfügung:

Wert / Schlüsselwort-Subparameter	Beschreibung
ON	Die Trace Recording-Funktion wird beim Start der Natural-Session automatisch gestartet.
OFF	Der Natural Data Collector ist deaktiviert und die Trace Recording-Funktion muss separat aktiviert werden.
EVENT	Zur Angabe von Ereignissen, die aufgezeichnet werden sollen. Wenn Sie ALL angeben, werden alle Ereignisse aufgezeichnet. Weitere Informationen siehe *EVENT - Natural Data Collector-Ereignisse, die aufgezeichnet werden sollen*. **Anmerkung:** Alternativ können Sie das Natural-Statement CALL 'CMRDC' 'T' benutzen, siehe ***Ereignistypen für Trace-Aufzeichnung auswählen***.
EXIT	Zur Definition von User Exits, siehe *EXIT - User Exits für den Natural Data Collector*.
FNAT	Zur Steuerung der Trace-Recording-Funktion, während Programme in der Natural-Systemdatei ausgeführt werden. Weitere Informationen siehe *FNAT - Trace Recording in der Natural-Systemdatei*.
SIZE	Zur Angabe der Größe des Pufferspeichers für den Natural Data Collector. Zusätzlich zur Steuerung der Trace-Recording-Funktion des Natural Data Collector. Weitere Informationen siehe *SIZE - Größe des Natural Data Collector-Puffers*. **Anmerkung:** Als Alternative zu diesem Subparameter können Sie den äquivalenten Natural-Profilparameter RDCSIZE benutzen.

Trace Recording-Funktion

Die gesammelten Ereignisdaten werden immer an **User Exits** geliefert, die, wie an anderer Stelle beschrieben, von externen Monitoring/Accounting-Programmen benutzt werden sollen. Sie können unabhängig von den User Exits die Ereignisdaten der aktuellen Natural Session im Natural Data Collector-Puffer aufzeichnen. Dies kann für Testzwecke nützlich sein. In diesem Abschnitt wird das Aufzeichnen von Daten im Natural Data Collector-Puffer als *Trace Recording-Funktion* bezeichnet.

Wenn Sie eine Natural Session starten, ist die Trace-Recording-Funktion standardmäßig nicht aktiv. Damit die Trace-Recording-Funktion beim Start einer Session (ohne Trace-Aufzeichnung) aktiviert wird, können Sie den Natural-Profilparameter RDC auf ON setzen, siehe auch *RDC - Konfiguration des Natural Data Collector*. Um außerdem die Trace-Aufzeichnung zu aktivieren, müssen Sie den RDC-Subparameter SIZE auf einen Wert größer als 2 setzen.

Sie können auch eine der folgenden Option nutzen, um die Trace-Aufzeichnung zu starten:

- Systemkommando `RDC ON`
- Terminalkommando `%<RDC+`
- Natural-Statement `CALL 'CMRDC' 'S'`

Anmerkungen:

1. Der Natural Data Collector-Puffer wird im Wrap-Around-Modus gefüllt, d. h. der älteste Datensatz wird überschrieben, wenn der Puffer voll wird. Am Ende der Sitzung wird der Inhalt des Puffers gelöscht.
2. Die Trace-Aufzeichnung umfasst standardmäßig nicht die Systemdatei FNAT. Um auch die Systemdatei FNAT in die Aufzeichnung einzubeziehen, müssen Sie den `RDC`-Subparameter `FNAT=ON` setzen.

Die Trace-Aufzeichnung wird beendet durch:

- Systemkommando `RDC OFF`
- Terminalkommando `%<RDC-`
- Natural-Statement `CALL 'CMRDC' 'P'`
- Beenden einer Session

User Exits für externes Monitoring/Accounting

Die Ereignisdaten können an externe Monitoring- und Accounting-Programme zur Auswertung der Aktivitäten in Natural Sessions übergeben werden. Dies wird erreicht mit den User Exits und den mitgelieferten Beispielen für User-Exit-Programme. User-Exit-Programme sind in Assembler geschrieben.

Der Natural Data Collector unterstützt eine beliebige Anzahl an User Exits. User Exits können mit dem Natural-Profilparameter RDCEXIT definiert werden (siehe *Parameter-Referenz*-Dokumentation). Mit jedem User Exit kann ein externes Monitoring- und Accounting-Programm verknüpft werden.

Es sind drei vordefinierte Exit-Namen vorhanden: RDCEX1, RDCEX2 und RDCEX3. Wenn Sie einen dieser Namen als Einstiegspunkt für Ihren Exit (der mit dem Natural-Nukleus verlinkt ist) benutzen, ist der Natural-Profilparameter RDCEXIT nicht erforderlich.

Bei jedem der unter ***Datenerfassung auslösende Ereignisse*** aufgelisteten Ereignisse übernehmen die User Exits die Steuerung. Dazu benutzen sie die in der folgenden Tabelle aufgeführten Standard Linkage-Aufrufkonventionen:

Register	Inhalt
1	Zeigt auf eine Parameteradressliste, die aus zwei Adressen besteht: Die eine Adresse zeigt auf die allgemeinen Daten und die andere auf die ereignisspezifischen Daten. Das Layout dieser beiden Bereiche wird durch den DSECT RDCGDATA bzw. RDCLDATA dargestellt. Beide DSECTs werden in Source-Form im Natural-Makro NAMRDC geliefert.
13	Zeigt auf einen 72-Byte Standard-Speicherbereich.
14	Enthält die Rückgabeadresse.
15	Enthält entweder die Einstiegspunktadresse oder den Rückgabecode des User-Exits.



Anmerkung: Der Aufruf der User Exits erfolgt unabhängig vom CMRDC-Interface.

Ein User-Exit-Programm muss die gleichen Attribute wie Natural haben, d.h., es muss den gleichen Adressierungsmodus haben und muss ablaufinvariant (*reentrant*) sein. Es muss mit dem Natural-Nukleus gemäß den Konventionen für statisch verlinkte Nicht-Natural-Programme verlinkt sein. Siehe auch Natural-Profilparameter CSTATIC und RCA in der *Parameter-Referenz*-Dokumentation.

Standardmäßig wird für jeden User Exit ein 400-Byte großer exklusiver Arbeitsbereich (pro Session) zur Verfügung gestellt (das Feld RDCGWRKA). Sollte für einen User Exit ein größerer Arbeitsbereich erforderlich sein, kann er im Natural-Profilparameter RDCEXIT (s. *Parameter-Referenz*-Dokumentation) nach dem Namen des User Exit angegeben werden. Die Übergabe der Länge des Arbeitsbereichs an die User Exits erfolgt im Feld RDCGWRKL. Sie kann zur Verifizierung genutzt werden. Der Standort dieses Arbeitsbereichs kann sich aufgrund einer Natural-Standortverlagerung ändern, sein Inhalt bleibt jedoch erhalten.

In TP-Monitor-Umgebungen wird die TP-Ankeradresse geliefert (das Feld RDCGANCH), z.B. die CSA-Adresse unter CICS. Sie kann benutzt werden, um Zugang zu Systeminformationen zu erhalten.

Falls während der Ausführung eines User Exit-Programms ein Program Check auftritt, wird die weitere Datenerfassung für den Rest der Session deaktiviert, um rekursive Abbruchsituationen zu vermeiden.

Anschließend werden folgende Themen behandelt:

- Rückgabecodes

- Beispiele für User-Exit-Programme

Rückgabecodes

Rückgabecodes ungleich Null werden nur für zwei Ereignisse unterstützt:

- Vor einem Datenbankaufruf, bei dem das Register 15 einen Adabas-Antwortcode enthalten kann, der im Steuerblock gespeichert ist. Der Adabas-Aufruf wird nicht ausgeführt.
- Beim Programmstart, wobei Register 15 einen Natural-Fehlermeldungscode enthalten kann. Das Programm wird nicht ausgeführt, sondern es tritt eine Fehlerbedingung mit der angegebenen Nummer auf.

Beispiele für User-Exit-Programme

Die folgenden Beispiele für User-Exit-Programme werden in den folgenden Natural-Source-Datasets (Dateien) zur Verfügung gestellt:

Programm	Data Set (Datei)	Benutzt für
NAMRDC	NAT*nnn*.SRCE	DSECT Makro für allgemeine und ereignisspezifische Daten
XNATRDC1	NAT*nnn*.SRCE	Natural TSO Interface und Natural Batch Interface unter dem Betriebssystemen z/OS
XNCFRDC1	NCF*nnn*.SRCE	Com-plete
XNCIRDC1	NCI*nnn*.SRCE	Natural CICS Interface

Aufruf des CMRDC-Interface

Die Anwendungsprogrammierschnittstelle CMRDC wird verwendet, um die im Natural Data Collector-Puffer aufgezeichneten Trace-Daten abzurufen und zu steuern. Dies kann für Testzwecke von Nutzen sein.

Um das CMRDC Interface aufzurufen:

- Setzen Sie in dem Natural-Programm ein `CALL`-Statement ab.

Die vom CMRDC Interface zur Verfügung gestellten Funktionen und die Syntax, die bei den entsprechenden Natural-`CALL`-Statements gilt, werden im folgenden Abschnitt behandelt.

Zusätzlich enthält die Natural-System-Library SYSRDC Beispielprogramme und den dazu passenden lokalen Datenbereich (LDA) RDCDATA, in dem die Struktur der Trace-Aufzeichnungen definiert ist. Gemäß dieser Definition hat jede Trace-Aufzeichnung zwei Hauptbestandteile: einen für die allgemeinen Daten und einen für ereignisspezifische Daten.

Folgende Themen werden behandelt:

- Trace-Aufzeichnungen abrufen
- Trace-Aufzeichnung stoppen und erneut starten
- Ereignistypen für Trace-Aufzeichnung auswählen
- Benutzerdefinierte Ereignisse
- Trace-Status abfragen
- CMRDC-Rückgabecodes

Trace-Aufzeichnungen abrufen

Um Daten aus dem Natural Data Collector-Puffer zu lesen, müssen Sie CMRDC mit dem folgenden Natural-Statement aufrufen:

```
CALL 'CMRDC' function event-time gen-data event-data seq.-number
```

Folgende Parameter werden übergeben:

Parameter	Format/Länge	Erklärung
function	A1	F Ersten Trace-Datensatz holen. G Nächsten Trace-Datensatz holen. N Datensatz mit angegebener Folgenummer holen.
event-time	N10 oder N12	Zeitpunkt des Ereignisses: Die Zeit wird in einem der folgenden Formate zurückgegeben: *HHMMSSXXXX* *HH* = Stunden, *MM* = Minuten, *SS* = Sekunden, *XXXX* = Sekundenbruchteil *HHMMSSXXXXXX* *HH* = Stunden, *MM* = Minuten, *SS* = Sekunden, *XXXXXX* = Sekundenbruchteil
gen-data	A252	Allgemeine Daten (Feld GDATA)
event-data	A252	Ereignisspezifische Daten (Feld LDATA)
seq.-number	I4	Folgenummer des Datensatzes Gilt nur bei Funktion **N** (siehe oben).

Die Abruf-Funktionen stoppen das Aufzeichnen von Daten im Natural Data Collector-Puffer implizit.

Trace-Aufzeichnung stoppen und erneut starten

Um das Aufzeichnen von Trace-Daten im Natural Data Collector-Puffer zu stoppen und erneut zu starten, müssen Sie CMRDC mit dem folgenden Natural-Statement aufrufen:

```
CALL 'CMRDC' function
```

Folgende Parameter werden übergeben:

Parameter	Format/Länge	Erklärung
function	A1	S Natural Data Collector-Puffer löschen und Trace-Aufzeichnung neu starten. P Trace-Aufzeichnung stoppen.

Ereignistypen für Trace-Aufzeichnung auswählen

Standardmäßig werden alle Ereignisse für die Trace-Aufzeichnung ausgewählt. Sie können diese Funktion benutzen, wenn nur bestimmte Ereignisse aufgezeichnet werden sollen.

Anmerkungen:

1. Diese Funktion dient nur zum Auswählen von Ereignissen, die im Natural Data Collector-Puffer aufgezeichnet werden sollen. Sie hat keine Auswirkung auf die Daten, die an User Exits übergeben werden, und beeinflusst auch nicht den Status (gestartet/gestoppt) der Trace-Aufzeichnung.
2. Ereignisse können auch mit dem Subparameter `EVENT` des Profilparameters `RDC` angegeben werden, siehe ***Natural Data Collector aktivieren und steuern***.
3. Diese Funktion bewirkt, dass jegliche früher erfolgte Ereignisdefinition ersetzt wird,

Um die aufzuzeichnenden Ereignistypen auszuwählen, müssen Sie CMRDC mit dem folgenden Natural-Statement aufrufen:

```
CALL 'CMRDC' function type ...
```

Folgende Parameter werden übergeben:

Parameter	Format/Länge	Erklärung
function	A1	T Ereignisse für Trace-Aufzeichnung auswählen.
type	A1, A2 oder A3	Aus einem oder zwei Buchstaben bestehender Code für den aufzuzeichnenden **Ereignistyp**, siehe ***Datenerfassung auslösende Ereignisse***. Sie können eine beliebige Anzahl an Parametern für die gewünschten Ereignistypen angeben. Alternativ können Sie einen Bereich von Ereignissen oder kein Ereignis wählen: ALL — Alle Ereignistypen. *value** — Alle Ereignistypen, die mit *value* beginnen. Beispiel: **P*** wählt alle Ereignisse des Typs Programm: PL, PS und PT. leer — Bei Angabe eines Leerzeichens wird kein Ereignis ausgewählt.

Benutzerdefinierte Ereignisse

Um ein benutzerdefiniertes Ereignis auszuwählen, müssen Sie CMRDC mit dem folgenden Natural-Statement aufrufen:

```
CALL 'CMRDC' function record
```

Folgende Parameter werden übergeben:

Parameter	Format/Länge	Erklärung
function	A1	U Benutzerdefiniertes Trace-Ereignis.
record	A*nnn*	Trace-Aufzeichnung mit einer Länge (*nnn*) von bis zu 250 Bytes.

Trace-Status abfragen

Um den Trace-Status und den zurzeit gesetzten Ereignistyp abzufragen, müssen Sie CMRDC mit dem folgenden Natural-Statement aufrufen:

```
CALL 'CMRDC' function status types
```

Folgende Parameter werden übergeben:

Parameter	Format/Länge	Erklärung
function	A1	C Trace-Status abfragen.
status	A1	Aktueller Trace-Status: S Trace aktiv P Trace inaktiv
types	A*nn*	Die aus zwei Buchstaben bestehenden Codes für die zurzeit gesetzten **Ereignistypen** Siehe *Ereignistypen für Trace-Aufzeichnung auswählen*. Die Variable *types* benötigt eine Mindestlänge (*nn*) an 64 Bytes, um genügend Platz für alle Ereignistypen zur Verfügung zu stellen. Dieser Parameter ist optional.

CMRDC-Rückgabecodes

Code	Bedeutung
0	Funktion erfolgreich ausgeführt.
4	Letzte Trace-Aufzeichnung. Bei Ereignistyp = 0 sind für die Funktion **F** keine Trace-Daten vorhanden. Gilt nur für die Funktionen **F** und **G** (siehe oben).
8	Zu wenige Parameter für diese Funktion.
12	Ungültiger Funktionscode.
16	Natural Data Collector nicht aktive, z.B. wegen RDCSIZE=0.
20	Natural Data Collector nach Fehler deaktiviert.
24	Kein Pufferplatz für Trace-Aufzeichnung verfügbar (RDCSIZE=2).
28	Ungültiger Parameterwert. Gilt nur für die Funktionen **T** und **N** (siehe oben).

Beispiel-Programme

Die Natural-System-Library SYSRDC enthält folgende Beispiel-Programme:

Programm	Funktion
RDCDISP	Anzeige aller Datensätze im Natural Data Collector-Puffer und der Felder wie im Programm angegeben. Siehe auch *Beispiel-Ausgabe des Programms RDCDISP* weiter unten.
RDCSTART	Trace-Aufzeichnung neu starten.
RDCSTOP	Trace-Aufzeichnung stoppen.
RDCSET	Ereignisse für Trace-Aufzeichnung auswählen.

Programm	Funktion
RDCUSER	Benutzerdefiniertes Ereignis.
RDCSTAT	Trace-Status abfragen.
RDCDISIO	Anzeige aller Ein-/Ausgabe bezogenen Datensätze im Natural Data Collector-Puffer und der Felder wie im Programm angegeben. Siehe auch *Interne Gültigkeitsprüfung und Tracing nach E/A-bezogenen Statements* und *Beispiel-Ausgabe des Programms RDCDISIO* weiter unten.

Beispiel-Ausgabe des Programms RDCDISP

Der folgende Beispiel-Bildschirm zeigt einen Auszug aus einem Ausgabe-Report, der vom Beispiel-Programm RDCDISP erstellt wurde.

```
 ETIME1   TY  GCUID    LV   GPGM    T  GCAPL    PRLIB    PRNAM    CO  DB   FN  RC
--------- -- -------- -- -------- - -------- -------- -------- -- ---- ---- ---

   0.0000 SI SAG                    SYSTEM
   0.0002 DB SAG                    SYSTEM                      OP   10
   0.0002 DA SAG                    SYSTEM                      OP   10
   0.0000 DB SAG                    SYSTEM                      S1   10 1640
   0.0005 DA SAG                    SYSTEM                      S1   10 1640
   0.0001 PL SAG                    SYSTEM   SYSLIB   MAINMENU
   0.0001 PS SAG       1 MAINMENU F SYSTEM   SYSLIB   MAINMENU      10 1640
   0.0001 PL SAG       1 MAINMENU   SYSTEM   SYSLIB   NAT00029
   0.0000 DB SAG       1 MAINMENU   SYSTEM                      S1   10 1640
   0.0002 DA SAG       1 MAINMENU   SYSTEM                      S1   10 1640
   0.0000 PS SAG       2 NAT00029 N SYSTEM   SYSLIBS  NAT00029      10 1640
   0.0000 CB SAG       2 NAT00029   SYSTEM            CMMPP    S
   0.0000 CA SAG       2 NAT00029   SYSTEM            CMMPP    S
   0.0000 PT SAG       2 NAT00029 N SYSTEM   SYSLIBS  NAT00029
   0.0000 PL SAG       1 MAINMENU   SYSTEM   SYSLIB   USR2003P
   0.0000 PS SAG       2 USR2003P N SYSTEM   SYSLIB   USR2003P      10 1640
   0.0001 PT SAG       2 USR2003P N SYSTEM   SYSLIB   USR2003P
   0.0000 CB SAG       1 MAINMENU   SYSTEM            CMUB     S
```

In der folgenden Tabelle werden die Spaltenüberschriften im Beispiel-Bildschirm und die Variablen erklärt, auf die sie sich beziehen. Bei einigen Ereignissen haben die zu PRLIB angezeigten Daten eine andere Bedeutung. Für weitere Informationen siehe die Kommentare in der Programm-Source und dem lokalen Datenbereich (LDA) RDCDATA. Erläuterungen zu den nachfolgend erwähnten Systemvariablen können Sie der *Systemvariablen*-Dokumentation entnehmen.

Spalte	Erklärung
ETIME1	Zeitintervall in Sekunden zwischen der Ausführung des aktuellen und des vorhergegangenen Ereignisses.
TY	Ereignistyp gemäß der Auflistung im Abschnitt *Datenerfassung auslösende Ereignisse*.
GCUID	Aktuelle Natural-Benutzerkennung gemäß Zuweisung durch die Natural-Systemvariable *USER.
LV	Programmebene.
GPGM	Name des aktuellen Programm gemäß Zuweisung durch die Natural-Systemvariable *PROGRAM.
T	Programmtyp.
GCAPL	Kennung (DBID) der aktuellen Anwendungs-Library gemäß Zuweisung durch die Natural-Systemvariable *APPLIC-ID.
PRLIB	Kennung (DBID) der Library, in der das Programm gespeichert ist. Gilt für Ereignisse des Typs Programm, z.B. Ereignistyp PL.
PRNAM	Name des Programms, das für den Ereignistyp geladen werden soll.
CO	Datenbank-Kommando.
DB	Datenbankkennung (DBID).
FN	Dateinummer der Datenbank (FNR).
RC	Datenbank- oder Programmaufruf-Antwortcode.

Interne Gültigkeitsprüfung und Tracing nach E/A-bezogenen Statements

Um `NAT1132`-Fehler festzustellen und das Debugging zu erleichtern, können durch **Trace Recording** Informationen gesammelt werden, die sich auf den Fehlertyp beziehen (z.B. Zeilen- und Feldlängen oder Puffer-Offsets). Nach der Verarbeitung eines Ein-/Ausgabe-Statements (z.B. `WRITE`, `DISPLAY`) wird eine Validierungsroutine ausgeführt, um die Puffer `IOPAGE` und `IOPATTR` auf Gültigkeit und Konsistenz zu prüfen, z.B. hinsichtlich des Vorhandenseins eines Zeilentrennungszeichens. Wenn ein möglicher Fehler entdeckt, wird die Meldung (`NAT1132 condition detected`) sowie Informationen zum Lokalisieren des Fehlers in den Natural Data Collector-Puffer geschrieben und die Trace-Aufzeichnung wird gestoppt, um zu vermeiden, dass dieses Ergebnis durch nachfolgende, irrelevante Daten verborgen wird.

Um das Aufzeichnen des Tracing nach NAT1132 zu aktivieren, geben Sie die folgenden Natural-Profilparameter an:

```
ITRACE=ON,RDCSIZE=128,TRACE=IO
```

Wenn ein Trace für ein von Ihnen ausgeführtes Programm erstellt wurde, können Sie die I/O-bezogenen Datensätze anzeigen, indem Sie `RDCDISIO` in der Library SYSRDC ausführen.

Beispiel-Ausgabe des Programms RDCDISIO

Der folgende Beispiel-Bildschirm zeigt einen Auszug aus einem Ausgabe-Report, der vom Beispiel-Programm RDCDISIO erstellt wurde. In der folgenden Tabelle werden die Spaltenüberschriften im Beispiel-Bildschirm erklärt.

```
 GCUID   LV   GPGM    GCAPL                      NDATA
-------- -- -------- -------- ----------------------------------------------
MK        1 LOGON    SYSTEM   INO2  08 F 005041700618 (R6)
MK        1 MYDEMO   MK       INO2  00 F 001A00200609 (R6)
MK        1 MYDEMO   MK       IOIPR    F 001A00200609 50 50 0050 0050
MK        1 MYDEMO   MK       INO2  04 F 001A00200609 (R6)
MK        1 MYDEMO   MK       INO2  00 F 001A00200609 (R6)
MK        1 MYDEMO   MK       IOIPR    F 001A00200609 50 50 0050 0050
MK        1 MYDEMO   MK       INO2     F 10C000C8C9C5 (R6) BEG OF  IOPRI
MK        1 MYDEMO   MK       IOEUP    F 000000A0 0F  (PATTRADF/LNG)
MK        1 MYDEMO   MK       IOEUP    F 000000A0 0F  (PATTRADF/LNG)
MK        1 MYDEMO   MK       IODRU    F 000000B0 3F  (PATTRADF/LNG)
MK        1 MYDEMO   MK       INO2     F 000A00301804 (R6) END OF  IOPRI
MK        2 MYDEMON  MK       INO2  00 N 001801700609 (R6)
MK        2 MYDEMON  MK       IOIPR    N 001801700609 4F 50 004F 0050
MK        2 MYDEMON  MK       IOIPR    N LINE SIZE HAS CHANGED
MK        2 MYDEMON  MK       INO2     N 058000800104 (R6) BEG OF  IOPRI
MK        2 MYDEMON  MK       IOEUP    N 000000F0 08  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 000000F0 08  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 000000F9 32  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 000000F9 32  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 0000012C 12  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 0000012C 12  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOPRI    N NAT1132 CONDITION DETECTED
MK        2 MYDEMON  MK       IOEUP    N 0000013F 32  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 0000013F 32  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 00000172 32  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 0000018F 1E  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IOEUP    N 0000018F 1E  (PATTRADF/LNG)
MK        2 MYDEMON  MK       IODRU    N 000001AE 30  (PATTRADF/LNG)
MK        2 MYDEMON  MK       INO2     N 00060180FF08 (R6) END OF  IOPRI
MK        2 MYDEMON  MK       INO2     N NAT1132 CONDITION 1 DETECTED
```

GCUID	Aktuelle Natural-Benutzerkennung gemäß Zuweisung durch die Natural-Systemvariable `*USER`.
LV	Programmebene.
GPGM	Name des aktuellen Programms gemäß Zuweisung durch die Natural-Systemvariable `*PROGRAM`.
GCAPL	Kennung der aktuellen Anwendungs-Library gemäß Zuweisung durch die Natural-Systemvariable `*APPLIC-ID`.
NDATA	Bis zu 250 Bytes Informationen zu Natural-Nukleus-Komponenten. Nur für Debugging-Zwecke durch den Support.

XXIII SYSRPC Utility

Das Dienstprogramm („Utility“) SYSRPC wird verwendet, um Natural Remote Procedure Calls auf der Client-Seite zu verwalten.

SYSRPC Utility aufrufen und beenden

Funktion Service Directory Maintenance benutzen

Elemente im Service Directory ersetzen

Interface-Objekte generieren - Allgemeine Aspekte

Einzelne Interface-Objekte mit Parameterangabe generieren

Mehrere Interface-Objekte generieren

Benötigte Puffergröße berechnen

Parameter verwalten (Parameter Maintenance)

Server-Kommandos ausführen (Server Command Execution)

Beim EntireX Broker registrierte Server auflisten

Übersicht über die SYSRPC-Direktkommandos und -Batch-Kommandos

Verwandte Themen:

- Informationen darüber, wie Sie die Funktionen der SYSRPC Utility anwenden, um einen Rahmen für die Kommunikation zwischen Server- und Client-Systemen zu schaffen, finden Sie in der *Natural Remote Procedure Call (RPC)*-Dokumentation.
- Erläuterungen zu Ausdrücken, die für die SYSRPC Utility relevant sind, finden Sie im Abschnitt *Natural RPC Terminology* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.
- Die Benutzung von SYSRPC kann durch Natural Security kontrolliert werden: siehe *Dienstprogramme (Utilities) schützen* in der *Natural Security*-Dokumentation.
- Informationen zu Anwendungsprogrammierschnittstellen, die zur Verwaltung von Remote Procedure Calls bereitgestellt werden, finden Sie unter *Anwendungsprogrammierschnittstellen zur Verwendung beim Natural RPC* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.
- Ausführliche Informationen zu den Funktionen und Komponenten von EntireX Broker finden Sie in der entsprechenden *EntireX Broker*-Dokumentation.

124 SYSRPC Utility aufrufen und beenden

- SYSRPC Utility aufrufen 1130
- SYSRPC Utility beenden 1131
- Online-Hilfe aufrufen 1131

Dieses Kapitel beschreibt, wie Sie die SYSRPC Utility aufrufen und beenden und wie Sie die Hilfe-Funktion aufrufen können.

SYSRPC Utility aufrufen

Zum Aufrufen der SYSRPC Utility können Sie entweder ein Systemkommando oder eine Menüfunktionen benutzen.

Um die SYSRPC Utility mittels Systemkommando aufzurufen:

- Geben Sie in der Kommandozeile das folgende Direktkommando ein:

  ```
  SYSRPC
  ```

  Drücken Sie ENTER.

  Das Menü **Client Maintenance** der SYSRPC Utility wird angezeigt.

Um die SYSRPC Utility über Menüfunktionen aufzurufen:

1. Wählen Sie im Natural **Main Menu** den Menüeintrag **Maintenance and Transfer Utilities** und drücken Sie ENTER.

   Das Menü **Maintenance and Transfer Utilities** wird angezeigt.
2. Wählen Sie den Menüeintrag **Maintain Remote Procedure Calls** und drücken Sie ENTER.

   Das Menü **Client Maintenance** der SYSRPC Utility wird angezeigt.

Im Menü **Client Maintenance** können Sie alle zur RPC-Wartung verfügbaren Funktionen aufrufen:

- **Service Directory Maintenance**
- **Generating Interface Objects - Allgemeine Aspekte**
- **Parameter Maintenance**
- **Server Command Execution**

Beschreibungen dieser Funktionen finden Sie in den verlinkten Kapiteln.

SYSRPC Utility beenden

Um die SYSRPC Utility zu beenden:

- Geben Sie im Feld **Code** im Menü **Client Maintenance** einen Punkt (.) ein.

  Oder:

  Drücken Sie PF3 (Exit).

Online-Hilfe aufrufen

Um die Online-Hilfe-Funktion aufzurufen:

- Drücken Sie PF1 (Help).

125 Funktion Service Directory Maintenance benutzen

- Service Directory-Konzept 1134
- Funktion Service Directory Maintenance aufrufen 1135
- Felder auf dem Service Directory-Bildschirm 1137
- Kommandos zur Pflege des Service Directory 1140
- API USR8216N zur Pflege des Service Directory 1144

Die Funktion **Service Directory Maintenance** wird verwendet, um ein Dienstverzeichnis zu pflegen, um das aufrufende Programm des Clients mit einem Subprogramm auf einem Server zu verbinden.

Die Informationen des Dienstverzeichnisses werden im Subprogramm `NATCLTGS` in der Library gespeichert, die mit dem `NTRPC/RPC`-Schlüsselwort-Subparameter `RPCSDIR` definiert ist (siehe *Parameter-Referenz*-Dokumentation). Wenn `RPCSDIR` gesetzt ist, verweist die **Service Directory Maintenance**-Funktion auf die mit `RPCSDIR` angegebene Library. Wenn `RPCSDIRRPCSDIR` nicht gesetzt ist (dies ist die Standardeinstellung), wird auf die Library verwiesen, in der Sie angemeldet sind. Melden Sie sich in diesem Fall bei der Library (oder einer ihrer Steplibs) an, die vom Client zur Laufzeit verwendet wird, bevor Sie die Funktion **Service Directory Maintenance** ausführen.

Der Name der Library, auf die bei der Pflege des Dienstverzeichnisses verwiesen wird, wird in der oberen rechten Ecke des Bildschirms **Service Directory** angezeigt (siehe ***Service Directory Maintenance aufrufen***). Wenn `RPCSDIR` gesetzt ist, steht im Bildschirmtitel **Central**, was darauf hinweist, dass die auf dem Bildschirm angezeigte Library nicht die Library ist, in der Sie gerade angemeldet sind, sondern die mit `RPCSDIR` angegebene zentrale Library.

Achtung!

Wenn das Subprogramm `NATCLTGS` in der Natural System Library SYSRPC gespeichert ist, empfehlen wir dringend, `NATCLTGS` in die vom Client verwendete Application Library (oder eine ihrer Steplibs) zu verschieben.

Weitere Informationen zur Anwendung der Funktion **Service Directory Maintenance** siehe *RPC-Server-Adressen festlegen* im Kapitel *Betrieb einer Natural RPC-Umgebung* in der *Natural RPC (Remote Procedure Call)* -Dokumentation.

Service Directory-Konzept

Ein Service Directory hat eine hierarchische Struktur mit einer kaskadierenden Liste zur Zuordnung von untergeordneten zu übergeordneten Feldern. Die höchste hierarchische Ebene ist Knoten (Node), die niedrigste ist Programm. Sie können Knoten, Server, Library und Programm nicht in derselben Zeile eingeben. Wenn Sie dies tun, erscheint eine entsprechende Fehlermeldung. Sie müssen den Wert eines untergeordneten Feldes in den Zeilen unterhalb des übergeordneten Feldes eingeben. Sie können einem Knoten mehrere Server, einem Server mehrere Libraries und einer Library mehrere Programme zuordnen.

- Knoten und Server

Knoten und Server

In *Beispiel 1 - Service Directory-Standardansicht* sind zwei Server bei einem Knoten definiert. Beide Server sind mit demselben Knoten verbunden: ETB045. Der Remote-CALLNAT zum Subprogramm SUB1 wird auf Server NRPC001 ausgeführt, während die Subprogramme SUB2 und SUB3 auf Server NRPC002 ausgeführt werden.

Die hier angegebenen Servernamen müssen mit den Servernamen identisch sein, die für den Server mit dem NTRPC/RPC-Schlüsselwort-Subparameter SRVNAME angegeben wurden, der in der *Parameter-Referenz*-Dokumentation beschrieben ist. Analog dazu muss der Knotenname im Service Directory mit dem Knotennamen identisch sein, der für den Server mit dem NTRPC/RPC-Schlüsselwort-Subparameter SRVNODE angegeben wurde, der in der *Parameter-Referenz*-Dokumentation beschrieben ist.

Funktion Service Directory Maintenance aufrufen

Vorsicht: Die Funktion **Service Directory Maintenance** ruft den Natural Editor auf. Daher können Daten, die im Quellcode-Arbeitsbereich gespeichert sind, beim Aufruf der Funktion **Service Directory Maintenance** verloren gehen. Eine entsprechende Meldung warnt Sie davor, bestehende Einträge unbeabsichtigt zu löschen: Drücken Sie PF12, um die Funktion abzubrechen oder drücken Sie ENTER, um die Aktion zu bestätigen und den Quellcode-Arbeitsbereich zu leeren.

Um die Funktion Service Directory Maintenance aufzurufen:

1 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
SM
```

2 Drücken Sie ENTER.

Wenn das Serviceverzeichnis bereits Servicedefinitionen enthält, erscheint ein Fenster mit der Meldung, dass vorhandene Dienstdefinitionen gefunden wurden:

```
Existing service definitions found
```

- Geben Sie im Fenster im Feld **Code** ein A (Standardwert) ein, um alte Definitionen zu behalten und neue anzuhängen, und drücken Sie ENTER.
- Oder: Geben Sie im Fenster im Feld **Code** ein I ein, um alle vorhandenen Definitionen zu ignorieren und aus dem Service Directory zu löschen, und drücken Sie ENTER.

Die Service Directory-Standardansicht wird angezeigt (Beispiel):

Beispiel 1 - Service Directory-Standardansicht

```
15:00:11            ***** NATURAL SYSRPC UTILITY *****              2016-07-18
                             - Service Directory -              Library SAGTST

            Node          Tr.        Server       Logon  Library     Program
1       ETB045_____________  B    ____________________   _   _________   _________
2       ___________________  _    NRPC001_____________   N   _________   _________
3       ___________________  _    ____________________   _   SYSTEM__   _________
4       ___________________  _    ____________________   _   _________   SUB1_____
5       ___________________  _    NRPC002_____________   Y   _________   _________
6       ___________________  _    ____________________   _   SYSTEM__   _________
7       ___________________  _    ____________________   _   _________   SUB2_____
8       ___________________  _    ____________________   _   _________   SUB3_____
9       ___________________  _    ____________________   _   _________   _________
10      ___________________  _    ____________________   _   _________   _________
11      ___________________  _    ____________________   _   _________   _________
12      ___________________  _    ____________________   _   _________   _________
13      ___________________  _    ____________________   _   _________   _________
14      ___________________  _    ____________________   _   _________   _________
15      ___________________  _    ____________________   _   _________   _________
16      ___________________  _    ____________________   _   _________   _________

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Long  Exit  Find  -H    +H    -P    +P    Top   Bot   >     Canc
```

Der Bildschirm **Service Directory** bietet maximal 500 Zeilen für die Eingabe.

3 Wenn Sie PF11 drücken oder das Kleiner-als-Zeichen (<) in die Kommandozeile eingeben, wird die erweiterte Knoten-/Serveransicht des **Service Directory**-Bildschirms ähnlich wie im folgenden Beispiel angezeigt:

Beispiel 2 - Erweiterte Service Directory-Ansicht

```
15:09:01           ***** NATURAL SYSRPC UTILITY *****            2016-07-18
                          - Service Directory -              Library SAGTST

                   Node                          Tr.              Server           Logon
1    ETB045_____________________________          B  ____________________________  _
2    ___________________________________          _  NRPC001_____________________  N
3    ___________________________________          _  ____________________________  _
4    ___________________________________          _  ____________________________  _
5    ___________________________________          _  NRPC002_____________________  Y
6    ___________________________________          _  ____________________________  _
7    ___________________________________          _  ____________________________  _
8    ___________________________________          _  ____________________________  _
9    ___________________________________          _  ____________________________  _
10   ___________________________________          _  ____________________________  _
11   ___________________________________          _  ____________________________  _
12   ___________________________________          _  ____________________________  _
13   ___________________________________          _  ____________________________  _
14   ___________________________________          _  ____________________________  _
15   ___________________________________          _  ____________________________  _
16   ___________________________________          _  ____________________________  _

Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Long  Exit  Find  -H    +H    -P    +P    Top   Bot   >     Canc
```

Wenn Sie PF11 drücken oder das Größer-als-Zeichen (>) in die Kommandozeile eingeben, wird die Standardansicht des **Service Directory**-Bildschirms angezeigt, wie in *Beispiel 1 - Service Directory-Standardansicht* dargestellt.

Felder auf dem Service Directory-Bildschirm

Der **Service Directory**-Bildschirm enthält die folgenden Eingabefelder (ein Eintrag pro Zeile):

Feld	Beschreibung
Node	Der Name des Knotens, an den der Remote `CALLNAT` gesendet wird. Siehe auch *Natural RPC Terminology* in der *Natural RPC (Remote Procedure Call)* -Dokumentation. Die maximale Länge der Eingabe ist wie folgt: Standardansicht des Bildschirms **Service Directory**: 16 characters Erweiterte Knoten-/Serveransicht des Bildschirms **Service Directory**: 32 Zeichen

Feld	Beschreibung
	Fenster **Long Name** (siehe PF2 in *Lokale Kommandos und PF-Tasten*) des Bildschirms **Service Directory**: bis zu 192 Zeichen.
Tr.	Das zu verwendende Transportprotokoll: `B` steht für EntireX Broker ACI-Protokoll.
Server	Der Name des Servers, an den der Remote `CALLNAT` gesendet wird. Siehe auch *Natural RPC Terminology* in der *Natural RPC (Remote Procedure Call)* -Dokumentation. Die maximale Länge der Eingabe ist wie folgt: Standardansicht des Bildschirms **Service Directory**: 16 Zeichen Erweiterte Knoten-/Serveransicht des Bildschirms **Service Directory**: 32 Zeichen
Logon	Initiiert eine Natural-Anmeldung beim Server. Dies ist auf Server- oder Knotenebene möglich und gilt für alle Definitionen, die auf einer hierarchisch niedrigeren Ebene vorgenommen wurden. Wenn die **Logon**-Option bei einem bestimmten Server gesetzt wurde, gilt sie für alle zugehörigen Library- und Subprogramm-Definitionen. Mögliche Werte: `Y` — Wenn die Option auf `Y` (Ja) gesetzt ist, initiiert der Client bei jeder nicht-konversationellen `CALLNAT`-Anforderung oder bei jedem Start einer Konversation eine Natural-Anmeldung beim Server unter Verwendung des aktuellen Library-Namens auf dem Client, unabhängig von den Libraries in der untergeordneten Spalte **Library**, die zum Feld **Server** gehört. Sie können die Anwendungsprogrammierschnittstelle `USR4008N` verwenden, um eine andere Library anzugeben (siehe auch *Bei einer anderen Library anmelden* im Kapitel *Logon-Option benutzen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation). `N` oder *`blank`* — Wenn `N` (Nein) oder kein Wert eingegeben wird, wird keine Anmeldung initiiert. Nach der Ausführung des Remote `CALLNAT` (erfolgreich oder nicht) oder am Ende einer Konversation wird die Server Library in ihren vorherigen Zustand zurückgesetzt. Weitere Informationen finden Sie unter *Logon-Option benutzen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation. Siehe auch ***Server-Kommandos ausführen (Server Command)***.
Library	Library SYSTEM oder der Name der Library, bei der Ihre Client-Anwendung während der Ausführung des Remote `CALLNAT` angemeldet ist.
Program	Der Name des Remote Subprogramms, auf das vom Client aus zugegriffen werden soll.

<table>
<tr><th>Feld</th><th>Beschreibung</th></tr>
<tr><td></td><td>Sie können einen Namen oder einen Namensbereich eingeben. Gültige Namen sind beliebige Kombinationen aus einem oder mehreren alphanumerischen Zeichen mit einem oder mehreren Stern-Zeichen (*) und/oder einem oder mehreren Fragezeichen (?).

Dabei bedeutet:

Stern (*) eine beliebige Zeichenkette, Fragezeichen (?) ein einzelnes Zeichen.

Ungültige Kombinationen sind:
<table><tr><td>*?</td><td>Ein Stern gefolgt von einem Fragezeichen wird in ?* umgewandelt.</td></tr><tr><td>**</td><td>Zwei oder mehr aufeinanderfolgende Sterne werden in einen einzelnen Stern umgewandelt..</td></tr></table></td></tr>
</table>

Auswahlkriterien für Knoten und Server

Zur Natural-Laufzeit hängt die Auswahl eines Knotens und Servers vom Wert der Felder **Program** und **Library** ab. Halten Sie die folgenden Bedingungen ein:

Nicht-konversationelles `CALLNAT`

1. Das Feld **Library** muss den Namen der aktuellen Application Library oder SYSTEM enthalten.
2. Der Name des in der `CALLNAT`-Anweisung angegebenen Subprogramms muss im Feld **Program** enthalten sein, das zum Feld **Library** in Punkt (1) gehört.

Konversationelles `CALLNAT`

1. Das Feld **Library** muss den Namen der aktuellen Application Library oder SYSTEM enthalten.
2. Alle im Statement `OPEN CONVERSATION` angegebenen Subprogramme müssen in einem **Program**-Feld enthalten sein, das zum Feld **Library** unter Punkt (1) gehört.

Der für ein nicht-konversationelles oder konversationelles `CALLNAT` verwendete Knoten und Server werden aus den übergeordneten Feldern **Node** und **Server** des Feldes **Library** in Punkt (1) übernommen.

Kommandos zur Pflege des Service Directory

Dieser Abschnitt enthält Informationen zu den Kommandos, die auf dem Bildschirm **Service Directory** zur Verfügung stehen.

- Zeilenkommandos
- Lokale Kommandos und PF-Tasten

Zeilenkommandos

Mit den Zeilenkommandos, die auf dem Bildschirm **Service Directory** zur Verfügung stehen, können Sie einzelne oder mehrere Zeilen, die Feldwerte enthalten, kopieren, verschieben oder löschen.

Geben Sie ein Zeilenkommando am Anfang einer Zeile ein, d.h. überschreiben Sie die fortlaufende Nummer und drücken Sie ENTER.

Siehe auch ***Um einen Zeilenblock zu kopieren oder zu verschieben*** und das lokale Kommando `RESET`.

Zeilenkommando	Funktion
`A`	Kopiert oder verschiebt den mit `CC` oder `MM` markierten Zeilenblock unter der Zeile, in der das Kommando eingegeben wurde.
`CC`	Markiert den zu kopierenden Zeilenblock.
`D`	Löscht die markierte Zeile.
`DD`	Markiert und löscht einen Zeilenblock. Markieren Sie einen Zeilenblock, indem Sie dieses Kommando in der ersten und letzten Zeile des Blocks eingeben und ENTER drücken, um das Kommando auszuführen.
`I`	Fügt fünf Leerzeilen unter der Zeile ein, in der das Kommando eingegeben wurde. Der Cursor wird in die erste neue Zeile unterhalb der Spalte **Server** oder **Library** gesetzt, je nachdem, welches Element (Knoten, Server, Library, Programm) in der Zeile enthalten ist, in der Sie das Kommando eingeben. Beispiele: Wenn die Zeile einen Knoten- oder Servernamen enthält, wird der Cursor unter die Spalte **Server** platziert. Enthält die Zeile den Namen einer Library oder eines Programms, wird der Cursor unter die Spalte **Library** gesetzt.
`MM`	Markiert den zu verschiebenden Zeilenblock.
`P`	Kopiert oder verschiebt den mit `CC` oder `MM` markierten Zeilenblock oberhalb der Zeile, in der der Befehl eingegeben wurde.

Um einen Zeilenblock zu kopieren oder zu verschieben:

1 Überschreiben Sie am Anfang der Zeile, in der der Block **beginnt**, die laufende Nummer mit einem der folgenden Zeilenkommandos.

Um den Block zu kopieren:

```
CC
```

Um den Block zu verschieben:

```
MM
```

2 Überschreiben Sie am Anfang der Zeile, in der der Block **endet**, die laufende Nummer mit einem der folgenden Zeilenkommandos.

- Um den Block zu kopieren:

```
CC
```

- Um den Block zu verschieben:

```
MM
```

3 Drücken Sie ENTER.

Die Zeilenkommandos verschwinden, die laufenden Nummern werden wieder angezeigt und der Zeilenblock ist markiert.

4 Geben Sie am Anfang der Zeile, unter oder über der Sie den markierten Zeilenblock platzieren wollen, eines der folgenden Zeilenkommandos ein, um den Block zu kopieren oder zu verschieben.

- Um den Block unter die angegebene Zeile zu kopieren oder zu verschieben:

```
A
```

- Um den Block über die angegebene Zeile zu kopieren oder zu verschieben:

```
P
```

Anmerkung: Die Zeilenkommandos `A` oder `P` sind nur in Zeilen ausführbar, in denen mindestens ein Feld gefüllt ist.

5 Drücken Sie ENTER.

Der Zeilenblock wird unter oder über die angegebene Zeile kopiert bzw. verschoben.

Lokale Kommandos und PF-Tasten

Im Bildschirm **Service Directory** können Sie die folgenden lokalen Kommandos in die Kommandozeile eingeben bzw. durch Drücken der entsprechend belegten PF-Tasten absetzen.

Lokales Kommando	PF-Taste	Funktion
EXPIRATION		Die Daten des Remote Directory werden zur Laufzeit geladen. Die Verfallszeit in Sekunden bestimmt die Gültigkeitsdauer dieser Daten. Wenn Directory-Daten nach der eingestellten Ablaufzeit angefordert werden, werden sie automatisch neu geladen. Wird die Verfallszeit auf 0 gesetzt, werden die Remote Directory-Daten nicht neu geladen. Mit dem lokalen Kommando EXPIRATION können Sie eine Verfallszeit in Sekunden eingeben, zum Beispiel EXPIRATION 86400. Das Maximum ist eine 8-stellige Zahl. Wenn Sie mit dem Kommando keinen Parameter angeben, erscheint das Fenster **Expiration Time**, in dem Sie die aktuelle Verfallszeit anzeigen oder ändern können.
RESET		Entfernt die mit einem Zeilenkommando gesetzten Zeilenmarkierungen, siehe *Zeilenkommandos*. Beachten Sie, dass, wenn Zeilen falsch markiert wurden, eine entsprechende Meldung erscheint und Sie den fehlerhaften Zeilenbefehl entfernen müssen, bevor Sie RESET eingeben.
	PF1	Ruft die Online-Hilfe des Editors auf.
	PF2	Öffnet das Fenster **Long Name**, in dem Sie einen Knotennamen mit bis zu 192 Zeichen eingeben können.
	PF3	Exit (Beenden): Fordert Sie auf, die Änderungen zu speichern und den Bildschirm **Service Directory** zu verlassen.
FIND	PF4	Ruft das Fenster **Find Item** auf, in dem Sie nach einem Namen suchen können:: **Find what** – Was suchen: Geben Sie eine alphanumerische Suchzeichenfolge mit bis zu 32 Zeichen ein. **Case sensitive** – Groß-/Kleinschreibung: Ersetzen Sie die Standardeinstellung N (Nein) durch Y (Ja), um zwischen Groß- und Kleinbuchstaben zu unterscheiden. **Whole words only** – Nur ganze Wörter: Ersetzen Sie die Standardeinstellung N (Nein) durch Y (Ja), um nur nach ganzen Wörtern zu suchen.

<table>
<tr><th>Lokales Kommando</th><th>PF-Taste</th><th>Funktion</th></tr>
<tr><td></td><td></td><td>Drücken Sie ENTER, um die Suche zu starten und von einem Treffer zum nächsten zu gelangen, falls ein solcher vorhanden ist. Drücken Sie PF4, um die Suche von Anfang an zu starten.

Die Treffer werden mit dem Cursor markiert.</td></tr>
<tr><td>REPLACE</td><td>PF16</td><td>Ruft das Fenster Replace Item (Element ersetzen) auf, in dem Sie nach einzelnen oder mehreren Namen suchen und diese ersetzen können (Groß- und Kleinschreibung wird dabei nicht beachtet):

<table>
<tr><td>Find</td><td>Suchen: Geben Sie eine alphanumerische Suchzeichenfolge mit bis zu 32 Zeichen ein.</td></tr>
<tr><td>Replace with</td><td>Ersetzen durch: Geben Sie eine alphanumerische Ersetzungszeichenfolge mit bis zu 32 Zeichen ein.</td></tr>
<tr><td>Whole words only</td><td>Nur ganze Wörter: Ersetzen Sie die Standardeinstellung N (Nein) durch Y (Ja), um nur nach ganzen Wörtern zu suchen.</td></tr>
<tr><td>Search only</td><td>Nur suchen: Standardmäßig werden alle Namen im Service-Directory nach Übereinstimmungen durchsucht (leerer Feldeintrag).

Sie können einen der folgenden Buchstaben eingeben, um die Suche auf eines der folgenden Elemente zu beschränken:
<table>
<tr><td>N</td><td>Nur Knotennamen</td></tr>
<tr><td>S</td><td>Nur Servernamen</td></tr>
<tr><td>L</td><td>Nur Library-Namen</td></tr>
<tr><td>P</td><td>Nur Programmnamen</td></tr>
</table>
</td></tr>
<tr><td>Replace all</td><td>Alles ersetzen: Ersetzt alle gefundenen Vorkommen der Suchzeichenfolge.</td></tr>
</table>

Drücken Sie ENTER, um mit der Suche zu beginnen und von einem Treffer zum nächsten zu gelangen, falls einer vorhanden ist.

Drücken Sie PF4, um die Suche von Anfang an zu starten.

Die Treffer werden mit dem Cursor markiert.</td></tr>
<tr><td>REPLACE replace-clause</td><td></td><td>Führt die im Fenster Replace Item angebotenen Ersetzungsfunktionen aus. Entspricht der replace-clause-Klausel des SYSRPC SM REPLACE-Kommandos.</td></tr>
</table>

Lokales Kommando	PF-Taste	Funktion
-H	PF5	Blättert eine halbe Seite vor/zurück.
+H	PF6	
-P	PF7	Blättert eine Seite rückwärts/vorwärts.
+P	PF8	
TOP	PF9	Blättert an den Anfang der Liste.
BOT	PF10	Blättert an das Ende der Liste.
	PF11	Schaltet zwischen der Standardansicht des **Service Directory**-Bildschirms (siehe *Beispiel 1 - Service Directory-Standardansicht*) und der erweiterten Ansicht der Felder **Knoten** und **Server** um (siehe *Beispiel 2 - Erweiterte Service Directory-Ansicht*).
>	PF11	Zeigt die erweiterte Ansicht der Felder **Node** und **Server** an. In der erweiterten Knoten-/Serveransicht werden die Felder **Library** und **Program** nicht angezeigt, wie in *Beispiel 2 - Erweiterte Service Directory-Ansicht* dargestellt. Zeigt die Standardansicht des Bildschirms Dienstverzeichnis an, wie in Beispiel 1 - Standardansicht des Dienstverzeichnisses dargestellt.
<	PF11	Zeigt die Standardansicht des Bildschirms **Service Directory** an, wie in *Beispiel 1 - Service Directory-Standardansicht* dargestellt.
CANCEL	PF12	Beendet den Bildschirm **Service Directory**, ohne Änderungen zu speichern.

API USR8216N zur Pflege des Service Directory

Sie können die Anwendungsprogrammierschnittstelle (API) USR8216N verwenden, um Service Directory Maintenance-Funktionen auszuführen.

USR8216N ruft ein vorhandenes Service Directory ab und fügt Einträge im Dienstverzeichnis hinzu, ändert oder löscht sie.

USR8216N wird in der Natural SYSEXT System Library ausgeliefert. Anleitung zur Handhabung siehe *Eine Natural API verwenden* in *SYSEXT Utility*-Beschreibung.

126 Elemente im Service Directory ersetzen

- Syntax des Direktkommandos SYSRPC SM REPLACE 1146

Mit dem Direktkommando SYSRPC SM REPLACE können Sie die Namen von Knoten, Servern, Libraries und Programmen ersetzen, die in einem Service Directory definiert sind.

Das Direktkommando SYSRPC SM REPLACE entspricht dem Kommando SM REPLACE, das Sie im Bildschirm **Service Directory Maintenance** eingeben können. Siehe *Kommandos zur Pflege des Service Directory*.

Das Direktkommando SYSRPC SM REPLACE kann online und im Batch-Modus verwendet werden.

Dieses Kapitel enthält folgende Informationen:

Syntax des Direktkommandos SYSRPC SM REPLACE

```
SYSRPC SM REPLACE replace-clause
```

replace-clause

Die Replace-Klausel (*replace-clause*) des Kommandos REPLACE entspricht der Replace-Klausel *replace-clause* des Direktkommandos SM REPLACE.

```
[ ANY     ]
[ NODE    ]
[ SERVER  ]  search-string WITH replace-string [ ALL   ] [WHOLE]
[ LIBRARY ]                                    [ FIRST ]
[ PROGRAM ]
```

Die Syntaxelemente werden in der folgenden Tabelle erläutert:

ANY	Sucht nach allen im Service Directory angegebenen Namen. Dies ist der Standardwert.
NODE	Sucht nur nach Knotennamen.
SERVER	Sucht nur nach Servernamen.
LIBRARY	Sucht nur nach Library-Namen.
PROGRAM	Sucht nur nach Programmnamen.
search-string	Eine alphanumerische Suchzeichenfolge mit bis zu 32 Zeichen.
WITH	Leitet den *replace-string* ein.
replace-string	Eine alphanumerische Ersetzungszeichenfolge mit bis zu 32 Zeichen.
ALL	Ersetzt alle gefundenen Vorkommen der Suchzeichenfolge.
FIRST	Ersetzt nur das erste Vorkommen der gefundenen Suchzeichenfolge. Dies ist der Standardwert.

WHOLE	Ersetzt nur Vorkommen, die mit der gesamten Suchzeichenfolge übereinstimmen.

Anmerkung: Bei der Suche wird *nicht* zwischen Groß- und Kleinschreibung unterschieden.

127 Interface-Objekte generieren - Allgemeine Aspekte

Ein Interface-Objekt ist ein Natural-Subprogramm, das dazu dient, das aufrufende Programm des Clients mit einem Subprogramm auf einem Server zu verbinden.

Interface-Objekte sind eigentlich nicht erforderlich, wenn die automatische Natural RPC-Ausführung verwendet wird, mit einer wichtigen Ausnahme, die unten beschrieben wird. Es kann jedoch von Vorteil sein, Interface-Objekte zu erzeugen, wie in *Interface Objects and Automatic RPC Execution* im Kapitel *Operating a Natural RPC Environment* in der *Natural RPC (Remote Procedure Call)*-Dokumentation erläutert.

Hinweis bezüglich EntireX RPC Servern:
Es wird empfohlen, ein Interface-Objekt zu generieren, wenn Sie einen EntireX RPC Server aufrufen wollen. In diesem Fall müssen Sie die in diesem Abschnitt beschriebene SYSRPC-Funktion **Interface Object Generation** verwenden, um die gleiche Gruppenstruktur und die gleichen Attribute (Parameterrichtung) wie in der IDL (Interface Definition Language) des Subprogramms zu definieren. Wenn die IDL keine Gruppenstrukturen enthält, wird empfohlen, vor der Generierung des Interface-Objekts das Direktkommando `COMPAT IDL` zu verwenden. Einzelheiten hierzu finden Sie unter ***Besondere Aspekte beim Aufrufen von EntireX RPC Servern*** Spezielle Überlegungen zum Aufruf von EntireX RPC-Servern.

Hinweis bezüglich Reliable RPC:
Es wird empfohlen, ein Interface-Objekt zu generieren, wenn Sie Reliable RPC verwenden möchten. Wenn die Parameterdefinitionen keine Gruppenstrukturen enthalten, müssen Sie vor der Generierung des Interface-Objekts das Kommando `COMPAT IDL` einstellen. Einzelheiten hierzu finden Sie unter ***Besondere Aspekte bei Reliable RPC***.

Sie können ein Interface-Objekt aus neuen Parameterdefinitionen oder aus bestehenden Definitionen in einem Subprogramm oder einem Parameterdatenbereich (PDA) generieren.

Vorsicht: Das für die Generierung eines Interface-Objekts verwendete Subprogramm oder PDA kann nicht mehr in der lokalen Umgebung auf der Client-Seite referenziert werden.

Die Funktion **Interface Object Generation** ändert den Quellcode des Subprogramms vollständig, so dass es für lokale Programmaufrufe unbrauchbar wird.

Die folgenden Abschnitte beschreiben die Funktionen und Kommandos, die für die Generierung einzelner oder mehrerer Interface-Objekte zur Verfügung stehen:

- **Einzelne Interface-Objekte mit Parameterangabe generieren**
- **Mehrere Interface-Objekte generieren**

128 Einzelne Interface-Objekte mit Parameterangabe generieren

- Funktion Interface Object Generation benutzen 1152
- Parameter angeben 1154
- Beispiele für die Generierung von Interface-Objekten 1157

Die Funktion **Interface Object Generation** bietet die Möglichkeit, einzelne Interface-Objekte online in einem separaten Bildschirm zu generieren. Die benötigten Parameterdefinitionen geben Sie entweder ein oder lesen sie aus einem bestehenden Subprogramm oder einem Parameterdatenbereich (s. Parameter Data Area, PDA) ein.

Die Generierung eines Schnittstellenobjekts aus einer PDA erspart Ihnen die Erstellung eines Subprogramms und die Definition eines Inline-Parameterdatenbereichs.

Funktion Interface Object Generation benutzen

Interface-Objekte werden in der aktuellen Natural Library in der aktuellen Systemdatei generiert. Es wird daher dringend empfohlen, sich zur Ausführungszeit des Remote `CALLNAT` in der vom Client verwendeten Application Library (oder einer ihrer Steplibs) anzumelden.

Wichtig: Die Funktion **Interface Object Generation** überschreibt alle im Quellcode-Arbeitsbereich enthaltenen Daten. Wenn Sie die Funktion aufrufen, werden Sie durch eine entsprechende Meldung davor gewarnt, vorhandene Daten unbeabsichtigt zu löschen: Drücken Sie PF12, um abzubrechen, oder drücken Sie ENTER, um die Aktion zu bestätigen und den Inhalt des Quellcode-Arbeitsbereich zu überschreiben.

Um ein einzelnes Interface-Objekt zu generieren:

1 Bevor Sie die SYSRPC Utility aufrufen, sollten Sie Folgendes beachten:

- Melden Sie sich bei der Library an, in der Sie das Interface-Objekt generieren wollen.
- Wenn Sie ein Interface-Objekt aus einem Parameter-Datenbereich (PDA) generieren: Benennen Sie den PDA, von dem Sie das Interface-Objekt generieren wollen, um oder kopieren Sie ihn, wenn es auf der Client-Seite noch Objekte gibt, die auf diesen PDA verweisen. Der neue Name des PDA muss mit dem Namen des Remote `CALLNAT`-Programms identisch sein.

2 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
IG
```

3 Drücken Sie ENTER.

Der Bildschirm **Generate Interface Object** erscheint.

4 Geben Sie im Feld **Program Name** den Namen des zu generierenden Interface-Objekts ein.

Der Name des Interface-Objekts muss mit dem Namen des Remote `CALLNAT`-Programms identisch sein.

Das Feld **Library** ist mit dem Namen der aktuellen Library vorbelegt und kann nicht geändert werden.

Geben Sie im Feld **Compression** die Komprimierungsart `0`, `1` oder `2` an (Standardwert ist `1`). Siehe *Komprimierung verwenden* in *Betrieb einer Natural RPC-Umgebung* in der *Natural RPC (Remote Procedure Call)* -Dokumentation.

5 Drücken Sie ENTER.

- Wenn der im Feld **Program Name** eingegebene Name mit dem Namen eines Objekts übereinstimmt, das bereits in der zugeordneten Library existiert, erscheint ein Fenster mit einer entsprechenden Meldung:

  Geben Sie ein `N` (Nein) ein und drücken Sie ENTER, wenn Sie den Vorgang abbrechen möchten. Sie kehren dann zum Menü **Client Maintenance** zurück.

  Oder: Geben Sie ein `Y` (Ja) ein und drücken Sie ENTER, wenn Sie mit der Generierung von Interface-Objekten fortfahren wollen.

  Wenn der angegebene Name mit einem katalogisierten Objekt vom Typ Subprogramm oder PDA identisch ist, werden die Parameterdefinitionen des entsprechenden Subprogramms oder PDA auf dem Bildschirm **Interface Generation** angezeigt.

  Ist der angegebene Name identisch mit einem bereits generierten Interface-Objekt, zu dem auch ein Source-Objekt existiert, werden alle Feldattribute (siehe auch ***Parameter angeben***) beibehalten.
- Wenn die im Feld **Program Name** eingegebene Zeichenkette nicht mit dem Namen eines Objekts übereinstimmt, das bereits in der zugewiesenen Library existiert, wird ein leerer **Interface Object Generation**-Bildschirm angezeigt.

6 Fügen Sie auf dem Bildschirm **Interface Object Generation** die im Interface-Objekt zu verwendenden Parameter hinzu oder ändern Sie sie, wie in ***Parameter angeben*** beschrieben.

Die Kommandos auf dem Bildschirm **Interface Object Generation** entsprechen den Kommandos, die unter ***Kommandos zur Pflege des Service Directory*** im Kapitel *Funktion Service Directory Maintenance benutzen* beschrieben sind.

7 Drücken Sie ENTER, um das Interface-Objekt zu generieren und zu beenden. Das Interface-Objekt wird in der zugewiesenen Library generiert.

Es erscheint das Fenster **SYSRPC - Information**, das die Größe des Interface-Objekts anzeigt, die für das Senden von Daten vom Client zum Server oder umgekehrt erforderlich ist. Die Größe beinhaltet die internen RPC-Informationen, die für das Interface-Objekt verwendet werden. Die Angabe der Größe hilft Ihnen bei der Konfiguration der verwendeten Middleware-Schicht, z. B. der Broker-Attributdatei, wenn EntireX Broker verwendet wird.

Die folgende Meldung erscheint im Fenster **SYSRPC - Information**, wenn Sie ein Interface-Objekt aus dem Beispiel-Subprogramm `TESTS5` generieren (siehe ***Beispiel 1*** weiter unten):

```
Interface Object TESTS5 is generated in library SAGTEST (99,49).
   It requires:
        Send length: 2249 bytes
     Receive length: 2221 bytes
```

Bei Verwendung von dynamischen Parametern, X-Arrays oder X-Gruppen-Arrays gibt diese Meldung nur die Mindestlängenanforderungen an. Der tatsächliche Längenbedarf kann nur während der Programmausführung ermittelt werden und kann von Aufruf zu Aufruf unterschiedlich sein. Überschreitet die Sendelänge oder die Empfangslänge die Entire-Net-Work-Grenze von 32000 Bytes, erscheint ein Fenster mit einer entsprechenden Warnung:

Geben Sie ein `Y` (Ja) ein, um fortzufahren, oder ein `N` (Nein), um die Generierung abzubrechen. Wenn Sie ein `Y` eingeben, wird diese Einstellung für die gesamte SYSRPC-Sitzung beibehalten, d.h. Sie können mit der Generierung von Interface-Objekten fortfahren, ohne weitere Warnungen zu erhalten.

Überschreitet die Gesamtzahl der gesendeten oder empfangenen Daten (ohne interne RPC-Informationen) die Grenze von 1073739357 Bytes (1 GB minus 2467 Bytes interne RPC-Informationen), bricht die SYSRPC Utility die Verarbeitung ab und gibt eine entsprechende Fehlermeldung aus. Diese Fehlermeldung zeigt die Zwischensumme der Daten in Bytes an, die in dem Feld, bis zu dem die Zwischensumme berechnet wurde, übertragen werden konnten. Das entsprechende Feld ist dann markiert. Reduzieren Sie in diesem Fall die Datenmenge, bevor Sie die Generierung fortsetzen.

Wenn das Interface-Objekt in der Natural System Library SYSRPC generiert wurde, müssen Sie es mit der Natural Transfer Utility **SYSMAIN** oder dem **Object Handler** in die Application Library oder Steplib verschieben. Beachten Sie, dass Sie eventuell den Quellcode des Interface-Objekts in der Zielumgebung neu katalogisieren müssen.

Parameter angeben

In den Eingabefeldern auf dem Bildschirm **Interface Object Generation** können Sie die Parameterdefinitionen eingeben, die im Interface-Objekt verwendet werden. Sie können bis zu 5000 Parameter angeben. Sofern in der nachstehenden Tabelle nicht anders angegeben, ist die Eingabe in die Felder obligatorisch.

Feld	Beschreibung
Level	Die Ebene des Feldes. Ebene kann eine Zahl im Bereich von `01` (höchste Ebene) bis `99` (niedrigste Ebene) sein. Die führende `0` ist optional. Für ein Beispiel einer Gruppendefinition siehe *Gruppen definieren* und *Beispiel 2*.

Feld	Beschreibung
Attr	Das Attribut des Parameters: `M` (Modifiable/änderbar - INOUT), `O` (Output/Ausgabe - OUT) oder `I` (Input/Eingabe - IN). Parameter, denen eine Ebenennummer von `2` oder höher zugewiesen wurde, werden als Teil einer Gruppe betrachtet. Parameter innerhalb einer Gruppe müssen dasselbe Attribut haben wie die unmittelbar vorangehende Gruppe, die eine Ebene höher zugeordnet ist. Bei verschachtelten Gruppen ist dies das Attribut der Gruppe mit der höchsten Ebene. Ein Beispiel für eine Gruppendefinition finden Sie unter ***Beispiel 2***. Wenn ein Interface-Objekt aus einem Subprogramm oder aus einem PDA generiert wurde, ist das Attribut standardmäßig `M`, was eventuell geändert werden muss. Wenn ein Interface-Objekt aus einem anderen Interface-Objekt generiert wurde, werden die für das ursprüngliche Objekt angegebenen Attributwerte beibehalten. Das generierte Interface-Objekt enthält einen Kommentar, der angibt, welches Attribut für den Parameter angegeben wurde: `IN`, `OUT` oder `INOUT`.
Type	Ein Natural-Datenformat wie `N` (numerisch) und `G` (Gruppe) oder `K` (Kanji). Die Natural-Datenformate `C` (Attributsteuerung) und `HANDLE OF OBJECT` sind nicht zulässig. Beschreibung der Natural-Datenformate siehe *Format und Länge von Benutzervariablen* und *Spezielle Formate* im Abschnitt *Benutzervariablen* im *Leitfaden zur Programmierung*.
Length	Die Länge des Parameters oder `DYNAMIC`. Dieses Feld gilt nicht für die folgenden Natural-Datenformate: `D` (Datum), `G` (Gruppe), `L` (logisch) und `T` (Zeit). Das Natural-Datenformat `A` ist auf 1073739357 Bytes beschränkt, das Natural-Datenformat `B` auf 536869678 Bytes. `DYNAMIC` bezeichnet einen dynamischen Parameter und gilt für die Natural-Datenformate `A` und `B`.
Prec	Gilt nur für die Natural-Datenformate `N` (numerisch) und `P` (gepackt). Optional. Die Genauigkeit des Parameters, d. h. die Anzahl der Stellen nach dem Dezimalzeichen.
Dimension 1/2/3	Gilt nur bei Arrays. Optional. Die erste, zweite und dritte Dimension des Parameters. Ein X-Array oder ein X-Gruppen-Array wird durch Eingabe eines Sterns (*) für eine Dimension angegeben. Siehe auch ***X-Arrays und X-Gruppen-Arrays definieren***.

Dieser Abschnitt enthält Informationen über:

- Gruppen definieren
- X-Arrays und X-Gruppen-Arrays definieren

- Besondere Aspekte beim Reliable RPC
- Besondere Aspekte beim Aufrufen von EntireX RPC Servern

Gruppen definieren

Sie müssen nur eine Gruppenstruktur für ein Client-Natural-Objekt definieren, das ein Nicht-Natural-Objekt auf einem EntireX-RPC-Server aufruft. Die Gruppenstruktur muss mit der IDL-Definition in EntireX übereinstimmen (siehe *Besondere Aspekte beim Aufrufen von EntireX RPC Servern*). Für ein Natural-Client-Objekt, das ein Subprogramm auf einem Natural-RPC-Server aufruft, ist eine Gruppenstruktur nicht erforderlich.

Gruppen-Arrays und X-Gruppen-Arrays, die von einem Client-Natural-Objekt an ein Interface-Objekt übergeben werden, müssen zusammenhängend sein. Daher wird dringend empfohlen, immer ein vollständiges Array an das Objekt zu übergeben, indem Sie für alle Dimensionen die Stern-Notation (*) verwenden. Außerdem wird dringend empfohlen, im Client-Natural-Programm, im Interface-Objekt und im Server-Programm identische Datendefinitionen zu verwenden.

Für ein Beispiel einer Gruppendefinition siehe *Beispiel 2*.

X-Arrays und X-Gruppen-Arrays definieren

Wenn eine Dimension eines Parameters erweiterbar ist, sind alle anderen Dimensionen des Parameters ebenfalls erweiterbar. Wenn Sie in einem Subprogramm erweiterbare und feste Dimensionen für einen Parameter definieren, gibt die Funktion **Interface Object Generation** eine Warnung aus und ändert automatisch die feste Dimension in eine erweiterbare Dimension, wie in *Beispiel 3* gezeigt. In einer Gruppenstruktur können Sie für jede Ebene entweder eine erweiterbare oder eine feste Dimension definieren. Ein automatischer Wechsel von einer festen Dimension zu einer erweiterbaren Dimension zwischen den Ebenen findet nicht statt.

Natural RPC unterstützt nur erweiterbare Obergrenzen. Alle X-Arrays und X-Gruppen-Arrays im generierten `DEFINE DATA PARAMETER`-Bereich des Interface-Objekts sind daher als `(1:*)` definiert.

Vorsicht: Wenn Sie ein Interface-Objekt aus einem Subprogramm oder einem PDA generieren, das ein X-Array oder X-Gruppen-Array mit einer erweiterbaren Untergrenze enthält, wird die erweiterbare Untergrenze in eine erweiterbare Obergrenze umgewandelt.

Für ein Beispiel einer Gruppendefinition siehe *Beispiel 3*.

Besondere Aspekte beim Reliable RPC

Wenn Sie Reliable RPC verwenden wollen und Ihre Parameterdefinitionen keine Gruppenstrukturen enthalten, müssen Sie `COMPAT IDL` vor der Generierung des Interface-Objekts setzen.

Besondere Aspekte beim Aufrufen von EntireX RPC Servern

Die Attributdefinitionen auf dem Bildschirm **Interface Object Generation** spiegeln die Perspektive des Clients wider. Umgekehrt spiegelt die Parameterrichtung in der IDL-Definition die Perspektive des Servers wider. Dies bedeutet:

- `OUT` im Bildschirm **Interface Object Generation** entspricht `IN` in der IDL-Definition.
- `IN` im Bildschirm **Interface Object Generation** entspricht `OUT` in der IDL-Definition.

Wenn Sie einen EntireX-RPC-Server aufrufen möchten und die Parameterdefinitionen auf dem Bildschirm **Interface Object Generation** Gruppenstrukturen enthalten, müssen die Gruppenstruktur- und Attributdefinitionen auf dem Bildschirm **Interface Object Generation** mit der Gruppenstruktur- und Parameterrichtung in der IDL-Definition übereinstimmen.

Wenn Sie einen EntireX-RPC-Server aufrufen möchten und die entsprechende IDL-Datei keine Gruppenstrukturen enthält, empfiehlt es sich, vor der Generierung des Interface-Objekts `COMPAT IDL` zu setzen. In diesem Fall müssen die Attributdefinitionen auf dem Bildschirm **Interface Object Generation** mit der Parameterrichtung in der IDL-Definition übereinstimmen.

Beispiele für die Generierung von Interface-Objekten

In diesem Abschnitt finden Sie Beispiele für Natural-Subprogramme und die daraus generierten Interface-Objekte.

Die unten angegebenen Parameterdefinitionen stammen aus Beispiel-Subprogrammen, die in der Natural System Library SYSRPC ausgeliefert werden.

Beispiel 1

Der folgende `DEFINE DATA PARAMETER`-Bereich (Beispiel-Subprogramm `TESTS5`) zeigt vier änderbare Parameter und die entsprechenden Parameterdefinitionen auf dem Bildschirm **Interface Object Generation**:

```
DEFINE DATA
PARAMETER
  01 #IDENTIFIER  (A10)
  01 #N-OF-ID     (I4)
  01 #FREQ        (P5.2)
  01 #A100        (A100/5,4)
```

Interface Object Generation								
	Level	Attr	Type	Length	Prec	Dimension 1	Dimension 2	Dimension 3
1	01	M	A	10				
2	01	M	I	4				
3	01	M	P	5	2			
4	01	M	A	100		5	4	

Beispiel 2

Der folgende `DEFINE DATA PARAMETER`-Bereich (Beispiel-Subprogramm `TESTS6`) zeigt eine geschachtelte Gruppenstruktur und die entsprechenden Parameterdefinitionen auf dem Bildschirm **Interface Object Generation**:

```
DEFINE DATA
PARAMETER
 01 GROUP-1(10)
   02 A (A20)
   02 B (A20)
   02 GROUP-2(20)
     03 C (A10/5)
     03 D (A10)
 01 LINE (A) DYNAMIC
```

Interface Object Generation								
	Level	Attr	Type	Length	Prec	Dimension 1	Dimension 2	Dimension 3
1	01	M	G			10		
2	02	M	A	20				
3	02	M	A	20				
4	02	M	G			20		
5	03	M	A	10		5		
6	03	M	A	10				
7	01	M	A	DYNAMIC				

Beispiel 3

Der folgende `DEFINE DATA PARAMETER`-Bereich (Beispiel-Subprogramm `TESTS7`) zeigt eine geschachtelte Gruppenstruktur mit erweiterbaren Dimensionen und die entsprechenden Parameterdefinitionen auf dem Bildschirm **Interface Object Generation**:

```
DEFINE DATA
PARAMETER
 01 GROUP-1(10)
   02 A (A20)
   02 B (A20)
   02 GROUP-2(0:*)
     03 C (A10/5)
     03 D (A10)
 01 LINE (A) DYNAMIC
```

Interface Object Generation								
	Level	Attr	Type	Length	Prec	Dimension 1	Dimension 2	Dimension 3
1	01	M	G			10		
2	02	M	A	20				
3	02	M	A	20				
4	02	M	G			*		
5	03	M	A	10		5		
6	03	M	A	10				
7	01	M	A	DYNAMIC				

129 Mehrere Interface-Objekte generieren

- Direktkommando SYSRPC SGMASS benutzen 1162
- Name und Komprimierungsart angeben 1164

Sie können einzelne oder mehrere Interface-Objekte entweder online oder im Batch-Modus erzeugen, indem Sie das Direktkommando `SYSRPC SGMASS` verwenden.

Interface-Objekte können Sie aus Subprogrammen oder Parameterdatenbereichen (PDAs) generieren.

Direktkommando SYSRPC SGMASS benutzen

Um Interface-Objekte online zu erzeugen, können Sie das Direktkommando `SYSRPC SGMASS` an jeder Natural-Eingabeaufforderung eingeben

Der folgende Abschnitt enthält Informationen über:

- Syntax des Direktkommandos SYSRPC SGMASS
- SYSRPC SGMASS-Report

Syntax des Direktkommandos SYSRPC SGMASS

Die Syntax, die beim Direktkommando `SYSRPC SGMASS` gilt, ist im folgenden Diagramm dargestellt:

```
SYSRPC SGMASS [name] [compression]
```

Die Syntax-Elemente *name* und *compression* werden im Abschnitt ***Name und Komprimierungsart angeben*** erläutert.

SYSRPC SGMASS-Report

Das Direktkommando `SYSRPC SGMASS` erzeugt einen Bericht, der die mit dem Kommando erzeugten Interface-Objekte auflistet, wie im folgenden Beispiel gezeigt:

```
Page       1                                              2018-01-17 11:58:33

    SYSRPC - Interface Object Generation in Library SAGTEST


Generation Criteria:

    Object name or range: RPC*
              Compression: 1


Generation Results (sizes in bytes):

 Number of objects found:       8
     Maximum send length:  200228
  Maximum receive length: 1024192


Object     Type      Send Length   Receive Length   Message
--------   ----   --------------   --------------   --------------
RPCCALL1   N                 209              202
RPCCALL2   N                 219              240   Compression=2
RPCCALL3   N                 204              193
MORE
```

Der Report ist in drei Abschnitte gegliedert, die die folgenden Informationen enthalten:

- **Generation Criteria - Generierungskriterien:**
  Die Kriterien, auf deren Grundlage das/die Interface-Objekt(e) generiert wurden: ein einzelner Objektname oder ein Namensbereich (hier: `RPC*`) und die Komprimierungsart (hier: `1`).
- **Generation Results (sizes in bytes) - Generierungsergebnisse (Größe in Bytes):**
  Die Anzahl der Objekte, die für die Generierung von Interface-Objekten ausgewählt wurden.

  Die maximalen Puffergrößen, die alle generierten Interface-Objekte zum Senden und Empfangen von Daten vom Client benötigen.
- **Object List - Objektliste:**
  Name und Typ (hier: `N` für Objekttyp Subprogramm) jedes generierten Interface-Objekts. Die Puffergrößen, die jedes Objekt zum Senden (**Send Length**) und Empfangen (**Receive Length**) von Daten vom Client benötigt. Ein möglicher Kommentar zu jeder Generierung eines Interface-Objekts in der Spalte **Message**. Im obigen Beispiel zeigt `Compression=2` an, dass das Objekt `RPCCALL2` nicht, wie im Kommando gefordert, mit der Komprimierung `1` erzeugt wurde. Die Objektliste ist in alphabetischer Reihenfolge nach Objektnamen sortiert.

Wenn die Aufforderung `MORE` erscheint, können Sie ENTER drücken, um zum Ende des Reports zu blättern.

Wenn die Generierung der Interface-Objekte für einzelne oder mehrere Objekte fehlschlägt, zeigt der Report die Anzahl der betroffenen Objekte und die entsprechenden Fehlermeldungen an.

Name und Komprimierungsart angeben

Sie können die Objekte (Subprogramme oder PDAs) angeben, die für die Generierung von Interface-Objekten ausgewählt werden sollen, sowie die Art der Komprimierung, die verwendet werden soll.

- Name
- Komprimierung

Name

Sie können einen Objektnamen oder einen Bereich von Namen angeben. Die Angabe eines Objektnamens oder eines Namensbereichs ist optional.

Vorsicht: Wenn Sie keinen Objektnamen oder einen Namensbereich angeben, werden bis auf wenige Ausnahmen (siehe unten) alle Subprogramme oder PDAs in der aktuellen Library in Interface-Objekte umgewandelt.

Gültige Namensangaben sind im Folgenden beschrieben, wobei *value* eine beliebige Kombination aus einem oder mehreren alphanumerischen Zeichen ist:

Eingabe	Ausgewählte Objekte
*	Alle Subprogramme oder PDAs. Dies ist die Standardeinstellung.
value	Ein Subprogramm oder ein PDA mit einem Namen, der gleich *value* ist.
*value**	Alle Subprogramme oder PDAs mit Namen, die mit *value* beginnen.
value<	Alle Subprogramme oder PDAs, deren Name kleiner oder gleich *value* ist.
value>	Alle Subprogramme oder PDAs, deren Name größer oder gleich *value* ist.

Ausnahmen bei Namen

In der Natural System Library SYSRPC nimmt `SYSRPC SGMASS` alle Subprogramme von der Generierung von Interface-Objekten aus, deren Namen mit einem der folgenden Präfixe beginnen: RDS, RPC, NAT, NAD oder NSC.

In User Libraries nimmt `SYSRPC SGMASS` das Subprogramm NATCLTGS von der Generierung von Interface-Objekten aus.

Komprimierung

Sie können eine der folgenden Komprimierungsarten angeben: `0`, `1`, `2`. Die Angabe der Komprimierung ist optional. Der Standardtyp für die Generierung von Interface-Objekten ist `1`.

Siehe auch *Komprimierung verwenden* in *Betrieb einer Natural RPC-Umgebung* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

130 Benötigte Puffergröße berechnen

- Direktkommando SYSRPC CSMASS benutzen 1168
- Namensangabe für Größenberechnung und Komprimierung 1170

Das Direktkommando `SYSRPC CSMASS` wird verwendet, um die Puffergröße zu berechnen, die RPC-Aufrufe ohne Interface-Objekte benötigen, um Daten vom Client zum Server oder umgekehrt zu senden. Die Angabe der Größe hilft Ihnen, die verwendete Middleware-Schicht zu konfigurieren, z.B. die Broker-Attributdatei, wenn EntireX Broker verwendet wird.

Falls gewünscht, können Sie auch Größenberechnungen für Interface-Objekte durchführen, auch wenn die Größen bereits bei der Generierung der Interface-Objekte berechnet werden.

Das Direktkommando `SYSRPC CSMASS` kann online oder im Batch-Modus verwendet werden.

Direktkommando SYSRPC CSMASS benutzen

Um die benötigte Puffergröße online zu berechnen, können Sie das Direktkommando `SYSRPC CSMASS` an jeder Natural-Eingabeaufforderung eingeben.

Der folgende Abschnitt enthält Informationen über:

- Syntax des Direktkommandos SYSRPC CSMASS
- SYSRPC CSMASS-Report

Syntax des Direktkommandos SYSRPC CSMASS

Die Syntax, die beim Direktkommando `SYSRPC CSMASS` gilt, ist im folgenden Diagramm dargestellt:

```
SYSRPC CSMASS [name] [compression]
```

Die Syntax-Elemente *name* und *compression* werden im Abschnitt *Name und Komprimierungsart angeben* erläutert.

SYSRPC CSMASS-Report

Das Direktkommando `SYSRPC CSMASS` erzeugt einen Bericht, der den mit dem Kommando angegebenen Bedarf an Sende- und Empfangslänge der Subprogramme (Objekte) anzeigt, wie im folgenden Beispiel gezeigt:

```
Page       1                                              2018-01-17 15:54:12

    SYSRPC - Calculation of Buffer Sizes for RPC Without Interface Objects


Calculation Criteria:

    Object name or range: RPC*
              Compression: 1


Calculation Results (sizes in bytes):

 Number of objects found:       8
     Maximum send length:  200228
  Maximum receive length: 1024192


Object     Type      Send Length   Receive Length   Message
--------   ----   --------------   --------------   --------------
RPCCALL1   N                 209              202
RPCCALL2   N                 219              240   Compression=2
RPCCALL3   N                 204              193
MORE
```

Der Report ist in drei Abschnitte gegliedert, die die folgenden Informationen enthalten:

- **Calculation Criteria - Berechnungskriterien:**
  Die Kriterien, auf deren Grundlage die Berechnung durchgeführt wurde: ein einzelner Objektname oder ein Bereich von Namen (hier: `RPC*`) und die Komprimierung (hier: `1`).
- **Calculation Results (sizes in bytes) - Berechnungsergebnisse (Größen in Bytes):**
  Die Anzahl der Objekte, die für die Größenberechnung ausgewählt wurden.

  Die maximalen Puffergrößen, die alle ausgewählten Objekte zum Senden und Empfangen von Daten vom Client benötigen.
- **Object List - Objektliste:**
  Der Name und der Typ (hier: `N` für den Typ Subprogramm) jedes für die Berechnung ausgewählten Objekts. Die Puffergrößen, die jedes Objekt zum Senden (**Send Length** - Sendelänge) und Empfangen (**Receive Length** - Empfangslänge) von Daten vom Client benötigt. Ein möglicher Kommentar zu jeder Objektberechnung in der Spalte **Message** - Nachricht. Im obigen Beispiel zeigt `Compression=2` an, dass das Objekt `RPCCALL2` nicht, wie im Kommando gefordert, mit Compression 1 berechnet wurde. Die Objektliste ist in alphabetischer Reihenfolge der Objektnamen sortiert.

Wenn die Eingabeaufforderung `MORE` erscheint, können Sie ENTER drücken, um zum Ende des Berichts zu blättern.

Wenn die Größenberechnung für einzelne oder mehrere Objekte fehlschlägt, zeigt der Bericht die Anzahl der betroffenen Objekte und entsprechende Fehlermeldungen an.

Namensangabe für Größenberechnung und Komprimierung

Sie können die Objekte (Subprogramme oder PDAs) angeben, die für die Größenberechnung ausgewählt werden sollen, sowie die Art der Komprimierung, die verwendet werden soll.

- Name
- Compression

Name

Sie können einen Objektnamen oder einen Bereich von Namen angeben. Falls Sie keinen Namen oder Namensbereich angeben, wird die Größe aller in der aktuellen Library enthaltenen Subprogramme oder PDAs berechnet.

Gültige Namensangaben sind im Folgenden beschrieben, wobei *value* eine beliebige Kombination aus einem oder mehreren alphanumerischen Zeichen ist:

Eingabe	Ausgewählte Objekte
*	Alle Subprogramme oder PDAs. Dies ist die Standardeinstellung.
value	Ein Subprogramm oder ein PDA mit einem Namen, der gleich *value* ist.
*value**	Alle Subprogramme oder PDAs mit Namen, die mit *value* beginnen.
value<	Alle Subprogramme oder PDAs, deren Name kleiner oder gleich *value* ist.
value>	Alle Subprogramme oder PDAs, deren Name größer oder gleich *value* ist.

Compression

Sie können eine der folgenden Komprimierungsarten angeben: `0`, `1`, `2`. Die Angabe der Komprimierung ist optional. Der Standardtyp für die Generierung von Interface-Objekten ist `1`.

Siehe auch *Komprimierung verwenden* in *Betrieb einer Natural RPC-Umgebung* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

131 Parameter verwalten (Parameter Maintenance)

- Funktion Parameter Maintenance aufrufen 1172
- NTRPC/RPC-Schlüsselwort-Subparameter angeben 1172

Gilt nur für Client-Sitzungen.

Die Funktion **Parameter Maintenance** wird verwendet, um dynamisch (innerhalb einer Sitzung) Schlüsselwort-Subparameter des RPC-Profilparameters oder des NTRPC-Makros zu ändern.

Vorsicht: Die Parameteränderungen bleiben nur so lange erhalten, wie die Benutzersitzung aktiv ist. Sie gehen verloren, wenn die Sitzung beendet wird.

Funktion Parameter Maintenance aufrufen

Um die Funktion Parameter Maintenance aufzurufen:

1 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
PM
```

Es erscheint der Bildschirm **Client Parameter Maintenance**.

2 Ändern Sie die Werte der Eingabefelder. Siehe *NTRPC/RPC-Schlüsselwort-Subparameter angeben*.

3 Drücken Sie PF3 (Exit), um die Änderungen zu speichern und den Bildschirm **Client Parameter Maintenance** zu verlassen.

Oder:

Drücken Sie PF12 (Canc), um den Bildschirm zu verlassen, ohne die Parameteränderungen zu speichern.

Das Menü **Client Maintenance** wird angezeigt.

NTRPC/RPC-Schlüsselwort-Subparameter angeben

In den Eingabefeldern des Bildschirms **Client Parameter Maintenance** können Sie, wie in der folgenden Tabelle beschrieben, die Einstellungen der Schlüsselwort-Subparameter des NTRPC-Makros oder des RPC-Profilparameters ändern:

Feld	Erläuterung
Timeout	**Zeitüberschreitung** Gibt die Anzahl der Sekunden an, die der Client auf eine Antwort des RPC-Servers warten soll. Siehe auch den `NTRPC/RPC`-Schlüsselwort-Subparameter `TIMEOUT` in der *Parameter-Referenz*-Dokumentation.
Try alternative servers	**Alternative Server ausprobieren** Gibt an, ob ein RPC-Client versuchen soll, einen Dienst auf einem alternativen Server auszuführen (`ON`) oder nicht (`OFF`). Siehe auch *Alternativen Server benutzen* in der *Natural Remote Procedure Call (RPC)*-Dokumentation. Siehe auch den `NTRPC/RPC`-Schlüsselwort-Subparameter `TRYALT` in der *Parameter-Referenz*-Dokumentation.
Compression for auto remote RPC	**Komprimierung für Auto-Remote-RPC** Gibt die Komprimierungsart für einen automatisch generierten RPC-Aufruf an. Siehe *Komprimierung verwenden* in der *Natural Remote Procedure Call (RPC)*-Dokumentation. Siehe auch den `NTRPC/RPC`-Schlüsselwort-Subparameter `COMPR` in der *Parameter-Referenz*-Dokumentation. Weitere Informationen zur automatischen RPC-Ausführung finden Sie unter *Mit automatischer Natural RPC-Ausführung arbeiten* in der *Natural RPC (Remote Procedure Call)*-Dokumentation.

Weitere Informationen zu Parametereinstellungen siehe *Schlüsselwort-Subparameter für Client und Server | nur für Client | nur für Server* im Kapitel *RPC - Remote-Procedure-Call Settings* in der *Parameter-Referenz*-Dokumentation.

132 Server-Kommandos ausführen (Server Command Execution)

- Funktion Server Command Execution benutzen 1176
- RPC Server anpingen 1179
- RPC-Server beenden 1181
- Kommandos für RPC-Server-Tasks ausführen 1183

Die SYSRPC Utility bietet die Server-Ausführungskommandos `ping` und `terminate`. Sie werden verwendet, um aktive Server zu kontrollieren, die im Service Directory definiert wurden:

- Das Kommando `ping` sendet eine interne Nachricht an den Server, um eine Serververbindung zu verifizieren.
- Das Kommando `terminate` sendet entweder eine interne Nachricht an den Server, um die Beendigung einer einzelnen Server-Task anzufordern, oder gibt ein Kommando an den EntireX Broker aus, das die Beendigung aller mit einem EntireX Broker-Dienst verbundenen Server-Tasks anfordert.

Die Server-Ausführungskommandos verweisen auf das Service Directory in der Library, das mit dem `NTRPC/RPC`-Schlüsselwort-Subparameter `RPCSDIR` definiert ist (siehe *Parameter-Referenz*-Dokumentation). Wenn `RPCSDIR` nicht gesetzt ist (dies ist die Standardeinstellung), wird die Library verwendet, in der Sie gerade angemeldet sind. Der Name der Library wird in der oberen rechten Ecke des Bildschirms **Server Command Execution** angezeigt, wie im folgenden Abschnitt dargestellt.

Funktion Server Command Execution benutzen

Um die Funktion Server Command Execution zu benutzen:

1 Geben Sie im Feld **Code** im Menü **Client Maintenance** das folgende Kommando ein:

```
XC
```

2 Drücken Sie ENTER.

Die Standardansicht des Bildschirms **Server Command Execution** erscheint (Beispiel):

```
 15:16:30           ***** NATURAL SYSRPC UTILITY *****             2016-07-18
                          - Server
             Command Execution -         Library SAGTRPC2

    Cmd Node                           Server                        Message
 1      ETB045
 2   __                                NRPC001                       Natural
 3   __                                NRPC002




Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  ERR   Exit        -H    +H    -P    +P    TOP   BOT    >   Canc
```

Die Standardansicht des Bildschirms zeigt die Felder **Node**, **Server** und **Message**. Die Felder in der Spalte **Message** werden abgeschnitten angezeigt und enthalten maximal 8 Zeichen.

3 Wenn Sie PF11 drücken oder das Größer-als-Zeichen (>) in die Kommandozeile unten im Bildschirm eingeben, wird die erweiterte Ansicht des Bildschirms **Server Command Execution** angezeigt (Beispiel):

```
 16:36:39           ***** NATURAL SYSRPC UTILITY *****             2016-07-18
                          - Server
             Command Execution -         Library SAGTRPC2

    Cmd  Server            Message
 1
 2   __  NRPC001           Natural RPC Server 8.3.7 on WNT-x86
 3   __  NRPC002




Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  ERR   Exit        -H    +H    -P    +P    TOP   BOT    <    Canc
```

In der erweiterten Ansicht können Sie maximal 50 Zeichen Nachrichtentext in der Spalte **Message** anzeigen. In dieser Ansicht wird die Spalte **Node** nicht angezeigt und die Felder in der Spalte **Server** werden abgeschnitten und zeigen maximal 16 Zeichen an (in der Standardansicht werden 30 Zeichen angezeigt).

Wenn Sie ein weiteres Mal PF11 wählen oder das Kleiner-als-Zeichen (<) in die Kommandozeile eingeben, wird wieder die Standardansicht des Bildschirms **Server Command Execution** angezeigt, wie im *Beispiel für eine Standardansicht* dargestellt.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Zeilenkommandos: Server Command Execution

Zeilenkommandos: Server Command Execution

Welche Zeilenkommandos auf dem Bildschirm **Server Command Execution** verfügbar sind, hängt davon ab, ob sie auf einem EntireX Broker-Knoten oder einem RPC-Server-Knoten ausgeführt werden. In der folgenden Tabelle zeigt ein X an, ob ein Kommando für einen Knoten verfügbar ist.

Zeilenkommando	Beschreibung	Broker	Server
PI	**Broker**-Knoten: Pingt alle für den ausgewählten EntireX-Broker definierten Server an. **Server:** Pingt den ausgewählten RPC-Server an. Siehe auch das Direktkommando SYSRPC PING.	X	X
TE	Beendet alle Server-Tasks des ausgewählten RPC-Servers (EntireX Broker Service). Siehe auch *RPC-Server beenden*.		X
TI	Trace-Informationen: Ruft den aktuellen Trace-Level-Wert ab. **Broker**-Knoten: Zeigt die Trace-Informationen zu allen für den ausgewählten Broker definierten RPC-Server. **Server:** Zeigt die Trace-Informationen zum ausgewählten RPC-Server.	X	X
T*n*	Setzt den Trace-Level für den ausgewählten RPC-Server dynamisch. *n* steht für 0 bis 9. T0 bedeutet, dass die Trace-Funktion abgeschaltet ist.		X
T*	Ruft ein Fenster auf, in dem Sie den Trace-Level-Wert *n* eingeben und die Option Write-on-Error (E) für den ausgewählten RPC-Server angeben können.		X
LN	Zeigt alle Server-Tasks mit ihren Folgenummern für einen ausgewählten RPC-Server in einer Liste an. In dieser Liste können Sie ein Kommando für einen einzelnen Server-Task oder ein Kommando für alle Server-Tasks ausführen. Siehe auch *Kommandos für RPC-Server-Tasks ausführen*.		X
TS	Beendet den ausgewählten EntireX Broker Service.		X

Zeilenkommando	Beschreibung	Broker	Server
LS	Listet die auf dem ausgewählten EntireX Broker registrierten Server auf. Siehe auch Direktkommando SRVLIST.	X	X
IV	Listet die Versionen des ausgewählten EntireX Brokers und seiner Command and Information Services (CIS) sowie die Version des EntireX Broker Stub auf.	X	X

RPC Server anpingen

Sie können einen RPC-Server über die Standardansicht oder die erweiterte Nachrichtenansicht des Bildschirms **Server Command Execution** oder mit dem Direktkommando SYSRPC PING anpingen.

Informationen zum Anpingen eines RPC-Servers über die Anwendungsprogrammierschnittstelle USR2073N finden Sie in der entsprechenden *Natural RPC (Remote Procedure Call)*-Dokumentation.

Der folgende Abschnitt enthält eine Anleitung für das Anpingen eines RPC-Servers über die Standardansicht des Bildschirms **Server Command Execution screen**.

Um einen RPC-Server über den Bildschirm Server Command Execution anzupingen:

1 Geben Sie im Bildschirm **Server Command Execution** in der Spalte **Cmd** neben dem/den Server(n), der/die angepingt werden soll(en), das folgende Zeilenkommando ein:

```
PI
```

Beispiel:

```
 16:41:32           ***** NATURAL SYSRPC UTILITY *****              2016-07-18
                          - Server
             Command Execution -         Library SAGTRPC2

    Cmd Node                           Server                          Message
 1   __ ETB045
 2   PI                                NRPC001
 3   PI                                NRPC002




Command ===>


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  ERR   Exit        -H    +H    -P    +P    TOP   BOT    >    Canc
```

2 Drücken Sie ENTER. Der Server (bzw. die Server) gibt die Meldung zurück:

```
Server version on operating system
```

Dabei bezeichnet *Server* den Servertyp und *version* die Version des Betriebssystems *operating system*, auf dem der Server läuft.

Beispiel-Nachricht:

```
Natural RPC Server 8.3.7.0 on WNT-x86
```

Wenn das Anpingen des Servers fehlschlägt und stattdessen ein Fehler auftritt, können Sie PF2 (ERR) wählen, um RPC-bezogene Natural- und EntireX-Broker-Meldungen anzuzeigen, wie unter *Programm RPCERR verwenden* (*Den Status einer RPC-Sitzung überwachen* in der *Natural RPC (Remote Procedure Call)*-Dokumentation beschrieben.

3 Die vollständigen Meldungstexte werden automatisch in der erweiterten Nachrichtenansicht des Bildschirms **Server Command Execution** angezeigt (und die Belegung von PF11 ändert sich in das Kleiner-als-Zeichen (<).

- Direktkommando SYSRPC PING verwenden

Direktkommando SYSRPC PING verwenden

Sie können einen RPC-Server im Online- und Batch-Modus mit dem Direktkommando `SYSRPC PING` anpingen.

Es gilt die folgende Kommandosyntax:

```
SYSRPC PING { server-name ON broker-name [[PORT] port-number][TRANSPORT
            {TCP|SSL|NET}]
            ALL                                                        }
```

Die im Syntax-Diagramm verwendeten Symbole werden im Abschnitt *Syntax-Symbole* in der *Statements*-Dokumentation erläutert.

Die Syntax-Elemente werden in der folgenden Tabelle erläutert:

Syntax-Element	Format/Länge	Beschreibung
server-name	A32	Name eines RPC-Servers oder ein Bereich von Namen. Ein Stern (*) wählt alle Namen aus. Stern-Notation wählt alle Namen aus, die mit dem angegebenen Wert beginnen.
broker-name	A32	Name des EntireX-Brokers oder ein Bereich von Namen Ein Stern (*) wählt alle Namen aus. Stern-Notation wählt alle Namen aus, die mit dem angegebenen Wert beginnen.

Syntax-Element	Format/Länge	Beschreibung	
port-number	N5	Port-Nummer der für die Serververbindung verwendeten Netzwerkadresse. Gültige Werte: `0` bis `65535`	
`TRANSPORT`	A3	Vom EntireX Broker verwendete Transportmethode::	
		`TCP`	TCP/IP-Protokoll
		`SSL`	SSL oder TLS
		`NET`	Entire Net-Work (nicht unterstützt unter Linux oder Windows)
`ALL`	-	Pingt alle RPC-Server an, die im Service Directory der aktuellen Library definiert sind.	

RPC-Server beenden

In der Standardansicht oder erweiterten Nachrichtenansicht des Bildschirms **Server Command Execution** stehen die folgenden Kommandos zur Verfügung, um einen RPC-Server oder EntireX Broker Service zu beenden:

`TE`	Beendet eine einzelne RPC-Server-Task, indem eine interne Nachricht an den RPC-Server gesendet wird. Wenn ein RPC-Server mit mehreren RPC-Server-Tasks (einschließlich Replikaten auf z/OS-Plattformen) verknüpft ist, können Sie entweder jede RPC-Server-Task einzeln mit `TE` beenden oder alle RPC-Server-Tasks in einem Zug mit dem Kommando `TS` beenden.
`TS`	Beendet alle Server-Tasks, die mit einem EntireX Broker Service verbunden sind, indem es die EntireX Broker Command and Information Services (ETBCIS; Details finden Sie in der *EntireX*-Dokumentation) aufruft. Der Begriff Service fasst hier alle Server-Tasks zusammen, die unter demselben Servernamen auf derselben oder auf verschiedenen Plattformen laufen.

Der folgende Abschnitt enthält Anleitungen zum Beenden einer einzelnen RPC-Server-Task oder eines EntireX Broker Service über die Standardansicht.

Alternative Methoden zur Beendigung von Servern finden Sie unter *Beenden eines Natural RPC Servers* in der *Natural RPC (Remote Procedure Call)* -Dokumentation.

Um eine einzelne RPC-Server-Task zu beenden:

1 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
XC
```

Die Standardansicht des Bildschirms **Server Command Execution** wird angezeigt.

2 Geben Sie in der Spalte **Cmd** neben dem/den zu beendenden Server(n) das folgende Zeilenkommando ein:

```
TE
```

(Dies ist vergleichbar mit der Eingabe des Kommandos `PI`, wie im **Beispiel** zum Anpingen eines Servers gezeigt).

3 Drücken Sie ENTER.

Der Server gibt die Meldung aus:

```
Terminating Server version on operating system
```

Server bezeichnet den Servertyp; *version* bezeichnet die vier- oder fünfstellige Produktnummer; *operating system* bezeichnet das Betriebssystem, das auf dem Server läuft.

Beispielmeldung:

```
Terminating Natural RPC Server 6.3.1.0 on WNT-x86
```

Wenn die Beendigung des Servers fehlschlägt und stattdessen ein Fehler auftritt, können Sie PF2 (ERR) wählen, um RPC-bezogene Natural- und EntireX-Broker-Meldungen anzuzeigen, siehe *Programm RPCERR verwenden* (*Den Status einer RPC-Sitzung überwachen*, *Natural RPC (Remote Procedure Call)*-Dokumentation).

Der vollständige Meldungsinhalt wird automatisch in der Standardansicht des Bildschirms **Server Command Execution** (und PF11 ist mit dem Kleiner-als- Zeichen (<) belegt).

4 Wenn Sie PF11 erneut drücken oder das Kleiner-als- Zeichen (<) in der Kommandozeile unten im Bildschirm eingeben, wird wieder die Standardansicht des Bildschirms **Server Command Execution** angezeigt (siehe auch ***Beispiel für eine Standardansicht***).

5 Wenn im Service Directory die Option **Logon** gesetzt ist, werden die Anmeldedaten (Benutzerkennung, Passwort und Library-Name) mit dem Kommando `TE` an den Server gesendet, wie es bei der Remote-Ausführung von `CALLNAT` üblich ist.

Das Fenster **Security Token Data** wird angezeigt und fordert zur Eingabe von Benutzerkennung und Kennwort auf, wenn auf der Client-Seite kein Natural Security installiert ist und keine Anmeldedaten mit der Anwendungsprogrammierschnittstelle USR1071N für die aktuelle Natural-Sitzung festgelegt wurden. Siehe auch USR1071N in *Verwendung von Security*, *Natural RPC mit Natural Security verwenden*, in der *Natural RPC (Remote Procedure Call)* -Dokumentation.

Wenn auf der Server-Seite `LOGONRQ=ON` gesetzt wurde (siehe auch *Verwendung von Security* in der *Natural RPC (Remote Procedure Call)*-Dokumentation), müssen die Anmeldedaten vom Client mit dem Kommando `TE` gesendet werden.

Wenn Natural Security auf dem Server installiert ist, müssen die übertragenen Anmeldedaten eine Anmeldung bei der Natural System Library SYSRPC ermöglichen.

Um einen EntireX Broker Service zu beenden:

1 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
XC
```

Die Standardansicht des Bildschirms **Server Command Execution** wird angezeigt.

2 Geben Sie in der leeren Spalte zwischen der laufenden Nummer und der Spalte **Node** in der Zeile, die zu dem zu beendenden Server gehört, das folgende Kommando ein:

```
TS
```

(Dies ist vergleichbar mit der Eingabe des Kommandos `PI`, wie es im **Beispiel** zum Anpingen eines Servers gezeigt wurde).

3 Drücken Sie ENTER.

Es erscheint das Fenster **SYSRPC - Terminating EntireX Broker Service**.

4 Falls für die Anmeldung erforderlich, geben Sie die entsprechende Benutzerkennung und das Passwort für den EntireX Broker ein.

Wenn Sie Server-Tasks, die an einer Konversation beteiligt sind, beenden wollen, geben Sie im Feld **Terminate immediately** ein `Y` ein, um die sofortige Beendigung anzufordern. Wenn Sie ein `N` eingeben (dies ist die Standardeinstellung), bleiben alle an einer Konversation beteiligten Server-Tasks in Betrieb.

Wenn Sie nicht möchten, dass dieses Fenster während der aktuellen SYSRPC-Sitzung wiederholt angezeigt wird, wählen Sie **Do not show this window again** (Dieses Fenster nicht mehr anzeigen).

5 Drücken Sie ENTER, um den EntireX Broker Service zu beenden.

Kommandos für RPC-Server-Tasks ausführen

Verschaffen Sie sich zunächst einen Überblick über alle verfügbaren RPC-Server-Tasks für einen RPC-Server auf der gleichen oder auf verschiedenen Plattformen, die auf einem EntireX Broker registriert sind.

Um eine Liste aller RPC-Server-Tasks mit Folgenummer zu erhalten:

1 Geben Sie im Feld **Code** des Menüs **Client Maintenance** das folgende Kommando ein:

```
XC
```

Drücken Sie ENTER.

Die Standardansicht des Bildschirms **Server Command Execution** wird angezeigt.

2 Geben Sie in der Spalte **Cmd** neben dem/den zu beendenden Server(n) das folgende Zeilenkommando ein:

```
LN
```

Drücken Sie ENTER.

Der Bildschirm **Server Command Execution** auf *Server-Task-Ebene* wird mit der Liste aller Server-Tasks für einen RPC-Server angezeigt.

Die folgenden Informationen werden angezeigt:

Spalte	Beschreibung
SequenceNo	Die der Server-Task zugewiesene Nummer in der Reihenfolge, in der die Task beim EntireX Broker registriert wurde.
Date	Das Datum und die Uhrzeit (UTC), zu der die Server-Task gestartet wurde.
Time	
Host Name	Der Knoten, auf dem der Server gehostet wird.
Status	Der Status des Servers. Mögliche Werte: `idle` (untätig), `busy` (beschäftigt), `term` (beendet) oder `active` (bei Ping- und Trace-Kommandos).

Der Anwendungsname und die IP-Adresse (falls vorhanden) sind in der Zeile unter jeder nummerierten Zeile angegeben.

Der Anwendungsname hängt von der Umgebung ab, in der der Server läuft. Er kann z. B. der Name eines Natural-Images (auf Linux), einer LPAR (auf z/OS) oder einer gestarteten CICS-Task sein.

Die IP-Adresse zeigt die IPv4- oder IPv6-Adresse des Knotens an.

Mit den PF-Tasten können Sie auf dem Bildschirm navigieren und zur *Liste* aller Server im Bildschirm **Server Command Execution**(PF3) oder zum Menü **Client Maintenance** (PF12) zurückkehren.

3 Wählen Sie eine der folgenden Möglichkeiten:

- **Für eine einzelne Server-Task**
  Geben Sie in der Spalte **Cmd** neben der Folgenummer der Server-Task eines der folgenden Kommandos ein.

- **Für alle Server-Tasks**
  Drücken Sie PF4 (Cmds) und geben Sie in dem Eingabefenster eines der folgenden Kommandos ein.

  Informationen zum Ausführen eines Kommandos für eine RPC-Server-Task mithilfe der Anwendungsprogrammierschnittstelle USR9204N finden Sie in der entsprechenden Dokumentation zum *Natural RPC (Remote Procedure Call)*.

Kommando	Beschreibung
`PI`	Pingt einen einzelnen Server-Tasks bzw. alle Server-Tasks eines RPC-Server an.
`TE`	Beendet einen einzelnen Server-Tasks bzw. alle Server-Tasks eines RPC-Server an.
`TI`	Trace-Informationen: Ruft den aktuellen Trace-Level-Wert für einen einzelnen Server-Task bzw. alle Server-Tasks eines RPC-Server ab.
`T`*n*	Setzt den Trace-Level für einen einzelnen Server-Task bzw. alle Server-Tasks eines RPC-Server dynamisch. *n* steht für `0` bis `9`. `T0` bedeutet, dass die Trace-Funktion abgeschaltet ist.
`T*`	Ruft ein Fenster auf, in dem Sie den Trace-Level-Wert *n* eingeben und die Option **Write-on-Error** (`E`) für einen einzelnen Server-Task eines RPC-Server angeben können. Die Trace-Level-Werte `T1E` bis `T9E` können auch für alle Server-Tasks eines RPC-Servers nach Drücken der Taste `PF4` (Cmds) eingestellt werden.



Wichtig: Die Kommandos `PI`, `TI`, `T0`, `T1` bis `T9` und `T1E` bis `T9E` werden abgelehnt, wenn die von der aktuellen Natural-Sitzung verwendete CIS-Version nicht hoch genug ist, um das angeforderte EntireX-Kommando zu unterstützen (erforderlich: EntireX Broker 11.0 mit CIS-Version 13 und höher). Sie können die CIS-Version des EntireX Brokers mit dem Kommando `IV` im Menü **Client Maintenance** überprüfen.

Drücken Sie ENTER.

Der RPC-Server gibt die Meldung/Fehlermeldung und den geänderten Status für die angeforderte(n) Server-Task(s) zurück.

In der ersten Informationszeile der Server-Task steht dann die Meldung `Successful reponse` (`Erfolgreiche Antwort` nach Beendigung), die Ping-Informationen oder der Trace-Level.

In der zweiten Informationszeile der Server-Task wird die Fehlermeldung *mit Fehlernummer und Fehlertext* angezeigt, z. B. `Participant not found` (`Teilnehmer nicht gefunden`, wenn die Server-Task bereits beendet ist), `Function disallowed by Natural Security` (Funktion durch Natural Security nicht erlaubt) oder `Command SEND-COMMAND not supported` (Kommando `SEND-COMMAND` nicht unterstützt, wenn die CIS-Version das Kommando noch nicht unterstützt, siehe wichtiger Hinweis oben).

133 Beim EntireX Broker registrierte Server auflisten

- Beispiel für ein SYSRPC SRVLIST-Direktkommando 1189
- Server-Liste anzeigen 1189
- Zusätzliche Server-Informationen anzeigen 1190
- Server-Listen anpassen 1191

Sie können Informationen über RPC-Server, die beim EntireX Broker registriert sind, mit dem Direktkommando `SYSRPC SRVLIST` anfordern.

Das Direktkommando `SYSRPC SRVLIST` sendet einen Aufruf an den EntireX Broker und fordert Informationen über RPC-Server an, die beim EntireX Broker mit den Attributen `SERVER-CLASS=RPC` und `SERVICE=CALLNAT` registriert sind.

Sie können `SYSRPC SRVLIST` online (von einer Natural-Kommandoeingabe aus) oder im Batch-Modus ausführen.



Anmerkung: Wenn Sie dieses Kommando online ausführen, werden Sie in einem Fenster aufgefordert, sich bei dem im Kommando angegebenen EntireX Broker anzumelden.

Die folgende Kommandosyntax gilt für `SYSRPC SRVLIST`:

```
SYSRPC SRVLIST server-name ON broker-name [[PORT] port-number][TRANSPORT
{TCP|SSL|NET}][USING{HEAD1MAP|object-name}]
```

Die im Syntax-Diagramm verwendeten Symbole werden im Abschnitt *Syntax-Symbole* in der *Statements*-Dokumentation erläutert.

Die Syntax-Elemente werden in der folgenden Tabelle erläutert:

Syntax-Element	Format/Länge	Beschreibung	
server-name	A32	Name eines RPC-Servers oder ein Bereich von Namen. Ein Stern (*) wählt alle Namen aus. Stern-Notation wählt alle Namen aus, die mit dem angegebenen Wert beginnen.	
broker-name	A32	Name des EntireX-Brokers oder ein Bereich von Namen Ein Stern (*) wählt alle Namen aus. Stern-Notation wählt alle Namen aus, die mit dem angegebenen Wert beginnen.	
port-number	N5	Port-Nummer der für die Serververbindung verwendeten Netzwerkadresse. Gültige Werte: `0` bis `65535`	
`TRANSPORT`	A3	Vom EntireX Broker verwendete Transportmethode::	
		`TCP`	TCP/IP-Protokoll
		`SSL`	SSL oder TLS
		`NET`	Entire Net-Work (nicht unterstützt unter Linux oder Windows)
object-name	A8	Name des Natural-Text-Objekts, das zum Anpassen eines Server-Reports verwendet wird. Siehe auch ***Server-Listen anpassen***.	

In diesem Kapitel werden die folgenden Themen behandelt:

Beispiel für ein SYSRPC SRVLIST-Direktkommando

```
SYSRPC SRVLIST SERV* ON BRK123
```

Dieses Kommando gibt Daten für alle Server zurück, deren Namen mit `SERV` auf EntireX Broker `BRK123` beginnen.

Server-Liste anzeigen

Wenn Sie das Direktkommando `SYSRPC SRVLIST` online ausführen, wird der Bildschirm **Servers registered on ...** mit einer Server-Liste angezeigt (Beispiel):

```
 13:03:53             ***** NATURAL SYSRPC SRVLIST *****             2016-07-14
                       - Servers registered on BRK123 -
 Cmd  Server           TransRoutine   Requests ConvTimeouts ServersActive Conv>
 ---  ---------------- ------------ ---------- ------------ ------------- ---->
  _   SERVRPC1                             0           60             1
  _   SERVRPC2         SAGTCHA             0           60             1
  _   SERVRPC3                             0           60             1
  _   SERVRPC4         SAGTCHA            76           60             1
  _   SERVRPC5                          2035           60             1
  _   QA42RPC6         RPCTRNS            25          600             2
  _   QA42RPC7                         11190           60             1
  _   QA42RPC8                           206           60             1


Command ===>


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

Die Spalten und Spaltenüberschriften auf dem Bildschirm **Servers** werden im Textobjekt `HEAD1MAP` beschrieben. Siehe auch *Server-Listen anpassen*.

Sie können durch die Liste navigieren und zusätzliche Server-Informationen anzeigen, indem Sie die in der folgenden Tabelle beschriebenen Kommandos verwenden.

Die Spalte ganz links, die den Namen des RPC-Servers enthält, bleibt immer an ihrer Position, wenn Sie in der Liste nach rechts oder links blättern.

Kommando	Beschreibung
PF3	Beendet das Kommando.
PF6	Blättert in den Daten zur Spalte ganz links.
PF7	Blättert in den Daten nach links.
PF8 oder PF11	Blättert in den Daten nach rechts.
PF9	Blättert in den Daten zur Spalte ganz rechts.
I	Zeilenkommando, das in der Spalte **Cmd** für einen aufgelisteten Server eingegeben wird. Zeigt zusätzliche Informationen über einen einzelnen RPC-Server an: Siehe auch *Zusätzliche Server-Informationen anzeigen*.

Zusätzliche Server-Informationen anzeigen

Sie können zusätzliche Informationen zu einem bestimmten RPC-Server anzeigen.

Um zusätzliche Informationen für einen einzelnen Server anzuzeigen:

- Geben Sie in der Spalte **Cmd** des Bildschirms **Servers registered on ...** das Zeilenkommando I neben dem Server ein, zu dem Sie zusätzliche Informationen anzeigen möchten.

  Es erscheint der Bildschirm **Information on Server** (Beispiel):

```
15:16:28            ***** NATURAL SYSRPC SRVLIST *****            2016-07-14
               - Information on Server SERVRPC4 on BRK123 -
Description                                                 Value
------------------------------------------------------------ --------------------
Character set used on platform.........................: EBCDIC
Endian type of platform................................: Big endian
Status of user.........................................: Waiting
Kind of conversation for which user waits..............: NEW
Server for which user waits (Class=RPC/Service=CALLNAT)..: SERVRPC4
Number of active conversations of this user............: 0
Number of services active (offered) by this server.....: 1
Elapsed time since the last activity of the user.......: 95
Non-activity timeout in seconds........................: 600
Accumulated time server waited for new conversations.....: 68856
Number of times server waited for new conversations......: 190
Accumulated time server or client waited for messages of>: 0
Number of times server or client waited for messages of >: 0
Sum of conversations for the user since start of session.: 76
Number of UOWs (units of work).........................: 0
IPv4 address of server.................................: 10.20.91.119

Command ===>
```

Der Bildschirm zeigt alle Informationen an, die EntireX Broker für den angeforderten Server zurückgegeben hat. Siehe auch ***Server-Listen anpassen***.

Server-Listen anpassen

Sie können eine Liste von Servern oder eine Liste mit Server-Informationen nach Bedarf neu anordnen, indem Sie das Textobjekt Natural `HEAD1MAP` bzw. `HEAD2MAP` verwenden. `HEAD1MAP` und `HEAD2MAP` werden in der SYSRPC System Library ausgeliefert.

Wir empfehlen Ihnen, `HEAD1MAP` (Liste der Server) aus der Library SYSRPC in eine benutzerdefinierte Library zu kopieren, bevor Sie mit der Bearbeitung der Liste beginnen. Sie können das Objekt dann umbenennen und im Kommando `SRVLIST` darauf verweisen.

Die Liste `HEAD2MAP` (Serverinformationen) können Sie nicht umbenennen.

Die zu verwendenden Textobjekte müssen in der aktuellen Library, in der mit dem `NTRPC/RPC`-Schlüsselwort-Subparameter `RPCSDIR` angegebenen Library (siehe *Parameter-Referenz*-Dokumentation) oder in der SYSRPC-Systembibliothek enthalten sein, wenn der Objektname `HEAD1MAP` verwendet wird.

`HEAD1MAP` und `HEAD2MAP` enthalten Anweisungen, wie Sie eine Liste nach Ihren Bedürfnissen ändern können. Sie können die Quellcodezeilen für Spalten und Überschriften, die in Ihrem Report nicht benötigt werden, auskommentieren. Sie können die Position der Codezeilen ändern, um die Spalten neu anzuordnen.

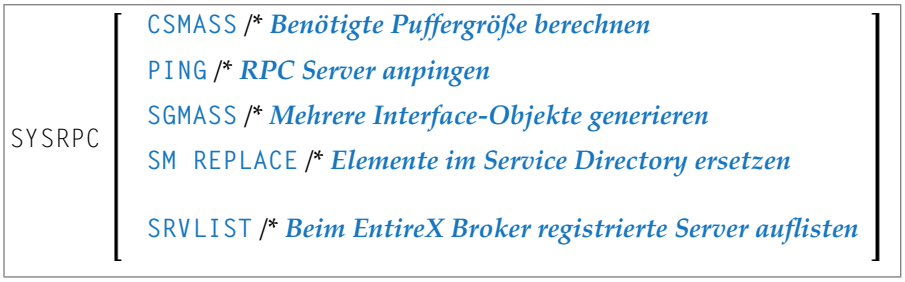
Ausnahmen:

- Bei `HEAD1MAP`: Sie dürfen die erste Quellcodezeile, die das Feld `SERVER-NAME` enthält, nicht auskommentieren oder verschieben. Sie dürfen den Namen eines Feldes in der Spalte **Field** nicht ändern.
- Bei `HEAD2MAP`: Sie dürfen den Namen eines Feldes in der Spalte **Field** nicht ändern.

134 Übersicht über die SYSRPC-Direktkommandos und -Batch-Kommandos

Das folgende Syntax-Diagramm gibt einen Überblick über die SYSRPC-Direktkommandos, die online und im Batch-Modus verfügbar sind.

Der Kommentar neben jedem Kommando erläutert den Verwendungszweck des jeweiligen Kommandos und referenziert die zugehörige Beschreibung.



XXIV SYSTP Utility

Mit dem Dienstprogramm SYSTP können Sie TP-Monitor-spezifische Natural-Eigenschaften überwachen.

Das SYSTP-Dienstprogramm bietet Funktionen, die in den meisten Umgebungen und unter den meisten TP-Monitoren verfügbar sind. Sie sind im Kapitel *Allgemeine SYSTP-Funktionen* beschrieben.

Das Dienstprogramm SYSTP bietet zusätzliche Funktionen für die TP-Monitore CICS und IMS TM.

Diese umgebungsabhängigen Funktionen sind in den entsprechenden Kapiteln dieses Dokuments beschrieben.

SYSTP aufrufen und Funktionen ausführen
SYSTP-Bildschirme benutzen
Allgemeine SYSTP-Funktionen
SYSTP-Funktionen unter CICS
SYSTP-Funktionen unter IMS TM
SYSTP in Batch für CICS-Sitzungen

135 SYSTP aufrufen und Funktionen ausführen

Dieses Kapitel enthält eine Anleitung zum Aufrufen des SYSTP-Dienstprogramms und zum Ausführen einer SYSTP-Dienstprogrammfunktion. Sie können eine Dienstprogrammfunktion entweder über ein SYSTP-Menü oder über ein SYSTP-Direktkommando ausführen.

Um SYSTP aufzurufen und eine Menüfunktion auszuführen:

1 Geben Sie an einer beliebigen Eingabeaufforderung das folgende Systemkommando ein:

```
SYSTP
```

Das Hauptmenü (**Main Menu**) des Dienstprogramms SYSTP wird angezeigt.

2 Führen Sie eine SYSTP-Funktion aus, indem Sie das Zeichen, das der gewünschten Funktion entspricht, in das Feld **Code** eingeben und ENTER drücken.

Wenn Sie `E` eingeben, erscheint das Menü für umgebungsabhängige TP-Monitorfunktionen. Diese Funktion ist unter Com-plete und TSO nicht verfügbar.

Um ein SYSTP-Direktkommando abzusetzen:

- Benutzen Sie an einer beliebigen Eingabeaufforderung oder im Batch-Modus das Kommando `SYSTP`, gefolgt von dem Funktionscode, der der gewünschten SYSTP-Menüfunktion entspricht. Sie können mehrere Funktionscodes in der Reihenfolge eingeben, in der sie aus einem Menü ausgeführt werden. Wenn Sie mehrere Funktionscodes eingeben, trennen Sie diese durch Punkte (.).

  Zum Beispiel:

  ```
  SYSTP  M.A
  ```

  Ruft das **Natural Monitor Menu** auf (Funktionscode `M` im SYSTP-**Main Menu**) und aktiviert den Natural Monitor (Funktionscode `A` im **Natural Monitor Menu**).

Siehe auch ***SYSTP in Batch für CICS-Sitzungen***.

136 SYSTP-Bildschirme benutzen

Die in einem SYSTP-Menü angebotenen Funktionen können folgendermaßen aufgerufen werden: Entweder durch Eingabe eines Funktionscodes oder durch Drücken der PF-Taste (falls vorhanden), die der gewünschten Funktion entspricht.

Die Zeilenkommandos und PF-Tasten (oder entsprechende Direktkommandos), die auf vielen der SYSTP-Bildschirme verfügbar sind, werden in der folgenden Tabelle beschrieben.

Ein Zeilenkommando geben Sie in der Spalte **C** neben dem gewünschten Listeneintrag auf dem Bildschirm ein. Ein Direktkommando geben Sie in der Kommandozeile ein.

PF-Taste oder Direktkommando	Zeilenkommando	Funktion
? oder *	? oder *	Bei Eingabe in der Spalte **C** werden alle auf dem aktuellen SYSTP-Bildschirm verfügbaren Zeilenkommandos angezeigt. Bei Eingabe in der Kommandozeile werden alle Direktkommandos angezeigt, die innerhalb des Dienstprogramms SYSTP verfügbar sind.
PF1 oder HELP	-	Anzeige von Hilfeinformationen zu der aktuell verwendeten SYSTP-Funktion.
PF3	.	Verlassen des aktuellen Bildschirms und Rückkehr zur vorherigen Ebene/Bildschirm.
PF5	/ oder P	Platzierung der Zeile, in der das Zeilenkommando eingegeben wird (oder in der sich der Cursor befindet), an den oberen Rand des Bildschirms.

PF-Taste oder Direktkommando	Zeilenkommando	Funktion
PF4	S oder U	Anzeige ausführlicher Informationen zu dem Element, das in der Zeile enthalten ist, in der das Zeilenkommando eingegeben wurde (oder in der sich der Cursor befindet).
PF6 oder --	-	An den Anfang einer Liste blättern.
PF7 oder -	-	In einer Liste um eine Seite zurück blättern.
PF8 oder +	-	In einer Liste um eine Seite vorwärts blättern.
PF9 oder ++	-	An das Ende einer Liste blättern.

Die Kopfzeilen aller SYSTP-Statistikbildschirme enthalten die folgenden Informationen:

- Das Feld **User** mit der Benutzerkennung des aktuellen Benutzers, wie sie durch die Natural-Systemvariable `*USER` zugewiesen wurde (siehe *Systemvariablen*-Dokumentation).
- Das Feld **TID** mit der Terminalkennung, die dem aktuellen Benutzer über die Natural-Systemvariable `*INIT-ID` zugewiesen wurde.

137 Allgemeine SYSTP-Funktionen

- Natural Monitoring (SYSMON) 1202
- Natural Print/Work Files (SYSFILE) 1206
- Buffer Usage Statistics (BUS) 1207
- Permanente Speichersegmente 1209
- Natural-Subsysteme und Roll Server-Informationen anzeigen 1210
- Natural Thread-Nutzungsstatistik 1211
- Natural-Lizenzinformationen 1214

Natural Monitoring (SYSMON)

Mit der Natural-Monitoring-Funktion können Sie Statistiken über die Natural-Programme und -Terminals anzeigen, die während der aktuellen Natural-Sitzung verwendet werden.

Die Aufgabenstellung des Natural-Monitors wird durch den Bereich bestimmt, in dem die statistischen Daten gesammelt werden:

- Wenn ein globaler oder lokaler Monitoring Buffer Pool verwendet wird (wie durch den Profilparameter `BPI` oder das Makro `NTBPI` des Natural-Parametermoduls zugewiesen), werden die Programm- und Terminalstatistiken aller Natural-Sitzungen gesammelt, die diesen Buffer Pool gemeinsam nutzen.
- Wenn ein Monitor-Pufferspeicher innerhalb des Threads verwendet wird (festgelegt durch den Profilparameter `MONSIZE`), werden nur Programmstatistiken und Informationen über das Terminal der aktuellen Natural-Sitzung gesammelt. Für einen Monitor-Pufferspeicher empfehlen wir einen Mindestwert von 64 KB.

Zusätzlich zur Definition des Umfangs für den Natural-Monitor müssen Sie den Profilparameter `RDCSIZE` setzen, der den Natural Data Collector aktiviert.

Vorsicht: Wenn die Monitoring-Funktion aktiv ist, kann sich das auf die gesamte Systemleistung auswirken.

Wenn Sie die Funktion **Natural Monitoring (SYSMON)** aufrufen, wird das **Natural Monitor Menu** angezeigt, das die folgenden Funktionen bietet:

- Monitor aktivieren/deaktivieren
- Informationen zum Monitorstatus
- Programm-/Terminalstatistik anzeigen

Monitor aktivieren/deaktivieren

Mit den Funktionen **Activate Monitor** und **Deactivate Monitor** können Sie die Monitoring-Funktion aktivieren bzw. deaktivieren.

Wenn die Monitoring-Funktion aktiviert ist, beginnt sie mit dem Sammeln statistischer Informationen über die aktuelle(n) Sitzung(en). Sobald die Monitoring-Funktion deaktiviert wird, wird eine statistische Zusammenfassung in die Systemprotokolldatei geschrieben.

Informationen zum Monitorstatus

Die Funktion **Monitor Status Information** liefert statistische Informationen darüber, ob Überwachungsdaten in einem Monitor Buffer Pool und/oder dem mit dem Profilparameter `MONSIZE` zugewiesenen Speicherplatz gesammelt werden, und gibt die Größe und die Adresse des zugeordneten Speicherplatzes an.

Programm-/Terminalstatistik anzeigen

Sie können statistische Informationen zu allen Natural-Programmen anzeigen, die seit dem Start des Monitors ausgeführt wurden, sowie zu den Terminals, die seitdem aktiviert wurden.

Um die Programm- oder Terminalstatistikfunktion auszuführen:

1 Geben Sie im **Natural Monitor Menu** im Feld **Code** ein `P` (**Display Program Statistics**) oder ein `T` (**Display Terminal Statistics**) ein.

Sie können Auswahlkriterien für die zu überwachenden Programme/Terminals und/oder Bibliotheken angeben:

- Geben Sie in das Feld **Name of LTERM or Program** und/oder das Feld **Name of Library** den betreffenden Namen ein oder geben Sie einen Bereich von Namen an, indem Sie einen Stern (*) und/oder einen Platzhalter (?) verwenden.
- Wenn Sie die Felder leer lassen oder nur einen Stern (*) eingeben, werden alle Programme/Terminals und Libraries ausgewählt.

Beispiele für Namensbereiche:

`*CD` wählt ABCD, ACD aus.
`AB*` wählt AB, AB1, ABC, ABEZ aus.
`ABC?` wählt ABCA, ABCZ aus.
`A?C*Z` wählt ABCZ, AXCBBBZ und ANCZ aus.

2 Wenn Sie die Eingabe der Auswahlkriterien abgeschlossen haben und ENTER drücken, wird ein Statistikübersichtsbildschirm ähnlich dem folgenden Beispiel angezeigt:

```
13:44:35              ***** NATURAL SYSTP UTILITY *****             2008-08-29
 User SAG           - Natural Monitor Program Statistics -       TID TCK8

 C Program  Library       NAT- ADA- Ext- Mean-  Screen I/O   User    Sys  Fetch
   *        *              time time time  time     No   KB    Acc     Acc
 ------------------------------------------------------------------------------
 _ MONMNU1M SYSTP             0    0    0   0.0     18    9      6       0     19
 _ SYSMAPOM SYSTP             0    0    0   0.0      0    0      0       0     52
 _ S2SCOM01 SYSTP             0    0    0   0.0      0    0      0       0     36
 * MONMNU1P SYSTP             0    0    0   0.0      1    0     28       0      6
 _ MONLST1P SYSTP             0    0    0   0.0      3    0     21       0     12
 _ SYSMAP1M SYSTP             0    0    0   0.0     19   13      2       0     20
 _ NAT00059 SYSTP             0    0    0   0.0      0    0      0       0    346
 _ STPMNU1P SYSTP             0    0    0   0.0      4    0     30       4      8
 _ NAT42004 SYSTP             0    0    0   0.0      0    0      0       0      9
 _ STPMMM1M SYSTP             0    0    0   0.0      7    5      4       0      9
 _ SYSTPRET SYSTP             0    0    0   0.0      0    0     13       0      4
 _ NAT00030 SYSTP             0    0    0   0.0      0    0      0       0      4
 _ NSCC0002 SYSTP             0    0    0   0.0      0    0     14       0      7
 _ LOGON    SYSTP             0    0    0   0.0      0    0      0       0      1
   Top of List                                                          1 of 3
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Cont  Help  Menu  Exit  Sel         --    -     +     ++    Term        Canc
```

Auf diesem Bildschirm werden alle Programme/Terminals und Libraries aufgelistet, die in Ihrer aktuellen Natural-Sitzung aktiv waren.

Mit PF4 können Sie ein Auswahlfenster aufrufen, in dem Sie Auswahlkriterien angeben können (siehe Schritt 1), um die Liste der auf dem Bildschirm angezeigten Elemente zu reduzieren.

Mit PF10 können Sie zwischen der Programmstatistik und der Terminalstatistik umschalten.

Die Spalten auf dem Statistikübersichtsbildschirm und die entsprechenden Feldnamen auf dem ausführlichen Statistikbildschirm (siehe Spalte **C** unten) werden in der folgenden Tabelle erläutert:

Spalte	Entsprechendes Feld	Erläuterung
C	-	Gilt nur beim Statistikübersichtsbildschirm. Über dieses Eingabefeld kann ein ausführlicher Statistikbildschirm für ein ausgewähltes Programm/Terminal aufgerufen werden: Geben Sie neben dem Listeneintrag, zu dem Sie ausführlichere Statistikinformationen wünschen, ein beliebiges Zeichen ein und drücken Sie ENTER. **Anmerkung:** Wenn die Statistikübersicht der aktiven Programme/Terminals wiederholt angezeigt wird,

Spalte	Entsprechendes Feld	Erläuterung
		erscheint in der Spalte **C** ein Stern (*) neben dem Programm/Terminal, das seit der letzten Wiederholung am aktivsten ist.
Program	**Name of program**	Gilt nur für die Terminalstatistik. Name des aktiven Programms.
Terminal	**Name of terminal**	Gilt nur für die Terminalstatistik. Name (ID) des aktiven Terminals.
Current Program	**Current program / library**	Gilt nur für die Terminalstatistik. Name des ausgeführten Programms und der Name der Library, die dieses Programm enthält.
Library	**Name of library**	Gilt nur für die Programmstatistik. Name der Library, die das zurzeit ausgeführte Programm enthält.
NAT-time	**Time in Natural**	Zeit im Natural Nukleus und in der Schnittstelle.
ADA-time	**Time in Adabas**	Wartezeit auf Antwort von Adabas.
Ext-time	**Time in external program**	Zeit, die ein benutzergeschriebenes Modul benötigt.
Mean-time	**Mean evaluation time**	Verstrichene Zeit einer Natural-Bildschirmtransaktion.
Screen I/O No	**Number of Screen I/Os**	Anzahl der Bildschirm-Ein-/Ausgaben.
Screen I/O KB	**Amount of data transmitted**	Menge der zum oder vom Bildschirm übertragenen Daten.
-	**Evaluation time > 3 sec**	Gilt nur für den ausführlichen Statistikbildschirm für ein Terminal. Prozentualer Anteil der Auswertungszeit, der länger als 3 Sekunden dauert.
-	**Evaluation time > 6 sec**	Gilt nur für den ausführlichen Statistikbildschirm eines Terminals. Prozentualer Anteil der Auswertungszeit, die länger als 6 Sekunden dauert.
User Acc	**Number of user file accesses**	Zähler für Zugriffe auf Adabas-Benutzerdateien.
Sys Acc	**Number of system file accesses**	Zähler für Zugriffe auf die Natural-Systemdatei, einschließlich Fetches.
Fetch	**Number of fetches**	Zähler für die Gesamtzahl der Fetches.

Natural Print/Work Files (SYSFILE)

Diese Funktion liefert Informationen über verfügbare Arbeitsdateien und Druckdateien.

Sie können diese Funktion auch mit dem Systemkommando `SYSFILE` aufrufen, das in der *Systemkommandos*-Dokumentation beschrieben ist.

Die vom Kommando `SYSFILE` bereitgestellten Informationen können auch über die Anwendungsprogrammierschnittstelle `USR1007N` abgerufen werden. Siehe Dienstprogramm SYSEXT.

Diese Funktion kann auch im Batch-Modus für CICS-Sitzungen verwendet werden.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Work File Information** mit einer Liste aller definierten Arbeits- und Druckdateien. Zu jeder Datei werden die folgenden Informationen angezeigt:

Spalte	Erläuterung
No.	Nummer der Arbeits-/Druckdatei.
Type	Art der Zuordnung, d. h. das Betriebssystem, der TP-Monitor oder die Natural-Produktdatei, der die Arbeits-/Druckdatei zugeordnet ist.
Name	Name der Arbeits-/Druckdatei.
Recfm	Satzformat der Arbeits-/Druckdatei.
Lrecl	Logische Satzlänge der Arbeits-/Druckdatei (falls zutreffend).
Blksz	Blockgröße der Arbeits-/Druckdatei.

Status	Eine der folgenden Statusmeldungen:
	`Available for Input/Output`
	oder
	`Open for Input/Output`

Kommandos für Natural Print/Work Files

Zusätzlich zu den Kommandos, die unter ***SYSTP-Bildschirme benutzen*** beschrieben sind, bietet der **Work File Information** die folgenden PF-Tasten und Zeilenkommando:

PF-Taste	Zeilenkommando	Funktion
PF10	-	Blättert zur Liste der Druckdateien.
PF11	-	Blättert zur Liste der Arbeitsdateien.
-	D	Zeigt den entsprechenden Natural Control Block (Arbeitsdateibereich) im Dump-Format an (zur internen Verwendung durch den Support).

Buffer Usage Statistics (BUS)

Die Funktion **Buffer Usage Statistics** (BUS) liefert statistische Informationen über die Nutzung der Natural-Puffer: welche Puffer für die aktuelle Natural-Sitzung zugewiesen sind und wie viel Pufferplatz verbraucht wird. Darüber hinaus können Sie anhand der Gesamtzahlen (**Total**) am Ende der Liste Rückschlüsse auf die Effizienz der Pufferkomprimierung ziehen.

In diesem Abschnitt werden die folgenden Themen behandelt:

- BUS-Funktion aufrufen
- Informationen auf dem BUS-Bildschirm
- BUS-Kommandos

BUS-Funktion aufrufen

Sie können diese Funktion entweder über das **SYSTP**-Menü oder über das Systemkommando `BUS` aufrufen. Siehe auch *SYSTP aufrufen und Funktionen ausführen*. Als Ergebnis erhalten Sie eine Liste mit allen Puffern, die in der aktuellen Natural-Sitzung tatsächlich benutzt werden.

Die Informationen, die das Kommando `BUS` liefert, können Sie auch über die Anwendungsprogrammierschnittstelle `USR1019N` abrufen. Siehe auch Dienstprogramm SYSEXT.

Informationen auf dem BUS-Bildschirm

Dieser Abschnitt gibt einen Überblick über die auf dem Bildschirm **Buffer Usage Statistics** angezeigten Elemente:

Spalte	Erläuterung
No.	Die Puffer werden in der Reihenfolge ihrer Zuweisung fortlaufend nummeriert.
Name	Name des Puffers. Es werden nur die Puffer aufgeführt, die in der aktuellen Sitzung tatsächlich angefordert wurden.
Type	Typ des Puffers: *leer* — Fester/statischer Natural-Puffer.

Spalte	Erläuterung
	V — Variabler Puffer, der im Natural-Thread zugeordnet wurde. O — Variabler Puffer, der derzeit außerhalb des Natural Thread zugeordnet ist. Der Puffer wird bei der nächsten Terminal-E/A in den Thread kopiert und komprimiert. P — Physischer Puffer, der in einem Natural-Workpool zugeordnet ist (siehe auch den Profilparameter WPSIZE. Der Puffer wird bei der nächsten Terminal-E/A freigegeben.
Size	Größe des Puffers (in Bytes).
Used	Anzahl der Bytes, die derzeit genutzt werden. Dieser Wert wird für die Pufferkomprimierung in Umgebungen mit Threads (z. B. CICS) verwendet.
Perc. (**Used**)	Prozentualer Anteil der aktuellen Nutzung, d. h. der Wert der Spalte **Used** im Verhältnis zum Wert der Spalte **Size**.
MaxUsed	Maximale Anzahl der Bytes, die im Laufe der aktuellen Sitzung bisher verwendet wurden (*nicht* die Größe, die derzeit genutzt wird).
Perc. (**MaxUsed**)	Prozentsatz der aktuellen Sitzungsnutzung, d. h. der Wert der Spalte **MaxUsed** im Verhältnis zum Wert der Spalte **Size**.
MaxSize	Maximale Größe (in Bytes), die dem Puffer im Laufe der aktuellen Sitzung bisher zugewiesen wurde (gilt nur für variable Puffer).
Perc. (**MaxSize**)	Maximale Größe, die bisher zugeordnet wurde (Wert der Spalte **MaxSize**) im Verhältnis zur aktuellen Größe (Wert der Spalte **Size**, gilt nur für variable Puffer). Ein Prozentsatz von 10000 oder mehr wird durch die hervorgehobene Anzeige 9999.9 dargestellt.
Am Ende der Liste werden die folgenden Informationen angezeigt:	
ThrdSize	Aktuelle Größe (in KB) des Natural Thread.
Total	Summe aller Puffergrößen (sowohl in Bytes als auch in KB) und Prozentsätze „verwendet/zugewiesen“. Diese Summen können auch mit PF10 angezeigt werden (siehe unten). Für **MaxSize** zeigt die Summe die maximale zusätzliche Größe des Thread an, die im Laufe der bisherigen Sitzung benötigt wurde.

BUS-Kommandos

Zusätzlich zu den Kommandos, die unter *SYSTP-Bildschirme benutzen* beschrieben sind, bietet der Bildschirm **Buffer Usage Statistics** die folgenden PF-Tasten und Zeilenkommandos:

PF-Taste	Zeilenkommando	Funktion
PF4	D	Zeigt den Inhalt des mit dem Cursor/Kommando markierten Puffers im Dump-Format an (zur internen Verwendung durch den Support).
-	S	Zeigt Details des ausgewählten Puffers in einer tabellarischen Übersicht an, z. B. **Thread-Offset**, **Adress** oder **Buffer Size**. Die Umschaltung zwischen dezimaler und hexadezimaler Anzeige ist durch Drücken von PF12 möglich.
PF10	-	Zeigt die Gesamtzahlen (**Total**) zur Puffernutzung an.
PF11	-	Zeigt folgende Informationen an: ■ Relative Adressen der Puffer an, d. h. relativ zum Eingabe-/Ausgabe-Steuerungspuffer (IOCB), ■ absolute Adressen der Puffer, ■ interne Kennung (ID) des Puffers, ■ Lücke (in rot) zwischen diesem und dem nächsten Puffer. **Anmerkung:** Wenn der Puffer unterhalb der 16-MB-Grenze liegt, ist es möglich, dass Sie einen unerwarteten Wert erhalten.
PF12	-	Zeigt den Grad der Pufferkomprimierungsoptimierung für alle Puffer in einer separaten Spalte **CmprOpt** an. **Anmerkung:** Durch zweimaliges Drücken von PF12 wird die Tabelle auf den ursprünglichen Zustand zurückgesetzt.

Permanente Speichersegmente

Die Funktion **Permanent Storage Segments** zeigt permanente Speichersegmente an, d.h. Speichersegmente, die auch dann noch existieren, wenn die Sitzung, die den Speicher zugeordnet hat, bereits beendet wurde.

Zu jedem zugewiesene Segment werden folgende Informationen angezeigt:

- Name
- Adresse
- Größe
- Datum und Uhrzeit der Zuweisung

- Kennung (ID) des Benutzers, der das Segment zugeordnet hat.

Das Zeilenkommando `D` zeigt den Inhalt eines Speichersegments an.

Natural-Subsysteme und Roll Server-Informationen anzeigen

Die Funktion **Natural Sub-Systems and Roll Server Information** zeigt eine Liste der in Ihrer Umgebung verfügbaren Natural-Subsysteme und den aktuellen Status (aktiv oder inaktiv) der Authorized Service Manager und Roll Server an, die mit den Subsystemen verbunden sind.

In der Spalte **GBP num** wird angezeigt, wie viele Buffer Pools mit diesem Subsystem verbunden sind. Es erscheint dort dieselbe Anzahl, die bei Verwendung des Zeilenkommandos `B` (s. unten) angezeigt wird.

In der Spalte **ECSA-D Address** wird die Adresse des ECSA-Verzeichnisse angezeigt. Die Adresse wird außerdem im Header eingeblendet, wenn Sie das Zeilenkommando `E` verwenden.

Zusätzlich zu den Kommandos, die unter ***SYSTP-Bildschirme benutzen*** beschrieben sind, stehen im Bildschirm **Natural-Sub-Systems** die folgenden Zeilenkommandos zur Verfügung:

PF-Taste	Zeilenkommando	Funktion
PF4	-	Schaltet die Sortierung der Listeneinträge um: sortiert nach physischer Reihenfolge, nach Name oder nach ECSA-Adresse.
-	B	Zeigt Informationen über den Buffer Pool und den Buffer Pool Cache (falls vorhanden) an, z.B. Name, Typ und Größe.
-	D	Zeigt Serververzeichniseinträge aus dem Session Information Pool (SIP) im Dump-Format an (zur internen Verwendung durch den Support). Informationen über SIP finden Sie unter *Authorized Services Manager unter z/OS* in der *Operations*-Dokumentation.
-	E	Zeigt den Speicher des ECSA-Verzeichnisses.
-	L	Zeigt Einträge im Dateiverzeichnis des Roll-Servers an und setzt sie zurück. Die Meldungszeile zeigt die Anzahl der genutzten und der gesamten Slots sowie den Belegungsgrad in Prozent an.
-	M	Zeigt die Adresse und die Version von Modulen, die mit `NATASM`*vr* verlinkt worden sind. Zurzeit betrifft es `NATZPX`, `NATFSSM` und `NATNDVX`.
PF5	P oder /	Das ausgewählte Subsystem bleibt an der ersten Position in der Liste.
	R	Zeigt Roll Server-Statistiken an, z. B. die Anzahl der Roll Server Slots und Roll Server-Dateien, Aus-/Einspeicherungsaktivitäten (Roll Out/Roll In), Roll-Datei-Ein-/Ausgaben sowie die zIIP-CPU-Zeitauslastung des Roll Server. Diese Informationen können bei der Optimierung der Roll Server-Funktionen nützlich sein, wie unter *Roll Server-Leistung optimieren* in der

PF-Taste	Zeilenkommando	Funktion
		Operations-Dokumentation beschrieben. Sie können die Statistikdaten beispielsweise verwenden, um eine optimale Thread-Größe oder Roll File-Größe für eine Natural-Anwendung zu ermitteln.
	S	Zeigt die Liste der Zaps, die auf beim Authorized Services Manager angewendet wurden.
	U	Zeigt Server-Verzeichniseinträge aus dem Session Information Pool (SIP) an. Informationen über SIP finden Sie unter *Authorized Services Manager unter z/OS* in der *Operations*-Dokumentation.
	Z	Zeigt eine Liste der beim Roll Server angewendeten Zaps an.
PF6	-	Sprung an den Anfang der Liste.
PF7	-	Rückwärts blättern.
PF8	-	Vorwärts blättern.
PF9	-	Sprung an das Ende der Liste.
PF11	-	Schaltet die Informationen zwischen Global, Detail und Details nur für das/die aktuell verwendete(n) Subsystem(e) um.

Natural Thread-Nutzungsstatistik

Die Funktion **Natural Thread Usage Statistics** ist nur unter CICS, Com-plete und IMS TM verfügbar. In einer z/OS Parallel Sysplex-Umgebung ist sie nicht verfügbar.

Mit dieser Funktion können Sie die optimale Thread- oder Roll-File-Größe für eine Natural-Anwendung ermitteln.

Sie sollten diese Funktion nur bei Bedarf aktivieren und sie deaktivieren, nachdem Sie die optimale Thread-Größe ermittelt haben, da diese Funktion Platz im Natural Buffer Pool beansprucht. Wenn Sie sie deaktivieren, wird der Platz im Buffer-Pool wieder frei.

Um die Funktion Natural Thread Usage Statistics auszuführen:

1 Definieren Sie einen überdimensionierten Thread im Bereich von 512 bis 1024 KB für Ihre Natural-Anwendung. Berücksichtigen Sie dabei die Anzahl der verwendeten Natural-Add-On-Produkte.

2 Starten Sie Ihre Natural-Anwendung, entweder im Produktions- oder im Testmodus.

3 Rufen Sie das Dienstprogramm SYSTP auf und wählen Sie die Funktion **Natural Thread Usage Statistics**, indem Sie im Feld **Code** des SYSTP-**Main Menu** ein T eingeben.

Sie gelangen auf den Bildschirm **Thread Usage Statistics**.

Die Spalten auf diesem Bild sind in der folgenden Tabelle beschrieben.

4 Geben Sie ein `A` in das Feld **Code** ein, um die Funktion zu aktivieren.

Sie erhalten die Meldung, dass die Funktion **Thread Usage Statistics** aktiviert wurde.

5 Benutzen Sie Ihre Natural-Anwendung unter typischen Produktionsbedingungen.

Die Funktion **Thread Usage Statistics** läuft im Hintergrund und protokolliert die verwendeten Puffergrößen.

6 Wenn Sie die Statistik anzeigen möchten, rufen Sie erneut den Bildschirm **Thread Usage Statistics** auf und geben einen der folgenden Werte in das Feld **Code** ein:

```
S
```

um die Statistik anzuzeigen, oder

```
P
```

um die Statistik zu drucken, oder

```
G
```

um die `GETMAIN`-Statistiken anzuzeigen (siehe *Physical GETMAIN Statistics anzeigen*), oder

```
D
```

um die Statistik zu drucken und die Funktion zu deaktivieren.

Wir empfehlen, den Funktionscode `D` nach Beendigung der Funktion zu verwenden, um Platz im Buffer Pool freizugeben.

Die folgenden Informationen werden auf dem Bildschirm **Thread Usage Statistics** angezeigt:

Spalte	Erläuterung
No.	Die Puffer werden in der Reihenfolge ihrer Zuordnung fortlaufend nummeriert.
Ext. Buffer	Die Größen dieser Puffer werden extern definiert (im Natural-Parametermodul).
Defined Size	Puffergröße wie im Natural-Parametermodul definiert.
Max. Allocated Size	Maximal zugeordnete Puffergröße. Beachten Sie, dass für den internen BB-Bereich 14368 Bytes zum Wert des Profilparameters `ESIZE` addiert werden (siehe *Parameter-Referenz*-Dokumentation).
Max. Used Size	Maximal genutzte Puffergröße.
Sum of external buffer sizes	Summe der externen Puffergrößen: Summe aller im Natural-Parametermodul definierten Puffergrößen.

Spalte	Erläuterung
Sum of internal buffer sizes	Summe der internen Puffergrößen: Summe aller von Natural intern angeforderten Puffergrößen.
Sum of physical GETMAINs	Summe aller physischen GETMAIN-Anforderungen für die Natural-Workpools und die variablen Natural-Puffer außerhalb der Natural-User-Threads.
Max. used thread length	Maximale von Natural verwendete Thread-Länge. Definieren Sie diese Länge als Ihre minimale (optimale) Natural-Thread-Länge. Runden Sie auf die nächste KB-Zahl auf, die durch 2 teilbar ist.
Max. compressed thread length	Maximale Länge eines komprimierten Natural-Threads, der in das Natural-Roll-File geschrieben wurde. Definieren Sie diese Länge als Ihre minimale (optimale) Natural-Roll-File-Länge.

Physical GETMAIN Statistics anzeigen

Physical GETMAIN-Statistics liefern Informationen über alle physischen GETMAIN-Anforderungen, die für die Natural-Workpools und die variablen Natural-Puffer außerhalb der Natural-User-Threads relevant sind. Sie zeigen die ursprünglichen Puffergrößen (**Org. Size**) während des Starts einer Natural-Sitzung, die Anzahl der physischen GETMAIN-Anforderungen (**No.**), die Pufferlänge für die physischen GETMAIN-Anweisung (**Bytes**) und die Pufferposition (**P**) oberhalb oder unterhalb der 16-MB-Grenze.

Die Statistikdaten beziehen sich immer auf die Puffer mit den größten Längen, die innerhalb einer Terminal-Ein-/Ausgabe angefordert wurden, für alle Benutzer der Natural-Anwendung. Die Statistik liefert maximal sechs Einträge für jeden Puffer. Diese Einträge können durch das Wrap-around-Verfahren überschrieben werden. Die höchste Zahl entspricht der maximalen Anzahl der physischen GETMAIN-Anforderungen innerhalb einer Terminal-Ein-/Ausgabe für jeden betroffenen Puffer.

Die ersten beiden Einträge in der Statistik beziehen sich auf die Natural Workpools (falls vorhanden) oberhalb (**WRKPOOLA**) bzw. unterhalb (**WRKPOOLB**) der 16-MB-Grenze. Dabei bezieht sich die höchste physikalische GETMAIN-Zahl auf die Anzahl der gleichzeitig verfügbaren Workpools während der Terminal-Ein-/Ausgabe. Die Summe aller Workpool-Längen ergibt den Gesamtspeicherbedarf der Workpools innerhalb einer Terminal-Ein-/Ausgabe.

Alle nachfolgenden Statistikeinträge beziehen sich auf die physischen GETMAIN-Anforderungen für die variablen Natural-Puffer, die entweder aus Platzgründen nicht im Natural-User-Thread definiert werden konnten oder außerhalb des Natural-User-Thread erhöht wurden. Bei diesen Puffern zeigt die höchste physische GETMAIN-Zahl den größten Platzbedarf für jeden Puffer innerhalb einer Terminal-Ein-/Ausgabe an. Der gesamte zuvor angeforderte Speicherplatz wurde vor jeder der folgenden physischen GETMAIN-Anforderungen freigegeben. Das heißt, die Summe aller physischen GETMAIN-Anforderungen mit der höchsten Zahl zeigt den maximalen

Speicherbedarf für die variablen Puffer außerhalb der Natural-User-Threads während einer Terminal-Ein-/Ausgabe für alle Benutzer der Natural-Anwendung an.

Natural-Lizenzinformationen

Die Funktion **Natural License Information** im SYSTP **Main Menu** (Code `L` oder PF5) zeigt den Inhalt der vorhandenen Lizenzdateien an.

Um die Funktion Natural License Information direkt aufzurufen und zu benutzen:

1 Setzen Sie in der Kommandozeile des Natural-Hauptmenüs das Systemkommando `SYSTP` und den Funktionscode `L` ab.

```
SYSTP L
```

Der SYSTP Utility-Bildschirm der Funktion **License Information** wird angezeigt.

Standardmäßig erscheint zuerst der Inhalt der Lizenzdatei des Basisprodukts Natural linksbündig im Format XML.

2 Durch wiederholtes Drücken der Taste PF5 (Form) können Sie das Anzeigeformat umschalten.

Verfügbare Formate:

Format	Buchstabencode
XML, linksbündig (Standardeinstellung)	`L`
Strukturiertes XML	`S`
Formatierter Text	`F`

Sie können die Funktion **Natural License Information** auch dynamisch mittels Direktkommando `SYSTP L` und mit dem Buchstabencode `S` oder `F` aufrufen, zum Beispiel:

```
SYSTP L.F
```

Optional können Sie zusätzlich den Produktcode des gewünschten Add-On-Produkts angeben, zum Beispiel:

```
SYSTP L.F.nnn
```

Dabei steht *nnn* für den Produktcode.

Beispiel 1:

```
SYSTP L.F
```

Zeigt direkt den Inhalt der Produktlizenzdatei für das Produkt Natural im Format "Formatierter Text".

Beispiel 2:

```
SYSTP L F.NOC
```

Zeigt direkt den Inhalt der Produktlizenzdatei für das Add-On-Produkt Natural Optimizer Compiler (NOC) als formatierten Text an.

3 Durch wiederholtes Drücken der Taste PF10 können Sie zwischen den Produktlizenzdateien für das Basisprodukt Natural und für Add-On-Produkte umschalten, zum Beispiel: Natural Batch for zIIP, Natural for CICS for zIIP, Natural for Com-plete for zIIP, Natural Optimizer Compiler oder Predict, (je nachdem, welches Produkt in Ihrer aktuellen Umgebung installiert ist).

Wen ein Produkt nicht gefunden werden kann, erscheint eine entsprechende Meldung, z.B.:

```
Requested product key (PRD) not found, take next license (NOC).
```

Drücken Sie PF10, um zur nächsten Produktlizenzdatei weiterzuschalten.

4 Drücken Sie PF3 (Exit), um die Funktion zu beenden und zum SYSTP **Main Menu** zurückzukehren.

138 SYSTP-Funktionen unter CICS

- Natural-Benutzersitzungen 1218
- Natural Roll Facilities 1223
- Natural-Thread-Gruppen 1225
- Natural-Speicher-Threads 1226
- Globale Systeminformationen zum Natural CICS Interface anzeigen 1227
- Natural/CICS-Generierungsoptionen 1229
- Natural Thread-Gruppendefinitionen 1229
- Eigene Natural-Benutzersitzung 1230
- CICS Task Information 1231
- System Administration Facilities 1231

Das Dienstprogramm SYSTP bietet Funktionen, die für die Verwendung unter CICS vorgesehen sind.

Um spezifische SYSTP-Funktionen unter CICS aufzurufen:

- Geben Sie im Feld **Code** im SYSTP **Main Menu** ein E für **Environment-Dependent Functions** ein.

  Aus dem dann angezeigten Menü können Sie die in diesem Kapitel erläuterten Funktionen auswählen.

Anmerkung: NCI ist der Produktcode für das Natural CICS Interface. Er wird in Bildschirmen gelegentlich aus Platzgründen anstelle des Produktnamens verwendet.

Natural-Benutzersitzungen

Mit der Funktion **Natural User Sessions** können Sie eine Liste der aktiven Benutzersitzungen in einer Natural-Umgebung anzeigen.

Wenn Sie diese Funktion aufrufen, wird der Bildschirm **Natural User Sessions** angezeigt, der die folgenden Informationen enthält:

Spalte	Erläuterung
Term ID	Eindeutige Terminalkennung innerhalb von CICS, die mit der Natural-Sitzung verbunden ist.
User ID	Natural-Benutzerkennung der Natural-Sitzung.
Tran	CICS-Transaktionskennung, unter der die Natural-Sitzung gerade läuft. Bei pseudo-konversationellen Sitzungen ist dies die Transaktionskennung des pseudo-konversationellen Neustarts.
Start Date / Time	Startdatum und -uhrzeit der Natural-Sitzung. Letzter Akt Zeitpunkt der letzten Bildschirmausgabe.
Last Act	Zeitpunkt der letzten Bildschirmausgabe.
Stat	Sitzungsstatus: siehe *Betriebsstatus*.
Program	Zurzeit aktives Natural-Programm.
Library	Natural-Library, in der der Benutzer gerade arbeitet.

Wenn Sie PF10 drücken, wird die Anzeige des Sitzungsdatums und der Uhrzeit durch die folgenden Sitzungsressourcen-Daten ersetzt:

Spalte	Erläuterung
Thrd Grp	Thread-Gruppe, der der Benutzer zugeordnet ist.
Thread	Name des zuletzt verwendeten Thread.
Roll Fac	Zugewiesene Roll Facility.

In diesem Abschnitt werden die folgenden Themen behandelt:

- Kommandos für Natural-Benutzersitzungen
- Statistik der Natural-Benutzer-Sitzungen

Kommandos für Natural-Benutzersitzungen

Zusätzlich zu den Kommandos, die unter *SYSTP-Bildschirme benutzen* beschrieben sind, können Sie für jedes Element, das auf dem Bildschirm **Natural User Sessions** angezeigt wird, eines der folgenden Zeilenkommandos ausführen.

Zeilenkommando	Funktion
`C`	**Cancel Session** Ruft ein Bestätigungsfenster auf, in dem Sie `YES` eingeben können, um eine Sitzung zur Beendigung zu markieren. Die ausgewählte Sitzung wird dann mit einem Nummernzeichen (#) gekennzeichnet, das in der Spalte **User ID** erscheint. Außerdem wird für die betreffende Sitzung der Betriebsstatus `Purged by Admin` (Vom Administrator gelöscht) auf dem Bildschirm **Natural User Session Statistics** des Benutzers (Administrators) angezeigt, der den Abbruchbefehl ausgeführt hat. Tatsächlich wird die Sitzung beendet, wenn der Eigentümer der zur Beendigung markierten Sitzung die nächste Terminal-Ein-/Ausgabe, den nächsten Adabas-Aufruf oder externen Programmaufruf durchführt. Der Eigentümer der Sitzung erhält dann eine entsprechende Beendigungsmitteilung.
`F`	**Flush Session** Ruft ein Bestätigungsfenster auf, in dem Sie `YES` eingeben können, um eine Sitzung sofort zu beenden. Die beendete Sitzung wird dann mit einem Nummernzeichen (#) gekennzeichnet, das in der Spalte **User ID** erscheint. Der Benutzer (Administrator), der die Sitzung beendet hat, erhält eine Beendigungsmeldung, wenn er versucht, den Bildschirm **Natural User Session Statistics** für die Sitzung aufzurufen, die beendet wurde, aber noch auf dem Bildschirm **Natural User Sessions** aufgeführt ist. Der Eigentümer der Sitzung erhält eine entsprechende Beendigungsmeldung.
`R`	**Reactivate Session** Macht ein Kommando C (Cancel, wie oben beschrieben) rückgängig. Das Kommando `R` entfernt die für eine Sitzung gesetzten Beendigungskennzeichen und setzt den Sitzungsstatus auf aktiv zurück. Beachten Sie, dass Sie eine Sitzung, die mit dem oben beschriebenen Kommando `F` (Flush) beendet wurde, nicht reaktivieren können.

Zeilenkommando	Funktion
W	**Wake up Session** Reaktiviert sofort eine Sitzung, die durch einen CMROLL-Aufruf mit einem Warteintervall ungleich Null, das mit dem Profilparameter MAXROLL angegeben wurde, unterbrochen wurde (siehe *Parameter-Referenz*-Dokumentation). Siehe auch das Beispiel-Programm SUSPEND, das in der Natural-System Library SYSEXTP enthalten ist.

Um die Anzahl der Listenelemente durch Angabe von Auswahlkriterien zu verringern:

- Drücken Sie im Bildschirm **Natural User Sessions** die Taste PF4.

  Sie gelangen zum Fenster **Selection for User Sessions**, in dem Sie Benutzersitzungen nach Inaktivitätsdatum und -uhrzeit, Inaktivitätszeitintervall, Natural-Server-Kennung, CICS-System-Kennung, Terminal-Kennung, Benutzerkennung oder Transaktionskennung auswählen können.

  Unter Inaktivitätsdatum und -zeit sowie Inaktivitätsintervall werden alle Sitzungen aufgelistet, die vor dem angegebenen Datum und der angegebenen Uhrzeit oder vor dem angegebenen Zeitintervall nicht aktiv waren.

  Format der Datumsangabe:

  YYYY-*MM*-*DD* (*YYYY* = Jahr, *MM* = Monat, *DD* = Tag)

  Format der Uhrzeitsangabe: *HH*:*II*:*SS* (*HH* = Stunden, *II* = Minuten, *SS* = Sekunden).

  Um einen Bereich von Kennungen anzugeben, verwenden Sie einen Stern (*) als vorangestelltes oder nachgestelltes Zeichen oder ein Fragezeichen (?) als Platzhalterzeichen. Siehe auch *Beispiele für Namensbereiche*.

Statistik der Natural-Benutzer-Sitzungen

Zu jeder Sitzung, die auf dem Bildschirm **Natural User Sessions** angezeigt wird, können Sie sich weitere Informationen anzeigen lassen, indem Sie den Bildschirm **Natural User Session Statistics** mit dem Zeilenkommando S oder U aufrufen.

Alle Größenangaben auf dem Bildschirm **Natural User Session Statistics** sind in KB, sofern in den nachfolgenden Feldbeschreibungen nicht anders angegeben ist.

Feld	Erläuterung
Started	Tag, Datum und Uhrzeit, als die Sitzung gestartet wurde.
Last Actions	Datum und Uhrzeit, als der Benutzer zuletzt aktiv war.
User	Natural-Benutzerkennung, die durch die Natural-Systemvariable `*USER` zugewiesen wird (siehe *Systemvariablen*-Dokumentation).
at Terminal	Terminalkennung des Terminals, das mit der Natural-Sitzung verbunden ist, wie von der Natural-Systemvariablen `*INIT-ID` zugewiesen (siehe *Systemvariablen*-Dokumentation).
Transid	Kennung der pseudo-konversationellen Transaktion, unter der Natural läuft.
Task # in	Von CICS zugewiesene Task-Nummer, gefolgt von der Kennung der CICS Region.
Cur Strg Used	Aktueller Speicherplatz, der von dieser Sitzung genutzt wird.
Max Strg Used	Maximal von dieser Sitzung genutzter Speicherplatz.
Thread Size	Größe dieses Threads.
Thread Name	Name des zuletzt verwendeten Thread. Bei Threads, die durch `GETMAIN`-Anforderung zugewiesen wurden, besteht der Thread-Name aus dem Präfix `NSCP`, gefolgt von der Terminalkennung.
Thread Group	Name der zugehörigen Thread-Gruppe (ausgelöst durch das Starten der Transaktionskennung).
of Type	Typ des Thread, der für die Thread-Gruppe verwendet wird: `SHR` – Permanente Speicher-Threads. `GETM` – Speicher-Threads, die durch `GETMAIN`-Anforderung zugewiesen werden. `NONE` – Keine Threads verwendet. Alle Speicheranforderungen von Natural werden an CICS weitergeleitet.
Natural Library	Natural Library-Kennung, die durch die Natural-Systemvariable `*LIBRARY-ID` zugewiesen wird (siehe *Systemvariablen*-Dokumentation).
Natural Program	Name des zurzeit von der Sitzung verwendeten Natural-Programms, wie er von der Natural-Systemvariablen *PROGRAM zugewiesen wird (siehe *Systemvariablen*-Dokumentation).
Line No.	Zeilennummer des Natural-Programms, das die Sitzung zurzeit benutzt.
Operational Status	Siehe ***Betriebsstatus*** im folgenden Abschnitt.
Roll Facility	Name der zugehörigen Roll Facility.
Roll Recs (Last)	Anzahl der Datensätze, die beim letzten Roll-Out in die Roll Facility geschrieben wurden.
Roll Recs (Max)	Maximale Anzahl von Datensätzen, die während des Roll-Out geschrieben wurden.
Roll Record Size	Datensatzgröße dieser Roll Facility.
Slot Size	Anzahl der Datensätze, die zum kompletten Roll-Out eines Threads erforderlich sind.

Feld	Erläuterung
Restart Rec. No.	Nummer des Datensatzes, der die Roll-Out-Kontrollinformationen enthält. Dieser Datensatz muss beim Roll-In zuerst eingelagert werden. **VSAM Roll Files:** Die folgenden Angaben gelten nur für VSAM Roll Files. Die Beziehung zwischen Neustart-Datensatznummer (RecNum), Slot-Nummer (SN) und Slot-Größe (SZ) ist wie folgt: RecNum = (SN-1) * SZ + 2 oder SN = (RecNum-2) / SZ + 1
Slot Number	Nummer des Slots in dem VSAM Roll File, das zu dieser Sitzung gehört (nur für VSAM). Siehe auch *VSAM Roll Files* in **Restart Rec. No.** oben.
Compressed Length	Menge an relevantem Speicher, der derzeit durch Rolling verlagert wird.
Session Resumes	Gesamtanzahl der Sitzungswiederaufnahmen.
Swap-Ins	Anzahl der Sitzungswiederaufnahmen mit Swap-In aus dem Swap-Pool.
Thread Switches	Anzahl der Sitzungswiederaufnahmen mit Rolling in einen anderen Thread als dem, in dem sich die Sitzung zuvor befand.
Roll-Ins	Anzahl der Sitzungswiederaufnahmen mit Roll-In aus der Roll Facility.
Region Switches	Anzahl der CICS Region Switches.
OpSys Switches	Anzahl der Betriebssystem-Image Switches in einer z/OS Parallel Sysplex-Umgebung.

Betriebsstatus

Dieses Feld zeigt einen der folgenden Betriebszustände an:

Status	Abkürzung	Beschreibung
`Active`	`Act`	Zurzeit aktiv.
`Inactive`	`Ina`	Inaktiv, noch im Thread.
`Swapped`	`Swp`	Ausgelagert, im Swap Pool.
`Rolled out`	`Rld`	Ausgelagert, in Roll Facility.
`Wait (Init)`	`WtI`	Wartet auf Thread zur Sitzungsinitialisierung.
`Wait (Resume)`	`WtR`	Wartet auf einen Thread zur Wiederaufnahme der Sitzung.
`Initializing`	`Int`	Initialisierung der Sitzung.
`Resuming`	`Res`	Wiederaufnahme der Sitzung, im Thread, noch nicht aktiv.
`Suspending`	`Sus`	Suspendieren der Sitzung.
`Terminating`	`Trm`	Beenden der Sitzung.
`Swapping out`	`Swo`	Ausspeichern der Sitzung.

Status	Abkürzung	Beschreibung
Swapping in	Swi	Einspeichern der Sitzung.
Rolling out	Out	Auslagern aus Thread oder Swap Pool.
Rolling in	In	Einlagern aus Roll Facility.

Die folgenden zusätzlichen Informationen können im **Operational Status** angezeigt werden:

Status	Beschreibung
Conversational	Dialogorientierte Sitzung (PSEUDO=OFF) im Gegensatz zur pseudo-konversationellen/transaktionsorientierten Sitzung. Siehe auch Profilparameter PSEUDO in der *Parameter-Referenz*-Dokumentation.
Forced Conversational	Die letzte Bildschirmeingabe einer Sitzung mit PSEUDO=ON war konversationell. Siehe auch Profilparameter PSEUDO in der *Parameter-Referenz*-Dokumentation.
No-Roll	Die Sitzung darf nicht ausgelagert werden.
Compressed	Sitzung ist komprimiert (im Swap Pool oder in der Roll Facility).
Thread Switched	Der aktuell verwendete Thread ist nicht derselbe wie der zuvor verwendete.
Thread Locked	Sitzung wird am Thread-Wechsel gehindert (z.B. RELO=OFF). Kann auch den Status No-Roll/Conversational erzwingen. Siehe auch Profilparameter RELO in der *Parameter-Referenz*-Dokumentation.
Purged by Admin	Sitzung wurde vom Administrator gelöscht (Flag gesetzt).
Spool Task	Task ist ein Spool-/Druck-Task.
Asynchronous Task	Task ist ein asynchroner Task, der nicht an ein Terminal gebunden ist.

Natural Roll Facilities

Mit der Funktion **Natural Roll Facilities** können Sie anzeigen, welche Roll Facilities für das Auslagern von Benutzerarbeitsbereichen verfügbar sind, um Platz für aktive Benutzer zu schaffen.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Natural Roll Facilities** für die aktuelle CICS Region (wie durch die CICS ID im Bildschirmtitel angegeben). Zu jeder Roll Facility werden die folgenden Informationen angezeigt:

Spalte	Erläuterung
Facility Name	TEMPSTOR wird für den temporären Zusatzspeicher verwendet, MAINSTOR für den temporären Hauptspeicher, und die übrigen Dateinamen sind VSAM Roll Files, wie sie in der CICS File Control Table (FCT) definiert sind. none bedeutet, dass keine Roll Facility verwendet wird.
Record Size	Datensatzgröße dieser Roll Facility.
Slot Size	Diese Spalte wird standardmäßig angezeigt. Anzahl der Datensätze, die erforderlich sind, um einen Thread vollständig auszulagern (maximale Thread-Größe geteilt durch Datensatzgröße, aufgerundet).
No. of Slots	Anzahl der Sitzungen, die in dieses Roll File passen (Anzahl der Dateisätze geteilt durch die Slot-Größe, abgerundet). Gilt nur für VSAM Roll Files.
Facility Users Cur / Max	Aktuelle und maximale Anzahl der Benutzersitzungen, die dieser Roll Facility zugewiesen sind.
Roll Counts Out / In	Anzahl der Sitzungs-Auslagerungsvorgänge von oder in diese Roll Facility.
CI Size*	Größe des Kontrollintervalls des VSAM Roll File.
No. of Records*	Anzahl der Datensätze im VSAM Roll File.
Roll File Initialization* Date and Time	Datum und Uhrzeit, zu der das VSAM Roll File vom Dienstprogramm NCISCPRI formatiert wurde.
Status	Zeigt voll (Full) an, wenn die Anzahl der Benutzer der Einrichtung der Anzahl der verfügbaren Plätze entspricht.

* Diese Spalten werden nur angezeigt, wenn Sie PF10 drücken. Wenn Sie die Taste erneut drücken, wird wieder die Standardanzeige angezeigt

Zusätzlich zu den Kommandos, die unter ***SYSTP-Bildschirme benutzen*** beschrieben sind, bietet der Bildschirm **Natural Roll Facilities** die folgenden PF-Tasten und Zeilenkommandos:

PF-Taste	Zeilenkommando	Funktion
PF4	U	Ruft den Bildschirm **Natural User Sessions** für die markierte Roll Facility auf, der alle Natural-Benutzersitzungen anzeigt, die diese Facility nutzen.
PF10	-	Schaltet zwischen der Anzeige der folgenden Spalten um: **Slot Size**, **No. of Slots**, **Facility Users** und **Roll Counts** (Standardanzeige) und **CI Size**, **No. of Records** und **Roll File Initialization**.

Natural-Thread-Gruppen

Mit der Funktion **Natural Thread Groups** können Sie anzeigen, welche Thread-Gruppen für Natural verfügbar sind.

Wenn Sie die Funktion aufrufen, wird der Bildschirm **Natural Thread Groups** für die aktuelle CICS Region angezeigt (wie durch die CICS ID im Bildschirmtitel angegeben). Für jede Thread-Gruppe werden auf diesem Bildschirm die folgenden Informationen angezeigt:

Spalte	Erläuterung
Group Name	Name der Thread-Gruppe.
Thread Type	Typ des verwendeten Threads: siehe **Natural User Session Statistics**.
Thread Size	Gemeinsame Thread-Größe der Thread-Gruppe.
Strg Used	Maximaler Speicherplatz, der je von einer Sitzung, die diese Thread-Gruppe verwendet, genutzt wurde.
TCBs	Maximale Anzahl an Sitzungen, die gleichzeitig aktiv sind.
Max Compr*	Maximale komprimierte Länge dieser Thread-Gruppe.
VSAM Aux / Main	Für die Gruppe definierte Roll Facilities. CICS-Zwischenspeicher (Aux oder Main) sichert immer VSAM, wenn die VSAM Roll Files nicht verfügbar oder voll sind.

* Diese Spalten werden nur angezeigt, wenn Sie PF10 drücken. Wenn Sie die Taste erneut drücken, wird wieder die Standardanzeige angezeigt

Kommandos für Natural Thread-Gruppen

Zusätzlich zu den Kommandos, die unter *SYSTP-Bildschirme benutzen* beschrieben sind, bietet der Bildschirm **Natural Thread Groups** die folgenden PF-Tasten und Zeilenkommandos:

PF-Taste	Zeilenkommando	Funktion
PF4	U	Zeigt alle **Natural-Benutzersitzungen** an, die die mit dem Cursor/Kommando markierte Thread-Gruppe benutzen.
PF10	-	Schaltet zwischen der Anzeige der folgenden Spalten um: **TCBs** und **Queue Sizes** (Standardanzeige) und **Max Compr** und **Active Tasks**.
-	T	Natural Storage Threads (siehe unten) für die mit dem Cursor/Kommando markierte Thread-Gruppe an.

PF-Taste	Zeilenkommando	Funktion
-	D	Zeigt die **Natural-Thread-Gruppen-Definitionen** für die mit dem Cursor/Kommando markierte Thread-Gruppe an.

Natural-Speicher-Threads

Mit der Funktion **Natural Storage Threads** können Sie Informationen über die Speicher-Threads in der Natural-Umgebung anzeigen.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Natural Storage Threads** für die aktuelle CICS Region (wie durch die CICS ID im Bildschirmtitel angegeben). Auf dem Bildschirm werden die folgenden Informationen angezeigt:

Spalte	Erläuterung
Thread Name	Name des Threads.
Grp No.	Nummer der Gruppe, zu der dieser Thread gehört.
Thrd Size	Nutzbare Thread-Größe.
Strg Used	Maximaler Speicherplatz, der jemals von einer Sitzung, die diesen Thread benutzt, genutzt wurde.
Use Count	Anzahl, wie oft dieser Thread zur Verarbeitung ausgewählt wurde.
Roll-Ins Log. / Phys.	Anzahl der Roll-In-Vorgänge: **Log.** – Wiederaufnahme der Sitzung. **Phys.** – Roll-in von Roll Facility.
Queue Sizes Cur / Max / AtMax	Anzahl der in der Warteschlange wartenden Benutzer: **Cur** – Aktuelle Anzahl der Benutzer in der Warteschlange des Threads. Wenn diese Zahl *n* größer als 1 ist, warten *n* minus 1 Benutzer. **Max** – Maximale Warteschlangenanzahl für diesen Thread. **AtMax** – Anzahl der maximalen Wartezeiten.
Term ID	Terminalkennung der Natural-Sitzung, deren Daten sich in diesem Thread befinden.
Task No.	Kennung (ID) der CICS-Task, die derzeit in diesem Thread aktiv ist. Wenn keine ID angezeigt wird, ist keine Sitzung in diesem Thread aktiv.

Kommandos für Natural-Speicher-Threads

Zusätzlich zu den Kommandos, die unter ***SYSTP-Bildschirme benutzen*** beschrieben sind, bietet der Bildschirm **Natural Storage Threads** die folgenden Zeilenkommandos und PF-Tasten:

Zeilenkommando	PF-Taste	Funktion
C	-	Siehe ***Sitzung abbrechen*** in *Natural-Benutzersitzungen*.
D	PF11	Zeigt die **Natural-Thread-Gruppendefinitionen** an.
F	-	Siehe ***Sitzung auflösen*** in *Natural-Benutzersitzungen*.
G	PF10	Zeigt die **Natural-Thread-Gruppe** an
R	-	Siehe ***Sitzung reaktivieren*** in *Natural-Benutzersitzungen*.
U	-	Zeigt Statistiken über die derzeit im Thread aktiven Natural-Benutzersitzungen an. Siehe auch ***Natural User Session Statistics***.

Globale Systeminformationen zum Natural CICS Interface anzeigen

Die Funktion **Global System Information** dient zur Anzeige von Daten über das Systemverzeichnis.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Global System Information** für die aktuelle CICS Region (wie durch die CICS ID im Bildschirmtitel angegeben). Der Bildschirm enthält die folgenden Informationen:

Feld	Erläuterung
Natural User Sessions	Aktuelle (**Cur**) und maximale (**Max**) Anzahl der Natural-Sitzungen im System.
Concurrent SCP Active	Aktuelle (**Cur**) und maximale (**Max**) Anzahl der gleichzeitigen Systemsteuerungsprogramm (SCP)-Anforderungen. SCP-Anforderungen sind: Sitzungsinitialisierung, Sitzungsunterbrechung, Sitzungswiederaufnahme und Sitzungsbeendigung.
SIR Block Extensions	Aktuelle (**Cur**) und maximale (**Max**) Anzahl der lokalen SIR-Block-Erweiterungen.
Slots in 1st SIR Block	Anzahl der Benutzersitzungen, die in den primären Benutzerkontrollblock passen (erster `USERS`-Subparameter im `NCMDIR`-Makro *).
Slots in SIR Block Extns	Anzahl der Benutzersitzungen, die in einen sekundären Benutzerkontrollblock passen (zweiter `USERS`-Subparameter im `NCMDIR`-Makro *).
VSAM Roll File Slots	Anzahl der zu prüfenden VSAM Roll Files.
Possible Roll Facilities	Anzahl der VSAM Roll Files plus zwei für CICS (`MAINSTOR` und `TEMPSTOR`).
Thread Groups	Anzahl der Thread-Gruppen, die durch Auswertung aller `NCMTGD`-Makro-Angaben beim Systemstart ermittelt werden. Siehe auch *NCMTGD Macro Parameters**.

Feld	Erläuterung
System Recoveries	Systemwiederherstellungen: Anzahl der Korrekturen von Statistikzählungen und/oder Kontrollblockkette.
Size of DIR Extension (B)	Größe der Verzeichnis-Erweiterung: Anzahl der Bytes, die beim Systemstart für Thread-Kontrollblöcke und VSAM Roll File-Online-Verzeichnisse verwendet werden.
Operating System Host ID	Name des Betriebssystem-Abbilds (Image).
Assembled Last	Datum und Uhrzeit, wann das Systemverzeichnis-Quellcodemodul zuletzt assembliert wurde.
CICS System ID	Kennung der CICS Region.
Available Resources: **Swap Pool** **Local Buff. Pool** **Sort Buffer Pool** **Edit Buffer Pool** **Monitor Pool** **RNM Buffer Pool**	In der aktuellen NCI-Systemumgebung verfügbare Ressourcen: Swap Pools, Natural Buffer Pools, Monitor Buffer Pools und RNM Buffer Pools. Typ, Größe (in KB) und Lage (unterhalb (`below`) oder oberhalb (`above`) der 16 MB-Grenze) aller unterstützten Buffer Pools.
Max Thread Size	Größte Thread-Größe für alle gültigen Thread-Gruppen.
VSAM Roll Files	Anzahl der VSAM Roll Files.
Main/Aux TempStor	Zeigt an, ob CICS-Haupt- oder Hilfszwischenspeicher für die Roll Facilities von Natural/CICS verfügbar ist.
Session Logging	Sitzungsprotokollierung: Zeigt an, ob die Natural/CICS-Protokoll-Destination in der CICS-DCT (Destination Control Table) definiert ist und ob die Protokoll-Destination verfügbar ist. Die Protokoll-Destination für Sitzungen wird mit dem Parameter `LOGDEST`* definiert.
Message Logging	Meldungsprotokollierung: Zeigt an, ob die Natural/CICS-Fehlermeldungsprotokoll-Destination in der CICS-DCT definiert ist und ob die Protokoll-Destination verfügbar ist. Die Protokoll-Destination für Meldungen wird mit dem Parameter `MSGDEST`* definiert.
Message Switching	Zeigt an, ob die Transaktionskennung (ID) für die Meldungsumschaltung im CICS definiert ist und ob die Transaktionskennung verfügbar ist. Die Transaktion für den Meldungswechsel wird mit dem Parameter `MSGTRAN`* definiert. Wenn keine Transaktionskennung verfügbar ist, ist ein SYSTP-Session-Flush (siehe ***Flush session*** in *Natural -Benutzer-Sitzungen*) nicht möglich.
Console Terminal	Gibt an, ob das CICS-Konsolenterminal für Natural/CICS verfügbar ist.

* Dieser Parameter ist im `NTCICSP`-Makro des Natural-Parametermoduls enthalten, je nachdem, welche Version der Natural-CICS-Schnittstelle installiert ist. Das Makro ist in der *Parameter-Referenz*-Dokumentationen beschrieben.

Natural/CICS-Generierungsoptionen

Mit der Funktion **Generation Options** können Sie die Einstellungen der Generierungsparameter für ein Natural, das unter CICS läuft, anzeigen. Die Werte dieser Parameter werden im Makro `NTCICSP` des Natural-Parametermoduls (abhängig von der installierten Natural CICS Interface-Version) festgelegt und bei der Installation angelegt.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Generation Options** für die aktuelle CICS Region (wie durch die CICS ID im Bildschirmtitel angegeben). Dieser Bildschirm zeigt eine Übersicht über die Einstellungen der Generierungsoptionen für Natural.

Hinter jeder Parametereinstellung auf dem Bildschirm **Generation Options** steht ein Parameter des Makros `NTCICSP`. Diese Parameternamen können durch Drücken von `PF10` angezeigt werden. Drücken Sie `PF10`, um zwischen dem Bildschirm mit den Parameternamen und den Erläuterungen zu den Parametern zu wechseln.

Verwandte Themen:

- *Installing Natural CICS Interface* in *Installation for z/OS*-Dokumentation
- `NTCICSP`-Makro in der *Parameter-Referenz*-Dokumentation

Natural Thread-Gruppendefinitionen

Mit der Funktion **Natural Thread Group Definitions** können Sie die Definitionen der Natural Thread Group anzeigen.

Wenn Sie diese Funktion aufrufen, erscheint der Bildschirm **Natural Thread Group Definitions** für die aktuelle CICS Region (wie durch die CICS ID im Bildschirmtitel angegeben). Dieser Bildschirm zeigt die folgenden Informationen an:

Spalte	Erläuterung	
Grp No.	Nummer der Thread-Gruppe.	
Group Type	Typ der Gruppendefinition:	
	`SHR`	Permanente Speicher-Threads, die für die Thread-Gruppe verwendet werden sollen.
	`GETM`	Speicher-Threads, die mit GETMAIN zugewiesen werden.

Spalte	Erläuterung	
	none	Keine zu verwendenden Threads. Alle Natural Speicheranforderungen werden an CICS weitergeleitet.
	Alias	Neudefinition der Thread-Gruppe zur Zuweisung einer anderen primären Roll Facility, ausgelöst durch Transaktionskennung/Task-Anforderungsschlüssel.
Roll Fac.	Primäre Roll Facility zugewiesen: VSAM, Aux (temporärer Hilfsspeicher), Main (temporärer Hauptspeicher) oder none (keine Roll Facility zugewiesen).	
Thread Size	GETMAIN-Größe des Thread-Speichers (für die Thread-Gruppen-Typen GETM und SHR).	
TCBs	Maximale Anzahl von Natural-Sitzungen, die gleichzeitig in dieser Thread-Gruppe aktiv sind.	
Transaction IDs / Task Request Keys	Wie in den CICS-Transaktionsdefinitionen für Natural definiert.	

Kommandos für Natural-Thread-Gruppen-Definitionen

Zusätzlich zu den Kommandos, die unter *SYSTP-Bildschirme benutzen* beschrieben sind, bietet der Bildschirm **Natural Thread Group Definitions** die folgenden PF-Tasten und die entsprechenden Zeilenkommandos:

PF-Taste	Zeilenkommando	Funktion
PF4	S	Zeigt die Thread-Gruppen-Definitionen für die mit dem Cursor/Kommando markierte Thread-Gruppe an.
PF10	G	Zeigt die **Natural-Speicher-Threads** an, die zu der mit dem Cursor/Kommando markierten Thread-Gruppe gehören.

Eigene Natural-Benutzersitzung

Mit dieser Funktion können Sie den Bildschirm **Natural User Session Statistics** aufrufen, der unter *Natural User Session Statistics* beschrieben ist.

CICS Task Information

Mit dieser Funktion können Sie das Fenster **SYSTP Task Information** aufrufen, das Statusinformationen über den Natural-Task in einer CICS-Umgebung anzeigt.

System Administration Facilities

Mit dieser Funktion haben Sie Zugriff auf Debug- und Tracing-Funktionen.

Wenn Sie diese Funktion aufrufen, erscheint ein Menü mit den folgenden Funktionen:

- Trace Facilities
- Debugging Facilities
- System Snapshot for Logging
- Reset System Highwater Marks
- Gemeinsame dynamische Parameter-Kontrollinformationen
- Angewendete Natural CICS Interface-Quellcodeänderungen
- Angewendete Natural CICS Interface Zaps

Trace Facilities

Diese Funktion ist nur für den internen Gebrauch durch den Support vorgesehen.

Debugging Facilities

Diese Funktion ist nur für den internen Gebrauch durch den Support vorgesehen.

System Snapshot for Logging

Diese Funktion liefert vollständige SYSTP-Batch-Reports (siehe auch *SYSTP in Batch für CICS-Sitzungen*) mit Informationen über alle SCP-Funktionen, unabhängig davon, ob sie verwendet wurden oder nicht. Solche Einrichtungen sind:

- Thread-Gruppen
- Threads des Typs `TYPE=SHR`
- Roll Facilities

Alle diese Informationen werden in der Natural/CICS-Protokolldatei protokolliert, sofern vorhanden.

Reset System Highwater Marks

Diese Funktion umfasst die zuvor beschriebene System Snapshot-Funktion. Zusätzlich können alle Höchstwertmarken des Systems zurückgesetzt werden, z. B.:

- Die Anzahl der Benutzersitzungen.
- Jede Thread-Gruppe und Roll Facility.
- Die Anzahl der UCB-Blockerweiterungen.
- Die Speicherplatzmenge.
- Alle Thread-Gruppen und `TYPE=SHR`-Threads.
- Alle Warteschlangenwerte und -zählungen.
- Alle Roll Facility-Roll-Zahlen.

Gemeinsame dynamische Parameter-Kontrollinformationen

Die Funktion **Common Dynamic Parms Control Information** zeigt die allgemeinen dynamischen Profilparameter an, wie sie von der `PRMDEST`-Destination abgerufen wurden, falls verfügbar. Siehe *PRMDEST - Name des Bestimmungsorts für Natural CICS-Profilparametereingaben* in der *Parameter-Referenz*-Dokumentation.

Angewendete Natural CICS Interface-Quellcodeänderungen

Die Funktion **Applied NCI Source Changes** ruft den gleichnamigen Bildschirm für die aktuelle CICS-Region auf (wie durch die CICS ID im Bildschirmtitel angegeben). Auf diesem Bildschirm werden die Nummern aller Quellcodeänderungen angezeigt, die in der aktuellen Natural-TP-Umgebung angewendet wurden.

Angewendete Natural CICS Interface Zaps

Die Funktion **Applied NCI Zaps** ruft den gleichnamigen Bildschirm für die aktuelle CICS-Region auf (wie durch die CICS ID im Bildschirmtitel angegeben). Auf diesem Bildschirm werden die Nummern aller Quellcodeänderungen angezeigt, die in der aktuellen Natural-TP-Umgebung angewendet wurden.

139 SYSTP-Funktionen unter IMS TM

- Nachricht an eine Benutzergruppe verwalten 1234
- Umgebungsdaten anzeigen 1234
- Monitoring 1235
- Angewandte NII-Zaps 1235

Das Dienstprogramm SYSTP bietet Funktionen, die speziell für IMS TM gelten.

Um spezifische SYSTP-Funktionen unter IMS TM aufzurufen:

- Geben Sie im SYSTP-Hauptmenü im Feld **Code** ein `E` für **Environment-Dependent Functions** ein.

  In dem dann angezeigten **NII Menu** können Sie die in diesem Abschnitt erläuterten Funktionen auswählen.



Anmerkung: NII ist der Produktcode für das Natural IMS TM Interface. Er wird in Bildschirmen gelegentlich aus Platzgründen anstelle des Produktnamens verwendet.

Nachricht an eine Benutzergruppe verwalten

Die Funktion **Broadcasting** wird verwendet, um Nachrichten an bestimmte Benutzergruppen in einer IMS-Umgebung zu senden und zu verwalten.

Wenn Sie diese Funktion aufrufen, wird das Menü **Broadcasting** angezeigt, in dem Sie die folgenden Funktionen auswählen können:

- **Create Broadcast Messages**
  Dient zum Anlegen einer Nachricht an eine Benutzergruppe.
- **List all Broadcast Messages**
  Dient zum Auflisten aller bisher angelegten Nachrichten.

Weitere Informationen über die Broadcasting-Funktion finden Sie in der **TP Monitor Interfaces**-Dokumentation unter *Natural under IMS TM* > *Special Functions* > *Broadcasting*.

Umgebungsdaten anzeigen

Mit der Funktion **Display Environment Data** können Sie Umgebungsdaten von der Natural IMS TM-Schnittstelle anzeigen.

Wenn Sie diese Funktion aufrufen, wird der Bildschirm **Environment Table** für die Umgebungstabelle angezeigt, die von der aktuellen Natural-Sitzung verwendet wird. Auf dem Bildschirm werden die aktuellen Parametereinstellungen der Natural IMS TM-Schnittstelle angezeigt.

Die Parameterwerte können hier nicht geändert werden. Weitere Informationen zu IMS-Parametern finden Sie im Teil *Natural under IMS TM* in der *TP Monitor Interfaces*-Dokumentation.

Monitoring

Mit der Funktion **Monitoring** können Sie Überwachungsdaten zu Natural-Benutzersitzungen anzeigen, die unter demselben Natural-Subsystem laufen.

Wenn Sie die Funktion **Monitoring** aufrufen, gelangen Sie auf den gleichnamigen Bildschirm, auf dem Sie die folgenden Funktionen zur Anzeige von Monitoring-Daten zu Benutzersitzungen auswählen können:

- **Active Sessions**
  Zeigt alle aktiven Natural-Sitzungen an, die unter IMS TM laufen.
- **Suspended Sessions**
  Zeigt alle Natural-Sitzungen an, die derzeit unter IMS TM angehalten sind.
- **User Selection**
  Ruft ein Fenster auf, in dem Sie Auswahlkriterien angeben können, um nur bestimmte Natural-Benutzersitzungen anzuzeigen.

Angewandte NII-Zaps

Diese Funktion ruft das Fenster **Applied NII Zaps** auf, in dem die Anzahl aller Zaps angezeigt wird, die für die aktuelle Natural TP-Umgebung angewendet wurden.

140 SYSTP im Batch-Modus für CICS-Sitzungen

- SYSTP im Batch-Modus aufrufen 1238
- Protokolldatei auswerten 1238

Das Dienstprogramm SYSTP kann auch verwendet werden, um statistische Daten über Natural/CICS-Sitzungen im Batch-Modus zu erhalten.

Die Natural-Protokolldatei, in die die statistischen Daten über Natural/CICS-Sitzungen geschrieben werden, muss dem Natural-Batch-Job als Work File 1 zugeordnet werden (d.h. über `CMWKF01`). Außerdem muss sie im Online-System, d.h. in der CICS-DCT (Destination Control Table) definiert sein. Siehe Parameter `LOGDEST` des Makros `NTCICSP` (je nach installierter Version der Natural-CICS-Schnittstelle), der in der *TP Monitor Interfaces*- bzw. der *Parameter-Referenz*-Dokumentation beschrieben ist.

SYSTP im Batch-Modus aufrufen

Um das SYSTP-Dienstprogramm im Batch-Modus aufzurufen:

- Geben Sie im Batch-Job eines der folgenden Kommandos an:

```
SYSTP xxx
```

oder

```
LOGON SYSTP
SYSBATCH xxx
```

wobei *xxx* die Art der zu verarbeitenden Daten angibt. Beispiel: *xxx*=`nci` gibt an, dass die Daten von einem Natural/CICS-Online-System gesammelt werden.

Protokolldatei auswerten

Daten werden in die Natural-Protokolldatei geschrieben, wenn Natural initialisiert oder zurückgesetzt wird und wenn eine Natural-Sitzung beendet wird.

Das Natural CICS Interface schreibt die folgenden Sätze in die Natural-Protokolldatei

- Einen Startprotokollsatz, wenn die Natural-Umgebung initialisiert oder zurückgesetzt wird.
- Einen Sitzungsprotokollsatz, wenn eine Natural-Sitzung beendet wird.

Wenn eine Natural-Umgebung initialisiert wird, wird eine Systemkennung in den Systemsteuerungsblock geschrieben. Diese Systemkennung gehört auch zu allen Protokollsätzen. Daher kann eine Natural-Protokolldatei von mehreren Natural/CICS-Online-Umgebungen gemeinsam genutzt werden.

Die protokollierten Informationen dienen dazu, die Nutzung der Natural/CICS-Online-Umgebung zu verfolgen. Daher beziehen sich die meisten Informationen auf Einrichtungen der Natural-Umgebung. Die Protokolldatei ist nicht als Accounting- oder Monitoring-Tool gedacht, das sich auf CICS-Funktionen bezieht.

Auf der Grundlage der Systemkennung werden mehrere Berichte mit Daten erstellt, die sich auf eine Natural-Sitzung beziehen:

- Protokolldateidaten in chronologischer Reihenfolge, d. h. die Sitzungsprotokolle sind nach Datum und Uhrzeit des Sitzungsendes sortiert.
- Statistiken darüber, wie die Natural-Umgebung eingerichtet und verwendet wurde.
- Statistiken über Thread-Gruppen (falls verwendet).
- Statistiken über Programmspeicher-Threads (falls verwendet).
- Statistiken über Roll Facilities (falls verwendet).

Diese Berichte werden für alle Natural-Umgebungen mit Aufzeichnungen über Natural/CICS-Sitzungen in der Natural-Protokolldatei erstellt.



Anmerkung: Die Protokollsätze zur Sitzungsbeendigung geben natürlich nur die Ressourcen wieder, die von den entsprechenden Sitzungen verwendet wurden. Daher spiegeln diese Aufzeichnungen möglicherweise nicht die gesamte Natural-Umgebung wider. Berichte über eine vollständige Natural-Umgebung erhalten Sie, indem Sie mit der Funktion **System Administration Facilities** eine Momentaufnahme der gesamten Natural-Umgebung erstellen.

Stichwortverzeichnis

A

activate
    utility, 153

B

batch condition code
    Object Handler, 1059
buffer
    calculate size with SYSRPC utility, 1167

I

invoke
    Profiler utility, 414

L

library
    example library for new features, 649

O

object list
    Object Handler, 1025
option-setting
    Object Handler, 1037

P

parameter-setting
    Object Handler, 1031
profile setting
    Object Handler, 1063
Profiler utility
    invoke, 414
    terminate, 414

S

select-clause
    Object Handler, 1001
server
    ping with SYSRPC utility, 1179
    terminate with SYSRPC utility, 1181
size requirement
    calculate with SYSRPC utility, 1167
stub
    generate with SYSRPC utility, 1149
SYSEXV
    example library for new features, 649

T

terminate
    Profiler utility, 414
tools
    Object Handler, 1073

U

user exit routine
    Object Handler, 1059
utility
    activate, 153