Natural for Mainframes

Editoren

Version 9.2.1

November 2024

ADABAS & NATURAL

Dieses Dokument gilt für Natural for Mainframes ab Version 9.2.1.

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

Copyright © 1979-2024 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

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

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

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

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

Dokument-ID: NATMF-NNATEDITORS-921-20241107DE

Inhaltsverzeichnis

Vorwort xi
1 Über diese Dokumentation 1
Dokumentationskonventionen 2
Online-Informationen und Support 2
Datenschutz 3
I 5
2 NaturalONE als Standard-Entwicklungsumgebung benutzen 7
3 Abgeschaltete CUI-Editoren 9
Von der Abschaltung betroffene Funktionalität 10
Von der Abschaltung nicht betroffene Funktionalität 10
4 Allgemeine Informationen zu den Editoren in NaturalONE 13
NaturalONE-Editor-Typen 14
Aufrufen eines NaturalONE-Editors 15
Bearbeiten von Datenbereichen 15
Anzeigen von Dialogs, Classes und Adapters 16
Probleme in Ihren Natural-Quellcodes 16
Parsen von abhängigen Objekten 17
Unicode- und Codepage-Unterstützung 19
Unterstützung bidirektionaler Sprachen 20
Quellcode-Header 23
Zeilennummern 24
5 Benutzung des Source Editors 27
Allgemeine Informationen zum Source Editor 28
Zugehörige Ansichten 28
Verwendung von Content Assist (Inhaltshilfe) 32
Benutzung der kontextabhängigen Hilfe 33
Code Snippets aus der Natural Community verwenden 33
Arbeiten mit Tasks (Aufgaben) 37
Arbeiten mit Bookmarks (Lesezeichen) 40
Fehlerbehandlung im Source Editor 41
Zu einer bestimmten Natural-Zeilennummer springen 41
Öffnen von referenzierten Objekten und Springen zu Variablen und internen Subroutinen-Definitionen 42
Code-Fragmente externalisieren 44
Einfügen von Call- oder Include-Statements 47
Importieren von Datenfeldern 49
Generierung von Zählerfeldern 52
Hinzufügen und Entfernen von Kommentaren 54
Umschalten zwischen Kommentieren und Entkommentieren 54
Schreibschutz für Quellcodezeilen 55
Geschützte Zeilen in Quellcodes, die mit Construct oder Code Generation generiert wurden 56
Umwandeln in Groß- oder Kleinbuchstaben 56

Einrücken der Quellcodezeilen ... 57
6 Benutzung des Map Editors ... 59
Allgemeines zum Map Editor ... 60
Zugehörige Ansichten ... 61
Hinzufügen von Controls (Steuerelementen) zur Map ... 64
Auswählen von Controls in der Map ... 66
Ändern des Referenz-Control für die Auswahl ... 67
Verwalten der Controls in der Map ... 68
Definieren von Regeln ... 69
Verschieben von Datendefinitionen in Datenfelder ... 71
Bearbeiten des Codes einer Map ... 72
Ändern der Eigenschaften der Map ... 72
Ändern der Eigenschaften einer Textkonstante ... 76
Ändern der Eigenschaften für ein Datenfeld ... 77
Ändern der Eigenschaften für eine Regel ... 85
Definieren von Arrays ... 86
Anzeigen der Statuseigenschaften ... 89
Speichern von Maps ... 90
Speichern von Maps (Stow) ... 90
Validierung von Maps ... 91
Fehlerbehandlung im Map-Editor ... 92
7 Benutzung des DDM Editors ... 95
Anlegen eines Datendefinitionsmoduls (DDM) ... 96
Informationen zum DDM-Editor ... 99
Zugehörige Ansichten ... 100
Verwaltung der Felder des DDM ... 103
Feld-Attribute ... 104
II Sperrmechanismen für Source-Objekte ... 107
8 Sperrmechanismen für Source-Objekte ... 109
Source-Objekte bearbeiten ... 111
Objekte speichern ... 111
Objekte entsperren ... 112
Objekte verschieben, löschen, umbenennen und ersetzen ... 112
III Editoren — Allgemeine Informationen ... 113
9 Editoren — Allgemeine Informationen ... 115
Split-Screen-Modus ... 116
Editor-Profil ... 119
IV Programm-Editor ... 131
10 Programm-Editor ... 133
Programm-Editor aufrufen ... 134
Editor-Kommandozeile ... 135
Obere Informationszeile ... 136
Editierbereich ... 136
Untere Informationszeile ... 138
Editor-Kommandos ... 138

Editor-Kommandos zum Positionieren ... 147
Zeilenkommandos ... 148
Spezielle PF-Tastenfunktionen ... 150
Cursorsensitive Kommandos ... 154
Quellcode speichern und katalogisieren ... 155
Exit-Funktion ... 156
V Datenbereich-Editor (Data Area Editor) ... 159
11 Datenbereich-Editor (Data Area Editor) ... 161
Data Area Editor aufrufen ... 162
Obere Informationszeile ... 164
Editor-Kommandozeile ... 165
Untere Informationszeile ... 166
Editierbereich benutzen ... 166
Spalten im Editierbereich ... 167
Erweiterte Felddefinition bearbeiten ... 172
Zeilenkommandos ... 177
Editor-Kommandos ... 183
Editor-Kommandos zum Positionieren ... 188
Data Area speichern und katalogisieren ... 189
User Exit für den Data Area Editor ... 190
Exit-Funktion ... 191
VI Masken-Editor (Map Editor) ... 193
12 Komponenten des Masken-Editors ... 195
13 In vier Schritten zur fertigen Maske ... 197
Schritt 1 - Masken-Profil definieren ... 198
Schritt 2 - Maske definieren ... 198
Schritt 3 - Maskenfelder definieren ... 198
Schritt 4 - Masken-Definition speichern ... 199
14 Masken-Editor aufrufen und beenden ... 201
Masken-Editor aufrufen und neue Maske erstellen ... 202
Masken-Editor aufrufen und vorhandene Maske bearbeiten ... 203
Masken-Editor beenden ... 204
15 Funktionen im Menü 'Edit Map' ... 205
Feld- und Variablen-Definitionen bearbeiten — Funktion "Field and Variable Definitions" ... 206
Maske editieren — Funktion "Edit Map" ... 207
Feldumrahmungen editieren — Funktion "Outline Editor" ... 208
Neue Maske erstellen — Funktion "Initialize New Map" ... 208
Neue Hilfe-Maske erstellen — Funktion "Initialize a New Help Map" ... 208
Profile und Geräte pflegen— Funktion "Maintenance of Profiles & Devices" ... 209
Maske speichern — Funktion "Save Map" ... 209
Maske testen — Funktion "Test Map" ... 209
Maske speichern und katalogisieren — Funktion "Stow Map" ... 210
Hilfe zum Masken-Editor aufrufen — Funktion "Help" ... 210

16 Maske erstellen — Funktion 'Initialize New Map' ... 211
Trennzeichen — "Delimiters" ... 213
Masken-Format-Einstellungen — "Format" ... 215
Kontext-Einstellungen — "Context" ... 219
Füllzeichen — "Filler Characters" ... 221
17 Maske bearbeiten — Funktion 'Edit Map' ... 223
Bildschirm-Modi ... 224
PF-Tasten und Kommandos zum Positionieren ... 225
Zeilenkommandos ... 226
Feld-Kommandos ... 228
18 Maskenfelder definieren ... 231
Maskenfelder direkt im Bildschirm definieren ... 232
Definitionen aus anderen Objekten übernehmen ... 233
Systemvariablen in einer Masken-Definition benutzen ... 238
19 Erweiterte Feld-Bearbeitung ... 241
Funktion zur erweiterten Feld-Bearbeitung aufrufen und beenden ... 242
Felder im erweiterten Feld-Bearbeitungsbereich ... 244
Felder im erweiterten Text-Feld-Bearbeitungsbereich ... 248
20 Feldumrahmungen erstellen — mit dem Outline-Editor ... 251
Voraussetzungen ... 252
Editor für Feldumrahmungen (Outline-Editor) aufrufen ... 254
Box-Codes für Feldumrahmungen benutzen ... 255
Ende-Markierungen für Feldumrahmungen benutzen ... 257
Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen ... 258
Beispiel für Feldumrahmungen — Box-Coding ... 259
Zeilenkommandos und PF-Tasten ... 262
21 Nachträgliches Zuweisen von Feldern ... 265
22 Array- und Tabellen-Definitionen ... 267
Array-Definition ... 268
Array-Tabellen-Definition ... 272
23 Verarbeitungsregeln ... 277
Feldbezogene Verarbeitungsregeln ... 278
Funktionstastenbezogene Verarbeitungsregeln ... 279
Verarbeitungsregel bearbeiten ... 280
VII Map Editor Tutorial ... 287
24 Masken-Editor öffnen ... 289
25 Maskenfelder anlegen, positionieren und löschen ... 295
Felder anlegen und zentrieren ... 296
Felder verschieben ... 301
Felder löschen und Zeilen einfügen ... 316
26 Maske testen und speichern ... 327
27 Verarbeitungsregeln definieren ... 331
28 Felder benennen und Maske speichern/katalogisieren ... 341
29 Feld-Eigenschaften definieren ... 345
30 Hilfe-Maske erstellen und testen ... 351

31 Maske mit INPUT USING MAP aufrufen 357
32 Maske für WRITE erstellen und Feld-Definitionen kopieren 363
33 Layout einer Maske wiederverwenden 371
34 Maske mit WRITE USING MAP aufrufen 375
VIII DDM-Editor (SYSDDM Utility) 379
35 Funktionsbeschreibung 381
DDMs speichern 382
Einschränkungen bei der Benutzung 383
36 SYSDDM aufrufen und beenden 385
37 SYSDDM Pflege- und Service-Funktionen benutzen 387
Hilfe zu Funktionen 388
Funktionen ausführen 389
Beschreibung der Funktionen 389
DDM-Parameter angeben 393
38 DDMs erstellen 397
39 DDM-Editor aufrufen und beenden 401
Editor aufrufen 402
Editor-Sitzung beenden 403
40 DDM-Editor benutzen 405
DDM-Kopfzeilen-Informationen 406
Spalten für Feld-Attribute 407
Kommandos zum Editieren und Ausführen von Funktionen 413
Erweiterte Feld-Attribute angeben 419
41 DDM katalogisieren 423
42 DDMs auflisten 425
43 DDMs in unterschiedlichen Umgebungen pflegen 427
IX Software AG Editor 429
44 Allgemeine Informationen zum Software AG Editor 431
45 Software AG Editor aufrufen 433
46 Benutzung des Editor-Bildschirms 435
Benutzung der Eingabebereiche 437
In den Daten im Editierbereich blättern 437
Zu einer bestimmten Zeile springen 440
Zeilen zeigen oder verbergen 441
Ränder, Tabulator- und Spaltenpositionen anzeigen 442
47 Kommandos benutzen 443
Ausführung eines Kommandos 444
48 Daten anlegen und ändern 447
Zeilen einfügen und löschen 448
Zeilen kopieren, verschieben, überlagern und wiederholen 450
Ein Fenster mit Daten kopieren oder verschieben 454
Horizontale und vertikale Bereichsgrenzen einstellen 460
Maskenzeile definieren 462
Daten zwischen Bereichsgrenzen anordnen 463
Daten zentrieren 464

Daten ausrichten ... 465
Physischen oder logischen Tabulator benutzen ... 468
Zeilen in alphabetischer Reihenfolge sortieren ... 473
Zeichenkette suchen ... 473
Zeichenkette ersetzen ... 477
49 Editor-Profil einstellen ... 481
Editor-Profil-Einstellungen ... 483
50 Daten speichern und den Software AG Editor verlassen ... 485
51 Zeilenkommandos (Übersicht) ... 487
52 Hauptkommandos (Übersicht) ... 493
Syntax-Elemente ... 495
Gemeinsame Kommando-Optionen ... 496
ADVANCE ... 498
AORDER ... 499
AUTOREN ... 499
AUTOSAVE ... 500
BNDS ... 500
BOTTOM ... 501
CANCEL ... 501
CAPS ... 501
CEnter ... 502
CHANGE ... 502
COLS ... 505
CURSOR ... 506
CWINDOW ... 506
DELETE ... 506
DOWN ... 508
DWINDOW ... 508
EMPTY ... 509
END ... 509
ESCAPE ... 509
EXCLUDE ... 510
FIND ... 512
FIX ... 515
HEX ... 515
INCLUDE ... 515
JLEFT ... 516
JRIGHT ... 516
JUSTIFY ... 517
LABEL ... 518
LC ... 519
LEFT ... 520
LIMIT ... 521
LOCATE ... 521
LOG ... 522

MASK 523
MWINDOW 523
NULLS 523
ORDER 524
POWER 525
PROFILE 525
PROTECT 525
RCHANGE 526
RECOVERY 526
RENUMBER 527
RESET 527
RFIND 528
RIGHT 528
SORT 528
TABS 529
TOP 530
UC 530
UNDO 531
UNREN 531
UP 532
WINDOW 532
XSWAP 533
Stichwortverzeichnis 535

Vorwort

Dieses Dokument beschreibt die Editoren, die in einer NaturalONE-Entwicklungsumgebung bzw. in einer Natural für Großrechner-Entwicklungsumgebung zur Verfügung stehen.

Anmerkung: Ab Natural Version 9.1 für z/OS und Linux ist NaturalONE die Standard-Entwicklungsumgebung. Der Natural Development Server ist ebenfalls in Natural integriert.

Die Dokumentation für die Natural-Editoren ist in folgende Hauptteile gegliedert:

- **GUI-Editoren in NaturalONE**
- **CUI-Editoren in Natural für Großrechner**

GUI-Editoren in NaturalONE	
NaturalONE als Standard-Entwicklungsumgebung benutzen	Beschreibt die drei Editoren mit grafischer Benutzeroberfläche (GUI), die Ihnen zur Verfügung stehen, wenn Sie NaturalONE benutzen. ■ ***Allgemeine Informationen zu den GUI-Editoren in NaturalONE*** ■ ***Benutzung des Source Editors*** ■ ***Benutzung des Map Editors*** ■ ***Benutzung des DDM Editors*** Die oben genannten Beschreibungen sind übersetzte Exzerpte aus der englischen NaturalONE-Dokumentation (Stand: Mai 2022). Weitere übersetzte Informationen zum Thema „NaturalONE als die Standard-Entwicklungsumgebung“ und damit in Verbindung stehende Funktionen finden Sie in der Natural-*System-Architektur*-Dokumentation. ■ *Was ist NaturalONE?* ■ *Verschiedene Modi für die Entwicklung von Natural-Anwendungen* ■ *Verwendung eines Versionskontrollsystems*
CUI-Editoren in Natural für Großrechner	
Abgeschaltete CUI-Editoren	Beschreibt, welche CUI-Editoren in Ihrer Natural für Großrechner-Umgebung abgeschaltet sein können.
Sperrmechanismen für Source-Objekte	Beschreibt die Sperrmechanismen, die bei Natural-Source-Objekten zur Anwendung kommen können.

Editoren — Allgemeine Informationen	Erläutert, welche Natural-Objekte mit welchem Natural-Editor bearbeitet werden. Außerdem wird der Split-Screen-Modus (geteilter Bildschirm) und das Editor-Profil erklärt. Eine Anleitung zur Benutzung der CUI-Editoren finden Sie im Dokument *Erste Schritte*. Informationen zur Unicode- und Codepage-Unterstützung bei den Natural-Editoren siehe *Editoren in der SPoD-Umgebung* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.
Programm-Editor	Beschreibt den Editor, der benutzt wird, um Natural-Programme, Subprogramme, Subroutinen, Klassen, Copycodes, Helproutinen, Functions und Text-Objekte zu erstellen und zu ändern.
Datenbereich-Editor *(Data Area Editor)*	Beschreibt den Editor, der zum Erstellen und Ändern von lokalen, globalen und Parameter-Datenbereichen (Data Areas) benutzt wird.
Masken-Editor *(Map Editor)*	Beschreibt den Editor, mit dem Sie Bildschirm-Masken (Eingabe-, Ausgabe-, Hilfemasken) erstellen und ändern können.
Map Editor Tutorial	Dieses Tutorial enthält eine Einführung in die Benutzung des Natural-Masken-Editors.
DDM-Editor *(SYSDDM Utility)*	Beschreibt, wie Sie Natural-Datendefinitionsmodule (Data Definition Modules/DDMs) erstellen, pflegen und löschen können. **Anmerkung:** Bei Natural für Großrechner ist der Editor, mit dem die Datendefinitionsmodule (Data Definition Modules/DDMs) erstellt, gepflegt und gelöscht werden können, als Utility SYSDDM implementiert.
Software AG Editor	Beschreibt den Software AG Editor, der verwendet wird, um Objekte in verschiedenen Natural-Utilities, Natural-Add-on-Produkten und sonstigen Software AG-Produkten zu erstellen und zu bearbeiten.

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

- 2 NaturalONE als Standard-Entwicklungsumgebung benutzen 7
- 3 Abgeschaltete CUI-Editoren 9
- 4 Allgemeine Informationen zu den Editoren in NaturalONE 13
- 5 Benutzung des Source Editors 27
- 6 Benutzung des Map Editors 59
- 7 Benutzung des DDM Editors 95

2 NaturalONE als Standard-Entwicklungsumgebung benutzen

Dieser Teil informiert über die in Natural für Großrechner ab Version 9.1 abgeschalteten Editoren mit zeichenorientierter Benutzeroberfläche (CUI) und beschreibt die Editoren mit grafischer Benutzeroberfläche (GUI), die Ihnen stattdessen zur Verfügung stehen, wenn Sie NaturalONE benutzen.

Anmerkung: Die folgenden Beschreibungen sind übersetzte Exzerpte aus der englischen NaturalONE-Dokumentation (Stand: Mai 2022). Weitere, möglicherweise aktuellere Informationen siehe dort.

Folgende Themen werden behandelt:

Abgeschaltete CUI-Editoren

Allgemeine Informationen zu den Editoren in NaturalONE

Benutzung des Source Editors

Benutzung des Map Editors

Benutzung des DDM Editors

3 Abgeschaltete CUI-Editoren

- Von der Abschaltung betroffene Funktionalität 10
- Von der Abschaltung nicht betroffene Funktionalität 10

Ab Natural Version 9.1 für z/OS und Linux ist NaturalONE die Standard-Entwicklungsumgebung. Der Natural Development Server ist ebenfalls in Natural integriert.

Bei NaturalONE stehen die Editierfunktionen in einer modernen Eclipse-Umgebung zur Verfügung. Die früheren Editoren mit ihrer zeichenbasierten Benutzungsoberfläche auf grünem Bildschirm (Programm-Editor, Datenbereich-Editor *(Data Area Editor)*, Masken-Editor *(Map Editor)*) sind in einer lokalen z/OS- oder Linux-Umgebung abgeschaltet (deaktiviert).

Sie können NaturalONE und den Natural Development Server mit dem Lizenzschlüssel für Natural (NAT) aktivieren. Dies gilt nicht für Natural for Windows.

Anmerkungen:

1. Um herauszufinden, ob in der aktuellen Natural-Umgebung CUI-Editoren abgeschaltet sind, können Sie das Systemkommando `TECH` oder die Systemvariable `*EDITOR` benutzen.
2. In einer Umgebung, in der Natural ISPF installiert ist, können Sie alle Natural-Editoren benutzen.

Von der Abschaltung betroffene Funktionalität

Wenn Natural-Editoren in der aktuellen Natural-Umgebung abgeschaltet sind, werden die mit den Editoren in Beziehung stehenden Funktionen der folgenden Kommandos und Merkmale nicht mehr unterstützt. Es erscheint eine entsprechende Fehlermeldung, wenn Sie eine Editierfunktion anfordern.

- Systemkommando `EDIT`
- Systemkommando `LIST`
- Systemkommando `SCAN`
- Systemkommando `MAINMENU`
- Utility SYSEXT

Von der Abschaltung nicht betroffene Funktionalität

Folgendes ist von diesem Wechsel **nicht** betroffen:

- DDM-Editor *(SYSDDM Utility)*
- Software AG Editor
- Systemkommando `EDT` für den Zeileneditor
- Batch-Modus-Verarbeitung (Batch-Programme laufen wie üblich)
- Predict-Systemkommando `EDIT DESCRIPTION`

- Super Natural-Systemkommando `EDIT REPORT`
- APIs und User Exits, die auf die Editor-Schnittstellen zugreifen.

4 Allgemeine Informationen zu den Editoren in NaturalONE

- NaturalONE-Editor-Typen 14
- Aufrufen eines NaturalONE-Editors 15
- Bearbeiten von Datenbereichen 15
- Anzeigen von Dialogs, Classes und Adapters 16
- Probleme in Ihren Natural-Quellcodes 16
- Parsen von abhängigen Objekten 17
- Unicode- und Codepage-Unterstützung 19
- Unterstützung bidirektionaler Sprachen 20
- Quellcode-Header 23
- Zeilennummern 24

NaturalONE-Editor-Typen

Die folgende Tabelle zeigt, welcher Editor-Typ für einen bestimmten Objekttyp verwendet wird:

Symbol	Objekttyp	Erweiterung	NaturalONE-Editor
	Programm	NSP	Source Editor
	Class	NS4	
	Subprogramm	NSN	
	Subroutine	NSS	
	Function	NS7	
	Copycode	NSC	
	Helproutine	NSH	
	Text	NST	
	Dialog	NS3	
	Adapter	NS8	
	Globaler Datenbereich (GDA)	NSG	
	Lokaler Datenbereich (LDA)	NSL	
	Parameter-Datenbereich (PDA)	NSA	
	Map (Maske)	NSM	Map Editor
	Datendefinitionsmodul (DDM)	NSD	DDM Editor

Die oben gezeigten Symbole werden für die Quellcodes in der **Project Explorer**-Ansicht, in der **Natural Navigator**-Ansicht und in der **Dependencies**-Ansicht verwendet.

Sie werden auch in der **Natural Server**-Ansicht verwendet. Auf einem Natural Server können Natural-Objekte jedoch aus dem Quellcode, dem generierten Objekt (z. B. einem generierten Programm) oder aus beidem bestehen. Der Unterschied zeigt sich im Symbol. Daher können die oben genannten Symbole in der **Natural-Server**-Ansicht wie folgt dargestellt werden (Beispiel für ein Programm):

Symbol	Beschreibung
	Symbol ohne grüne Kugel und nicht grau: Nur das Quellcodeobjekt ist verfügbar.
	Symbol mit grüner Kugel und nicht grau: Quellcodeobjekt und generiertes Objekt sind verfügbar.
	Symbol mit grüner Kugel und grau: Nur ein generiertes Objekt ist verfügbar und kein Quellcodeobjekt.

Aufrufen eines NaturalONE-Editors

Ein NaturalONE-Editor wird aufgerufen, wenn Sie ein bestehendes Natural-Objekt öffnen oder ein neues Natural-Objekt anlegen.

Sie können ein bestehendes Objekt entweder aus einem Natural-Projekt im Eclipse-Arbeitsbereich (**Project Explorer**-Ansicht oder **Natural Navigator**-Ansicht) oder direkt auf dem Natural-Server (**Natural Server**-Ansicht) öffnen. Weitere Informationen finden Sie in den folgenden Abschnitten:

- *Editing Objects* in *Working with Natural Projects in Local Mode*.
- *Editing Objects* in *Working with Natural Objects in Natural Server Mode*.

Sie können ein neues Objekt in einem Natural-Projekt im Eclipse-Arbeitsbereich anlegen (**Project Explorer**-Ansicht oder **Natural Navigator**-Ansicht). Weitere Informationen hierzu finden Sie im folgenden Abschnitt:

- *Creating Natural Objects* (Anlegen von Natural-Objekten) in *Working with Natural Projects in Local Mode* in der *NaturalONE*-Dokumentation.

Bearbeiten von Datenbereichen

NaturalONE verwendet den Natural Source Editor für die Bearbeitung von Datenbereichen. Der Assistent zum Anlegen neuer Objekte erzeugt ein Skelett für das `DEFINE DATA`-Statement. Innerhalb dieses Skeletts können Sie die Variablendefinition schreiben. Die Bearbeitung des Datenbereichs wird durch den Natural-Parser wirkungsvoll unterstützt.

Datenbereiche, die von einem Natural-Server heruntergeladen werden, werden automatisch normalisiert. Das bedeutet, dass das interne Datenbereichsformat in ein `DEFINE DATA`-Statement umgewandelt wird. Dies hat den Vorteil, dass der Datenbereich schnell definiert werden kann und dass eine Versionierung des Quellcodeobjekts des Datenbereichs möglich ist. Wenn Sie das Kommando **Build Natural Project** oder **Upload** verwenden, wird die Normalisierung umgekehrt: Das interne Datenformat des Natural-Servers wird dann aus dem `DEFINE DATA`-Statement generiert.

Anzeigen von Dialogs, Classes und Adapters

NaturalONE unterstützt keine Dialogs und Classes (siehe auch *Are all Natural object types supported?*). Es ist jedoch möglich, den Quellcode eines solchen Objekts mit dem Source Editor anzusehen (und sogar zu bearbeiten, obwohl dies nicht empfohlen wird). Mit NaturalONE ist es nicht möglich, diese Objekte von Grund auf neu zu erstellen. Das Gleiche gilt für Adapter, die von Natural for Ajax generiert werden.

Probleme in Ihren Natural-Quellcodes

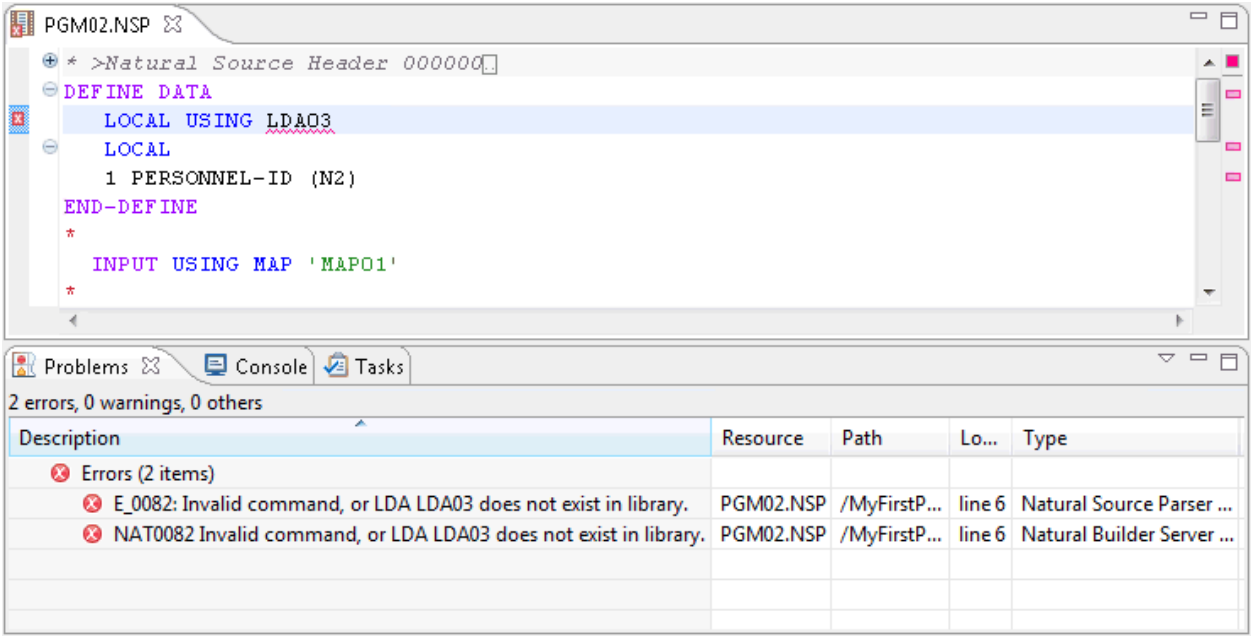
Wenn in einem Projekt ein Syntaxfehler festgestellt wird, nachdem ein Objekt gespeichert wurde, oder wenn in einem Projekt während eines Server-Updates ein Compiler-Problem festgestellt wird (d.h. wenn eine Quellcodeobjekt nicht auf dem Server katalogisiert werden konnte), wird dies in der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht angezeigt.

Anmerkung: Das Syntax-Parsing von abhängigen Objekten kann aktiviert und deaktiviert werden. Weitere Informationen finden Sie unter *Parsing Dependent Objects*.

Im Falle eines Problems enthält das Symbol des betreffenden Projekts eine entsprechende Decoration. Die Unterknoten, die das Problem enthalten, enthalten ebenfalls die Decoration in ihren Symbolen.

Anmerkung: Die Label-Decorations für die Server-Probleme werden über die Voreinstellungen unter **General > Appearance > Label Decorations** gesteuert. Wenn Sie diese Label Decorations nicht wünschen, gehen Sie einfach zu der oben erwähnten Einstellungsseite und deaktivieren Sie **Natural Compiler Problems**.

Die **Problems**-Ansicht zeigt die entsprechende(n) Fehlermeldung(en) an. Wenn Sie auf einen Fehler in der **Problems**-Ansicht doppelklicken (oder wenn Sie den Fehler markieren und dann im Kontextmenü **Go to** wählen), wird die entsprechende Zeile im Editor markiert.



In der Problems-Ansicht werden zwei Arten von Fehlern angezeigt. Fehler, die mit `NAT` beginnen, sind Natural-Systemfehlermeldungen, die von Natural auf dem Server ausgegeben werden. Fehler, die mit `E_` beginnen, werden vom lokalen NaturalONE-Parser ausgegeben, wenn ein Fehler im Editor gefunden wird.

Wenn Sie einen Fehler in der **Problems**-Ansicht auswählen, wird die Fehlermeldung auch unten links im Eclipse-Fenster angezeigt.

In den Natural-Voreinstellungen können Sie festlegen, ob die Informationen in der **Problems**-Ansicht angezeigt werden sollen. Siehe *Source Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Parsen von abhängigen Objekten

Wenn das Syntax-Parsing abhängiger Objekte in den Natural-Voreinstellungen aktiviert ist (siehe *Natural* > *Builder* in den Voreinstellungen), werden alle abhängigen Objekte, die Syntaxfehler enthalten, mit einer Fehlermarkierung in der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht angezeigt. Außerdem wird in der **Problems**-Ansicht eine entsprechende Fehlermeldung angezeigt.

Ein abhängiges Objekt ist z.B. ein Programm, das die Felder eines Datenbereichs verwendet. Wenn der Datenbereich geändert wird, muss diese Änderung auch im abhängigen Programm berücksichtigt werden.

In den Natural-Voreinstellungen können Sie auch festlegen, dass das Syntax-Parsing nach dem ersten Fehler gestoppt werden soll (siehe *Natural > Builder* in *Setting the Preferences* in der *NaturalONE*-Dokumentation).

Abhängig von Ihren Einstellungen wird die Syntaxanalyse entweder automatisch durchgeführt oder kann manuell gestartet werden. Dies wird im Folgenden erläutert:

- Automatisches Parsing
- Manuelles Parsen

Automatisches Parsing

Wenn das automatische Parsen aktiviert ist, werden alle abhängigen Objekte automatisch geparst, nachdem ein Natural-Quellcode gespeichert wurde.

Das automatische Parsen kann mit dem Kommando **Parse Syntax Automatically** oder durch Setzen der entsprechenden Option in den Natural-Voreinstellungen aktiviert und deaktiviert werden (siehe *Natural > Builder* in *Setting the Preferences* in der *NaturalONE*-Dokumentation). Wenn Sie das Kommando **Parse Syntax Automatically** wählen, wird die Einstellung in den Voreinstellungen entsprechend angepasst und umgekehrt.

Wenn das automatische Parsen aktiviert ist, wird neben dem Kommando **Parse Syntax Automatically** ein Häkchen angezeigt.

Um das automatische Parsen abhängiger Objekte zu aktivieren oder zu deaktivieren:

- Wählen Sie im Menü **Project** das Kommando **Parse Syntax Automatically**.

Manuelles Parsen

Wenn das automatische Parsing deaktiviert ist, ist das Kommando **Parse All** aktiviert und kann verwendet werden, um den Parser manuell aufzurufen. Alle abhängigen Objekte, die geparst werden müssen, seit Sie das Syntax-Parsing aktiviert haben, werden dann vom Parser berücksichtigt.

Wenn Sie Ihre Quellcodes korrigieren und speichern, werden die Fehlermarkierungen in der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht und die Einträge in der **Problems-Ansicht** nicht automatisch entfernt. Um sie zu entfernen, müssen Sie das Kommando **Parse All** erneut aufrufen (oder Sie müssen das automatische Parsen einschalten).

Um abhängige Objekte manuell zu parsen:

- Wählen Sie im Menü **Project** das Kommando **Parse All**.

  Oder:

Drücken Sie CTRL+ALT+P.

Unicode- und Codepage-Unterstützung

Wenn ein Quellcode von einem Natural-Server in den Eclipse-Arbeitsbereich heruntergeladen wird, wird der Quellcode immer von der Server-Kodierung in die Eclipse-Textdatei-Kodierung UTF-8 konvertiert.

Die Server-Kodierung des Quellcodes wird im Quellcode-Header gespeichert (weitere Informationen zum Quellcode-Header finden Sie weiter **unten**).

Vorsicht: Wenn Sie einen Quellcode im Eclipse-Arbeitsbereich bearbeiten und dieser Quellcode die Kodierung UTF-8 hat, können Sie zwar beliebige Unicode-Zeichen eingeben, jedoch können nur Zeichen, die in der Server-Kodierung enthalten sind, auf den Server hochgeladen werden.

In der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht können Sie die Serverkodierung des Quellcodes im Dialogfeld **Properties** des Quellcodes ändern. Die unterstützten ICU-Kodierungen hängen vom Natural-Server ab. Siehe *Changing the Object Properties* in der *NaturalONE*-Dokumentation.

Wenn ein Quellcode zurück auf einen Natural-Server übertragen wird, wird er in die Serverkodierung konvertiert, die im Dialogfeld **Properties** des Quellcodes definiert ist. Wenn der Quellcode „nicht unterstützte“ Zeichen enthält, kann er nicht auf dem Natural-Server gespeichert werden und der Server gibt einen Fehler zurück.

Anmerkung: Der Profilparameter `SRETAIN` (auf dem Natural-Server eingestellt) wird beim Hochladen eines Quellcodes auf einen Natural-Server nicht ausgewertet. Sie müssen eine geeignete Projektkodierung in den Natural-Projekteigenschaften oder eine geeignete Serverkodierung in den Objekteigenschaften der einzelnen Natural-Objekte festlegen. Wenn Sie z. B. das Textfeld für die Projektkodierung in den Projekteigenschaften leer lassen, wird nicht allen neu angelegten Natural-Objekten eine Codepage zugewiesen. Dies gilt auch für ein temporäres Projekt, das angelegt wird, wenn Sie ein Quellcodeobjekt in der **Natural Server**-Ansicht bearbeiten.

In der **Natural Server**-Ansicht (nur auf UNIX- und Windows-Servern) können Sie die Kodierung eines Natural-Objekts auch im Dialogfeld **Properties** (auf der Seite **Object Details**) ändern. Das Ändern der Kodierung eines Objekts bedeutet, dass die auf dem Server gespeicherten Kodierungsinformationen geändert werden; dies bedeutet nicht, dass der Inhalt des Objekts in irgendeiner Weise konvertiert wird.

Weitere Informationen finden Sie unter *Unicode- und Codepage-Unterstützung* bzw. *Unicode and Code Page Support* in der Natural-Dokumentation für die entsprechende Plattform.

Fehlermeldungen

Fehlermeldungen, die in Natural-Projekten im Eclipse-Arbeitsbereich angelegt oder in diese heruntergeladen werden, werden in UTF-8-Dateikodierung gespeichert. Die Fehlermeldungen enthalten jedoch nicht die ursprüngliche Serverkodierung.

Beim Herunterladen von einem Natural-Server wird die Fehlermeldung von der Server-Kodierung in UTF-8 konvertiert. Wird eine Fehlermeldung auf einen Natural-Server hochgeladen, wird sie von UTF-8 in die Server-Kodierung konvertiert. Alle Zeichen, die nicht in der Serverkodierung enthalten sind, gehen dabei verloren.

Daher sollte die Laufzeitumgebung Ihres Natural-Projekts mit der richtigen Kodierung gestartet werden.

Unterstützung bidirektionaler Sprachen

Einige Sprachen, z. B. Arabisch und Hebräisch, werden von rechts nach links (RTL = Right-To-Left) geschrieben, während die meisten Sprachen, z. B. Englisch und Deutsch, von links nach rechts (LTR) geschrieben werden. Text, der sowohl Zeichen von links nach rechts als auch von rechts nach links enthält, wird als bidirektionaler Text bezeichnet.

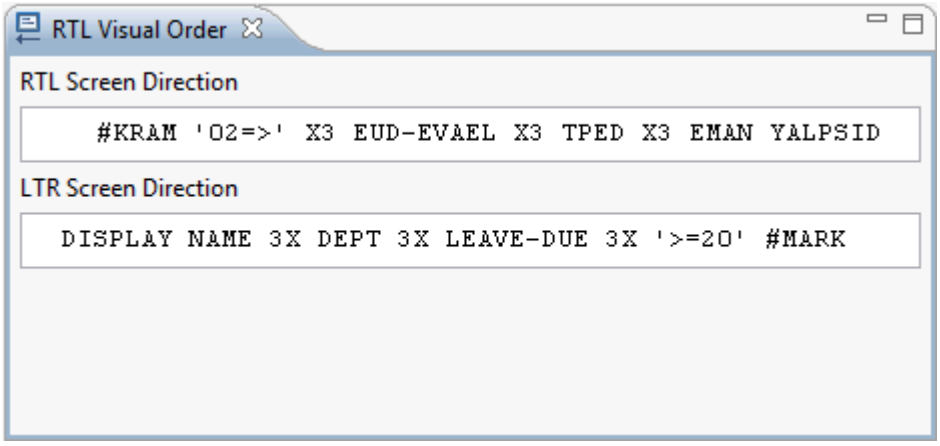
Zusätzlich zur Unterstützung bidirektionaler Sprachen in der Eclipse-Workbench (weitere Informationen finden Sie in der Eclipse-Online-Hilfe) bietet NaturalONE eine spezielle bidirektionale Unterstützung, die in den folgenden Abschnitten beschrieben wird:

- RTL Visual Order-Ansicht
- Option Visuelle Reihenfolge
- Map Editor und Parameter PM=I
- Arabic Shaping (Arabische Formgebung)

RTL Visual Order-Ansicht

Die Ansicht **RTL Visual Order** dient zur Unterstützung von Anwendungen, die ursprünglich für Terminals geschrieben wurden, die den inversen (von rechts nach links) Druckmodus, aber keine bidirektionalen Daten unterstützen. Diese Anwendungen erstellen die Anzeigereihenfolge der bidirektionalen Daten im Anwendungscode, d. h. die Anwendungsdaten werden in visueller Reihenfolge gespeichert.

Anmerkung: Diese Ansicht wird standardmäßig nicht angezeigt, wenn Sie die NaturalONE-Perspektive öffnen. Wie Sie sie anzeigen können, erfahren Sie unter *Showing a View of the NaturalONE Perspective* in der *NaturalONE*-Dokumentation.



Die **RTL Visual Order**-Ansicht kann in den folgenden Fällen benutzt werden:

- Beim **Source Editor**.
- Bei der **Source**-Seite des **Map Editor**, die den Code für eine **Verarbeitungsregel** anzeigt.

Die im aktiven Editor (entweder im Source Editor oder auf der **Source**-Seite des Map Editor) markierte Zeile wird vom visuellen in das logische Format umgewandelt und im Bildschirm von rechts nach links (erstes Textfeld in der Ansicht) und von links nach rechts (zweites Textfeld in der Ansicht) angezeigt.

Wenn Sie die aktuell ausgewählte Zeile im aktiven Editor ändern möchten, können Sie dies in der Ansicht **RTL Visual Order** tun. Jede Änderung wird sofort in den Editor übernommen.

Option Visuelle Reihenfolge

Wenn die Option **Visual order** in den Natural-Einstellungen ausgewählt ist (siehe *Regional Settings > RTL Languages* in den Natural-Voreinstellungen in der *NaturalONE*-Dokumentation), wird angenommen, dass die Anwendungsdaten (d. h. Natural-Quellcodes und Daten aus Datenbanken) in visueller Reihenfolge vorliegen. In diesem Fall konvertieren manche Editoren und Werkzeuge die Daten von der visuellen Reihenfolge in die logische Reihenfolge, bevor sie die Daten anzeigen. Wenn die Daten gespeichert werden, werden sie wieder in die visuelle Reihenfolge umgewandelt.

Map Editor und Parameter PM=I

Die Einstellungen für die bidirektionale Sprachunterstützung werden in den Natural-Voreinstellungen festgelegt. Siehe *Map Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Der inverse Druckmodus wird bei Maps und Datenfeldern unterstützt. Der entsprechende Parameter (`PM=I`) kann in der **Properties**-Ansicht eingestellt werden. Siehe ***Ändern der Eigenschaften der Map*** in der Beschreibung des Map Editor.

Arabic Shaping (Arabische Formgebung)

Wichtig: In der aktuellen Version von NaturalONE ist die Unterstützung für Arabic Shaping als Preview-Version enthalten. Diese Vorabversion wird nicht mehr unterstützt. Wir freuen uns über jedes Feedback unserer Kunden zu dieser neuen Funktion. Da die Unterstützung für Arabic Shaping noch in der Entwicklung ist, kann es sein, dass eine spätere Version von NaturalONE wesentliche Änderungen an dieser Art der Unterstützung enthält.

In arabischem Text sind normalerweise alle Zeichen einer Zeichenkette miteinander verbunden. Aus diesem Grund haben arabische Zeichen bis zu 4 Darstellungsformen: die isolierte, die finale, die initiale und die mediale Form. Welche Form verwendet wird, hängt von der Position des Zeichens in der Zeichenkette ab. Zum Beispiel hat das arabische Zeichen "MEEM" die folgenden Formen in Unicode:

U+0645	م	ARABIC LETTER MEEM
U+FEE1	ﻡ	ARABIC LETTER MEEM ISOLATED FORM
U+FEE2	ﻢ	ARABIC LETTER MEEM FINAL FORM
U+FEE3	ﻣ	ARABIC LETTER MEEM INITIAL FORM
U+FEE4	ﻤ	ARABIC LETTER MEEM MEDIAL FORM

Außerdem werden einige Zeichen zu einer neuen Form kombiniert, wenn sie nacheinander in einer Zeichenkette erscheinen. Dies wird als „Ligatur“ bezeichnet. Zum Beispiel die Zeichen

U+0644	ل	ARABIC LETTER LAM
U+0627	ا	ARABIC LETTER ALEF

haben die folgende kombinierte Form:

U+FEFB	ﻻ	ARABIC LIGATURE LAM WITH ALEF ISOLATED FORM

Unicode-Zeichenfolgen sollten nur die arabischen Zeichen des arabischen Blocks (U+0600 bis U+06FF) oder des arabischen Ergänzungsblocks (U+0750 bis U+077F) enthalten. Es wird nicht empfohlen, die Darstellungsformen in normalem arabischem Text zu verwenden. Es ist Aufgabe der Benutzeroberfläche, die korrekten Formen der Zeichen anzuzeigen. Anwendungen, die ursprünglich auf einfachen Terminals geschrieben wurden, verwenden jedoch häufig arabische Darstellungsformen in den Zeichenketten.

Wenn Daten, die Zeichen aus den arabischen Darstellungsformen enthalten, im Eclipse-Kontext verarbeitet werden müssen, ist es notwendig, die arabischen Zeichen zu entformen („unshape“), um volle bidirektionale Unterstützung von Eclipse zu erhalten. „Unshape“ bedeutet, dass jedes arabische Darstellungsformzeichen in sein Basiszeichen umgewandelt wird. Zum Beispiel wird U+FEE2 (ARABIC LETTER MEEM FINAL FORM) in U+0645 (ARABIC LETTER MEEM) umgewandelt.

Wenn die Daten zurück auf den ursprünglichen Server übertragen werden, müssen die arabischen Zeichenfolgen erneut geformt werden, um eine vollständige bidirektionale Unterstützung auf der Serverplattform zu erhalten. „Shape“ bedeutet, dass jedes arabische Basiszeichen in die entsprechende arabische Darstellungsform umgewandelt wird. Wenn zum Beispiel U+0645 (ARABIC LETTER MEEM) als letztes Zeichen einer Zeichenkette verwendet wird, wird es in U+FEE2 (ARABIC LETTER MEEM FINAL FORM) umgewandelt.

NaturalONE unterstützt Arabic Shaping bei der Quellcode-Übertragung. Über die Natural-Voreinstellungen für Arabic Shaping (siehe *Regional Settings > RTL Languages* unter *Changing the Project Properties* und unter *Setting the Preferences* in der *NaturalONE*-Dokumentation) kann die Umwandlung für Arabic Shaping gesteuert werden. Die „Unshape“-Umformungsoperation wird durchgeführt, wenn die Quellcodes von einem Server heruntergeladen werden, und die „Shape“-Umformungsoperation wird durchgeführt, wenn die Quellcodes auf einen Server hochgeladen werden.

Anmerkungen:

1. Es ist nicht garantiert, dass ein Quellcode nach einer Unshape/Shape-Umformungsoperation absolut identisch mit dem ursprünglichen Quellcode ist. Bei der Umformung werden nach Möglichkeit immer Ligaturformen verwendet, auch wenn der ursprüngliche Quellcode die Einzelzeichenformen enthält. Wenn der ursprüngliche Quellcode eine unangebrachte Form enthält (z. B. eine Endform, wo eine Anfangsform erwartet wird), kann die Umformungsoperation das ursprüngliche Zeichen nicht wiederherstellen.
2. Quellcodes, die arabische Zeichenketten sowohl in visueller Links-nach-Rechts-Reihenfolge (LTR) als auch in logischer Reihenfolge enthalten, können mit NaturalONE nicht bearbeitet werden. NaturalONE geht davon aus, dass alle Zeichenketten in einem Quellcode in der gleichen Reihenfolge gespeichert sind.

Quellcode-Header

Ein Quellcode-Header wird verwendet, um einige Quellcode-Eigenschaften zu erhalten (z. B. den Programmiermodus oder die Codepage).

Der Quellcode-Header ist Teil eines Natural-Quellcodes. Er wird entfernt, wenn der Quellcode auf den Natural-Server übertragen wird, und wird wieder eingefügt, wenn der Quellcode vom Server in den Eclipse-Arbeitsbereich eingelesen wird. Da der Header zum Quellcode gehört, bleibt er auch im Quellcode enthalten, wenn dieser im Repository Ihres Versionskontrollsystems gespeichert wird.

Bei der Verwendung eines Natur-Editors in Eclipse ist der Quellcode-Header geschützt. In diesem Fall kann er nicht geändert oder gelöscht werden. Die Hintergrundfarbe für die geschützten Zeilen, aus denen der Source Header besteht, kann in den Natural-Einstellungen geändert werden. Siehe *Syntax Coloring* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Vorsicht: Wenn Sie einen Nicht-Natural-Editor verwenden, dürfen Sie den Quellcode-Header nicht ändern oder löschen. Dies kann zu Kompilierfehlern oder zu Eigenschaften führen, die nicht mehr korrekt zugewiesen werden können.

Zeilennummern

Standardmäßig sind in Eclipse keine Zeilennummern mit einem Quellcode verbunden. Zeilennummern machen es nahezu unmöglich, den Quellcode in einem Versionskontrollsystem zu versionieren: Jede neue oder entfernte Zeile würde zu einer Neunummerierung der nachfolgenden Zeilen führen, und diese Zeilen würden als geändert markiert werden. Daher entfernt NaturalONE die Natural-Zeilennummern aus dem Quellcode.

Es ist jedoch möglich, die Anzeige der Eclipse-Zeilennummern einzuschalten (**Preferences > General > Editors > Text Editors > Show line numbers**). Dies kann vor allem im Falle eines Fehlers nützlich sein, da der Parser die Fehlerposition anhand der Zeilennummer angibt. Dies kann auch beim Ausführen einer Natural-Anwendung nützlich sein: Im Falle eines Fehlers verwendet die Natural-Laufzeitumgebung die Zeilennummer, um den Benutzer über die Fehlerposition zu informieren.

Eclipse verwendet für die Zeilennummerierung eine Inkrementierung von 1. Wenn Natural-Objekte in den Eclipse-Workspace heruntergeladen werden, werden alle Zeilennummern-Referenzen im Quellcode entsprechend neu nummeriert. Wenn die Objekte auf den Natural-Server hochgeladen werden, werden alle Zeilennummern auf die ursprüngliche Inkrementierung zurückgeändert.

Die folgenden Muster werden als gültige Zeilennummernreferenzen erkannt (wobei *nnnn* eine vierstellige Zahl ist):

```
(nnnn)
(nnnn/
(nnnn,
```

Der Source Editor unterstützt die Zeilennummernreferenzen.

Vorsicht: Wenn die Natural-Quellcodes mit einem anderen Editor als dem Source Editor geändert werden, können Zeilennummernreferenzen zerstört werden.

Standardmäßig werden Zeilennummernreferenzen in Konstanten nicht umnummeriert, wenn Sie einen Quellcode bearbeiten. Wenn sie während der Bearbeitung neu nummeriert werden sollen, müssen Sie die Option **Renumber line numbers in constants** in den Projekteigenschaften aktivieren. Siehe *Editor* unter *Changing the Project Properties* in der *NaturalONE*-Dokumentation. Wenn Sie jedoch einen Quellcode mit dem Kommando **Add to New Project** oder **Add to Existing Project** hinzufügen von einem Server herunterladen, auf dem `RNCONST` auf `ON` gesetzt ist, werden die Zeilennummernreferenzen in Konstanten immer neu nummeriert.

Während des Downloads ist es möglich, die Zeilennummernreferenzen durch Labels zu ersetzen. Dies kann in den Natural-Einstellungen aktiviert werden. Siehe *Natural > Options* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Wichtig: Wenn die oben genannte Option in den Natural-Einstellungen aktiviert ist, werden alle gültigen Zeilennummern (wie oben beschrieben) durch Labels ersetzt. Es wird nicht geprüft, ob das Ergebnis einer gültigen Natural-Syntax entspricht. Es liegt daher bei Ihnen, das Ergebnis der Ersetzung zu überprüfen.

5 Benutzung des Source Editors

- Allgemeine Informationen zum Source Editor 28
- Zugehörige Ansichten 28
- Verwendung von Content Assist (Inhaltshilfe) 32
- Benutzung der kontextabhängigen Hilfe 33
- Code Snippets aus der Natural Community verwenden 33
- Arbeiten mit Tasks (Aufgaben) 37
- Arbeiten mit Bookmarks (Lesezeichen) 40
- Fehlerbehandlung im Source Editor 41
- Zu einer bestimmten Natural-Zeilennummer springen 41
- Öffnen von referenzierten Objekten und Springen zu Variablen und internen Subroutinen-Definitionen 42
- Code-Fragmente externalisieren 44
- Einfügen von Call- oder Include-Statements 47
- Importieren von Datenfeldern 49
- Generierung von Zählerfeldern 52
- Hinzufügen und Entfernen von Kommentaren 54
- Umschalten zwischen Kommentieren und Entkommentieren 54
- Schreibschutz für Quellcodezeilen 55
- Geschützte Zeilen in Quellcodes, die mit Construct oder Code Generation generiert wurden 56
- Umwandeln in Groß- oder Kleinbuchstaben 56
- Einrücken der Quellcodezeilen 57

Allgemeine Informationen zum Source Editor

Der Natural Source Editor basiert auf dem Eclipse-Texteditor. Er unterstützt beispielsweise Syntax-Highlighting und Content Assist. Darüber hinaus bietet er Natural-spezifische Funktionen.

Das Verhalten des Source Editors kann durch Ändern der Natural-Voreinstellungen beeinflusst werden. Siehe *Source Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Es wird ein intelligenter Parser verwendet. Bei der Eingabe des Quellcodes erhalten Sie im Falle eines Syntaxfehlers eine Meldung. In den Projekteigenschaften können Sie festlegen, für welche Plattform die Natural-Syntax geparst werden soll (siehe die Beschreibung der Parser-Optionen im Abschnitt *Changing the Project Properties* in der *NaturalONE*-Dokumentation).

Informationen zur Programmiersprache Natural (z. B. zu Statements, die im Natural Source Editor angegeben werden können) finden Sie über Links in der vorliegenden Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.

Anmerkung: In der *NaturalONE*-Dokumentation (und -Hilfe) finden Sie die (englische) Dokumentation für die Natural-Programmiersprache auf den verschiedenen Plattformen. Siehe *Natural Language for Mainframes*, *Natural Language for UNIX* und *Natural Language for Windows*.

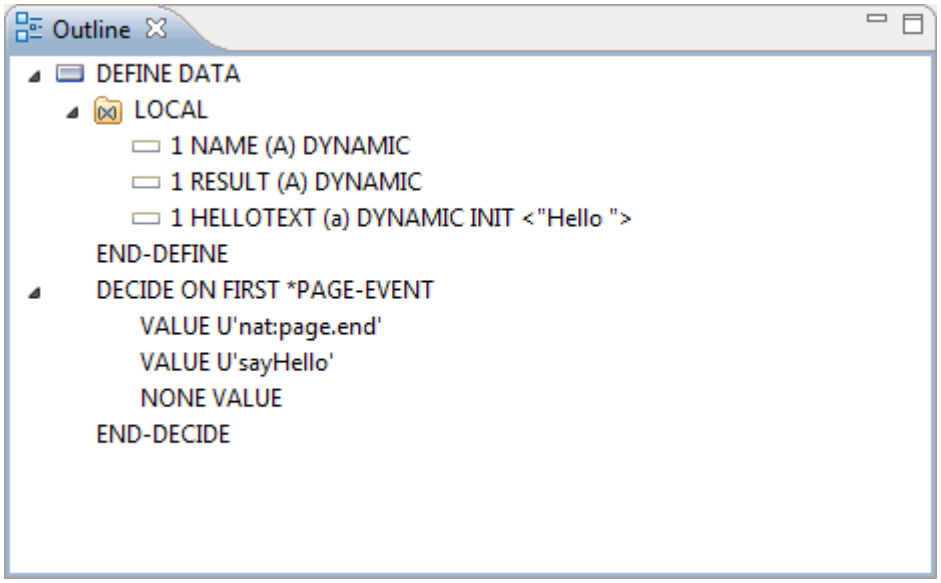
Zugehörige Ansichten

Der Source Editor verwendet die folgenden Ansichten der NaturalONE-Perspektive:

- Outline-Ansicht
- Dependencies-Ansicht

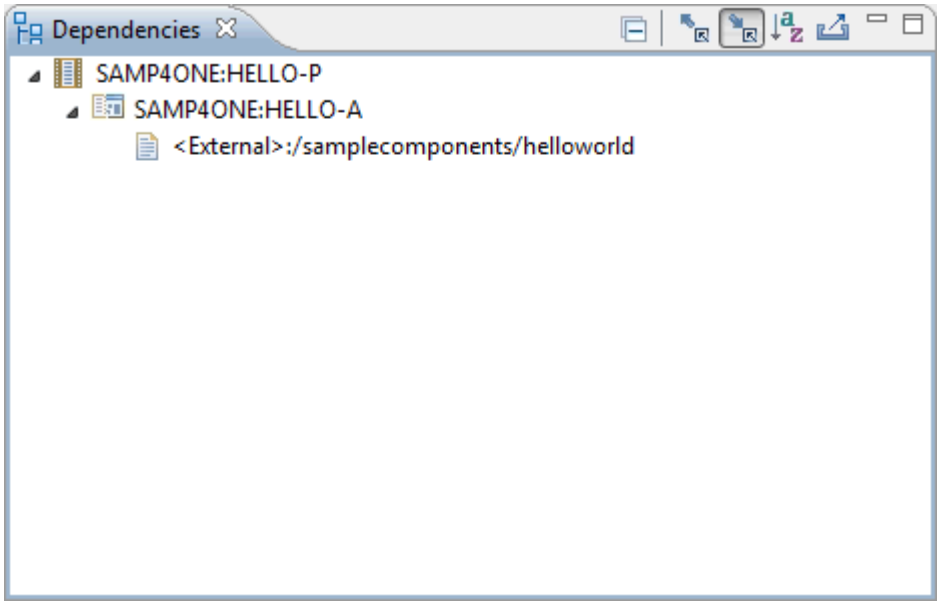
Outline-Ansicht

Die Gliederungsansicht zeigt alle Kontrollstrukturen, die in dem Quellcode verwendet werden, der gerade im aktiven Editorfenster angezeigt wird. Dazu gehören alle Strukturen, die im Editorfenster eingeklappt dargestellt werden können, wie z.B. `DEFINE DATA`, `IF` und `DECIDE`. Wenn Sie ein Objekt in der Gliederungsansicht auswählen, wird die entsprechende Position im Editor angezeigt.



Dependencies-Ansicht

Diese Ansicht zeigt die Abhängigkeiten zwischen dem Natural-Objekt, das gerade im aktiven Editorfenster angezeigt wird, und anderen Objekten. Sie wird jedes Mal aktualisiert, wenn das Objekt gespeichert wird.



Anders als in der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht werden in der **Dependencies**-Ansicht immer die Natural-Objektnamen angezeigt (nicht die alternativen Dateinamen, die bis zu 255 Zeichen lang sein können). Bei Subprogrammen, Funktionen und DDMs werden jedoch immer die so genannten Langnamen angezeigt (d. h. der Name, der im Quellcode selbst definiert ist, nicht der Name, unter dem der Quellcode gespeichert wurde).

- **Anzeige von aktiven und passiven Cross-Referenzen**
  Sie können die Informationen in der **Dependencies**-Ansicht in zwei verschiedenen Modi anzeigen lassen:

Symbol	Modus	Beschreibung
	Aktive Cross-Referenzen (Aufgerufene/Callees)	Zeigt die Objekte an, die im aktuellen Quellcode referenziert werden.
	Passive Cross-Referenzen (Aufrufer/Callers)	Zeigt die Objekte an, in denen der aktuelle Quellcode referenziert wird.

  Sie können zwischen diesen Modi umschalten, indem Sie das entsprechende Symbol in der lokalen Symbolleiste oder das entsprechende Kommando (**Active XRefs** oder **Passive XRefs**) aus dem Kontextmenü wählen.

- **Öffnen von Objekten**
  Wenn Sie ein Objekt in der **Dependencies**-Ansicht auswählen, können Sie das Kommando **Open** aus dem Kontextmenü wählen. Oder Sie können auf das Objekt doppelklicken. Dadurch wird das Objekt in dem entsprechenden Editor geöffnet.

  Wenn Sie in den Einstellungen des Quelltexteditors die Option **Mark occurrences of the selected called object in the current source** (Ausprägungen des ausgewählten aufgerufenen Objekts im aktuellen Quelltext markieren) aktiviert haben, wird der Cursor auf die erste Ausprägung des entsprechenden aufgerufenen Objekts positioniert und alle Referenzen werden markiert.

  Wenn Sie einen Verweis auf ein Objekt in Ihren Code eingefügt haben (z.B. wenn Sie ein `INCLUDE` *`object-name`*-Statement hinzugefügt haben) und das referenzierte Objekt noch nicht in Ihrem Arbeitsbereich verfügbar ist, wird das Objekt in der **Dependencies**-Ansicht wie folgt angezeigt (dieses Beispiel geht davon aus, dass Sie "TEST" als Objektnamen angegeben haben):

```
<Unknown>:TEST
```

  Wenn Sie das Kommando **Open** (oder einen Doppelklick) bei einem unbekannten Objekt verwenden, erscheint ein Dialogfeld mit der Frage, ob Sie das Objekt vom Natural-Server herunterladen möchten. Wenn Sie zustimmen, versucht NaturalONE, eine Verbindung zum Server herzustellen (unter Verwendung der im zugehörigen Natural-Projekt gespeicherten Verbindungseigenschaften). Wenn die Verbindung hergestellt werden kann, wird das Objekt heruntergeladen und anschließend im entsprechenden Editor geöffnet.

  Anmerkung: Über die Option **Handling of missing objects** (Behandlung fehlender Objekte) in den Natural-Voreinstellungen können Sie festlegen, dass bestimmte Typen unbekannter Objekte dem Parser automatisch zur Verfügung gestellt werden. Siehe *Natural > Project* unter *Setting the Preferences* (Festlegen der Voreinstellungen) in der *NaturalONE*-Dokumentation.

- **Alphabetisch sortieren**
  Sie können die Objekte in der Dependencies-Ansicht alphabetisch sortieren, indem Sie das Symbol (Alphabetisch sortieren) in der lokalen Symbolleiste benutzen. Die Sortierreihenfolge ist zuerst der Library-Name, dann der Objektname.

- **Kopieren von Knoten als Text in die Zwischenablage**
  Wenn Sie einen Knoten in der **Dependencies**-Ansicht auswählen, können Sie das Kommando **Copy** aus dem Kontextmenü wählen. Oder Sie drücken STRG+C. Dadurch werden der ausgewählte Knoten und alle seine sichtbaren Unterknoten als Text in die Zwischenablage kopiert, abhängig vom aktuellen Modus (aktive oder passive Cross-Referenzen). Unsichtbare Knoten (d.h. Knoten, die nicht aufgeklappt sind) werden nicht kopiert. Sie können dann den Text aus der Zwischenablage in jede andere Anwendung (z. B. einen Texteditor) einfügen, indem Sie die Einfügefunktion der Zielanwendung verwenden (z. B. CTRL+V).

  Zusätzlich zu den Angaben in der **Dependencies**-Ansicht enthält der kopierte Text auch die Dateierweiterungen der Natural-Objekte. Die Dateierweiterungen werden in Klammern hinter den Natural-Objektnamen angezeigt. Im Falle von Subprogrammen, Functions und DDMs werden sie hinter den Langnamen der Objekte angezeigt.

- **Exportieren von Knoten als Text in eine Datei**
  Sie können den Inhalt der **Dependencies**-Ansicht auch als Text in eine Datei exportieren, abhängig vom aktuellen Modus (aktive oder passive Cross-Referenzen). Diese Datei wird in einem Projekt Ihres Eclipse-Arbeitsbereichs gespeichert. In diesem Fall ist es nicht erforderlich, dass Sie zuerst einen Knoten auswählen. Alle sichtbaren Knoten werden in die Datei geschrieben. Unsichtbare Knoten (d.h. Knoten, die nicht aufgeklappt sind) werden nicht geschrieben. Um den Export zu starten, wählen Sie das Symbol (als Text exportieren) in der lokalen Symbolleiste. Wählen Sie im daraufhin angezeigten Dialogfeld **Save As** (Speichern unter) das Projekt aus, in dem Sie die Datei speichern möchten (Sie können auch einen beliebigen Ordner in diesem Projekt auswählen), geben Sie einen Dateinamen ein und wählen Sie dann die Schaltfläche **OK**. Wenn der angegebene Dateiname am gewählten Zielort bereits existiert, wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, ob Sie die vorhandene Datei ersetzen möchten.

  Zusätzlich zu dem, was in der **Dependencies**-Ansicht angezeigt wird, enthält der exportierte Text auch die Dateierweiterungen der Natural-Objekte. Die Dateierweiterungen werden in Klammern hinter den Natural-Objektnamen angezeigt. Im Falle von Subroutinen, Functions und DDMs werden sie nach den Langnamen der Objekte angezeigt.

Verwendung von Content Assist (Inhaltshilfe)

Content Assist ist eine Standardfunktion in Eclipse. In NaturalONE wird eine grundlegende Content-Assist-Unterstützung bereitgestellt. Sie bietet Natural Syntax- und Variablenunterstützung, basierend auf dem aktuellen Kontext der Position innerhalb des Quellcodes. Diese Funktion verwendet die Vorlagen (Templates), die in den Natural-Voreinstellungen definiert sind. Siehe *Source Editor > Templates* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Um die Inhaltshilfe aufzurufen:

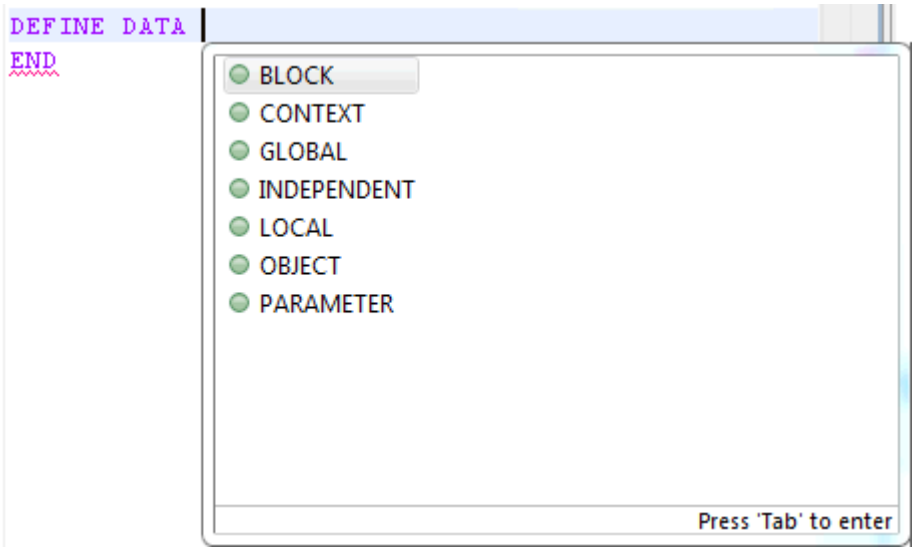
1 Geben Sie ein Schlüsselwort oder einen Teil davon ein und drücken Sie die Tastenkombination STRG+LEERTASTE.

Oder:

Geben Sie nichts ein und drücken Sie STRG+LEERTASTE in einer leeren Zeile.

Es erscheint ein Fenster mit einer Reihe von Auswahlmöglichkeiten, aus denen Sie den Code auswählen können, der in Ihr Programm aufgenommen werden soll. Die angezeigten Auswahlmöglichkeiten hängen vom Kontext ab und können z. B. Variablennamen, Statements, Programmnamen oder Vorlagen (Templates) sein.

Wenn Sie z. B. „DEFINE DATA “ (einschließlich des Leerzeichens nach den Buchstaben) eingeben, werden alle gültigen Schlüsselwörter angezeigt.



2 Wählen Sie den Code, den Sie in Ihr Programm einfügen möchten.

Weitere Informationen zur Inhaltshilfe finden Sie in der Eclipse-Online-Hilfe.

Benutzung der kontextabhängigen Hilfe

Zusätzlich zur Inhaltshilfe bietet der Source Editor kontextabhängige Hilfe für die folgenden Syntaxelemente:

- Statements (z.B. `WRITE`).
- Systemvariablen (z.B. `*USER`).
- Systemfunktionen (z.B. `COUNT`).
- Parameter (z.B. `AD`).

Um die kontextabhängige Hilfe aufzurufen:

1 Bewegen Sie die Einfügemarke im Source Editor auf das Schlüsselwort innerhalb eines Syntaxelements, zu dem Sie Hilfe benötigen.

2 Drücken Sie F1.

Der entsprechende Abschnitt in der Dokumentation zur Programmiersprache Natural wird in der **Help**-Ansicht aufgelistet.

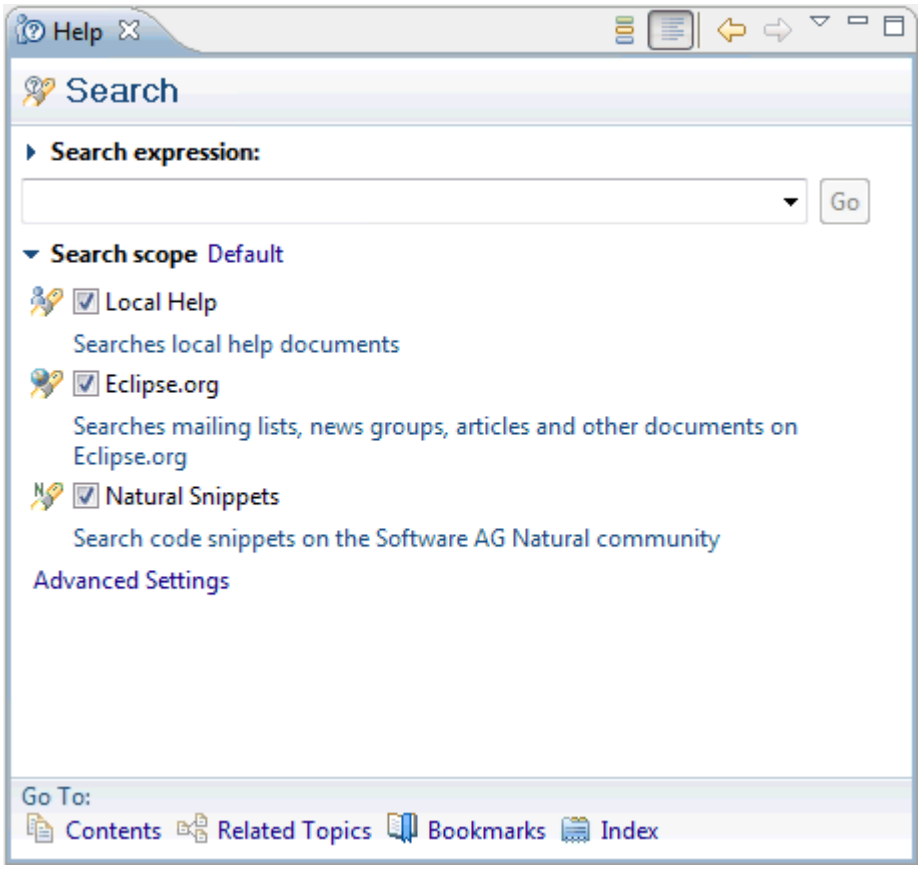
NaturalONE verwendet den Host-Typ, der aktuell auf der Seite **Runtime** der Projekteigenschaften definiert ist, um festzustellen, ob die Natural Language-Dokumentation für die Mainframe-, UNIX- oder Windows-Plattform angezeigt werden soll. Wenn der Host-Typ "<unknown>" ist, wird standardmäßig die Language-Dokumentation für den Mainframe angezeigt. Die Beschreibungen in der Natural-Language-Dokumentation können Verweise auf Teile der Natural-Dokumentation enthalten, die nicht im NaturalONE-Dokumentationssatz enthalten sind. Wenn Sie die referenzierten Seiten einsehen möchten, wechseln Sie zur Natural-Dokumentation unter *http://documentation.softwareag.com/* (Anmeldung bei Empower erforderlich).

Code Snippets aus der Natural Community verwenden

Das Natural Code Samples Forum der Natural-Community enthält Natural-Codebeispiele (auch „Code Snippets“ genannt), die Sie in Ihren eigenen Quellcode einbinden können. Die Codebeispiele wurden von Mitarbeitern, Kunden oder Partnern der Software AG geschrieben. Wenn Sie Codebeispiele haben, die Sie weitergeben möchten, können Sie dem Natural Code Samples Forum unter *http://techcommunity.softwareag.com/ecosystem/communities/public/adanat/products/natural/code-samples/* beitreten.

Sie können die Natural Community nach Code-Snippets durchsuchen - direkt von Eclipse aus. Dazu verwenden Sie die Standard-Suchfunktion der Eclipse-**Help**-Ansicht. NaturalONE fügt

automatisch einen **Natural Snippets**-Eintrag in den Suchbereich ein. Mit Copy-und-Paste können Sie Informationen aus einem gefundenen Code-Snippet in Ihren eigenen Quellcode kopieren.



Um nach Code-Snippets zu suchen:

1 Wählen Sie im Menü **Help** die Option **Search**.

2 Klicken Sie auf den Link **Search scope**, um den Abschnitt **Search scope** zu erweitern.

3 Stellen Sie sicher, dass das Kontrollkästchen neben dem Eintrag **Natural Snippets** aktiviert ist.

> **Tipp:** Wenn Sie nur Natural Snippets im Suchergebnis anzeigen lassen möchten, deaktivieren Sie alle anderen Kontrollkästchen im Abschnitt **Search scope**.

4 Geben Sie in das Textfeld **Search expression** die Zeichenfolge ein, nach der Sie suchen möchten.

Wenn Sie beispielsweise nach Code-Snippets suchen, die sich mit dem Statement `REQUEST DOCUMENT` befassen, können Sie die Zeichenfolge "request" eingeben.

5 Wählen Sie die Schaltfläche **Go**.

Wenn Code Snippets gefunden werden, werden sie unter der Überschrift **Natural Snippets** aufgelistet. Die Anzahl der gefundenen Treffer wird neben dieser Überschrift angezeigt.

6 Klicken Sie auf die Überschrift eines Suchergebnisses.

Dadurch wird das Code Snippet in der **Help**-Ansicht geöffnet.

7 Kopieren Sie den gesamten oder einen Teil des Codes und fügen Sie ihn in Ihren eigenen Quellcode ein.

Um die Suchparameter zu ändern:

1 Wählen Sie im **Help**-Menü die Option **Search**.

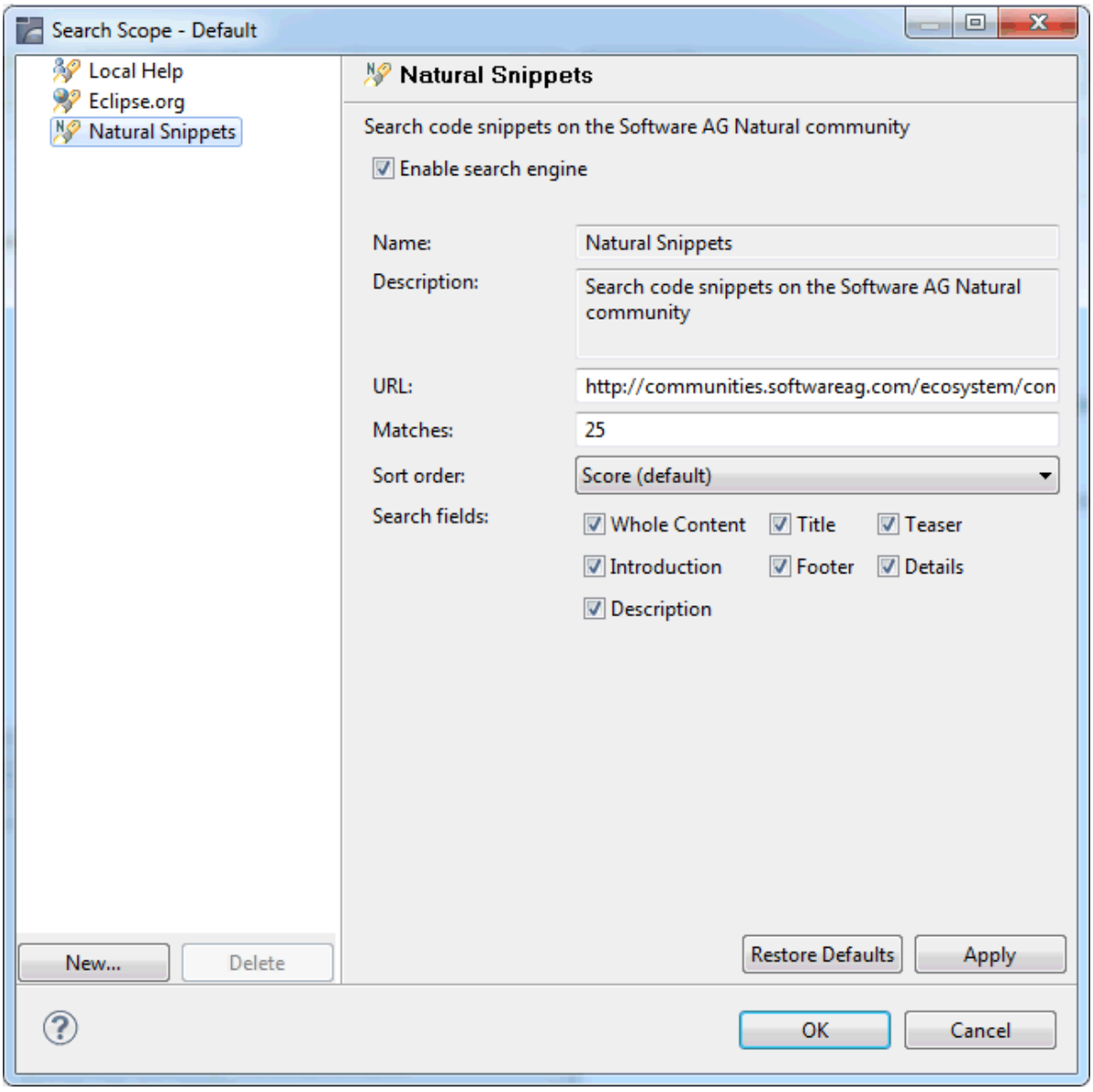
2 Klicken Sie auf den Link **Search scope**, um den Abschnitt **Search scope** zu erweitern.

3 Klicken Sie auf den Link **Advanced Settings** (Erweiterte Einstellungen).

Das Dialogfeld **Search Scope** wird angezeigt.

4 Wählen Sie in dem Baum auf der linken Seite **Natural Snippets**.

Die folgenden Informationen werden angezeigt:



5 Stellen Sie sicher, dass das Kontrollkästchen **Enable search engine** (Suchmaschine aktivieren) aktiviert ist.

Vorsicht: Ändern Sie die angegebene URL nicht. Andernfalls können Sie nicht nach Code Snippets suchen. Über die Schaltfläche **Restore Defaults** (Standardwerte wiederherstellen) können Sie jederzeit zu den Standardeinstellungen zurückkehren.

6 Geben Sie im Textfeld **Matches** (Übereinstimmungen) die maximale Anzahl der Code Snippets an, die zurückgegeben werden sollen. Es wird empfohlen, nicht mehr als 50 Code Snippets zurückzugeben.

7 Wählen Sie im Dropdown-Listenfeld **Sort order** (Sortierreihenfolge) die Reihenfolge aus, in der die Suchergebnisse sortiert werden sollen. Standardmäßig wird die Reihenfolge "Score" verwendet.

8 Aktivieren Sie das Kontrollkästchen bei jedem Suchfeld, das bei der Suche berücksichtigt werden soll, oder deaktivieren Sie das Kontrollkästchen bei jedem Suchfeld, das bei der Suche nicht berücksichtigt werden soll.

9 Wählen Sie die Schaltfläche **OK**.

Arbeiten mit Tasks (Aufgaben)

Wenn Sie mit dem Source Editor arbeiten und Erinnerungen setzen möchten (z. B. für Programmierschritte, die noch zu erledigen sind), können Sie zu Ihrem Quellcode Tasks hinzufügen. Die Aufgaben werden auch in der Wenn Sie mit dem Source Editor arbeiten und Erinnerungen setzen möchten (z. B. für Programmierschritte, die noch zu erledigen sind), können Sie zu Ihrem Quellcode Tasks hinzufügen. Die Tasks werden auch in der Task-Ansicht angezeigt. Sie können Tasks auf verschiedene Arten hinzufügen, wie unten beschrieben. Weitere Informationen zu Tasks finden Sie in der Online-Hilfe von Eclipse.

Folgenden Themen werden behandelt:

- Hinzufügen eines Task-Tag zum Quellcode
- Hinzufügen eines Task über ein Dialogfeld
- Benutzung der Tasks-Ansicht

Hinzufügen eines Task-Tag zum Quellcode

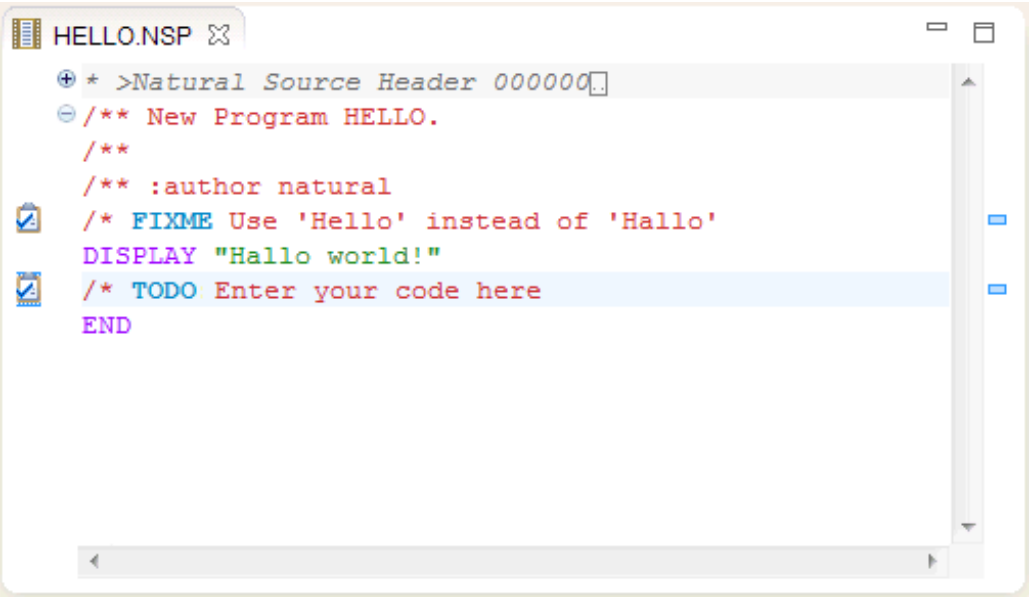
NaturalONE verwendet die Task-Tags, die für den Java-Compiler definiert sind (**Preferences > Java > Compiler > Task Tags**). Die Standard-Java-Task-Tags sind `TODO` und `FIXME`, aber Sie können auch Ihre eigenen Task-Tags definieren. Siehe auch die Seite **Parser > Task Tags** in den Natural-Voreinstellungen.

Anmerkung: Im Unterschied zum Java-Compiler unterstützt NaturalONE keine projektspezifischen Konfigurationen.

Die Farbe für die in den Natural-Quellcodes verwendeten Task-Tags kann in den Natural-Voreinstellungen geändert werden. Siehe *Source Editor > Syntax Coloring* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Um ein Task-Tag zum Quellcode hinzuzufügen:

1 Geben Sie im Source Editor einen Kommentar ein, der ein Task-Tag enthält, gefolgt von einem beliebigen Text. Beispiel:



2 Speichern Sie Ihre Änderungen, um das neue Task-Tag in der Tasks-Ansicht anzuzeigen. Eine solche Task ist vom Typ "Natural Task".

Hinzufügen eines Task über ein Dialogfeld

Wenn Sie kein sichtbares Task-Tag in Ihren Quellcode einfügen möchten, können Sie ein Task über ein Dialogfeld hinzufügen. In diesem Fall wird nur das Task-Tag auf der linken Seite des Source Editors angezeigt.

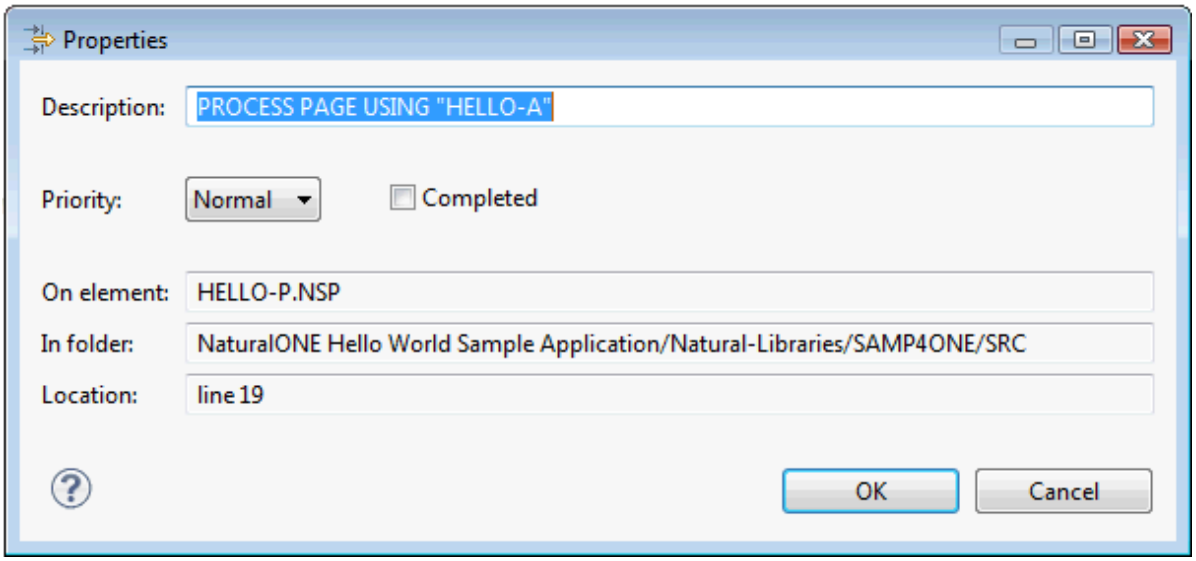
Um ein Task über ein Dialogfeld hinzuzufügen:

1 Positionieren Sie den Mauszeiger im Source Editor auf der Markierungsleiste, direkt neben der Zeile, bei die Sie ein Task hinzufügen möchten.

Anmerkung: Die Markierungsleiste (Marker Bar) ist der schmale graue Bereich ganz links im Source Editor.

2 Öffnen Sie das Kontextmenü für die Markierungsleiste und wählen Sie **Add Task**.

Es erscheint ein Dialogfenster.



Das Textfeld **Description** enthält zunächst den Text der ausgewählten Zeile. Dies ist der Text, der auch in der Tasks-Ansicht angezeigt wird. Sie können diesen Text in einen beliebigen Text ändern.

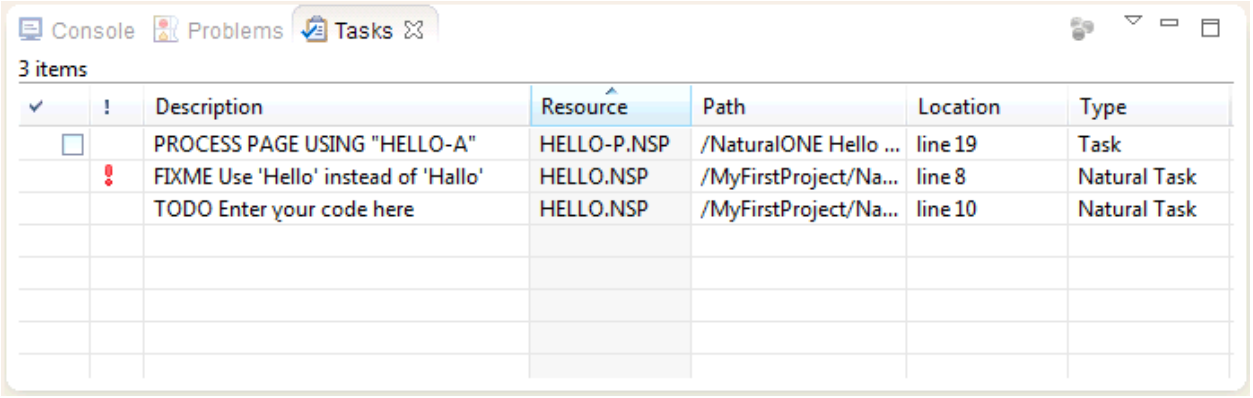
Der Dateiname, der Pfad und die Zeilennummer werden automatisch angegeben.

3 Wählen Sie die Schaltfläche **OK**.

Das Task wird nun in der Task-Ansicht angezeigt. Diese Aufgabe ist vom Typ "Task", d.h. es handelt sich um eine Eclipse-Standard-Task.

Benutzung der Tasks-Ansicht

Die **Tasks**-Ansicht ist eine Standardansicht von Eclipse. Sie ist Teil der NaturalONE-Perspektive. Wenn sie derzeit nicht angezeigt wird, können Sie sie mit **Window > Show View > Other > General > Tasks** wieder einblenden.



Wenn Sie in der **Tasks**-Ansicht auf ein Task doppelklicken, wird die entsprechende Datei automatisch geöffnet und die entsprechende Zeile ausgewählt. Weitere Informationen zur **Tasks**-Ansicht finden Sie in der Eclipse-Online-Hilfe.

Arbeiten mit Bookmarks (Lesezeichen)

Wenn Sie mit dem Source Editor arbeiten, können Sie ein Lesezeichen zu einer Quellcodezeile hinzufügen. Ein Lesezeichensymbol für diese Zeile wird dann in der Markierungsleiste angezeigt, und ein neuer Eintrag wird zur **Bookmarks**-Ansicht hinzugefügt. Die **Bookmarks**-Ansicht ermöglicht es Ihnen, schnell zu einer Datei zurückzukehren.

Die **Bookmarks**-Ansicht ist eine Standardansicht von Eclipse. Sie ist nicht Teil der NaturalONE-Perspektive. Sie können sie mit **Window > Show View > Other > General > Bookmarks** anzeigen.

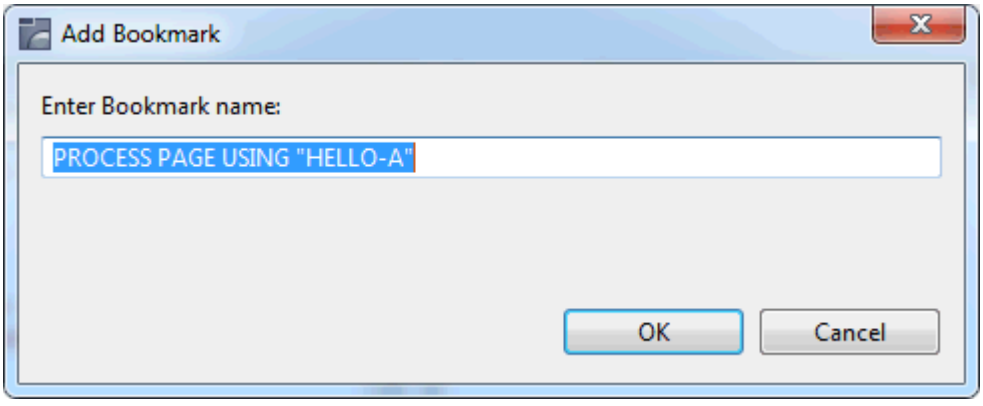
Um ein Lesezeichen für eine bestimmte Quellcodezeile hinzuzufügen:

1 Positionieren Sie den Mauszeiger im Source Editor auf der Markierungsleiste, direkt neben der Zeile, für die Sie ein Lesezeichen hinzufügen möchten.

> **Anmerkung:** Die Markierungsleiste ist der schmale graue Bereich ganz links im Source Editor.

2 Öffnen Sie das Kontextmenü für die Markierungsleiste und wählen Sie **Add Bookmark**.

Es erscheint ein Dialogfenster:



Das Textfeld enthält zunächst den Text aus der markierten Zeile. Dieser Text wird auch in der **Bookmarks**-Ansicht gezeigt. Sie können diesen Text in einen beliebigen Text ändern.

3 Wählen Sie die Schaltfläche **OK**.

Weitere Informationen zu Bookmarks finden Sie in der Eclipse-Online-Hilfe.

Fehlerbehandlung im Source Editor

Im Falle eines Syntaxfehlers enthält das vertikale Lineal auf der linken Seite des Editors eine rote Fehlermarkierung, die die Fehlerposition anzeigt. Es können mehrere Fehler in einem Quellcode vorkommen. Der inkrementelle Parser ist in der Lage, unterschiedliche Fehlersituationen zu behandeln und parst den Code bis zum Ende des Quellcodes. Wenn der Quellcode mehrere Syntaxfehler enthält, können Sie mit einer der roten Fehlermarkierungen im Übersichtslineal auf der rechten Seite des Editors direkt zu einer anderen Fehlerstelle wechseln.

Die allgemeinen Voreinstellungen für die Annotationen (**Preferences > General > Editors > Text Editors > Annotations**) gelten auch für NaturalONE. So erscheinen bei entsprechender Konfiguration z.B. geschlängelte Linien unter dem Wort, das den Fehler verursacht, und eine Anmerkung wird neben der betroffenen Zeile angezeigt. Nachdem ein fehlerhaftes Programm gespeichert wurde, werden die Fehler dauerhaft in der **Problems**-Ansicht angezeigt.

Ein Tooltip mit der Natural-Fehlernummer und dem Fehlertext erscheint, wenn Sie den Mauszeiger über eine Anmerkung bewegen.

Siehe auch *Probleme in Ihren Natural-Quellcodes*.

Zu einer bestimmten Natural-Zeilennummer springen

Das Kommando **Go to Natural Line** ähnelt dem Eclipse-Standardkommando **Go to Line**. Das Kommando **Go to Line** bezieht sich auf die Eclipse-Zeilennummern. Der Befehl **Go to Natural Line** hingegen bezieht sich auf die vierstelligen Natural-Zeilennummern, die auf einem Natural-Server verwendet werden. Da im Eclipse-Arbeitsbereich ein Source Header vorhanden ist und auf einem Natural-Server unterschiedliche Inkremente verwendet werden können, sind die Natural-Zeilennummern nicht identisch mit den Eclipse-Zeilennummern. Siehe auch *Line Numbers*.

Das Kommando **Go to Natural Line** ist z.B. dann hilfreich, wenn in der Produktionsumgebung ein Laufzeitfehler auftritt und sich die entsprechenden Natural-Quellcodes in einem Eclipse-Workspace befinden. Sie können die Informationen nutzen, die mit dem Laufzeitfehler geliefert wurden (also den Objektnamen und die vierstellige Natural-Zeilennummer): Öffnen Sie den entsprechende Natural-Quellcode in Ihrem Workspace und verwenden Sie dann das Kommando **Go to Natural Line**, um im Source Editor zu der entsprechenden Zeile zu springen.

Um zu einer bestimmten Natural-Zeilennummer zu springen:

1 Öffnen Sie den Quellcode, der die gewünschte Zeile enthält, im Source Editor.

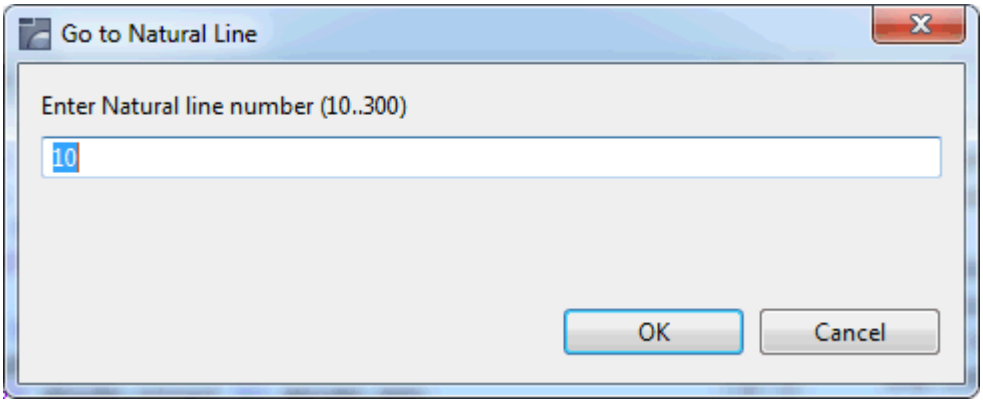
2 From the **Navigate** menu, choose **Go to Natural Line**.

Wählen Sie aus dem **Navigate**-Menü das Kommando **Go to Natural Line**.

Oder:

Drücken Sie CTRL+G.

Das Dialogfeld **Go to Natural Line** wird angezeigt.



3 Geben Sie die Natural-Zeilennummer ein.

Sie müssen nicht alle vier Ziffern der Natural-Zeilennummer angeben, führende Nullen können weggelassen werden.

4 Wählen Sie die Schaltfläche **OK**.

Anmerkung: Die Schaltfläche **OK** ist nicht aktiviert, wenn die angegebene Natural-Zeilennummer in der aktuellen Quelle nicht vorhanden ist oder wenn ein anderes Zeichen als eine Zahl angegeben wurde.

Öffnen von referenzierten Objekten und Springen zu Variablen und internen Subroutinen-Definitionen

Gilt nur für den lokalen Modus (d. h., wenn die Quellcodes in einem Projekt im Eclipse-Arbeitsbereich verfügbar sind).

Sie können ein referenziertes Objekt direkt vom Source Editor aus öffnen. Wenn Ihr Quellcode zum Beispiel das folgende Statement enthält, können Sie direkt den lokalen Datenbereich mit dem Namen "LDA01" öffnen:

```
LOCAL USING LDA01
```

Wenn das referenzierte Objekt noch nicht im Eclipse-Arbeitsbereich verfügbar ist (d. h. wenn es in der **Dependencies**-Ansicht als "<Unknown>" aufgeführt ist), wird ein Dialogfeld mit der Frage angezeigt, ob Sie das Objekt vom Server herunterladen möchten.

Anmerkung: Sie können ein referenziertes Objekt auch aus der **Dependencies**-Ansicht öffnen.

Neben dem Öffnen von referenzierten Objekten können Sie auch zu Variablendefinitionen springen und Hyperlinks verwenden.

Um ein referenziertes Objekt zu öffnen:

1 Setzen Sie im Source Editor den Cursor auf den Namen des referenzierten Objekts.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Open**.

Oder:

Drücken Sie F3.

Das referenzierte Objekt wird nun in dem entsprechenden Editor geöffnet.

Um zu einer Variablendefinition zu springen:

1 Setzen Sie im Source Editor den Cursor auf den Namen der Variable.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Open**.

Oder:

Drücken Sie F3.

Wenn sich die Variablendefinition im `DEFINE DATA`-Block des aktuellen Objekts befindet, wird sie dort ausgewählt. Wenn sie sich in einem anderen Objekt befindet (z. B. in einem lokalen Datenbereich), wird das Objekt geöffnet und die Variablendefinition dort ausgewählt.

Um zu einer internen Subroutinendefinition zu springen:

1 Stellen Sie den Cursor im Source Editor auf den Namen der internen Subroutine.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Open**.

Oder:

Drücken Sie F3.

Der Subroutinenname der internen Subroutinendefinition (`DEFINE SUBROUTINE`-Statement) wird gewählt.

Um Hyperlinks zu verwenden:

1 Drücken Sie im Source Editor die Umschalttaste (standardmäßig ist das die STRG-Taste) und bewegen Sie die Maus über Ihren Quellcode.

Wenn **Hyperlinking** (Verwendung von Hyperlinks) aktiviert ist, wird ein Hyperlink angezeigt, wenn Sie die Maus über ein referenziertes Objekt oder einen Variablennamen bewegen.

Anmerkung: Standardmäßig ist das Hyperlinking in den allgemeinen Voreinstellungen aktiviert (**Preferences > General > Editors > Text Editors > Hyperlinking**). Für NaturalONE ist eine spezielle Option verfügbar: **Natural Element**.

2 Klicken Sie auf den Hyperlink.

Die entsprechende Aktion wird ausgeführt: Entweder wird ein referenziertes Objekt geöffnet oder Sie springen zu der Variablendefinition.

Wie der Sprung zu einer Variablen- oder Subroutinendefinition in einem Copycode realisiert wird

Wenn ein Copycode geöffnet ist, kann er nur ohne Kontext geparst werden. Um einen Sprung zu Variablen- und internen Subroutinendefinitionen zu realisieren, wird über die **Dependencies**-Ansicht die erste von möglicherweise mehreren möglichen Root-Sources ausgewählt, die diesen Copycode enthalten.

Diese ausgewählte Root-Source wird dann intern geparst. Anhand des Ergebnisses des internen Parsing wird die Position der Variablen- oder internen Subprogrammdefinition bestimmt.

Code-Fragmente externalisieren

Gilt nur für den lokalen Modus (d. h., wenn die Quellcodes in einem Projekt im Eclipse-Arbeitsbereich verfügbar sind).

Sie können ein Codefragment aus dem aktuellen Quellcode in ein neues Objekt in derselben Bibliothek verschieben. Dies ist hilfreich, wenn Sie dieses Stück Code wiederverwenden möchten. Das neue Objekt kann von folgendem Objekttyp sein: Subprogramm, Subroutine, Copycode oder Programm.

Solange Sie den Editor, aus dem Sie das Codefragment externalisiert haben, nicht schließen, können Sie Ihre letzte Externalisierung rückgängig machen/wiederholen (dies ist eine Standardfunktion von Eclipse). Dazu muss der Editor aktiv sein.

Um ein Codefragment zu externalisieren:

1 Wählen Sie im Source Editor das Codefragment aus, das Sie externalisieren möchten.

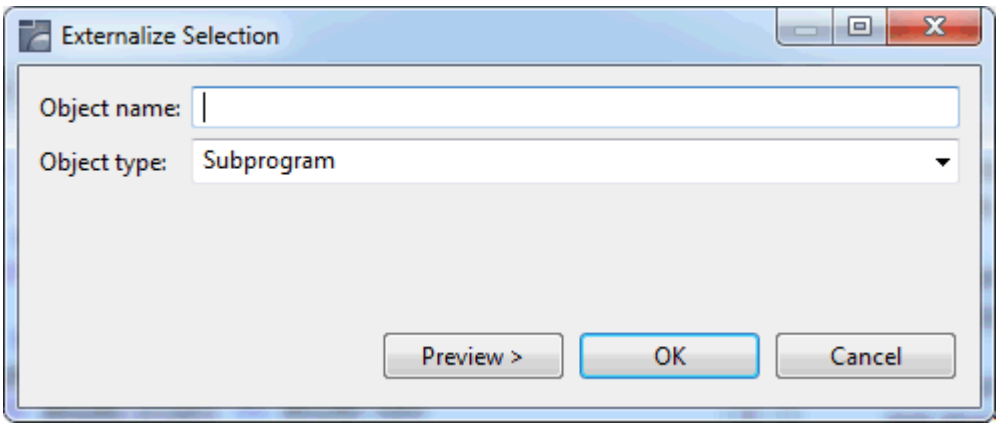
Anmerkung: Wenn Sie nicht alle Zeilen ausgewählt haben, die z. B. zu einer Schleife gehören, wird die Auswahl automatisch erweitert, um alle benötigten Zeilen aufzunehmen.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Advanced Edit Features > Externalize Selection**.

Oder:

Drücken Sie ALT+SHIFT+S.

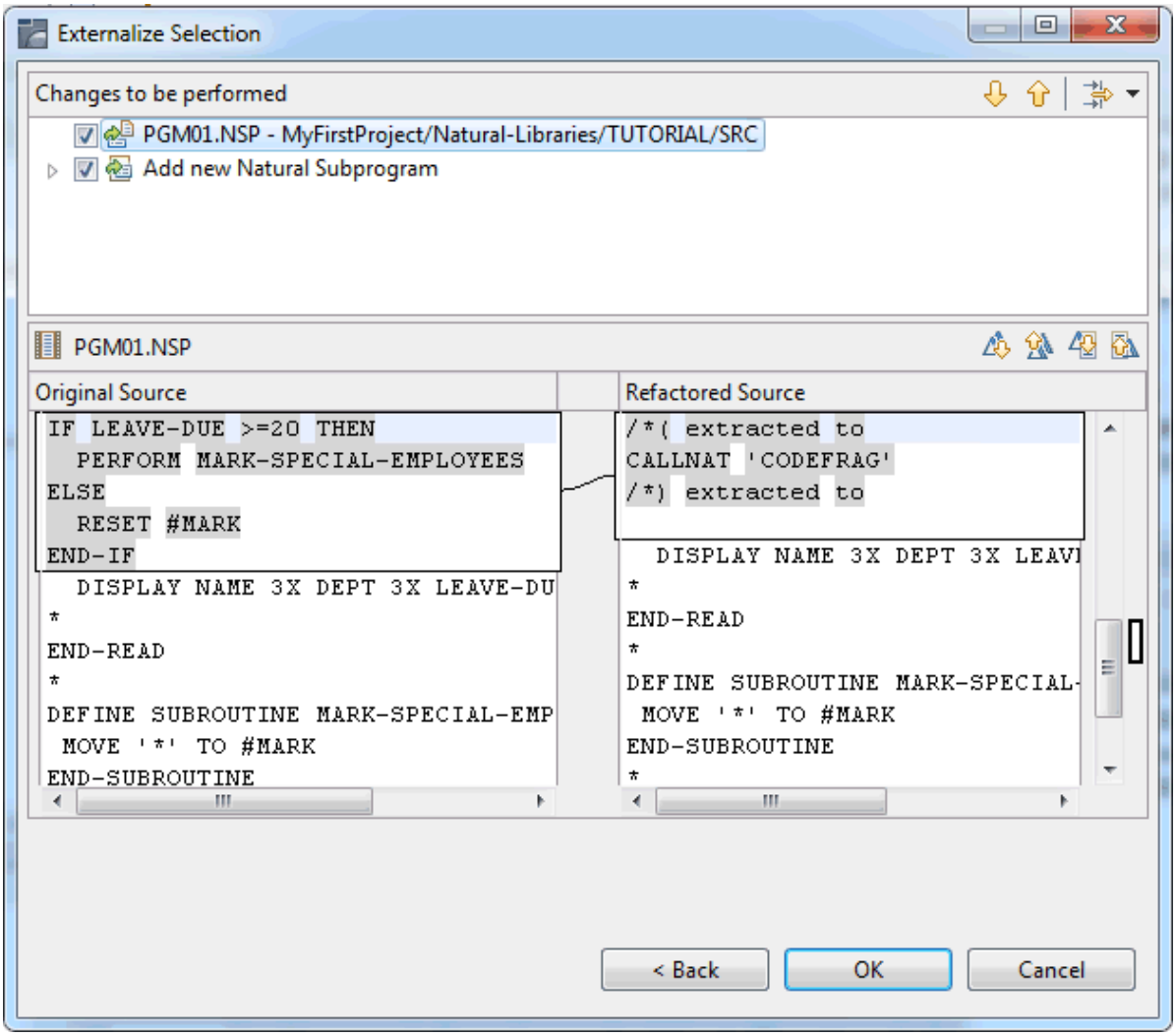
Ein Dialogfenster erscheint:



3 Geben Sie einen Namen für das neue Objekt an und wählen Sie einen Objekttyp.

4 Optional. Wählen Sie die Vorschau-Schaltfläche **Preview**.

Ein Dialogfenster erscheint:



In diesem Dialogfenster wird angezeigt, wie Ihr Coding geändert wird. An der Position des ausgewählten Codefragments wird der entsprechende Aufruf eingefügt. Wenn Sie z.B. Code in ein Subprogramm auslagern wollen, wird ein `CALLNAT`-Statement eingefügt. Außerdem wird ein Kommentar eingefügt, der anzeigt, dass der Code extrahiert wurde.

Anmerkung: Wenn Sie ein Kontrollkästchen am oberen Rand des Dialogfelds deaktivieren, wird die entsprechende Änderung nicht durchgeführt.

5 Wählen Sie die Schaltfläche **OK**.

Der ausgewählte Code wird aus dem aktuellen Quellcode entfernt, und ein neues Objekt, das diesen Code enthält, wird in der Projekt Explorer-Ansicht oder in der **Natural Navigator**-Ansicht angezeigt.

Einfügen von Call- oder Include-Statements

Gilt nur für den lokalen Modus (d. h., wenn die Quellcodes in einem Projekt im Eclipse-Arbeitsbereich verfügbar sind).

Sie können ein Dialogfenster aufrufen, in dem Sie ein Natural-Objekt angeben können, das in dem aktuellen Quellcode referenziert werden soll. Die entsprechende Anweisung wird dann automatisch in Ihren Quellcode eingefügt.

Die folgende Tabelle gibt einen Überblick über die verschiedenen Statements, die für die verschiedenen Objekttypen eingefügt werden:

Objekttyp	Statement

Adapter	`PROCESS PAGE USING "adapter-name"`
	`...`
	`END-DECIDE`

Copycode	`INCLUDE copycode-name`
Function	`function-name(< [parameter]... >)`
Global data area	`GLOBAL USING gda-name`
Local data area	`LOCAL USING lda-name`
Map	`INPUT USING MAP "map-name"`
Parameter data area	`PARAMETER USING pda-name`
Program	`FETCH RETURN "program-name"`
Subprogram	`CALLNAT "subprogram-name" [parameter]...`
Subroutine	`PERFORM subroutine-name [parameter]...`

Um ein Call- oder ein Include-Statement einzufügen:

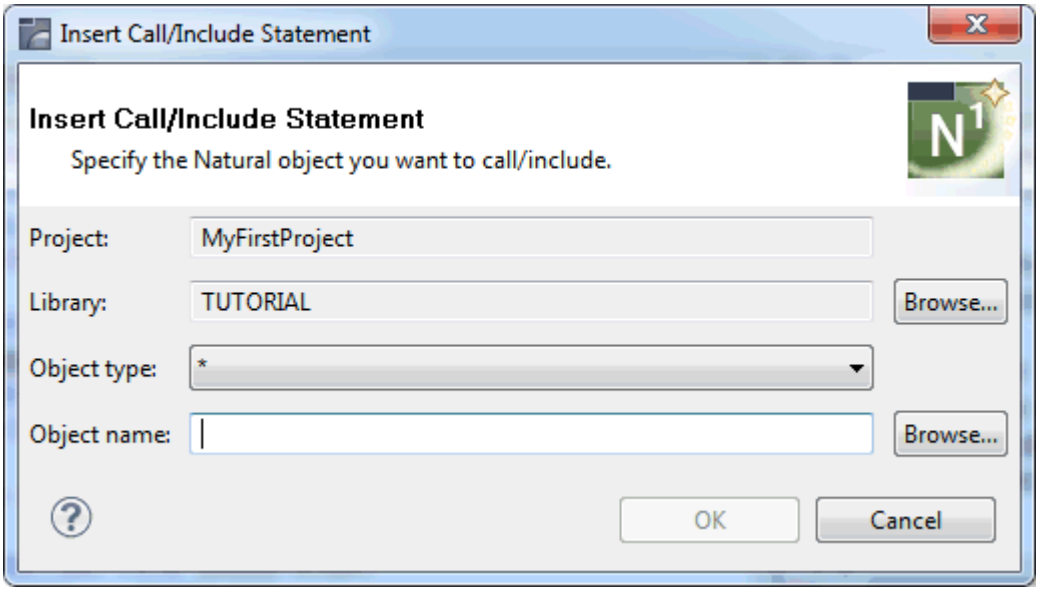
1 Positionieren Sie den Cursor im Source Editor an der Stelle, an der Sie die Objektreferenz einfügen möchten.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Advanced Edit Features > Insert Call/Include Statement**.

Oder:

Drücken Sie ALT+SHIFT+C.

Es erscheint das folgende Dialogfenster.



3 Optional. Benutzen Sie die Schaltfläche **Browse**, um die Library auszuwählen, die das Objekt enthält, das Sie referenzieren möchten.

4 Wählen Sie aus dem Dropdown-Listenfeld **Object type** den Typ des Objekts aus, das Sie referenzieren möchten.

Der Stern (*), der standardmäßig angezeigt wird, dient als Filter für das Dialogfeld **Select Natural Object** (Natural-Objekt auswählen), das Sie über die Schaltfläche **Browse** (Durchsuchen) für den Objektnamen aufrufen können. Wenn Sie die Schaltfläche **Browse** nicht benutzen möchten, ist es wichtig, dass Sie den richtigen Objekttyp für den von Ihnen angegebenen Objektnamen wählen. Andernfalls wird die Schaltfläche **OK** nicht aktiviert.

5 Geben Sie in das Textfeld **Object name** den Namen des Objekts ein, das Sie referenzieren möchten.

Wenn Sie die Schaltfläche **Browse** benutzen möchten, können Sie Platzhalterzeichen eingeben: ein Stern (*) für optional mehrere Zeichen und/oder ein Fragezeichen (?) für ein einzelnes Zeichen. Diese Angabe wird nur als Filter für das Dialogfeld **Select Natural Object** verwendet.

Oder:

Wählen Sie die Schaltfläche **Browse**, um das Objekt aus dem Dialogfeld **Select Natural Object** auszuwählen.

Wenn kein Natural-Objekt ausgewählt ist, werden im Dialogfeld **Select Natural Object** nur die Objekte angezeigt, die dem Typ entsprechen, der im Dropdown-Listenfeld **Object type** angezeigt wird (falls vorhanden). Wenn das Textfeld **Object name** (Objektname) außerdem Platzhalterzeichen enthält, werden im Dialogfeld **Select Natural Object** (Natural-Objekt auswählen) nur die Objekte angezeigt, die diesem Namensmuster entsprechen.

6 Wählen Sie die Schaltfläche **OK**.

Das entsprechende Statement wird an der Cursorposition eingefügt.

Importieren von Datenfeldern

Gilt nur für den lokalen Modus (d. h., wenn die Quellcodes in einem Projekt im Eclipse-Arbeitsbereich verfügbar sind).

Sie können ein Dialogfenster aufrufen, in dem Sie Datenfelddefinitionen aus einem anderen Objekt auswählen können, die in den aktuellen Quellcode importiert werden sollen.

Anmerkung: Die Datenfelddefinitionen sind nur für Natural-Objekte verfügbar, die ein `DEFINE DATA`-Statement enthalten.

Um Datenfelder zu importieren:

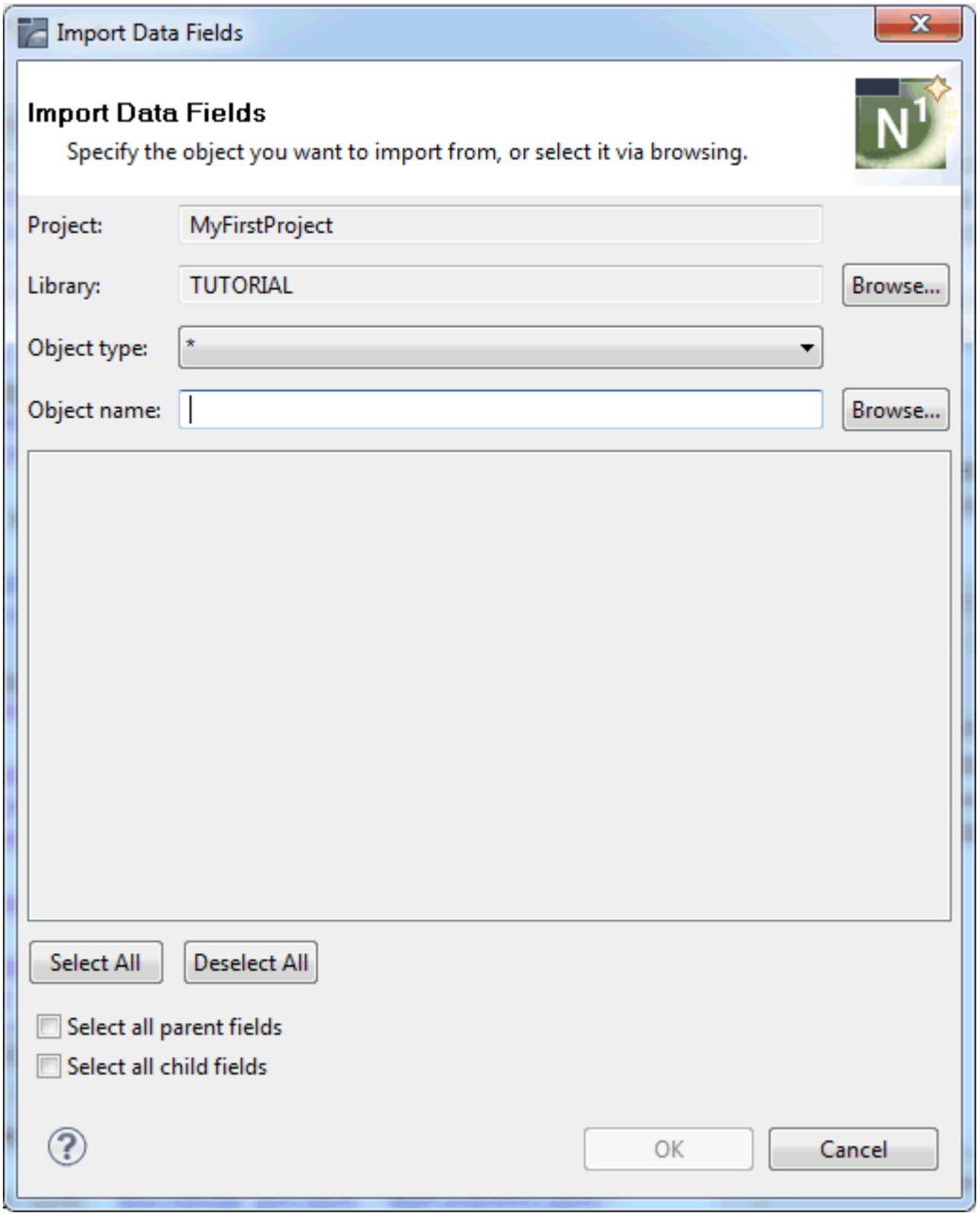
1 Positionieren Sie den Cursor im Source Editor an der Stelle, an der Sie das/die importierte(n) Datenfeld(er) platzieren möchten. Dies ist in der Regel die erste Position einer Zeile innerhalb des `DEFINE DATA`-Blocks.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Advanced Edit Features > Import Data Fields**.

Oder:

Drücken Sie ALT+SHIFT+I.

Es erscheint das folgende Dialogfenster.



3 Optional. Wählen Sie über die Schaltfläche **Browse** die Library aus, die das Objekt enthält, aus dem Sie Datenfelder importieren möchten.

4 Wählen Sie aus dem Dropdown-Listenfeld **Object type** den Typ des Objekts aus, aus dem Sie Datenfelder importieren möchten.

Der Stern (*), der standardmäßig angezeigt wird, dient als Filter für das Dialogfeld **Select Natural Object** (Natural-Objekt auswählen), das über die Schaltfläche **Browse** (Durchsuchen) für den Objektnamen aufgerufen werden kann. Wenn Sie die Schaltfläche **Browse** nicht benutzen möchten, ist es wichtig, dass Sie den richtigen Objekttyp für den von Ihnen angege-

benen Objektnamen wählen. Andernfalls werden die Felder, die importiert werden können, nicht angezeigt.

5 Geben Sie in das Textfeld **Object name** den Namen des Objekts ein, das Sie importieren möchten.

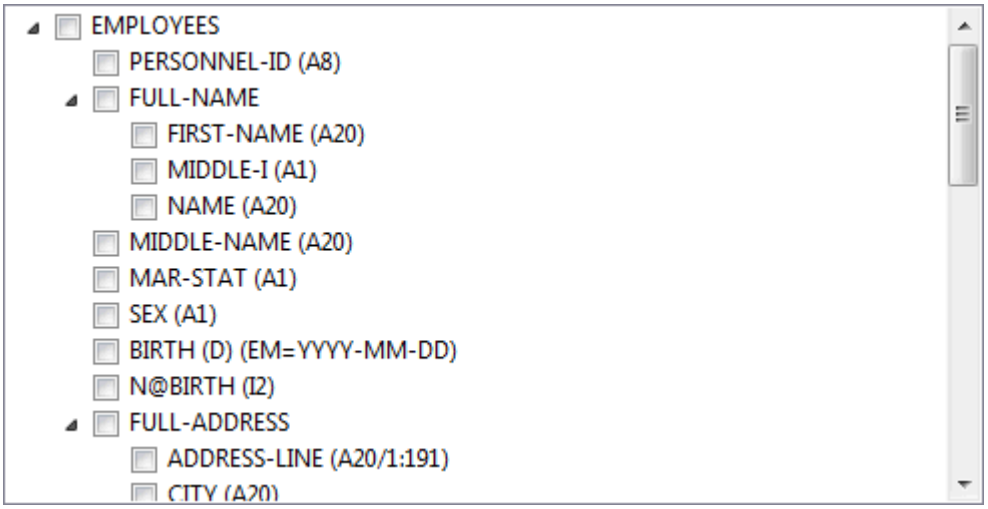
Wenn Sie die Schaltfläche Browse (Durchsuchen) benutzen möchten, können Sie Platzhalterzeichen eingeben: ein Stern (*) für optionale mehrere Zeichen und/oder ein Fragezeichen (?) für ein einzelnes Zeichen. Diese Angabe wird nur als Filter für das Dialogfeld **Select Natural Object** verwendet.

Oder:

Wählen Sie die Schaltfläche **Browse**, um das Objekt aus dem Dialogfeld **Select Natural Object** auszuwählen.

Wenn kein Natural-Objekt ausgewählt ist, werden im Dialogfeld **Select Natural Object** nur die Objekte angezeigt, die dem Typ entsprechen, der (falls vorhanden) im Dropdown-Listenfeld **Object type** angezeigt wird. Wenn das Textfeld **Object name** Platzhalterzeichen enthält, werden im Dialogfeld **Select Natural Object** nur die Objekte angezeigt, die diesem Namensmuster entsprechen.

Alle Datenfelder, die aus dem ausgewählten Objekt importiert werden können, werden im Dialogfeld **Import Data Fields** aufgelistet. Zum Beispiel:



6 Markieren Sie das Kontrollkästchen bei jedem Datenfeld, das Sie importieren möchten.

Wenn das Kontrollkästchen **Select all parent fields** markiert ist, werden automatisch alle übergeordneten Felder markiert, wenn Sie ein untergeordnetes Feld markieren.

Wenn das Kontrollkästchen **Select all child fields** aktiviert ist, werden automatisch alle untergeordneten Felder markiert, wenn Sie ein übergeordnetes Feld markieren.

Sie können auch die Schaltflächen **Select All** und **Deselect All** benutzen, um alle Kontrollkästchen zu markieren oder alle Markierungen rückgängig zu machen.

7 Wählen Sie die Schaltfläche **OK**.

Anmerkung: Diese Schaltfläche ist nur aktiviert, wenn mindestens ein zu importierendes Datenfeld markiert wurde.

Die ausgewählten Datenfelder werden in den Quellcode importiert.

Generierung von Zählerfeldern

Gilt nur für den lokalen Modus (d. h., wenn die Quellcodes in einem Projekt im Eclipse-Arbeitsbereich verfügbar sind).

Ein Zählerfeld (C*-Notation) wird verwendet, um die Anzahl der Ausprägungen eines multiplen Feldes oder einer Periodengruppe aus einer Adabas-Datenbank abzurufen. Das Zählerfeld `C*INCOME` gibt zum Beispiel die Anzahl der Ausprägungen für die Periodengruppe `INCOME` zurück. Weitere Informationen zum Referenzieren der internen Zählung für ein Datenbank-Array finden Sie im Leitfaden zur Programmierung.

Sie können Zählerfelder für Periodengruppen und multiple Felder erzeugen. Sie können dies in allen Objekten tun, in denen ein `DEFINE DATA`-Statement verwendet werden kann (z.B. in Programmen, Datenbereichen oder Verarbeitungsregeln in Maps).

Um ein Zählerfeld zu generieren:

1 Positionieren Sie den Cursor im Source Editor auf den Namen eines multiplen Feldes oder einer Periodengruppe innerhalb des `DEFINE DATA`-Blocks.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Advanced Edit Features > Generate Counter Field**.

Anmerkung: Dieses Kommando ist nur für ein multiples Feld oder eine Periodengruppe sichtbar.

Oder:

Drücken Sie ALT+SHIFT+G.

Eine neue Zeile wird in den `DEFINE DATA`-Block eingefügt, und zwar vor dem markierten multiplen Feld oder der Periodengruppe.

Die folgenden Beispiele zeigen, wo ein generiertes Zählerfeld auftritt.

- Periodengruppe:

```
DEFINE DATA LOCAL
   1 EMP-VIEW VIEW OF EMPLOYEES
     2 NAME
     2 CITY
     2 C*INCOME
     2 INCOME (1:12)
       3 SALARY
       ...
```

- Multiples Feld::

```
DEFINE DATA LOCAL
   1 EMP-VIEW VIEW OF EMPLOYEES
     2 NAME
     2 C*LANG
     2 LANG (1:3)
     ...
```

- Multiples Feld innerhalb einer Periodengruppe:

```
DEFINE DATA LOCAL
   1 EMP-VIEW VIEW OF EMPLOYEES
     2 NAME
     2 C*BONUS (1:12)
     2 INCOME (1:12)
       3 SALARY
       3 BONUS (1:2)
       ...
```

In diesem Fall wird der Zähler direkt über der Periodengruppe generiert, und zwar mit dem gleichen Index wie die Periodengruppe.

Der Index der Periodengruppe wird immer verwendet, auch dann wenn die Periodengruppe nicht innerhalb der View definiert ist.

```
DEFINE DATA LOCAL
   1 EMP-VIEW VIEW OF EMPLOYEES
     2 NAME
     2 C*BONUS (1:12)
     2 BONUS (1:12,1:2)
     ...
```

Hinzufügen und Entfernen von Kommentaren

Sie können mehrere Zeilen in Ihrem Quellcode auf einmal als Kommentarzeilen markieren, oder Sie können die Kommentarzeichen von mehreren Zeilen entfernen.

Um mehrere Zeilen als Kommentarzeilen zu markieren:

1 Markieren Sie im Source Editor alle Zeilen, die Sie in Kommentarzeilen umwandeln wollen.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Add Natural Comment Lines**.

Oder:

Drücken Sie CTRL+ALT+C.

Alle markierten Zeilen werden in Kommentarzeilen umgewandelt, wobei jede Kommentarzeile mit einem Stern (*), gefolgt von einem Leerzeichen, beginnt.

Um die Kommentarzeichen aus mehreren Zeilen zu entfernen:

1 Markieren Sie im Source Editor alle Kommentarzeilen, aus denen Sie die Kommentarzeichen entfernen möchten.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Remove Natural Comment Lines**.

Oder:

Drücken Sie CTRL+ALT+U.

Die Kommentarzeichen (* gefolgt von einem Leerzeichen, ** oder /*) werden von den Anfängen aller markierten Zeilen entfernt.

Anmerkung: Wenn Sie einen Teil einer Zeile auswählen, wird die Auswahl automatisch erweitert, um die gesamte Zeile einzuschließen.

Umschalten zwischen Kommentieren und Entkommentieren

Sie können mehrere Zeilen in Ihrem Quellcode markieren und zwischen dem Kommentieren und Entkommentieren der markierten Zeilen hin- und herschalten.

Um Kommentarzeilen umzuschalten:

1 Markieren Sie im Source Editor alle Zeilen, in denen Sie zwischen Kommentieren und Nicht-Kommentieren umschalten wollen.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Toggle Natural Comment Lines**.

Oder:

Drücken Sie CTRL+7.

Alle markierten Zeilen werden in Kommentarzeilen umgewandelt.

3 Wenn Sie bei denselben markierten Zeilen erneut CTRL+7 drücken, machen Sie die vorherige Aktion rückgängig und der ursprüngliche Status der markierten Zeilen wird wieder angezeigt.

Anmerkung: Wenn Sie einen Teil einer Zeile auswählen, wird die Markierung automatisch so erweitert, dass sie die gesamte Zeile umfasst.

Schreibschutz für Quellcodezeilen

Sie können eine oder mehrere Zeilen in Ihrem Quellcode durch die Verwendung von Read-Only-Tags schützen (nur Lesen). Geschützte Quellcodezeilen können nicht bearbeitet werden.

Schreibgeschützte Zeilen haben eine spezielle Hintergrundfarbe. Diese Hintergrundfarbe kann in den Natural-Voreinstellungen geändert werden. Siehe *Syntax Coloring* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Um eine einzelne Zeile als schreibgeschützt zu markieren:

- Geben Sie am Ende der zu schützenden Zeile ein Leerzeichen ein und fügen Sie das folgende Tag an:

```
/*<RO>
```

Um einen Codeblock als schreibgeschützt zu markieren:

1 Geben Sie am Ende der Zeile, in der der zu schützende Block beginnt, ein Leerzeichen ein und fügen Sie das folgende Tag an:

```
/*<RO>>
```

2 Geben Sie am Ende der Zeile, in der der zu schützende Block endet, ein Leerzeichen ein und fügen Sie das folgende Tag an:

```
/*<<RO>
```

Geschützte Zeilen in Quellcodes, die mit Construct oder Code Generation generiert wurden

Wenn die erste Zeile eines Quellcodes (d. h. die Zeile direkt unter dem Natural-Quellcode-Header) mit `**SAG GENERATOR` beginnt, bedeutet dies, dass der Quellcode mit Construct oder der Code-Generierungskomponente von NaturalONE generiert wurde. Die meisten Zeilen eines solchen Quellcodes sind geschützt und können nicht bearbeitet werden. Es gelten die folgenden Regeln:

- Sie können die Zeilen bearbeiten, die mit `**SAG` beginnen.
- Sie können die Zeilen zwischen `**SAG DEFINE EXIT` und `**SAG END-EXIT` bearbeiten.
- Alle anderen Zeilen sind schreibgeschützt.

Der Schutz kann in den Natural-Voreinstellungen aktiviert oder deaktiviert werden. Siehe *Protection* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Umwandeln in Groß- oder Kleinbuchstaben

Wenn die Groß-/Kleinschreibung in den Voreinstellungen von Natural aktiviert ist, können Sie den Quellcode von Großbuchstaben in Kleinbuchstaben bzw. von Kleinbuchstaben in Großbuchstaben umwandeln. Siehe *Case Translation* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Die Umwandlung von Groß- und Kleinschreibung kann entweder manuell mit den unten beschriebenen Kommandos erfolgen oder (wenn die entsprechende Option in den Voreinstellungen aktiviert ist) automatisch, wenn Sie Ihren Quellcode speichern oder ablegen.

Um Text von Klein- in Großbuchstaben umzuwandeln:

1 Markieren Sie im Source Editor den Text, den Sie umzuwandeln möchten.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Upper Case**.

Oder:

Drücken Sie ALT+U.

Um Text von Großbuchstaben in Kleinbuchstaben umzuwandeln:

1 Markieren Sie im Source Editor den Text, den Sie umzuwandeln möchten.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Lower Case**.

Oder:

Drücken Sie ALT+L.

Anmerkungen:

1. Wenn Sie den gesamten Quellcode markieren und dann eines der oben genannten Kommandos wählen, werden nur die Teile des Quellcodes in eine andere Schreibung übersetzt, für die die entsprechende Option in den Voreinstellungen aktiviert wurde.
2. Wenn Sie z. B. nur eine Zeichenkette in einfachen Anführungszeichen auswählen und die entsprechende Option in den Voreinstellungen nicht aktiviert ist, werden die oben genannten Kommandos nicht aktiviert.

Einrücken der Quellcodezeilen

Der Natural Source Editor verwendet den Standard-Quelltextformatierer von Eclipse. Es ist jedoch möglich, die Quellcodezeilen so einzurücken, dass die Einrückung die Struktur des Programms widerspiegelt. Diese Funktion entspricht dem Natural-Systemkommando `STRUCT`.

Die Anzahl der Positionen, um die eine Zeile eingerückt wird, wird in den Natural-Voreinstellungen festgelegt. Siehe in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Die folgenden Statement-Typen sind von einer strukturellen Einrückung betroffen:

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

Sie können Abschnitte Ihres Quellcodes von der strukturellen Einrückung ausnehmen, indem Sie die speziellen Statements `/*STRUCT OFF` und `/*STRUCT ON` verwenden. Diese Statements müssen am Anfang einer Quellcodezeile eingegeben werden. Die Quellcodezeilen zwischen diesen beiden Anweisungen bleiben so, wie sie vor der Einrückung waren.

Um die Quellcodezeilen einzurücken:

- Rufen Sie im Source Editor das Kontextmenü auf und wählen Sie **Struct**.

  Oder:

Drücken Sie CTRL+ALT+S.

Der gesamte Quellcode wird entsprechend den Einstellungen in den Natural-Einstellungen eingerückt.

6 Benutzung des Map Editors

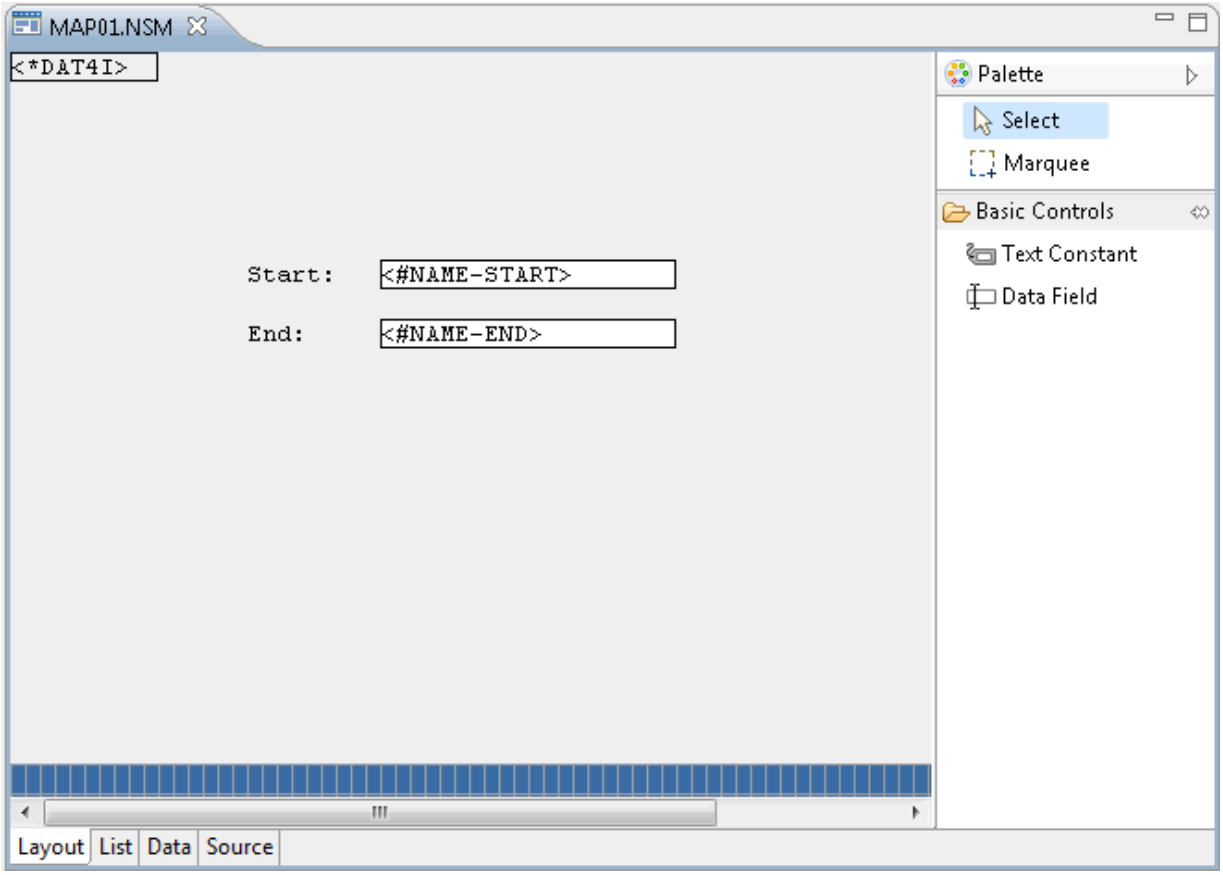
- Allgemeines zum Map Editor 60
- Zugehörige Ansichten 61
- Hinzufügen von Controls (Steuerelementen) zur Map 64
- Auswählen von Controls in der Map 66
- Ändern des Referenz-Control für die Auswahl 67
- Verwalten der Controls in der Map 68
- Definieren von Regeln 69
- Verschieben von Datendefinitionen in Datenfelder 71
- Bearbeiten des Codes einer Map 72
- Ändern der Eigenschaften der Map 72
- Ändern der Eigenschaften einer Textkonstante 76
- Ändern der Eigenschaften für ein Datenfeld 77
- Ändern der Eigenschaften für eine Regel 85
- Definieren von Arrays 86
- Anzeigen der Statuseigenschaften 89
- Speichern von Maps 90
- Speichern von Maps (Stow) 90
- Validierung von Maps 91
- Fehlerbehandlung im Map-Editor 92

Allgemeines zum Map Editor

Das `INPUT`-Statement bietet die Möglichkeit, vordefinierte Masken-Layouts (Natural-Objekttyp Map) zu verwenden.

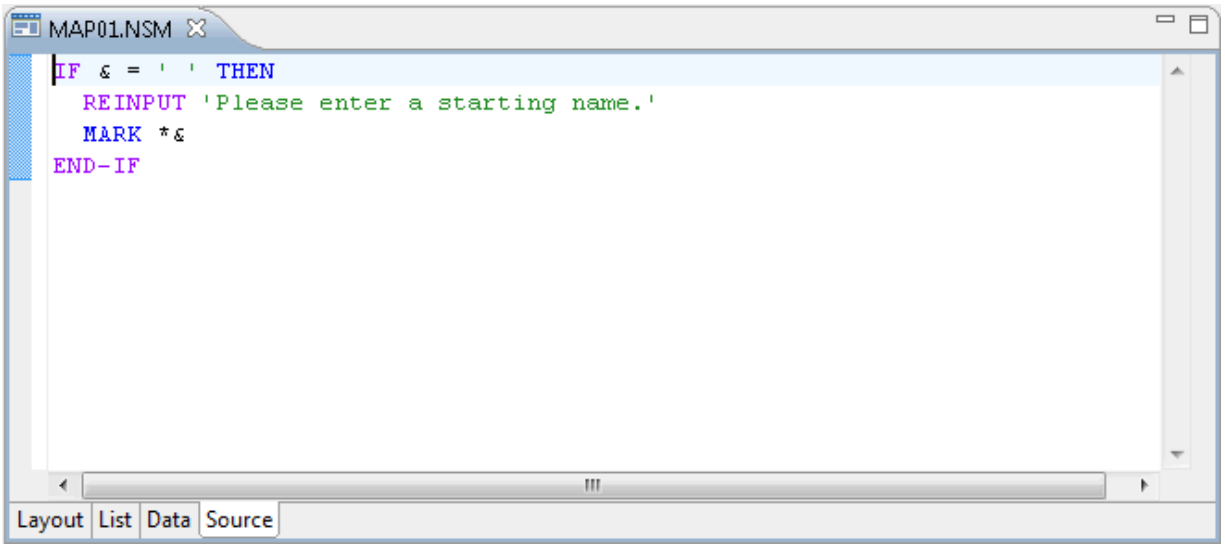
Wenn Sie eine Map erstellen oder bearbeiten, wird der Map Editor (Masken-Editor) aufgerufen. Der Map Editor ist ein mehrseitiger Editor, der die folgenden Seiten umfasst:

- **Layout**
  Zeigt die grafische Darstellung der Map und bietet eine Palette zum Einfügen von Controls (Steuerelementen) in die Map.



- **List**
  Zeigt den Natural-Code an, der dem aktuellen Zustand der gesamten Map entspricht.
- **Data**
  Zeigt die Datendefinition für den Datenbereich an, der gerade in der **Outline**-Ansicht ausgewählt ist.

- **Source**
  Wird nur angezeigt, wenn mindestens eine Verarbeitungsregel definiert wurde. Zeigt den Code für eine Verarbeitungsregel an.



Das Verhalten des Map Editor kann durch Ändern der Natural-Voreinstellungen beeinflusst werden. Siehe *Map Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Weitere Informationen zum Natural-Objekttyp Map und dessen Verwendung finden Sie unter *Map* im *Leitfaden zur Programmierung* bzw. in der Natural-Dokumentation für die entsprechende Plattform.

Zugehörige Ansichten

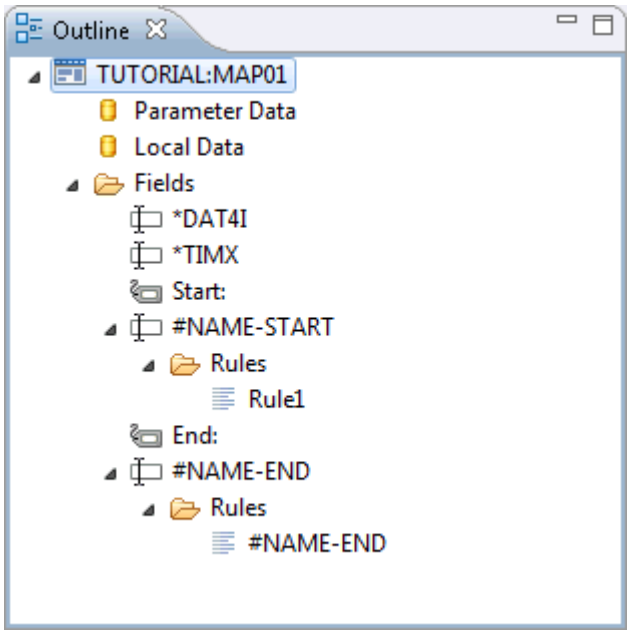
Der Map Editor verwendet die folgenden Ansichten der NaturalONE-Perspektive::

- Outline-Ansicht
- Dependencies-Ansicht

- Properties-Ansicht

Outline-Ansicht

In dieser Ansicht werden die verschiedenen Komponenten der Map in einer Baumstruktur dargestellt. Dazu gehören Datendefinitionen, Felddefinitionen und Verarbeitungsregeln (map- und feldbasiert). Wenn ein dynamisches Layout vorhanden ist, werden dessen Komponenten ebenfalls in einer separaten Hierarchie angezeigt.



Sie können ein Kontextmenü aufrufen, um z. B. das ausgewählte Element zu löschen oder eine neue Verarbeitungsregel zu erstellen.

Sie können zu einem bestimmten Element in der Map navigieren, indem Sie es in der **Outline**-Ansicht auswählen, und umgekehrt.

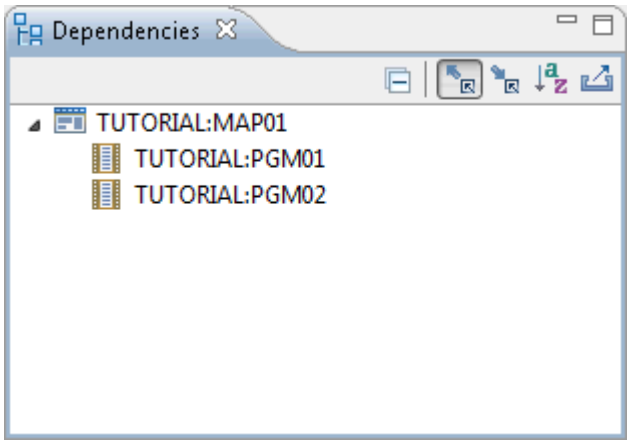
Wenn die Definition eines Elements im Baum geändert oder ein neues Element erstellt wurde, wird für dieses Element eine als „Label Decoration“ bezeichnete Beschriftung in Form eines Sterns angezeigt.

Anmerkung: Die Beschriftungen für Änderungen im Map Editor werden über die Einstellungen unter **General > Appearance > Label Decorations** gesteuert. Wenn Sie diese Label Decorations nicht wünschen, gehen Sie einfach auf die oben genannte Einstellungsseite und deaktivieren Sie **Natural Map Editor Modifications**.

Fehler und Warnungen werden ebenfalls in Form von Label Decorations angezeigt. Weitere Informationen finden Sie unter *Fehlerbehandlung im Map-Editor*.

Dependencies-Ansicht

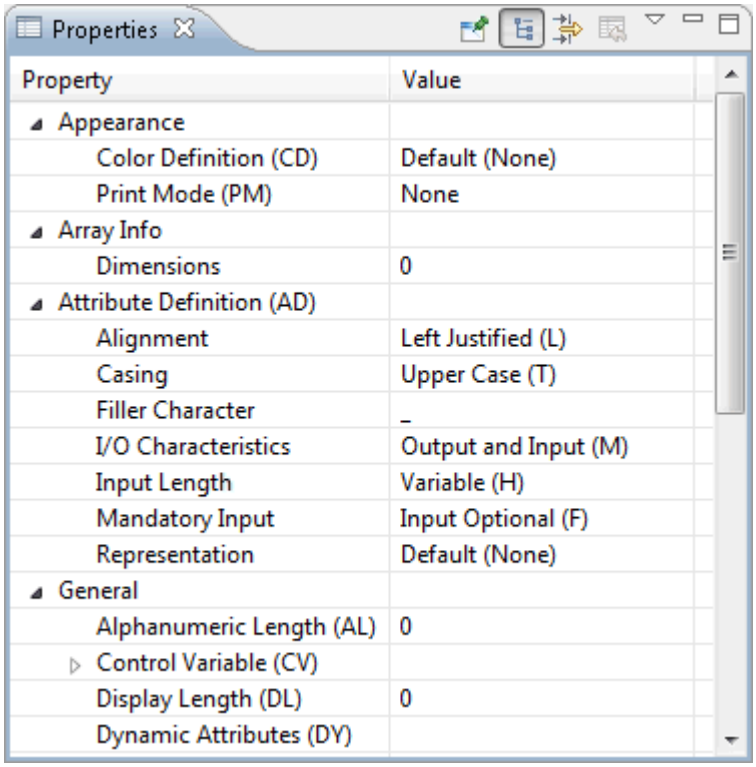
Diese Ansicht zeigt die Abhängigkeiten zwischen der Map, die gerade im aktiven Editorfenster angezeigt wird, und anderen Objekten. Wenn beispielsweise die passiven Cross-Referenzen gerade angezeigt werden, können Sie alle Natural-Objekte sehen, die die bearbeitete Map referenzieren.



Weitere Informationen zu dieser Ansicht finden Sie unter *Dependencies-Ansicht* in der Beschreibung des Source Editors.

Properties-Ansicht

In dieser Ansicht werden die Eigenschaften des Elements angezeigt, das gerade im aktiven Map Editor-Fenster oder in der **Outline**-Ansicht ausgewählt ist. Sie können die in dieser Ansicht angezeigten Werte ändern. Beispiel für ein Datenfeld:



Weitere Informationen zu den verschiedenen Arten von Eigenschaften finden Sie weiter unten.

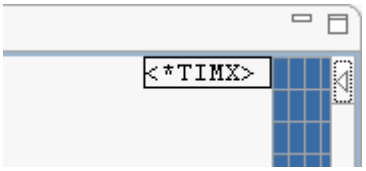
Hinzufügen von Controls (Steuerelementen) zur Map

Sie können Controls mit Hilfe der Palette, durch Klonen vorhandener Controls oder durch Importieren von Datendefinitionen hinzufügen.

Um ein Control zur Map hinzuzufügen

1 Vergewissern Sie sicher, dass die **Layout**-Seite angezeigt wird.

2 Wenn die Palette derzeit ausgeblendet ist, blenden Sie sie über die Pfeil-Schaltfläche ein.



Anmerkung: Standardmäßig wird die Palette auf der rechten Seite des Map Editors angezeigt. Sie können sie auch auf die andere Seite des Map Editors verschieben.

3 Klicken Sie auf die grafische Darstellung des gewünschten Control in der Palette.

4 Klicken Sie auf die Position in der Map, an der Sie das Control platzieren möchten. Lassen Sie die Maustaste nicht sofort los. Halten Sie sie gedrückt und zeichnen Sie den Umriss des Control, bis es die gewünschte Größe hat. Lassen Sie dann die Maustaste los.

Das neue Control wird nun in der **Outline**-Ansicht und in der **Properties**-Ansicht angezeigt.

Um ein Control zu klonen:

1 Wählen Sie ein oder mehrere Controls auf der **Layout**-Seite aus. Dabei kann es sich um Textkonstanten oder Datenfelder handeln.

2 Drücken Sie die STRG-Taste und ziehen Sie das/die ausgewählte(n) Control(s) an eine andere Position in der Map.

Oder:

Verwenden Sie Kopieren und Einfügen. Dies funktioniert auch zwischen zwei verschiedenen Maps. Wenn Controls in dieselbe Map eingefügt werden, überlagern sie die Positionen der kopierten Controls und Sie müssen sie an die gewünschten Positionen verschieben.

Um ein Control auf der Grundlage einer externen Datendefinition zu erstellen (Methode 1):

1 Vergewissern Sie sich, dass die **Layout**-Seite angezeigt wird.

2 Rufen Sie das Kontextmenü an der Stelle auf, an der Sie das/die importierte(n) Datenfeld(er) platzieren möchten, und wählen Sie **Import Data Fields** (Datenfelder importieren).

Oder:

Drücken Sie ALT+SHIFT+I.

Der Dialog **Import Data Fields** wird angezeigt.

3 Wählen Sie im Dialog die zu importierende(n) Datendefinition(en) aus, wie unter *Importieren von Datenfeldern* beschrieben.

Notes:

1. Wenn das ausgewählte Feld bereits als änderbares (**Modifiable**) Feld in der Map vorhanden ist und angepasst werden kann, wird ein Bestätigungsdialog angezeigt. Dieses Verhalten kann durch Ändern der Natural-Voreinstellungen beeinflusst werden. Siehe *Map Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.
2. Die Warnung erscheint nicht, wenn das Feld als reines Ausgabefeld (**Output Only**) in der Map vorhanden ist. In diesem Fall wird das Feld als neues Feld an der ausgewählten Position importiert. Um die Datendefinition anzupassen, führen Sie die folgenden Schritte durch, bevor Sie das Datenfeld importieren:

a. Wählen Sie die Ein-/Ausgabe-Merkmale in der **Properties**-Ansicht aus.
b. Ändern Sie die Einstellung von **Output Only** auf einen anderen Attributwert.
c. Schalten Sie die Ein-/Ausgabe-Merkmale wieder auf die alte Einstellung zurück.

Um ein Control auf der Grundlage einer externen Datendefinition zu erstellen (Methode 2):

1 Wechseln Sie zur **Data**-Seite.

2 Rufen Sie das Kontextmenü an der Stelle auf, an der Sie das/die importierte(n) Datenfeld(er) platzieren möchten, und wählen Sie **Advanced Edit Features > Import Data Fields**.

Oder:

Drücken Sie ALT+SHIFT+I.

Der **Import Data Fields** wird angezeigt.

3 Wählen Sie in diesem Dialogfeld die zu importierende(n) Datendefinition(en) aus, wie unter *Importieren von Datenfeldern* beschrieben.

Für jede importierte Datendefinition wird ein neuer Feldknoten für den aktuellen Datenabschnitt in der **Outline**-Ansicht erstellt.

4 Verschieben Sie die Datendefinitionen in Datenfelder, wie unter *Verschieben von Datendefinitionen in Datenfelder* beschrieben.

Auswählen von Controls in der Map

Sie können Controls entweder auf der **Layout**-Seite oder in der **Outline**-Ansicht auswählen.

Die Palette auf der Layout-Seite bietet verschiedene Werkzeuge für die Auswahl: das **Select** Tool/Auswahlwerkzeug (Standard) und das **Marquee** Tool/Markierungswerkzeug.

Um Controls mit dem Auswahlwerkzeug auszuwählen:

1 Vergewissern Sie sich, dass die Option **Select** in der Palette ausgewählt ist.

2 Klicken Sie auf der **Layout**-Seite auf das Control, das Sie auswählen möchten.

Oder:

Um mehr als ein Control auszuwählen, halten Sie die STRG-Taste gedrückt und klicken Sie dann auf jedes gewünschte Control auf der **Layout**-Seite.

Die Auswahl wird automatisch in der **Outline**-Ansicht übernommen.

Um Controls mit dem Markierungswerkzeug auszuwählen:

1 Wählen Sie **Marquee** in der Palette.

2 Ziehen Sie auf der **Layout**-Seite den Mauszeiger um alle Controls, die Sie in Ihre Auswahl aufnehmen möchten.

Die Auswahl wird automatisch in der **Outline**-Ansicht übernommen.

Um mehrere Controls in der Outline-Ansicht auszuwählen:

- Halten Sie die STRG-Taste gedrückt und klicken Sie dann auf jedes gewünschte Control in der **Outline**-Ansicht.

Die Auswahl wird automatisch auf der **Layout**-Seite übernommen.

Um ein ganzes Array auszuwählen:

1 Drücken Sie ALT.

2 Klicken Sie auf der **Layout**-Seite auf eines der Feldelemente des Arrays.

Anmerkung: Um eine Zwischenablage-Operation (**Cut**/Ausschneiden, **Copy**/Kopieren, **Delete**/Löschen) bei einem Array auszuführen, genügt es, eines oder mehrere seiner Elemente auszuwählen. Standardmäßig wird ein Dialog angezeigt, in dem Sie aufgefordert werden, den Vorgang zu bestätigen. Sie können diesen Dialog für die Zukunft unterdrücken, entweder im Bestätigungsdialog selbst oder in den Natural-Voreinstellungen (siehe *Map Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation). Mit dieser Funktion kann die Zwischenablage in Verbindung mit der Markierungsauswahl (siehe oben) verwendet werden.

Ändern des Referenz-Control für die Auswahl

Obwohl mehrere Controls ausgewählt werden können, ist immer nur eines dieses Control das Referenz-Control, das sich von den anderen ausgewählten Controls dadurch unterscheidet, dass die Ziehpunkte nicht hohl, sondern gefüllt sind. Für einige Operationen, wie z. B. das Löschen von Controls, ist das Referenz-Control unerheblich. Bei anderen Operationen, wie z. B. dem Ausrichten von Feldern, muss jedoch eines der ausgewählten Controls als Grundlage für die Operation verwendet werden, und das ist unveränderbar das Referenz-Control.

Um das Referenz-Control zu ändern:

1 Wählen Sie auf der **Layout**-Seite alle gewünschten Controls aus.

2 Drücken Sie SHIFT und halten Sie die Taste gedrückt.

3 Klicken Sie auf das neue Referenz-Control.

4 Führen Sie die gewünschte Aktion aus.

Verwalten der Controls in der Map

Um die Größe eines Control zu ändern oder es zu verschieben:

1 Wählen Sie das Control auf der **Layout**-Seite der Map aus.

2 Benutzen Sie die Maus, um die Größe des Controls zu ändern oder es zu verschieben.

Oder:

Ändern Sie die entsprechenden Werte in der **Properties**-Ansicht.

Anmerkung: Wenn Sie die Größe eines Arrays ändern, wird nicht die Breite oder Höhe der einzelnen Elemente, sondern die Anzahl der Array-Elemente geändert. Da nur ganze Zeilen und Spalten angezeigt werden können, kann dies dazu führen, dass nichts passiert, wenn das Begrenzungsrechteck für das Array in einer der beiden Richtungen nicht genug erweitert wurde, um eine zusätzliche Zeile oder Spalte aufzunehmen. Informationen darüber, wie man ein ganzes Array vor der Größenänderung auswählt, siehe *Selecting Controls in the Map* in der *NaturalONE*-Dokumentation.

Um eine Textkonstante umzubenennen:

1 Wählen Sie die Textkonstante auf der **Layout**-Seite der Map oder in der **Outline**-Ansicht aus.

2 Ändern Sie den **Label**-Wert in der **Properties**-Ansicht.

Oder:

Klicken Sie auf der **Layout**-Seite noch einmal auf die Textkonstante und geben Sie dann die neue Bezeichnung ein.

Um ein Control zu löschen:

1 Wählen Sie das Control auf der **Layout**-Seite der Map oder in der **Outline**-Ansicht aus.

2 Drücken Sie DEL.

Oder:

Rufen Sie in der **Outline**-Ansicht das Kontextmenü auf und wählen Sie **Delete**.

Um mehrere Controls auszurichten:

1 Markieren Sie die Controls auf der **Layout**-Seite der Map.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Align >** *`alignment-option`*.

Die Controls werden relativ zum Referenz-Control ausgerichtet. Siehe auch *Changing the Reference Control for Selection* in der *NaturalONE*-Dokumentation.

Definieren von Regeln

Sie können Verarbeitungsregeln und freie Predict-Regeln für Datenfelder definieren. Siehe auch *Ändern der Eigenschaften für eine Regel*.

Um eine Verarbeitungsregel zu erstellen:

1 Wählen Sie ein Datenfeld in der **Outline**-Ansicht aus.

Oder:

Markieren Sie den obersten Knoten in der **Outline**-Ansicht, wenn Sie eine Verarbeitungsregel für die gesamte Map erstellen möchten.

2 Rufen Sie das Kontextmenü auf und wählen Sie **New Rule**.

Oder:

Drücken Sie CTRL+ALT+R.

Eine neue Regel wird in der **Outline**-Ansicht angezeigt und die **Source**-Seite des Map-Editors wird automatisch geöffnet.

3 Geben Sie die Verarbeitungsregel auf der **Source**-Seite ein. Zum Beispiel:

```
IF & = ' ' THEN
  REINPUT 'Please enter a starting name.'
  MARK *&
END-IF
```

Ein kaufmännisches Und (&) in der Verarbeitungsregel wird dynamisch durch den Namen des Feldes ersetzt.

4 Definieren Sie den Rang für die Regel in der **Properties**-Ansicht.

Der Rang definiert die Reihenfolge, in der die Regeln für die verschiedenen Felder abgearbeitet werden sollen.

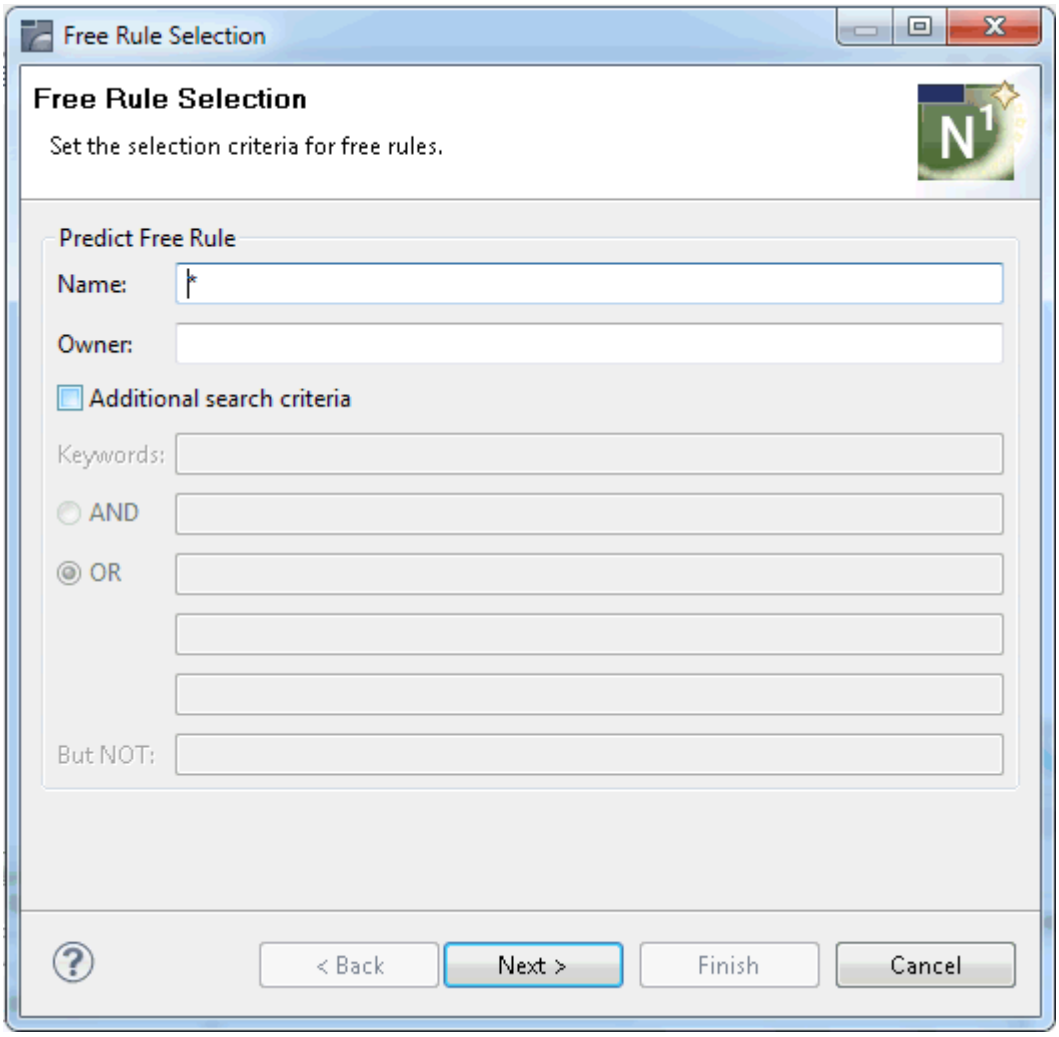
Um freie Predict-Regeln zu verlinken:

ONur verfügbar, wenn Predict in der entsprechenden Natural-Server-Umgebung verfügbar ist.

1 Markieren Sie ein Datenfeld in der **Outline**-Ansicht.

2 Rufen Sie das Kontextmenü auf und wählen Sie **Free Rule Wizard**.

Es erscheint das folgende Dialogfenster.



3 Geben Sie den Namen der Regel, den Predict-Eigentümer und bis zu fünf Schlüsselwörter ein, unter denen die Regel in Predict gespeichert wird.

Für den Regelnamen kann die Stern-Notation verwendet werden, die Schlüsselwörter können mit den Booleschen Operatoren `AND` und `OR` kombiniert werden, und es kann auch ein `BUT NOT`-Schlüsselwort angegeben werden.

4 Wählen Sie die Schaltfläche **Next**.

Es wird eine Liste mit freien Regeln angezeigt.

5 Markieren Sie die Regel(n), die Sie mit dem Feld verknüpfen wollen.

6 Optional: Wählen Sie die Schaltfläche **Next**, um eine Vorschau der ausgewählten Regel(n) anzuzeigen.

7 Wählen Sie die Schaltfläche **Finish**, um die Regel(n) mit dem Feld zu verknüpfen.

Um die Verknüpfung von freien Predict Regeln aufzuheben:

1 Markieren Sie die Regel in der **Outline**-Ansicht.

2 Drücken Sie DEL.

Oder:

Rufen Sie das Kontextmenü auf und wählen Sie **Delete**.

Anmerkungen:

1. Wenn Sie die Verknüpfung einer Regel aufheben, wird die Regel auf dem Predict-Server nicht gelöscht.
2. Die Verknüpfung einer automatischen Regel, die implizit für ein View-Feld erstellt wurde, kann nicht aufgehoben werden.

Verschieben von Datendefinitionen in Datenfelder

Sie können eine Datendefinition aus einem für die Map definierten lokalen Datenbereich (LDA) oder Parameterdatenbereich (PDA) in ein Datenfeld verschieben.

Um eine Datendefinition in ein Datenfeld zu verschieben:

1 Markieren Sie eine Datendefinition in der **Outline**-Ansicht (d.h. markieren Sie einen Eintrag unter **Local Data** oder **Parameter Data**).

2 Rufen Sie das Kontextmenü auf und wählen Sie **Create Map Field(s)**.

Die Datendefinition wird unter **Local Data** oder **Parameter Data** entfernt.

Ein neues Datenfeld ist nun auf der **Layout**-Seite verfügbar. Es befindet sich zunächst in der linken oberen Ecke der Map (Zeile 1 und Spalte 1), von wo aus Sie es an die gewünschte Position verschieben können.

Bearbeiten des Codes einer Map

Wenn Sie das Kontextmenü auf der **List**-, **Data**- oder **Source**-Seite des Map-Editors aufrufen, sind auch mehrere Standard-Kommandos für den Source Editor verfügbar. Ausführliche Informationen zu diesen Kommandos finden Sie in den entsprechenden Beschreibungen unter *Benutzung des Source Editors*.

Ändern der Eigenschaften der Map

Wenn Sie die Eigenschaften (Properties) für die Map selbst anzeigen (z. B. durch Auswahl des obersten Eintrags in der **Outline**-Ansicht), können Sie die Folgendes ändern:

Eigenschaft	Beschreibung
Erscheinungsbild (Appearance)	
Control Variable	**Kontrollvariable** Der Name einer Attribut-Kontrollvariable, deren Inhalt die Attributeigenschaften von Feldern und Texten bestimmt, die die Attributdefinition `AD=Y` haben. Die Attribut-Kontrollvariable, auf die in der Map verwiesen wird, muss in dem Programm, das diese Map verwendet, definiert sein. Das Entfernen einer Attribut-Kontrollvariablen aus den Map-Eigenschaften bedeutet, dass die Attribut-Kontrollvariable auch aus der Map entfernt wird, sofern sie nicht mit einem anderen Map-Feld verknüpft ist.
Print Mode (PM)	**Druck-/Anzeige-Modus** Der Standard-Druck-/Anzeige-Modus für Variablen. Dieser Wert wird in die Felddefinition kopiert, wenn ein neues Feld erstellt wird. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: None Alternative character set / Alternativer Zeichensatz (C) Inverse print direction / Invertierte Druckrichtung (I) No hardcopy can be made / Es kann keine Hardcopy erstellt werden (N) None / Keine bedeutet, dass der Standardzeichensatz verwendet wird. For further information, see the description of the session parameter `PM` in the Natural documentation for the appropriate platform. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters `PM` in der *Parameter-Referenz-Dokumentation* bzw. in der Natural-Dokumentation für die entsprechende Plattform.

Eigenschaft	Beschreibung
Filler Characters / Füllzeichen	
Label Filler	**Label-Füllzeichen** Nachgestelltes Füllzeichen, das zum Auffüllen von Textkonstanten-Labels verwendet wird, wenn die Länge des Labels kleiner ist als die Breite der Textkonstante.
Optional, complete	**Optional, vollständig** Füllzeichen für Felder, die optional sind, aber vollständig ausgefüllt werden müssen, wenn sie verwendet werden.
Optional, partial	**Optional, teilweise** Füllzeichen für Felder, die optional sind und teilweise ausgefüllt werden können.
Required, complete	**Erforderlich, vollständig** Füllzeichen für Felder, die erforderlich sind (Pflichtfelder) und die obligatorisch sind und vollständig ausgefüllt werden müssen.
Required, partial	**Erforderlich, teilweise** Füllzeichen für Felder, die erforderlich sind (Pflichtfelder) und teilweise ausgefüllt werden dürfen.
General / Allgemein	
Auto Rule Rank	**Autom. Regel Rang** Der Standard-Rang für jede neu erstellte Verarbeitungsregel in der Map. (Später können Sie die Ränge individuell ändern).
Column Shift	**Spaltenverschiebung** Spaltenverschiebung (0 oder 1), die bei der Map angewendet werden soll. Diese Funktion kann verwendet werden, um alle 80 Spalten auf einem 80-spaltigen Bildschirm anzusprechen (Spaltenverschiebung = 1, Zeilengröße = 80).
Decimal Character	**Dezimalzeichen** Das Zeichen, das als Dezimalzeichen verwendet werden soll. Das Standardzeichen ist ein Punkt (.).
Field Sensitive	**Feldsensitiv** Wenn dieser Wert auf true gesetzt ist, wird die Konsistenzprüfung für ein Map-Feld durchgeführt, sobald das Feld vom Benutzer ausgefüllt wird. Ist der Wert auf false gesetzt, wird die Feldprüfung durchgeführt, wenn die Map vollständig ausgefüllt ist.
Help Map	**Hilfe-Map** Wenn diese Option auf true gesetzt ist, wird die Map als Hilfetext markiert. Bei der Einstellung false wird die Map nicht als Hilfetext gekennzeichnet.

Eigenschaft	Beschreibung
Manual Skip	**Manuelles Überspringen** Wenn diese Option auf true gesetzt ist, wird der Cursor zur Ausführungszeit nicht automatisch zum nächsten Feld in der Map bewegt, selbst wenn das aktuelle Feld vollständig ausgefüllt ist. Bei false wird der Cursor zur Ausführungszeit automatisch zum nächsten Feld in der Map bewegt.
Output Map	**Ausgabe-Maske** Wenn diese Option auf true gesetzt ist, ist das Ergebnis des Map-Definitionsprozesses ein `WRITE`-Statement und die resultierende (Ausgabe-)Map kann von einem Programm aus mit einem `WRITE USING FORM`-Statement aufgerufen werden. Leerzeilen am Ende der Map werden automatisch gelöscht, so dass die Map mehrfach auf einer Seite ausgegeben werden kann. Bei false ist das Ergebnis der Map-Definition ein `INPUT`-Statement und die resultierende Map kann von einem Programm aus mit einem `INPUT USING MAP`-Statement aufgerufen werden.
Right Justified	**Rechtsbündig** Wenn diese Option auf true gesetzt ist, werden numerische und alphanumerische Felder, die aus Datendefinitionen anderer Natural-Objekte kopiert werden, rechtsbündig ausgerichtet. Bei der Einstellung false werden numerische und alphanumerische Felder, die aus Datendefinitionen anderer Natural-Objekte kopiert werden, nicht rechtsbündig ausgerichtet.
Standard Keys	**Standard-Tasten** Wenn diese Option auf true gesetzt ist, bleiben die letzten beiden Zeilen der Map leer, damit zur Ausführungszeit Funktionstastenspezifikationen eingegeben werden können. Ist der Wert auf false gesetzt, werden alle Zeilen für die Map verwendet.
Upper Case	**Großbuchstaben** Wenn diese Option auf true gesetzt ist, wird die Eingabe bei der Ausführung der Map in Großbuchstaben umgewandelt. Bei false wird die Eingabe bei der Ausführung der Map nicht in Großbuchstaben umgewandelt.
Zero Printing	**Anzeige von Nullwerten** Wenn diese Option auf true gesetzt ist, werden numerische Felder, die ausschließlich Nullen enthalten, ausgegeben (Anzeige einer Null, rechtsbündig). Bei false wird die Ausgabe von numerischen Feldern, die nur Nullen enthalten, unterdrückt.

Eigenschaft	Beschreibung
Help / Hilfe	
Helproutine Name	**Name der Helproutine** Der Name der Helproutine oder Help-Map, die zur Ausführungszeit aufgerufen wird, wenn die Hilfefunktion für diese Map aufgerufen wird (globale Hilfe für die Map).
Help Field Default	**Hilfefeld Standard** Wird nur angezeigt, wenn der Name einer Helproutine oder einer Help-Map definiert wurde. Bei true gilt die für die Map angegebene Helproutine oder Help-Map als Standard für jedes einzelne Feld der Map, d.h. der Name jedes Feldes wird einzeln an die Helproutine oder Help-Map übergeben. Bei false wird der Name der Map an die Helproutine bzw. Help-Map übergeben.
Help Parameters	**Hilfe-Parameter** Wird nur angezeigt, wenn der Name einer Helproutine oder einer Help-Map definiert wurde. Geben Sie den/die Namen des Parameters/der Parameter ein, die zur Ausführungszeit aufgerufen werden, wenn die Hilfefunktion aufgerufen wird.
Die Syntax, die für die Angabe von Helproutine-Namen und -Parametern gilt, entspricht der Syntax des Session-Parameters HE (siehe *Parameter-Referenz*-Dokumentation bzw. Natural-Dokumentation für die entsprechende Plattform).	
Layout	
Layout Name	**Name des Layouts** Der Name der Map, die als Standard-Layout für die aktuelle Map dient. Sie können diese Eigenschaft verwenden, um die Erstellung vieler ähnlicher Maps zu vereinfachen, indem Sie eine Map als grundlegende Layout-Map mit einer Reihe von Feldern erstellen, die von den anderen Feldern verwendet werden sollen. In allen ähnlichen Maps geben Sie den Namen der Standard-Layout-Map an und fügen nur die Felder hinzu, die für die neue Map spezifisch sind.
Dynamic Layout	**Dynamisches Layout** Wenn diese Option auf true gesetzt ist, werden die Felder zur Laufzeit dynamisch in das Layout importiert. Das bedeutet, dass Sie Ihr Standard-Layout ändern können und diese Änderung automatisch in allen ähnlichen Maps, die dieses Layout verwenden, übernommen wird. Bei false wird das Standard-Layout nicht dynamisch verwendet.
RTL Support / RTL-Unterstützung	

Eigenschaft	Beschreibung
Visual Order	**Visuelle Anordnung** Bei true wird angenommen, dass die Textkonstanten der Map in visueller Reihenfolge angeordnet sind. Bei false (Standardeinstellung) wird angenommen, dass die Textkonstanten in logischer Reihenfolge angeordnet sind. Siehe auch *Unterstützung bidirektionaler Sprachen*.
Size / Größe	
Columns	**Spalten** Die Anzahl der Spalten, aus denen sich die Breite der Map zusammensetzt.
Rows	**Zeilen** Die Anzahl der Zeilen, aus denen sich die Höhe der Map zusammensetzt.

Ändern der Eigenschaften einer Textkonstante

Wenn Sie eine Textkonstante auswählen, können Sie die folgenden Eigenschaften ändern:

Eigenschaft	Beschreibung
Erscheinungsbild (Appearance)	
Color Definition (CD)	**Farbdefinition** Legt das Farbattribut für die Textkonstante fest. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Default (None) - Standard Blue (BL) - blau Green (GR) - grün Neutral (NE) - neutral Pink (PI) - rosa Red (RE) - rot Turquoise (TU) - türkis Yellow (YE) - gelb Weitere Informationen siehe Beschreibung des Session-Parameters `CD` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Attribut-Definition (AD)	

Eigenschaft	Beschreibung
Representation	**Feldanzeige** Legt fest, wie der Wert angezeigt wird. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Default (None) - Standard Blinking (B) - blinkend Italic (C) - kursiv Intensified (I) - intensiviert Non-display (N) - nicht sichtbar Underlined (U) - unterstrichen Reverse Video (V) - invers Dynamic (Y) - dynamische Attribute Weitere Informationen siehe Beschreibung des Session-Parameters `AD` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
General / Allgemein	
Label	**Bezeichnung** Das Label, das auf der Map erscheinen soll.
Position	
Column	**Spalte** Die Spalte in der Map, in der die Textkonstante beginnt. Wenn Sie die Textkonstante mit der Maus verschieben, wird dieser Wert automatisch angepasst.
Row	**Zeile** Die Zeile in der Map, die die Textkonstante enthält. Wenn Sie die Textkonstante mit der Maus verschieben, wird dieser Wert automatisch angepasst.
Größe	
Width	**Breite** Die Länge der Textkonstante. Wenn Sie die Größe der Textkonstante mit der Maus ändern, wird dieser Wert automatisch angepasst.

Ändern der Eigenschaften für ein Datenfeld

Wenn Sie ein Datenfeld auswählen, können Sie die folgenden Eigenschaften (Properties) ändern:

Eigenschaft	Beschreibung
Erscheinungsbild (Appearance)	
Color Definition (CD)	**Farbdefinition** Legt das Farbattribut für das Datenfeld fest. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Default (None) - Standard Blue (BL) - blau Green (GR) - grün Neutral (NE) - neutral Pink (PI) - rosa Red (RE) - rot Turquoise (TU) - türkis Yellow (YE) - gelb Weitere Informationen siehe Beschreibung des Session-Parameters `CD` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Print Mode (PM)	**Druck-/Anzeige-Modus** Legt den Druckmodus für das Datenfeld fest. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: None Alternative character set / Alternativer Zeichensatz (C) Inverse print direction / Invertierte Druckrichtung (I) No hardcopy can be made / Es kann keine Hardcopy erstellt werden (N) None / Keine bedeutet, dass der Standardzeichensatz verwendet wird. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters `PM` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Array-Informationen	
Dimensions	**Dimensionen** Sie können ein Array mit bis zu drei Dimensionen definieren. Weitere Informationen finden Sie unter *Definieren von Arrays*.
Attribut-Definition (AD)	
Alignment	**Ausrichtung des Feldwertes** Legt fest, wie der Inhalt des Feldes angezeigt wird. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: None - keine Left-justified (L) - linksbündig

Eigenschaft	Beschreibung
	Right-justified (R) - rechtsbündig Leading zeros (Z) - vorangestellte Nullen „Keine“ bedeutet, dass die Standardausrichtung verwendet wird, die vom Format eines Feldes abhängt. „Linksbündig“ ist die Standardeinstellung für alphanumerische Felder. „Rechtsbündig“ ist der Standard für numerische Felder. „Vorangestellte Nullen“ können für numerische Werte definiert werden, die dann rechtsbündig angezeigt werden. Weitere Informationen siehe Beschreibung des Session-Parameters AD in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Casing	**Groß-/Kleinschreibung** Upper case (T) - Kleinbuchstaben werden in Großbuchstaben umgesetzt Mixed case (W) - Kleinbuchstaben werden akzeptiert Weitere Informationen siehe Beschreibung des Session-Parameters AD in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Filler Character	**Füllzeichen** Sie können ein Zeichen angeben, das zum Füllen leerer Eingabefelder oder veränderbarer Ausgabefelder verwendet werden soll. Wenn Sie ein Eingabefeld bearbeiten, das bis zu seiner maximalen Länge mit Leerzeichen aufgefüllt ist, kann es so aussehen, als ob der Text nicht bearbeitet werden kann. In solchen Fällen müssen die Leerzeichen vom Ende des Feldes her explizit gelöscht werden.
I/O Characteristics	**Ein-/Ausgabe-Eigenschaften** Legt fest, ob das Feld ein Eingabe- oder ein Ausgabefeld ist. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Input Only (A) - Nur Eingabe Output and Input (M) - Ausgabe und Eingabe Output Only (O) - Nur Ausgabe
Input Length	**Eingabe-Länge** Legt fest, ob der in das Feld eingegebene Wert die gleiche Länge wie das Feld haben muss (feste Länge - nur für Nur-Eingabe-Felder relevant) oder ob der in das Feld eingegebene Wert kürzer sein darf als das Feld (variable Länge). Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Fixed (G) - feste Länge Variable (H) - variable Länge

Eigenschaft	Beschreibung
Mandatory Input	**Obligatorische Eingabe (Pflichtfeld)** Legt fest, ob ein Wert in das Feld eingegeben werden muss (nur relevant für Nur-Eingabe-Felder (AD=A)) oder ob ein Wert in das Feld eingegeben werden kann, aber nicht muss. Sie können den erforderlichen Wert aus dem Dropdown-Listenfeld auswählen: Input Mandatory (E) - Eingabe obligatorisch Input Optional (F) - Eingabe fakultativ
Representation	**Darstellung** Legt fest, wie der Wert angezeigt wird. Sie können den gewünschten Wert aus dem Dropdown-Listenfeld auswählen: Default (None) - Standard Blinking (B) - blinkend Italic (C) - kursiv Intensified (I) - intensiviert Non-display (N) - nicht sichtbar Underlined (U) - unterstrichen Reverse Video (V) - invers Dynamic (Y) - dynamische Attribute
Weitere Informationen siehe Beschreibung des Session-Parameters AD in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.	
General / Allgemein	
Alphanumeric Length (AL)	**Alphanumerische Länge (AL)** Wird nur bei den Natural-Datenformaten A (alphanumerisch) und U (Unicode) angezeigt. Sie können die Standardausgabelänge für ein alphanumerisches Feld angeben, d.h. wenn sie kürzer als die Feldlänge angegeben wird, wird das Feld rechts abgeschnitten. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters AL in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Control Variable (CV)	**Kontrollvariable** Sie können eine Kontrollvariable für ein dynamisches Feldattribut eingeben. Dies ist die Kontrollvariable, die die Attribute enthält, die für das Datenfeld verwendet werden sollen. Die Variable muss in dem Programm, das die Map referenziert, im Natural-Datenformat C (für Attributsteuerung) definiert sein. Die Kontrollvariable enthält auch ein MODIFIED-Daten-Tag, das anzeigt, ob das Feld nach der Ausführung der Map geändert wurde. Eine einzelne Kontrollvariable kann auf mehrere Map-Felder angewendet werden. In diesem

Eigenschaft	Beschreibung
	Fall wird das MODIFIED-Daten-Tag gesetzt, wenn eines der Felder, die die Kontrollvariable referenzieren, geändert wurde. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters CV in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Display Length (DL)	**Anzeigelänge** Wird nur bei den Natural-Datenformaten A (alphanumerisch) und U (Unicode) angezeigt. Mit dieser Eigenschaft können Sie die Ausgabelänge für Unicode-Strings angeben, die zusätzlichen Platz benötigen können. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters DL in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Date Format (DF)	**Datumsformat** Wird nur für das Natural-Datenformat D (Datum) angezeigt. Diese Eigenschaft bestimmt die Länge eines Datums, wenn es in eine alphanumerische Darstellung umgewandelt wird, ohne dass eine Editiermaske angegeben ist. Sie können die gewünschte Option aus dem Dropdown-Listenfeld auswählen: Default (none) - Standard (keine) Short (S) - Kurz Internal (I) - Intern Long (L) - Lang Weitere Informationen finden Sie in der Beschreibung des Session-Parameters DF in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Dynamic Attributes (DY)	**Dynamische Attribute** Sie können Attribute für die dynamische Anzeige von Attributfeldern zuweisen. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters DY in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Edit Mask (EM)	**Editiermaske** Sie können die Editiermaske eingeben, die für das Datenfeld verwendet werden soll.

Eigenschaft	Beschreibung
	Weitere Informationen finden Sie in der Beschreibung des Session-Parameters EM und EMU in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform. **Anmerkung:** Die Editiermaske hat Vorrang vor der Anzeigelänge.
Float Length (FL)	**Gleitkomma-Mantissenlänge** Wird nur für das Natural-Datenformat F (Gleitkomma) angezeigt. Sie können die Mantissenlänge einer Gleitkommavariablen bei der Eingabe oder Ausgabe angeben. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters FL in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Format	**Datenformat** Sie können das gewünschte Natural-Datenformat aus dem Dropdown-Listenfeld auswählen: Alphanumeric (A) - Alphanumerisch Binary (B) - Binär Date (D) - Datum Floating point (F) - Gleitkomma Integer (I) - Ganzzahl Logical (L) - Logisch Numeric (N) - Numerisch Packed numeric (P) - Gepackt numerisch Time (T) - Zeit Unicode (U) Das Standardformat ist A. **Anmerkung:** Da die zur Definition eines Feldes erforderlichen Informationen vom Natural-Datenformat abhängen, können andere Eigenschaften erscheinen oder Eigenschaften nicht erscheinen, wenn sich das Format ändert.
Length	**Länge** Wird nur für die Natural-Datenformate A (alphanumerisch), B (binär), F (Fließkomma), I (Ganzzahl), N (numerisch), P (gepackt numerisch) und U (Unicode) angezeigt. Sie können die interne Länge des Feldes angeben, die von einem Programm verwendet wird, das dieses Feld referenziert. Gültige Eingabewerte finden Sie unter *Format and Length of User-Defined Variables* im *Leitfaden zur Programmierung* bzw. in der Natural-Dokumentation für die entsprechende Plattform.

Eigenschaft	Beschreibung
Name	**Name** Der Name des Feldes. **Anmerkung:** Der Standardname, der für das Feld beim ersten Anlegen generiert wird, hängt von den zulässigen Bezeichnerzeichen ab, die für die Natural-Serverumgebung definiert sind.
Numeric Length (NL)	**Numerische Länge der Ausgabe** Wird nur für die Natural-Datenformate B (binär), I (ganzzahlig), N (numerisch) und P (gepackt numerisch) angezeigt. Sie können die Standard-Ein-/Ausgabelänge für ein numerisches Feld angeben, das in einem `DISPLAY`-, `INPUT`-, `PRINT`- oder `WRITE`-Statement verwendet wird. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters `NL` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Sign Position (SG)	**Vorzeichen-Stelle** Wird nur für die Natural-Datenformate F (Gleitkomma), I (Ganzzahl), N (numerisch) und P (gepackt numerisch) angezeigt. Über das Dropdown-Listenfeld können Sie angeben, ob für das Feld eine Stelle zur Anzeige des Vorzeichens zulässig sein soll. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters `SG` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Zero Printing (ZP)	**Anzeige von Nullwerten** Wird nur für die Natural-Datenformate F (Gleitkomma), I (Ganzzahl), N (numerisch) und P (gepackt numerisch) angezeigt. Wenn diese Option aktiviert ist, werden Nullwerte als eine Null angezeigt. Ist die Option deaktiviert, werden Nullwerte unterdrückt. Weitere Informationen finden Sie in der Beschreibung des Session-Parameters `ZP` in der *Parameter-Referenz*-Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.
Help / Hilfe	
Helproutine Name	**Name der Helproutine** Sie können den Namen einer Helproutine oder einer Help-Map eingeben, die dem Map-Feld zugeordnet werden soll. Die Helproutine oder Help-Map wird dann zur Ausführungszeit für das Map-Feld aufgerufen, wenn eine Hilfeanfrage für dieses Map-Feld gestellt wird.

Eigenschaft	Beschreibung
Help Parameters	**Hilfe-Parameter** Wird nur angezeigt, wenn der Name einer Helproutine oder einer Help-Map definiert wurde. Geben Sie den Namen des/der Parameter(s) ein, der/die an die Helproutine oder Help-Map übergeben werden soll(en). Das Entfernen eines Parameters aus diesem Textfeld bedeutet, dass der Parameter auch aus der Map entfernt wird, es sei denn, der Parameter ist ein Map-Feld oder mit einem anderen Map-Feld als Hilfe-Parameter verknüpft oder ist ein Startindex für ein Array.
Die Syntax, die für die Angabe von Helproutine-Namen und -Parametern gilt, entspricht der Syntax des Session-Parameters HE (siehe *Parameter-Referenz*-Dokumentation bzw. die Natural-Dokumentation für die entsprechende Plattform). Zusätzlich zu den Syntaxerklärungen für HE gilt bei der Verwendung des Map-Editors Folgendes: *operand1*: ■ Wenn ein Variablenname angegeben wird, der dem Namen eines Map-Feldes entspricht, muss dieses Feld mit Natural-Datenformat/Länge A8 vorliegen. ■ Wird ein Variablenname angegeben, für den noch kein Map-Feld existiert, so wird automatisch ein Map-Parameter mit diesem Namen im Natural-Datenformat/Länge A8. *operand2*: ■ Wird ein Variablenname angegeben, zu dem noch kein Map-Feld existiert, wird automatisch ein Map-Parameter mit diesem Namen im Natural-Datenformat/Länge N7 definiert.	
Position	
Column	**Spalte** Die Spalte in der Map, in der das Feld beginnt. Wenn Sie das Feld mit der Maus verschieben, wird dieser Wert automatisch angepasst.
Row	**Zeile** Die Zeile in der Map, die das Feld enthält. Wenn Sie das Feld mit der Maus verschieben, wird dieser Wert automatisch angepasst.
Größe	
Width	**Breite** Die Länge des Feldes. Wenn Sie die Größe des Feldes mit der Maus ändern, wird dieser Wert automatisch angepasst.
Die folgenden Attribute bestimmen die Breite eines Feldes mit der folgenden Prioritätsreihenfolge:	

Eigenschaft	Beschreibung
DL > EM > { AL, NL, FL } > [SG] Length	
	Wenn die Eigenschaft **Breite** geändert wird, wird das höchste verwendete Prioritätsattribut (d. h. das nicht Null oder nicht leer ist) angepasst. Ausnahme: Die Attribute Editiermaske (EM) und Länge werden nicht automatisch angepasst. **Beispiele:** Gegebene Feldlänge = 30 1. Die Breite wird auf 25 und DL=0 => AL wird auf 25 gesetzt. 2. Die Breite wird auf 25, AL=18 und DL ist ungleich Null => DL wird auf 25 gesetzt. 3. EM=X(25), DL=0 und AL=18 => Eigenschaft **Breite** kann nicht geändert werden.

Ändern der Eigenschaften für eine Regel

Wenn Sie eine Regel auswählen, können Sie die folgenden Eigenschaften (Properties) ändern:

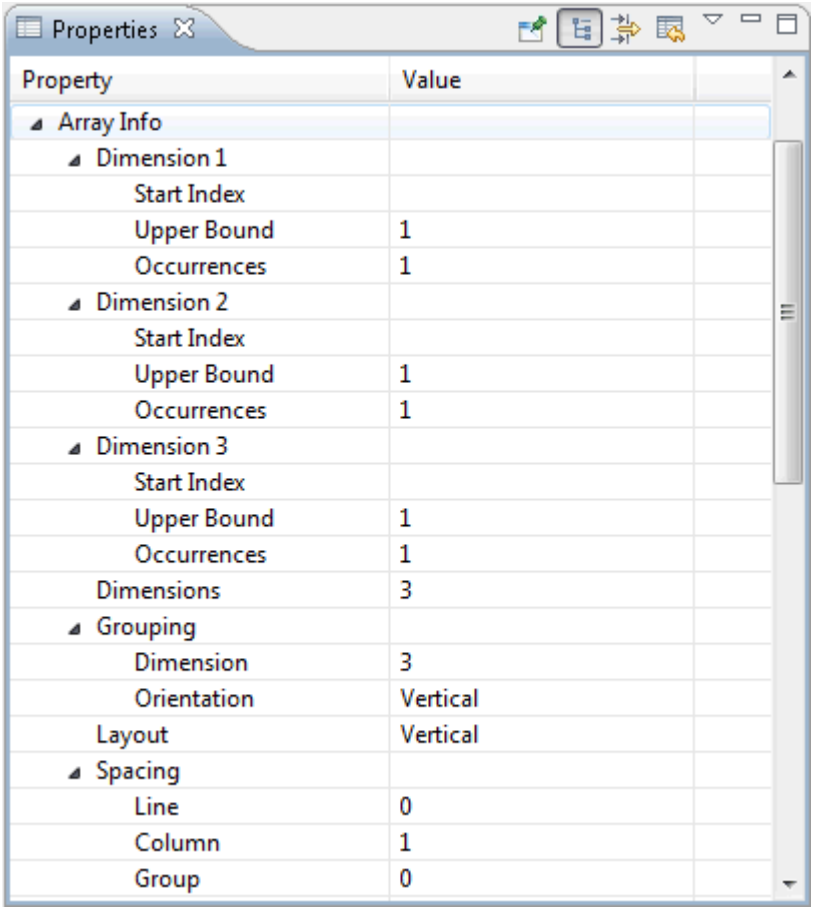
Eigenschaft	Beschreibung
Type	**Typ der Regel** Kann einer der folgenden Werte sein: ■ **Inline Rule** Eine Regel, deren Code im Quellcode der Map gespeichert ist. ■ **Free Rule** Eine benannte Predict-Regel, die explizit mit einem Datenfeld oder mit der Map selbst verbunden sein kann. ■ **Automatic Rule** Automatische Predict-Regeln gelten für Datenbankfelder und werden vom Predict-Administrator definiert. Wenn ein Datenfeld aus einer View-Datendefinition erstellt wird (siehe *Verschieben von Datendefinitionen in Datenfelder*), werden alle automatischen Regeln für dieses Feld implizit mit der Map-Definition verknüpft. Alle automatischen Regeln werden verkettet und als *eine* Map-Regel behandelt. Eine freie Regel kann in eine Inline-Regel umgewandelt werden, indem der Typ **Inline Rule** aus dem Dropdown-Listenfeld ausgewählt wird. Bei allen anderen Regeltypen ist dieses Dropdown-Listenfeld nicht verfügbar. **Anmerkung:** Der Code für Predict-Regeln wird auf dem Predict-Server gespeichert und gepflegt und kann im Map Editor nicht geändert werden.

Eigenschaft	Beschreibung
Rank	**Rang** Ein numerischer Wert (0 - 99), der die Reihenfolge bestimmt, in der die Regel relativ zu den anderen Regeln ausgeführt wird (falls vorhanden). Ein Feld kann bis zu 100 Verarbeitungsregeln haben, von denen jede einen eindeutigen Rang haben muss. Zur Ausführungszeit der Map werden die Verarbeitungsregeln in aufsteigender Reihenfolge ausgeführt, zunächst nach Rang, dann nach der Position des Feldes auf dem Bildschirm. Zu diesem Zweck wird davon ausgegangen, dass Verarbeitungsregeln, die mit der Map selbst verknüpft sind, immer die erste Bildschirmposition haben. Der Standardrang für automatische Regeln wird durch die Eigenschaft **Auto Rule Rank** in den Eigenschaften (Properties) der Map definiert. Siehe auch *Ändern der Eigenschaften für die Map*.

Definieren von Arrays

Arrays werden in den Eigenschaften (Properties) eines Datenfelds definiert. Sie können ein Array mit bis zu drei Dimensionen definieren. Die Reihenfolge, in der die Dimensionen des Arrays auf das Map-Layout abgebildet werden, wird durch die von Ihnen eingegebenen Werte bestimmt.

Die Anzahl der angezeigten array-spezifischen Eigenschaften hängt von der Dimension ab, die Sie angeben. Wenn Sie z. B. den Wert 3 in der Eigenschaft **Dimensions** angeben, werden die folgenden Eigenschaften angezeigt:



Sie können die folgenden Array-Eigenschaften ändern:

Eigenschaft	Beschreibung
Dimensions	**Dimensionen** Wenn dieser Wert auf 0 gesetzt ist, werden keine Dimensionen verwendet (Standard) und dies ist die einzige Eigenschaft, die unter **Array Info** angezeigt wird. Sie können den Wert auf 1, 2 oder 3 setzen, um ein ein-, zwei- oder dreidimensionales Array zu verwenden. In diesem Fall werden zusätzliche Eigenschaften angezeigt (siehe unten).
Layout	**Darstellung im Map-Layout** Legt fest, wie das Array im Map-Layout dargestellt wird (vertikal oder horizontal).
Dimension *n*	
Start Index	**Startindexwert für jede Dimension.** Sie können eine Zahl oder den Namen einer Variablen eingeben. Der tatsächliche Wert wird in dem Programm angegeben, das die Map-Definition aufruft. Wenn nicht als ein Feld in der Map definiert, wird davon ausgegangen, dass die Variable mit Natural-Datenformat/Länge N7 definiert ist.

Eigenschaft	Beschreibung
	Das Entfernen eines Startindexwerts aus einem Array hat zur Folge, dass die Variable auch aus der Map entfernt wird, es sei denn, sie ist ein Map-Feld oder sie ist mit einem anderen Map-Feld als Startindexwert oder Hilfeparameter verknüpft.
Upper Bound	**Obere Begrenzung** Die obere Begrenzung des Feldes. Diese Zahl ist die höchste Ausprägung in der ersten, zweiten und dritten Dimension.
Occurrences	**Ausprägungen** Die Anzahl der Ausprägungen, die in der Map angezeigt werden sollen. Sie können die Größe des Feldes auch mit der Maus vertikal verändern. Der entsprechende Wert wird automatisch in den Eigenschaften berücksichtigt. Dies gilt nicht für die dritte Dimension eines dreidimensionalen Arrays, da nur zwei Bereiche von Ausprägungen auf dem Bildschirm dargestellt werden können. Eindimensionale Arrays können als mehrzeilige/mehrspaltige Felder angezeigt werden. Für diese Arrays können die zweite Anzahl von Ausprägungen und das Layout für die zweite Dimension definiert werden.
Spacing / Abstände	
Line	**Zeile** Wird nur für Arrays angezeigt, die eine Dimension mit vertikaler Ausrichtung haben. Dies ist die Anzahl der Leerzeilen, die horizontal zwischen jeder Dimensionsausprägung im Array eingefügt werden.
Column	**Spalte** Wird nur für Arrays angezeigt, die eine Dimension mit horizontaler Ausrichtung haben. Die Anzahl der leeren Spalten, die vertikal zwischen jeder Dimensionsausprägung im Array eingefügt werden.
Group	**Gruppe** Wird nur bei Arrays mit einer gruppierten Dimension angezeigt. Die Anzahl der Leerzeilen, die vertikal, oder Spalten, die horizontal zwischen jeder Ausprägung der gruppierten Dimension eingefügt werden sollen.
Gruppierung	
Dimension	**Dimension** Wird nur bei einem zwei- oder dreidimensionalen Array angezeigt. Der Index der Dimension, für die die Array-Elemente innerhalb der anderen Dimension mit der gleichen Ausrichtung gruppiert werden.
Orientation	**Ausrichtung** Wird nur bei einem dreidimensionalen Array angezeigt.

Eigenschaft	Beschreibung
	Die Ausrichtung, in der die Elemente, die der gruppierten Dimension entsprechen, angezeigt werden.

Weitere Informationen zu Arrays finden Sie im *Leitfaden zur Programmierung* in der vorliegenden Dokumentation bzw. in der Natural-Dokumentation für die entsprechende Plattform.

Anzeigen der Statuseigenschaften

Die folgenden Eigenschaften (Properties) sind für die Map selbst und für alle Controls in der Map verfügbar:

Eigenschaft	Beschreibung
Modified	**Geändert** Änderungsstatus (nur Lesezugriff). Wenn dieser Wert auf true gesetzt ist, wurde das ausgewählte Element seit dem letzten Speichern der Map geändert.
Error Status	**Fehlerstatus** Fehlerstatus und Warnstatus (nur Lesezugriff). Mögliche Werte: ■ **Critical Error** Die Map enthält einen kritischen Fehler. ■ **Error** Das ausgewählte Element oder eines seiner untergeordneten Elemente enthält einen Fehler. ■ **Warning** Das ausgewählte Element oder eines seiner untergeordneten Elemente enthält eine Warnung. ■ **None** Keine: Weder das ausgewählte Element noch eines seiner untergeordneten Elemente enthält Fehler oder Warnungen. Fehler und Warnungen werden in Form von Label-Decorations in der **Outline**-Ansicht angezeigt. Informationen über den Schweregrad eines Fehlers oder einer Warnung finden Sie unter *Fehlerbehandlung im Map-Editor*.

Speichern von Maps

Maps werden mit den Standardfunktionen von Eclipse gespeichert.

Wenn jedoch ein dynamisches Layout vorhanden ist, gilt Folgendes:

Wenn Sie eine Map speichern, wird nur die sogenannte Host Map gespeichert. Erst wenn Sie den Editor schließen, wird die Layout Map gespeichert. In diesem Fall erscheint ein Dialog, in dem Sie gefragt werden, ob Sie die Layout Map speichern möchten.

Wenn Sie die Layout Map zu einem früheren Zeitpunkt speichern möchten, gehen Sie wie folgt vor:

Um die Layout Map zu speichern (nur dynamisches Layout):

1 Markieren Sie in der **Outline**-Ansicht die Layout Map.

2 Rufen Sie das Kontextmenü für die Map auf und wählen Sie **Save**.

Oder:

Drücken Sie CTRL+S.

Oder:

Wählen Sie im Menü **File** die Schaltfläche **Save**.

Speichern von Maps (Stow)

Wenn Sie eine Map in der **Natural Server**-Ansicht geöffnet haben, können Sie die Map direkt aus dem Map Editor heraus sowohl in Quellcode- als auch in Objektform abspeichern.

Um die Map per Stow zu speichern (nur Natural Server-Ansicht):

1 Stellen Sie sicher, dass die **Layout**-Seite angezeigt wird.

2 Invoke the context menu for the map and choose **Stow**.

Rufen Sie das Kontextmenü für die Map auf und wählen Sie **Stow**.

Oder:

Drücken Sie CTRL+T.

Validierung von Maps

Wenn Sie eine Map speichern, werden automatisch die folgenden Konsistenzprüfungen durchgeführt, um die Gültigkeit der Map zu bestimmen:

Konsistenzprüfung	Beschreibung
Critical Errors	**Kritische Fehler** Ermittelt, ob die Map einen oder mehrere kritische Fehler enthält, was bedeutet, dass die Map nicht erfolgreich geladen werden konnte. Ist dies der Fall, wird eine Fehlermeldung angezeigt. Dieser Zustand wird in der Regel durch Syntaxfehler innerhalb der `DEFINE DATA`- oder `INPUT`/`WRITE`-Statements verursacht, die durch Öffnen der Map mit dem Source Editor überprüft werden können.
Invalid Data Definitions	**Ungültige Datendefinitionen** Erkennt, ob einer oder mehrere der Datenabschnitte in der Map Fehler enthalten. Ist dies der Fall, wird eine Fehlermeldung angezeigt. Dieser Zustand wird in der Regel dadurch verursacht, dass eine oder mehrere der eingegebenen Datendefinitionen einen Syntaxfehler aufweisen.
Empty Map	**Leere Map** Prüft, ob die Map keine Felder enthält. Ist dies der Fall, wird eine Warnmeldung angezeigt. Beachten Sie, dass leere Maps in Natural syntaktisch nicht gültig sind, da die zugrundeliegenden `INPUT`- oder `WRITE`-Statements das Vorhandensein von mindestens einem Operanden erfordern. Außerdem können Maps, die eine oder mehrere Verarbeitungsregeln, aber keine Felder enthalten, aufgrund dieser Einschränkung nicht korrekt geöffnet werden. Um das Risiko einer solchen Situation zu minimieren, ist das Speichern solcher Maps nicht erlaubt.
Overlapping Fields	**Überschneidungen bei Feldern** Erkennt, ob die Map ein oder mehrere sich überschneidende Felder enthält (und damit syntaktisch ungültig ist). Ist dies der Fall, wird eine Fehlermeldung angezeigt. Außerdem werden die sich überschneidenden Felder automatisch ausgewählt, um Sie bei der Suche nach der Fehlerquelle zu unterstützen.

Wenn Sie die Map zu einem früheren Zeitpunkt validieren wollen, gehen Sie wie unten beschrieben vor.

Um die Map zu validieren:

1 Stellen Sie sicher, dass die **Layout**-Seite angezeigt wird.

2 Rufen Sie das Kontextmenü für die Map auf und wählen Sie **Validate Map**.

Oder:

Drücken Sie CTRL+ALT+V.

Im Falle eines Fehlers wird ein Meldungsdialog angezeigt, der Informationen über den Fehler enthält.

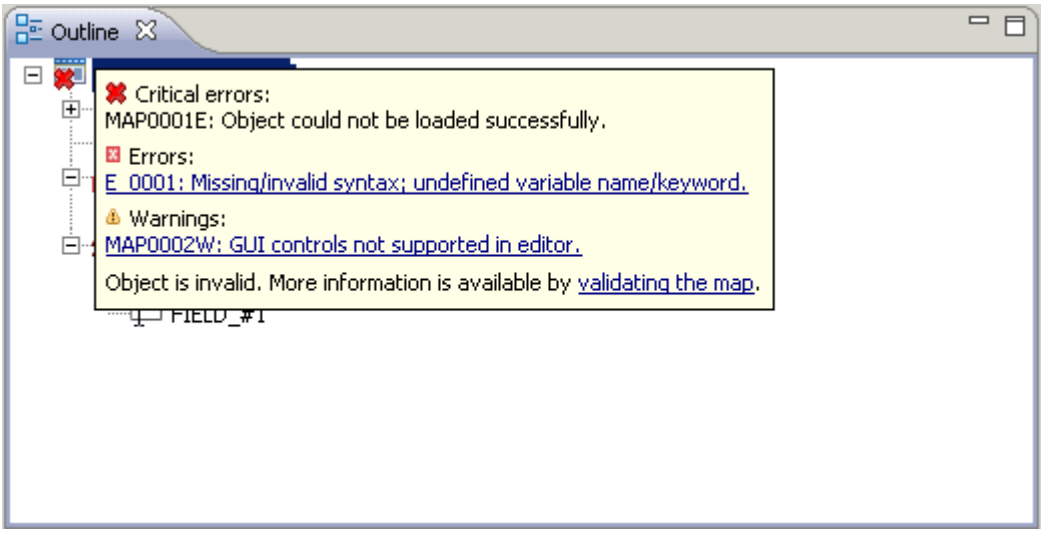
Standardmäßig wird ein Meldungsdialog angezeigt, auch wenn die Validierung erfolgreich war. Dieses Verhalten kann in den Natural-Voreinstellungen geändert werden. Siehe *Map Editor* in *Setting the Preferences* in der *NaturalONE*-Dokumentation.

Fehlerbehandlung im Map-Editor

In der **Outline**-Ansicht werden Fehler und Warnungen in Form von Label-Decorations angezeigt (siehe auch *Viewing the Status Properties* in der *NaturalONE*-Dokumentation). Welche Decoration verwendet wird, hängt vom Schweregrad ab (siehe unten):

Decoration	Beschreibung
	Kritischer Fehler Diese Decoration wird beim Top-Level-Knoten angezeigt, wenn eine oder mehrere Komponenten der Map (z.B. Feld- oder Datendefinitionen) nicht korrekt geladen werden konnten, typischerweise aufgrund eines Syntaxfehlers. Kritische Fehler können nicht innerhalb der aktuellen Editor-Sitzung korrigiert werden. Die genaue Position des Fehlers/der Fehler kann durch Öffnen der Map mit dem Source Editor ermittelt werden.
	Fehler Diese Decoration wird angezeigt, um Fehler darzustellen, die in der aktuellen Editor-Sitzung behoben werden können.
	Warnung Diese Decoration warnt den Benutzer vor Konflikten, die noch nicht schwerwiegend genug sind, um als Fehler angesehen zu werden.

Wenn Sie den Mauszeiger über ein Element in der **Outline**-Ansicht bewegen, das einen Fehler oder eine Warnung verursacht, erscheint ein Tooltip. Dieser Tooltip kann einen oder mehrere Links enthalten. Sie können zu einem Fehler in einer Map navigieren, indem Sie auf den entsprechenden Link klicken. In bestimmten Fällen steht auch ein Link zur Verfügung, der es Ihnen ermöglicht, die Map zu validieren und damit Informationen über den Fehler in einem Meldungsfeld anzuzeigen (siehe auch *Validating Maps* in der *NaturalONE*-Dokumentation).



Sie können auch zu einem Fehler in einer Map navigieren, indem Sie die **Problems**-Ansicht verwenden (siehe die Beschreibung unter *Problems in Your Natural Sources* in der *NaturalONE*-Dokumentation). Die Einstellungen für den Source Editor, insbesondere die Option Show error list, gelten auch für die Seiten List, Data und Source des Map Editors.

Wenn Sie zu einem Fehler navigieren, reagiert der Map Editor auf eine solche Anforderung, indem er versucht, die erste Map-Komponente (Datendefinition, Textkonstante, Datenfeld, Array-Element oder Verarbeitungsregel) zu finden, die die durch die gewählte Fehlermarkierung dargestellte Zeile (oder einen Teil davon) enthält, und (wenn eine solche Komponente gefunden wird) sie auf der **List**-Seite des Map Editors auswählt, oder im Falle einer Verarbeitungsregel auf der **Source**-Seite auswählt. Wenn keine Komponente gefunden wird oder die Navigation aus einem anderen Grund fehlschlägt, wird ein Meldungsdialog angezeigt.

Es ist wichtig zu wissen, dass sich die im Tooltip der **Outline**-Ansicht angezeigten Fehler auf den aktuellen Zustand der Map im Editor beziehen, während sich die in der **Problems**-Ansicht angezeigten Ressourcenmarkierungen auf die Map beziehen, wie sie im Arbeitsbereich gespeichert ist.

Informationen zu dem Fehler oder der Warnung können auch in die **Error Log**-Ansicht geschrieben werden.

Anmerkungen:

1. Die Informationen im Abschnitt *Fehlerbehandlung im Source Editor* gelten auch für die Seiten **List**, **Data** und **Source** des Map Editor.
2. Die Label Decorations für Fehler und Warnungen im Map Editor werden über die Voreinstellungen unter **General > Appearance > Label Decorations** gesteuert. Wenn Sie diese Label Decorations nicht wünschen, gehen Sie einfach auf die oben genannte Voreinstellungsseite und deaktivieren Sie die Option **Natural Map Editor Errors and Warnings**.

7 Benutzung des DDM Editors

- Anlegen eines Datendefinitionsmoduls (DDM) 96
- Informationen zum DDM-Editor 99
- Zugehörige Ansichten 100
- Verwaltung der Felder des DDM 103
- Feld-Attribute 104

Anlegen eines Datendefinitionsmoduls (DDM)

Damit Natural auf eine Datenbankdatei zugreifen kann, ist eine logische Definition der physischen Datenbankdatei erforderlich. Eine solche logische Dateidefinition wird als *Datendefinitionsmodul* (DDM) bezeichnet.

Weitere Informationen zu DDMs finden Sie unter *Zugriff über Datendefinitionsmodule* unter *Natural und Datenbankzugriff* im *Leitfaden zur Programmierung*.

Sie können DDMs direkt aus den Felddefinitionen in einer Datenbank erstellen. Dabei kann es sich um eine Adabas-, SQL- oder XML-Datenbank oder eine andere Datenbank (z. B. VSAM) handeln.

Anmerkung: DDMs aus XML-Datenbanken können nicht für Großrechner-Server erstellt werden.

Um ein DDM zu erstellen:

1 Markieren Sie in der **Project Explorer**-Ansicht oder in der **Natural Navigator**-Ansicht die Library, in der Sie das neue DDM speichern möchten.

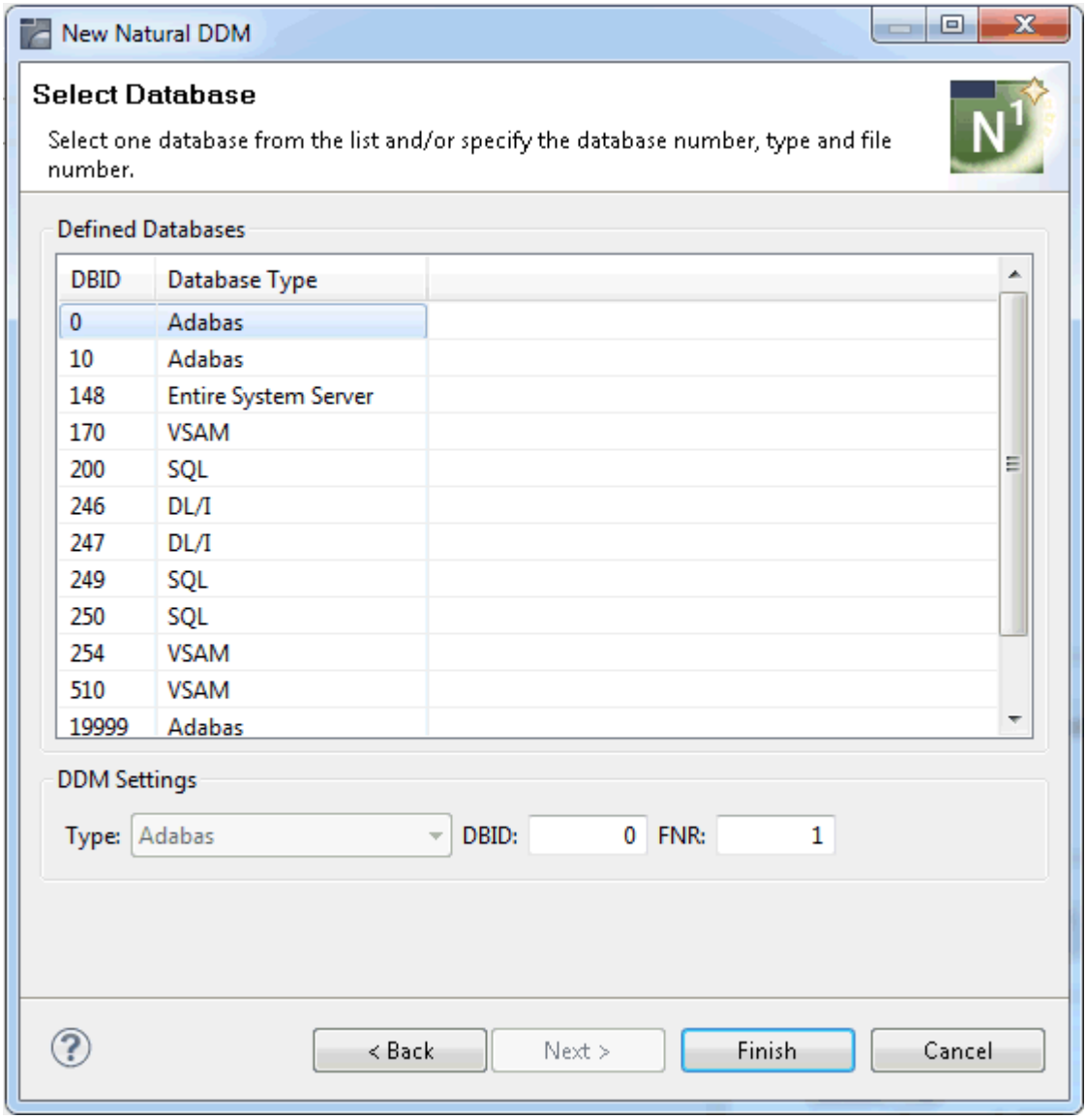
2 Wählen Sie im Menü **File** den Menüpunkt **New > DDM**.

Der Assistent zum Erstellen eines neuen DDMs wird angezeigt (siehe auch *Creating Natural Objects* (*Anlegen von Natural-Objekten*). Außer der Seite, auf der Sie den DDM-Namen eingeben, sind im Assistenten mehrere Seiten vorhanden.

3 Geben Sie einen Namen für das neue DDM ein.

4 Wählen Sie die Schaltfläche **Next**.

NaturalONE prüft nun die verfügbaren Datenbanken auf dem Natural-Server, der für das aktuelle Projekt definiert ist. Wenn der entsprechende Server zurzeit in der **Natural Server**-Ansicht durch Mapping eingebunden ist, werden alle definierten Datenbanken auf diesem Server auf der nächsten Seite angezeigt. Wenn der entsprechende Server nicht eingebunden ist oder gerade nicht aktiv ist, bleibt die Tabelle der definierten Datenbanken leer.



5 Wählen Sie die Datenbank, aus der das DDM erstellt werden soll, in der Tabelle der definierten Datenbanken aus.

Oder:

Geben Sie die entsprechenden DDM-Einstellungen unten auf der Seite ein (dies ist hilfreich, wenn die Tabelle der Datenbanken leer ist):

1. Wählen Sie den gewünschten Datenbanktyp aus dem Dropdown-Listenfeld aus.
2. Geben Sie als Datenbankkennung (DBID) einen numerischen Wert im Bereich von 0 bis 65535 (außer 255) ein.

Wenn Sie als Datenbankkennung eine 0 (Null) eingeben, wird die mit dem Natural-Profilparameter `UDB` im Natural-Parametermodul `NATPARM` angegebene Datenbankkennung verwendet.

3. Eine Dateinummer (FNR) ist nur für Adabas-Datenbanken und alle anderen Datenbanktypen außer SQL und XML erforderlich. Geben Sie einen numerischen Wert im Bereich von `1` bis `5000` ein.

6 Für SQL- und XML-Datenbanken wählen Sie die Schaltfläche **Next**.

Die daraufhin angezeigte Seite hängt von dem von Ihnen gewählten Datenbanktyp ab:

- **SQL**
  Es wird eine Seite angezeigt, auf der Sie die Auswahlkriterien für die Abfrage von SQL-Tabellen angeben müssen:
  - Um alle SQL-Tabellen zur Auswahl aufzulisten, verwenden Sie Sterne (*) in den Textfeldern **Owner** (Eigentümer) und **Name**. Die Sterne (*) sind standardmäßig eingetragen.
  - Um einen bestimmten Bereich von SQL-Tabellen aufzulisten, benutzen Sie die Stern-Notation (*) (z.B. AB* wählt alle SQL-Tabellen aus, deren Namen mit "AB" beginnen).

    Anmerkung: Wenn über die ODBC-Schnittstelle auf eine SQL-Datenbank zugegriffen wird, wird kein Tabellenkatalog bereitgestellt. Das bedeutet, dass Sie nur einen Tabellennamen, nicht aber einen Tabelleneigentümer angeben können.

  Wählen Sie die Schaltfläche **Next**.

  Je nach den Einstellungen Ihrer SQL-Datenbank erscheint ein Dialog zur Datenbankanmeldung, wenn Sie in dieser Sitzung zum ersten Mal auf diese SQL-Datenbank zugreifen. Geben Sie die Benutzerkennung und das Passwort für die Datenbank ein und wählen Sie die Schaltfläche **OK**.

  Auf der nächsten Seite können Sie nun die gewünschte SQL-Tabelle auswählen.
- **XML**
  Es wird eine Seite angezeigt, auf der Sie einen Tamino-Doctype auswählen müssen.

  Ein Datenbank-Anmeldedialog erscheint, wenn Sie in dieser Sitzung zum ersten Mal auf diese Tamino-Datenbank zugreifen. Geben Sie die Benutzerkennung und das Passwort für die Datenbank ein und wählen Sie die Schaltfläche **OK**.

Anmerkung: Für Adabas und andere Datenbanktypen werden keine zusätzlichen Seiten angezeigt.

7 Wählen Sie die Schaltfläche **Finish**.

Wenn die angegebene Datenbank und die Datei verfügbar sind, wird der DDM-Editor aufgerufen und die in dieser Datenbankdatei enthaltenen Felder werden in den DDM-Editor eingelesen.

Ist die angegebene Datenbank nicht aktiv oder kann nicht darauf zugegriffen werden oder existiert die Datei nicht, wird eine entsprechende Fehlermeldung ausgegeben. Dennoch erscheint ein leeres DDM-Editor-Fenster (mit einem einzigen Dummy-Feld), in dem Sie neue Feldattributdefinitionen eingeben und den DDM-Quellcode speichern können. In diesem Fall können Sie jedoch keine Definitionen mit der Beschreibung der Datenbankdatei abgleichen.

Informationen zum DDM-Editor

Wenn Sie ein DDM anlegen oder öffnen, wird der DDM-Editor aufgerufen. Beispiel:

EMPLOYEE.NSD

Type	Level	Short...	Name	Format	Length	Sup...	Descriptor	Header
	1	AA	PERSONNEL-ID	A	8		D	PERSONNEL/ID
*			CNNNNNNN					
*			C=COUNTRY					
G	1	AB	FULL-NAME					
	2	AC	FIRST-NAME	A	20	N		
	2	AD	MIDDLE-I	A	1	N		
	2	AE	NAME	A	20		D	
	1	AD	MIDDLE-NAME	A	20	N		
	1	AF	MAR-STAT	A	1	F		MARITAL/ST...
*			M=MARRIED					
*			S=SINGLE					
*			D=DIVORCED					
*			W=WIDOWED					
	1	AG	SEX	A	1	F		S/E/X
	1	AH	BIRTH	N	6.0		D	DATE/OF/BIR...
	1	AH	N@BIRTH	I	2		D	
G	1	A1	FULL-ADDRESS					
M	2	AI	ADDRESS-LINE	A	20	N		ADDRESS
	2	AJ	CITY	A	20	N	D	

Wenn der DDM-Editor aktiv ist, ist eine entsprechende Symbolleiste verfügbar. Siehe *Verwaltung der Felder des DDM*.

Das Verhalten des DDM-Editors kann durch Ändern der Natural-Voreinstellungen beeinflusst werden. Siehe *DDM Editor* unter *Setting the Preferences* in der *NaturalONE*-Dokumentation.

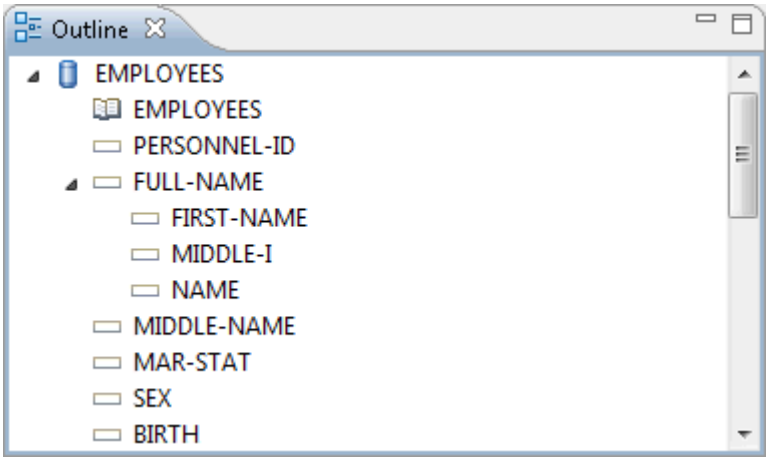
Zugehörige Ansichten

Der DDM-Editor verwendet die folgenden Ansichten der NaturalONE-Perspektive:

- Outline-Ansicht
- Dependencies-Ansicht
- Properties-Ansicht

Outline-Ansicht

Diese Ansicht zeigt die hierarchische Struktur des DDM und der DDM-Felder in einem Baum. Der oberste Knoten enthält den Langnamen des DDM, der erste untergeordnete Knoten stellt den DDM-Header dar und die anderen untergeordneten Knoten stellen die DDM-Felder dar.

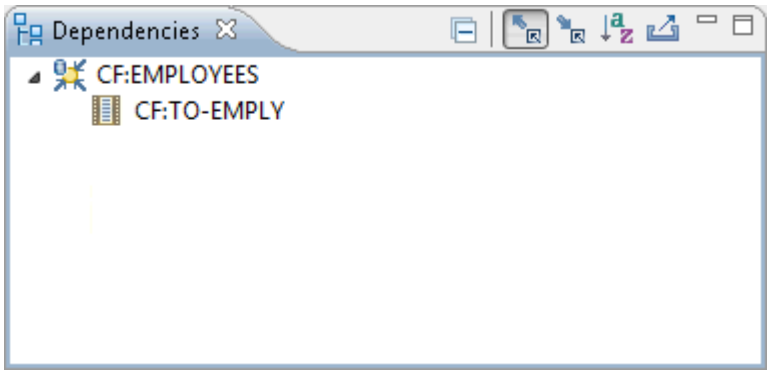


Sie können ein Kontextmenü aufrufen, um Felder hinzuzufügen, auszuschneiden, zu kopieren, einzufügen oder zu löschen. Siehe auch *Verwaltung der Felder des DDM*.

Sie können zu einem bestimmten Element im DDM-Editor navigieren, indem Sie es in der **Outline**-Ansicht auswählen, und umgekehrt. In der **Properties**-Ansicht (siehe unten) wird die Auswahl entsprechend geändert.

Dependencies-Ansicht

Diese Ansicht zeigt die Abhängigkeiten zwischen dem DDM, das gerade im aktiven Editorfenster angezeigt wird, und anderen Objekten. Wenn beispielsweise gerade die passiven Cross-Referenzen angezeigt werden, können Sie alle Natural-Objekte sehen, die auf das gerade in Bearbeitung befindliche DDM verweisen.

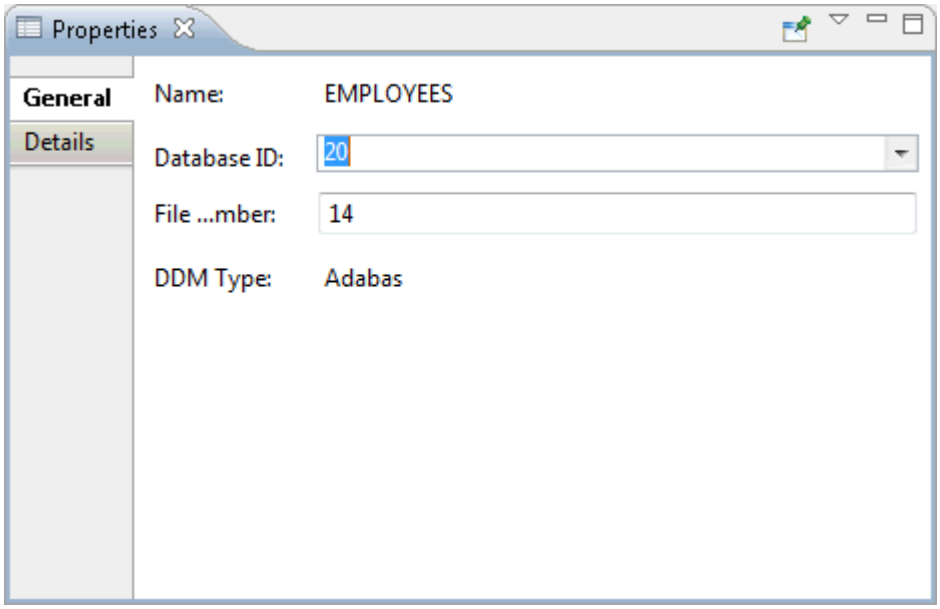


Weitere Informationen zu dieser Ansicht finden Sie unter *Dependencies-Ansicht* in der Beschreibung des Source-Editors.

Properties-Ansicht

In dieser Ansicht werden Informationen zum DDM, zum DDM-Header oder zu dem Feld angezeigt, das gerade im DDM-Editor oder in der **Outline**-Ansicht ausgewählt ist.

Im Allgemeinen sind die in der **Properties**-Ansicht angezeigten Attributwerte nicht änderbar. Ausnahmsweise können hier auch Informationen angegeben werden, die in der Tabellenansicht nicht bearbeitet werden können. Dazu gehören für alle Datenbanktypen die Datenbankkennung (DBID), die Dateinummer (FNR), die Standardreihenfolge und insbesondere die für VSAM-Datenbanken erforderlichen Definitionen.



Die Seite **General** zeigt allgemeine Informationen an. Wenn ein DDM oder DDM-Feld ausgewählt ist, wird der Name des DDM oder DDM-Feldes angezeigt. Wenn der DDM-Header ausgewählt ist, werden der Name des DDM-Headers, die Datenbankkennung, die Dateinummer und der Datenbanktyp angezeigt. Datenbankkennung und Dateinummer können hier geändert werden, der Name des DDM und der Datenbanktyp sind nicht änderbar.

Auf der Seite **Details** werden, falls vorhanden, zusätzliche, speziellere Informationen angezeigt, die von dem aktuell ausgewählten Objekt abhängen:

- Für Adabas-DDMs werden hier Informationen zu gekoppelten Dateien angezeigt. Für Header von Adabas-DDMs wird die Standardreihenfolge angezeigt. Bei Adabas-DDM-Feldern werden die übergeordneten Felder des ausgewählten Superdeskriptors angezeigt.
- Für SQL-DDM-Header wird die Standardreihenfolge angezeigt.
- Für Tamino-DDM-Header werden Informationen über die XML-Schemadefinition, Doctype, Collection und Namespaces angezeigt.

  Für Tamino-DDM-Felder werden feldspezifische Informationen aus dem XML-Schema wie XML-Tag und XML-Pfad angezeigt.
- Für VSAM-DDMs werden erweiterte Informationen über den VSAM-Header und die VSAM-Felddefinitionen angezeigt.

Verwaltung der Felder des DDM

Mit den Kommandos aus dem Menü **Edit** (Bearbeiten) können Sie im DDM-Editor Felder ausschneiden, kopieren, einfügen und löschen sowie Ihre letzte Änderung rückgängig machen und wiederherstellen. Einige dieser Kommandos sind auch verfügbar, wenn Sie das Kontextmenü für eine Zeile im DDM-Editor aufrufen, und sie sind auch in einer Symbolleiste verfügbar. Der DDM-Editor bietet auch eine eigene Symbolleiste, mit der Sie Felder in den DDM-Editor einfügen können.

Darüber hinaus ist es möglich, in der **Outline**-Ansicht ein Kontextmenü aufzurufen. Über die Kommandos in diesem Kontextmenü können Sie ebenfalls Felder hinzufügen, ausschneiden, kopieren, einfügen und löschen.

Die Kommandos **Copy**/**Cut** und **Paste** können verwendet werden, um ein oder mehrere Felder innerhalb des aktuellen DDM-Quellcodes oder zwischen verschiedenen DDM-Quellcodes zu kopieren oder zu verschieben. Die kopierten oder ausgeschnittenen Felder werden in der Zwischenablage abgelegt und können in den aktuellen oder einen anderen DDM-Quellcode eingefügt werden. Die Felder werden entweder vor oder nach einem ausgewählten Feld eingefügt, je nachdem, welche Einfügeposition in den Einstellungen für den DDM-Editor festgelegt wurde.

Im DDM-Editor können Sie mit der TAB-Taste von einem Feld zum anderen wechseln.

Um eine Zeile im DDM-Editor zu markieren (z. B. um eine Zeile zu kopieren oder zu löschen), können Sie entweder auf die Zeile doppelklicken oder die ESC-Taste drücken, damit die gesamte Zeile markiert wird.

Um einen bestimmten Wert hinzuzufügen oder zu ändern, klicken Sie einfach auf die entsprechende Zelle im DDM-Editor. Wenn eine Zeile markiert ist, können Sie auch F2 drücken, um die Zelle zu bearbeiten, die vor der Markierung der Zeile aktiv war (wenn vorher keine Zelle aktiv war, wird die erste Zelle zur Bearbeitung aktiviert). Bei einigen Spalten sind einfache Textfelder vorgesehen, in die Sie Informationen eingeben müssen. Bei anderen Spalten stehen Dropdown-Listenfelder zur Verfügung, aus denen Sie einen Wert auswählen können.

Wenn Sie die Spalten im DDM-Editor neu anordnen möchten, ziehen Sie einfach die Spaltenüberschrift an die gewünschte Position.

Feld-Attribute

Jede Zeile im DDM-Editor stellt ein DDM-Feld dar. Die folgenden Feldattribute können in den verschiedenen Spalten des DDM-Editors angezeigt oder definiert werden.

Alle DDM-Typen haben die folgenden Spalten gemeinsam:

Spalte	Beschreibung
Type	Typ des Feldes. *leer*: Elementares Feld. G: Gruppe. M: Multiples Feld. P: Periodengruppe. *: Kommentarzeile.
Level	Ebenennummer, die dem Feld zugewiesen ist.
Name	Name des Feldes.
Format	Natural-Datenformat eines elementaren Feldes, z. B. A (alphanumerisch).
Length	Standardlänge eines elementaren Feldes.
Descriptor	Typ des Deskriptors. *leer*: Kein Deskriptor. D: Elementarer Deskriptor. H: Hyperdeskriptor. N: Nicht-Deskriptor. P: Phonetischer Deskriptor. S: Subdeskriptor oder Superdeskriptor.
Header	Header, der für jedes in einem `DISPLAY`-Statement angegebene Feld erzeugt werden soll.
Edit Mask	Zu verwendende Editiermaske.
Remarks	Kommentar, der sich auf ein Feld und/oder das DDM bezieht.

Adabas-DDMs und SQL-DDMs zeigen zusätzlich die folgenden Spalten:

Spalte	Beschreibung
Short Name	Zweistelliger Kurzname des entsprechenden Feldes in der Datenbankdatei.
Suppression	Option zur Unterdrückung von Nullwerten. *leer*: Adabas-Standard-Unterdrückung. F: Adabas-Option Festspeicherung. N: Adabas-Option zur Unterdrückung von Nullwerten. M: SQL-Nullwert-Option.
Format Option	Wird nur für den Datenbanktyp ADA2 angezeigt. Adabas-Formatoption für alphanumerische Felder.

Spalte	Beschreibung
SQL Type	Wird nur für SQL-DDMs angezeigt. Informationen, die aus den Datentypen BLOB (binary large object) oder CLOB (character large object) generiert werden, wenn sie in einer Oracle-Datenbank enthalten sind.

Ausführliche Informationen zu den Feldattributen, die in den verschiedenen Spalten definiert werden können, finden Sie in der Beschreibung des DDM-Editors in der Natural-Dokumentation für die entsprechende Plattform.

II Sperrmechanismen für Source-Objekte

8 Sperrmechanismen für Source-Objekte

- Source-Objekte bearbeiten 111
- Objekte speichern 111
- Objekte entsperren 112
- Objekte verschieben, löschen, umbenennen und ersetzen 112

Natural bietet Sperrmechanismen, durch die ein konkurrierendes Ändern von Natural-Source-Objekten verhindert wird. Durch diese Mechanismen können Source-Objekte gesperrt werden, die in einer lokalen Großrechnerumgebung und/oder in einer SPoD-Umgebung (SPoD = Natural Single Point of Development) bearbeitet werden, welche mit einem Großrechner-Server verbunden ist.

Die verschiedenen Sperrmechanismen können Sie aktivieren und deaktivieren, indem Sie den Profilparameter `SLOCK` entsprechend setzen. Weitere Informationen zu diesem Profilparameter siehe *Parameter-Referenz-Dokumentation*.

- **Source-Objekte in lokalen Umgebungen und in SPoD-Umgebungen sperren**
  `SLOCK=PRE` aktiviert die Sperrung von Source-Objekten, die entweder lokal oder in einer SPoD-Umgebung oder mit Natural ISPF oder in gemischten Umgebungen editiert werden.

  `SLOCK=PRE` ist die empfohlene Einstellung für das Arbeiten in gemischten Umgebungen.
- **Source-Objekte in SPoD-Umgebungen sperren**
  Die Standardeinstellung `SLOCK=SPOD` aktiviert die Sperrung von Source-Objekten nur in einer SPoD-Umgebung. Ein Source-Objekt wird dann nur gesperrt, wenn es mit Natural Studio bearbeitet wird. Weitere Informationen siehe *Remote Development Using SPoD* in der *Natural for Windows*-Dokumentation.

  In einer SPoD-Umgebungen sorgt die Einstellung `SLOCK=SPOD` für Kompatibilität mit früheren Natural-Versionen, die die Sperrung unter SPoD unterstützten.
- **Auf letzte Änderungen prüfen**
  Wenn `SLOCK=POST` gesetzt wird, kann das Source-Objekt, das momentan bearbeitet wird, in den Editor-Arbeitsbereich gelesen und von mehreren Benutzern geändert werden. Es kann dann aber nur der Benutzer, der eine Änderung zuerst speichert, das Source-Objekt aktualisieren. Dies geschieht durch den Vergleich der Zeitmarke des in der Datenbank gespeicherten Source-Objekts mit der Zeitmarke des Source-Objekts beim Einlesen in den Editor-Arbeitsbereich. Alle anderen Benutzer erhalten eine entsprechende Fehlermeldung, wenn sie die Source zu speichern versuchen.

  Anmerkung: Dieses Konzept ist nicht kompatibel mit dem SPoD-Sperrkonzept früherer Natural-Versionen.

- **Sperrmechanismen deaktivieren**
  `SLOCK=OFF` deaktiviert alle Sperrmechanismen.

Die folgenden Abschnitte beschreiben das Prinzip der Source-Objekt-Sperrung, wenn der Profilparameter `SLOCK=PRE` in einer lokalen Umgebung gesetzt ist.

Source-Objekte bearbeiten

Bei folgenden Natural-Objekttypen werden die Source-Objekte gesperrt, während sie mit dem entsprechenden Natural-Editor bearbeitet werden:

- Programm
- Subprogramm
- Subroutine
- Copycode
- Helproutine
- Text
- Map
- Local Data Area
- Global Data Area
- Parameter Data Area
- DDM (Datendefinitionsmodul)
- Function

Wenn Sie einen Natural-Editor aufrufen, wird die im Source-Arbeitsbereich enthaltene Source gesperrt. Falls das Source-Objekt, das Sie bearbeiten möchten, schon von einem anderen Benutzer gesperrt worden ist (worauf Sie durch eine entsprechende Meldung hingewiesen werden, könne Sie die Source zwar im Editor anzeigen, jedoch wird oben im Bildschirm kein Objektname angezeigt. Wenn Sie die Source ändern und die Änderung aufbewahren möchten, müssen Sie die Source als neues Source-Objekt unter einem neuen Namen speichern.

Anmerkung: Wenn Sie eine Source mit dem Systemkommando `READ` in den Arbeitsbereich einlesen, erfolgt *keine* Sperrung des Source-Objekts. Die Sperrung einer Source erfolgt nur dann, wenn Sie den Natural-Editor aufrufen.

Objekte speichern

Eine Source, die von einem anderen Benutzer gesperrt worden ist, können Sie nicht mittels `SAVE`- und `STOW`-Kommandos speichern.

Objekte entsperren

Eine im Source-Arbeitsbereich enthaltene gesperrte Source wird entsperrt, wenn eine der folgenden Maßnahmen ausführen:

- Löschen des Source-Arbeitsbereichs
- Einlesen der Source eines anderen Source-Objekts in den Source-Arbeitsbereich
- Anmelden bei einer anderen Library
- Beenden der Natural Session
- Verlassen des Natural-Editors, während die Option **Leave Editor with Unlock** im Editor-Profil gesetzt ist (siehe auch *Allgemeine Standardwerte* in *Editorprofil* im Abschnitt *Allgemeine Informationen*). Mit dieser Option wird festgelegt, ob die im aktuellen Source-Arbeitsbereich enthaltene Source beim Verlassen des Editors entsperrt wird. Dies gilt bei Maps und DDMs.

Um sich die gesperrten Source-Objekte anzeigen zu lassen oder um sie, falls erforderlich, zu entsperren, können Sie das Systemkommando `UNLOCK` benutzen (siehe *Systemkommandos*-Dokumentation.

Objekte verschieben, löschen, umbenennen und ersetzen

Die Sperrung von Objekten wird auch berücksichtigt, wenn Sie das Systemkommando `DELETE` oder `RENAME`, den Object Handler oder die Utility SYSMAIN benutzen.

Wenn Sie ein Source-Objekt verschieben, löschen, umbenennen und ersetzen, wird der Sperrzustand des Objekts geprüft:

- Wenn das Source-Objekt gesperrt ist, wird die Kommandoausführung verweigert.
- Ist das Source-Objekt nicht gesperrt, wird das Kommando ausgeführt.

Einschränkungen

Bei den Utilities SYSRPC, SYSPARM und SYSERR wird die Sperrung von Objekten nicht unterstützt.

III Editoren — Allgemeine Informationen

9 Editoren — Allgemeine Informationen

- Split-Screen-Modus 116
- Editor-Profil 119

Dieser Abschnitt erläutert, welche Natural-Objekte mit welchem Natural-Editor bearbeitet werden. Außerdem wird der Split-Screen-Modus (geteilter Bildschirm) und das Editor-Profil erklärt.

Um einen Natural-Editor aufzurufen, benutzen Sie das Systemkommando `EDIT`. Weitere Informationen siehe *Objekte erstellen und bearbeiten* im Dokument *Natural benutzen*. Hinweise zur Vergabe von Namen beim Bearbeiten oder Speichern von Objekten siehe *Namenskonventionen für Objekte* im Dokument *Natural benutzen*.

Welcher Editor aufgerufen wird, hängt davon ab, welches Objekt Sie bearbeiten möchten.

- Programme, Subprogramme, Subroutinen, Helproutinen, Copycode, Textobjekte, Klassen und Functions werden mit dem **Programm-Editor** erstellt und bearbeitet.
- Globale Datenbereiche (GDAs), lokale Datenbereiche (LDAs) und Parameter-Datenbereiche (PDAs) werden mit dem **Data Area Editor** erstellt und bearbeitet.
- Bildschirmmasken (Maps) und Hilfe-Masken (Help Maps) werden mit dem **Masken-Editor** (Map Editor) erstellt und bearbeitet.
- Predict-Beschreibungen werden im Predict Description Editor bearbeitet. Weitere Informationen siehe (separate) *Predict*-Dokumentation.

Bei jedem der genannten Editoren steht Ihnen ein Online-Hilfe-System zur Verfügung.

Übungen, die Sie in die Benutzung der Hauptfunktionen der Editoren einführen, finden Sie im Dokument *Erste Schritte* und im ***Map Editor Tutorial***.

Zusätzlich zu den Natural-Editoren gibt es den Software AG Editor, der bei mehreren Natural Utilities und Software AG-Produkten verwendet wird. Sie können den Software AG Editor als Alternative zum Natural-Programm-Editor benutzen.

Split-Screen-Modus

Sie können alle drei Natural-Editoren im Split-Screen-Modus (geteilter Bildschirm) benutzen: Sie können die eine Hälfte des Bildschirms zum Bearbeiten eines Objekts benutzen und sich gleichzeitig in der anderen Hälfte ein anderes Natural-Objekt anzeigen lassen. So können Sie im Split-Screen-Modus in der unteren Bildschirmhälfte eine Natural View (siehe auch Datendefinitionsmodule/DDMs), einen Datenbereich / Data Area, eine Predict-Programm-Beschreibung oder ein Natural-Programm betrachten. Darüber hinaus können Sie Einzelheiten aus dem Anzeigebereich in den Bearbeitungsbereich, d.h., in das von Ihnen zurzeit bearbeitete Objekt, übernehmen.

Beispiel:

Die folgende Abbildung zeigt den Programm-Editor-Bildschirm im Split-Screen-Modus, wobei der Bearbeitungsbereich (obere Hälfte) den Quellcode eines Programms und der Anzeigebereich (untere Hälfte) eine Local Data Area (LDA) enthält.

```
>                                      > +  Program     SAGDEMO  Lib SAGTEST

  Top    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7.
    0010 DEFINE DATA LOCAL USING L-INVOIC
    0020             LOCAL USING L-INV-LN
    0030 END-DEFINE
    0040 *
    0050 READ INVOICE-VIEW BY INVOICE-NO FROM 1
    0060 *
    0070   FIND INVOICE-LINE-VIEW WITH INVOICE-NO = INVOICE-NO (0050)
    0080     DELETE
    0090   END-FINE
    0100 *
         ....+....1....+....2....+....3....+....4....+....5....+.. S 16   L 1
  Split All       Local       L-INVOIC  Library SAGTEST
    0010 V 1 INVOICE-VIEW                          INVOICE
    0020   2 CUST-NO                          N   8
    0030   2 INVOICE-NO                       N   8
    0040   2 DATE                             A   8
    0050   2 AMOUNT                           N 9.2
    0000
    0000
    0000
    0000
```

Split-Screen-Kommandos

Die folgenden Kommandos können Sie bei Benutzung des Programm-Editors oder des Data Area Editors im Split-Screen-Modus ein Objekt anzeigen und positionieren. Informationen zum Anzeigen von Objekten im Split-Screen-Modus beim **Masken-Editor** siehe ***Definitionen aus anderen Objekten übernehmen*** in der *Masken-Editor*-Dokumentation.

Alle Kommandos beginnen mit einem `S` oder mit `SPLIT`, um den Split-Screen-Modus anzuzeigen. Das `SPLIT`-Kommando ist ein **cursorsensitives Kommando**. Weitere Informationen siehe im Abschnitt *Programm-Editor*.

Wenn ein Teil eines Kommandos unterstrichen ist (kein Hyperlink!), kann der unterstrichene Teil als Abkürzung für das Kommando verwendet werden.

Kommando	Funktion
SPLIT ++	Mit diesem Kommando können Sie den geteilten Bildschirm an das Ende des Objekts positionieren.
SPLIT B	
SPLIT --	Mit diesem Kommando können Sie den geteilten Bildschirm an den Anfang des Objekts positionieren.
SPLIT T	
SPLIT +	Mit diesem Kommando können Sie den geteilten Bildschirm um eine Seite vorwärts positionieren.
SPLIT +P	

Kommando	Funktion
SPLIT - SPLIT -P	Mit diesem Kommando können Sie den geteilten Bildschirm um eine Seite rückwärts positionieren.
SPLIT +*nnn*	Mit diesem Kommando können Sie den geteilten Bildschirm um *nnn* Zeilen vorwärts positionieren. Diese Funktion gibt es nur beim Programm-Editor.
SPLIT -*nnn*	Mit diesem Kommando können Sie den geteilten Bildschirm um *nnn* Zeilen rückwärts positionieren. Diese Funktion gibt es nur beim Programm-Editor.
SPLIT . or SPLIT END	Split-Screen-Modus beenden.
SPLIT 7 *function-name* [*library*]	Objekttyp Function anzeigen. Dabei ist *function-name* der Name der Function, so wie er in dem Statement DEFINE FUNCTION benutzt wird (nicht der Objektname der Function). Diese Funktion gibt es nur beim Programm-Editor.
SPLIT CLASS *class-name* [*library*]	Objekttyp Class anzeigen. Dabei ist *class-name* der Name der Class, so wie er in dem Statement DEFINE CLASS benutzt wird (nicht der Objektname der Class). Diese Funktion gibt es nur beim Programm-Editor.
SPLIT DATA *name* [*library*]	Objekttyp Data Area (Global, Local, Parameter) anzeigen. Im Data Area Editor können Sie bei *name* Stern-Notation (*) benutzen, um eine Liste des so angegebenen Bereichs von Objekten anzuzeigen.
SPLIT DESCRIPTION *program-name* [*library*]	Die Predict-Beschreibung aus dem Predict-Datendiktionär zu dem bei *program-name* angegebenen Programm anzeigen.
SPLIT FUNCTION *subroutine-name* [*library*]	Objekttyp Subroutine anzeigen. Dabei ist *subroutine-name* der Name der Subroutine, so wie er in dem Statement DEFINE SUBROUTINE benutzt wird (nicht der Objektname der Subroutine). Diese Funktion gibt es nur beim Programm-Editor.
SPLIT PROGRAM *name* [*library*]	Objekttyp Programm, Subprogramm, Subroutine, Helproutine, Copycode, Text, Map, Class, Natural-Kommando-Prozessor, Recording, Adapter, Function oder Resource anzeigen. Im Data Area Editor können Sie bei *name* Stern-Notation (*) benutzen, um eine Liste des so angegebenen Bereichs von Objekten anzuzeigen.

Kommando	Funktion
SPLIT SCAN [*scan-value*]	Quellcode durchsuchen. Dabei ist *scan-value* die Zeichenkette, nach der im aktuellen Quellcode gesucht werden soll. Jede Zeile, die den *scan-value* enthält, wird mit einem Größer-als-Zeichen (>) markiert. Um weiter nach demselben *scan-value* zu suchen, brauchen Sie nur S SC einzugeben.
SPLIT VIEW *name* [SHORT]	Objekttyp View (DDM, entsprechend Definition in Predict oder SYSDDM Utility) anzeigen. Wenn Sie SHORT angeben, wird das DDM in Kurzform ohne Feldüberschrift- und Feld-Editiermarkeninformationen aufgelistet (Kurzform bedeutet, es werden nur die Adabas-Kurznamen und die entsprechenden Natural-Feldnamen angezeigt). Im Data Area Editor können Sie bei *name* Stern-Notation (*) benutzen, um eine Liste des so angegebenen Bereichs von Objekten anzuzeigen.
SPLIT	Vorheriges Objekt anzeigen. Wenn Sie das Kommando SPLIT ohne Parameter absetzen, wird wieder der Quellcode des zuvor im Split-Screen-Bereich angezeigten Objekts angezeigt. Dies gilt für die Dauer der aktuellen Editor-Sitzung. Diese Funktion gibt es nur beim Programm-Editor.

name ist der Name des Objekts, so wie in der Natural-Library und/oder in der Natural-Systemdatei enthalten.

Eine *library* kann nur beim Programm-Editor angegeben werden. Wenn Natural Security eingesetzt wird, kann keine Library angegeben werden.

Editor-Profil

Für den Natural Programm-Editor und den Data Area Editor gibt es ein Editor-Profil, in dem die Einstellungen festgelegt sind, die während der aktuellen Programm-Editor- oder Data Area Editor-Sitzung gelten. Dazu gehören zum Beispiel die Standardbelegungen der PF- und der PA-Tasten. Sie können hier die von Ihnen bevorzugten Editor-Profil-Einstellungen definieren.

Dieser Abschnitt behandelt folgende Themen:

- Editor-Profil aufrufen
- Editor-Profil-Bildschirm
- Zusätzliche Optionen
- Editor-Standardwerte - Editor Defaults

- Allgemeine Standardwerte - General Defaults
- Farbdefinitionen - Color Definitions
- Direktkommandos
- Editor-Profil-Verwaltungsbildschirme verlassen

Editor-Profil aufrufen

Um Ihr aktuelles Editor-Profil aufzurufen:

- Setzen Sie in der Kommandozeile des Programm-Editors bzw. des Data Area Editors folgendes Kommando ab:

```
PROFILE ↩
```

Der **Editor Profile**-Bildschirm wird angezeigt (Beispiel):

```
11:06:16                ***** NATURAL EDITORS *****                2010-11-22
                            - Editor Profile -

Profile Name .. SAG_____

PF and PA Keys
  PF1 ... HELP__________   PF2 ... _________________   PF3 ... EXIT__________
  PF4 ... _________________   PF5 ... _________________   PF6 ... _________________
  PF7 ... -_________________   PF8 ... +_________________   PF9 ... _________________
  PF10 .. SC=_______________   PF11 .. _________________   PF12 .. CANCEL_____
  PF13 .. _________________   PF14 .. _________________   PF15 .. MENU__________
  PF16 .. _________________   PF17 .. _________________   PF18 .. SCAN__________
  PF19 .. --_______________   PF20 .. ++_______________   PF21 .. _________________
  PF22 .. _________________   PF23 .. _________________   PF24 .. _________________
  PA1 ... _________________   PA2 ... SCAN__________   PA3 ... _________________

Automatic Functions
  Auto Renumber .. Y   Auto Save Numbers .. 0__   Source Save into .. EDITWORK

Additional Options .. N

Command ===>                                                                ↩
                                                                            ↩

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

Dieser Bildschirm zeigt Ihr eigenes Editor-Profil, wenn zu Ihrer Benutzerkennung ein individuelles Profil existiert (siehe Feld **Profile Name**). Falls ein solches Profil nicht existiert, wird stattdessen das Standard-Profil `SYSTEM` angezeigt. Das Profil `SYSTEM` wird aus der User Exit Routine USR0070P gelesen (siehe *Operations*-Dokumentation).

Um ein spezifisches Editor-Profil aufzurufen:

- Setzen Sie in der Kommandozeile des Programm-Editors bzw. des Data Area Editors folgendes Kommando ab:

  ```
  PROFILE profile-name
  ```

  dabei ist *profile-name* eine gültige Benutzerkennung oder SYSTEM.

  Wenn Sie eine Editor-Sitzung beginnen und das Kommando PROFILE zusammen mit Ihrer Benutzerkennung als Profilnamen eingeben, wird Ihr Profil immer direkt aus der Datenbank aufgerufen. Während der aktuellen Editor-Sitzung vorgenommene, aber nicht in der Datenbank gespeicherte Änderungen gelten dann nicht. Deshalb sollten Sie, um Ihr aktuelles Profil aufzurufen, nur das Kommando PROFILE absetzen.

Vorsicht: Änderungen am aktuellen Editor-Profil gehen verloren, wenn Sie das Systemkommando LOGON absetzen oder wenn Sie einen anderen Editor öffnen, bevor Sie die Profiländerungen in der Datenbank gespeichert haben.

Editor-Profil-Bildschirm

Der **Editor Profile**-Bildschirm enthält folgende Felder:

Feld	Erläuterung
Profile Name	Der Name des Editor-Profils. Es wird Ihr eigenes Editor-Profil angezeigt. Falls ein solches Profil nicht existiert, können Sie das Standard-Profil so anpassen, dass es Ihren Erfordernissen entspricht. Dazu müssen Sie den vorgegebenen Profilnamen SYSTEM mit Ihrer Benutzerkennung überschreiben und das umbenannte Profil in der Datenbank speichern. Wenn Sie den Namen Ihres Profils mit einem anderen gültigen Profilnamen (d.h., eine andere gültige Benutzerkennung) überschreiben und dann Enter drücken, wird das Profil des entsprechenden Benutzers aufgerufen. Pro Benutzerkennung kann nur ein Profil eingerichtet werden, und die am Profil eines anderen Benutzers vorgenommenen Änderungen gelten nur für die aktuelle Editor-Sitzung. Sie können nicht in der Datenbank gespeichert werden. Sie können aber den Profilnamen des Profils eines anderen Benutzers mit Ihrer eigenen Benutzerkennung überschreiben und dann das umbenannte Profil in der Datenbank speichern.
PF and PA Keys	In den Feldern unter dieser Überschrift werden die Kommandos angezeigt, mit denen die PF- und PA-Tasten belegt sind. Sie können diesen Tasten jedes beliebige Natural-Editor oder Natural-Systemkommando zuweisen. Sie können auch eine Kombination aus mehreren Kommandos (durch ein Komma voneinander abgetrennt) angeben.

Feld	Erläuterung
Auto Renumber	Ein `Y` gibt an, dass der Quellcode im Programm-Editor automatisch neu nummeriert werden soll, wenn einer der folgenden Fälle eintritt: ■ Das Kommando `CATALOG`, `CHECK`, `RUN`, `SAVE` oder `STOW` wird abgesetzt. ■ Das Zeilenkommando `.I` wird abgesetzt und für die einzufügende Zeile steht keine Zeilennummer zur Verfügung. **Anmerkung:** Siehe auch *Sourcecode-Zeilennummern umnummerieren* im *Leitfaden zur Programmierung*.
Auto Save Numbers	Wenn ein numerischer Wert eingeben wird, dann wird nach der hier angegebenen Anzahl an Änderungen, d.h. nach Eingabe von Source-Daten verändernden Informationen am Bildschirm, automatisch einer Kopie der aktuellen Source in dem Source-Objekt gespeichert, welches im Feld **Source Save into** angeben ist. **Auto Save Numbers** gilt auch für den **Masken-Editor**.
Source Save into	Der Name des Source-Objekts, in das automatisch eine Kopie der Source in der aktuellen Library gespeichert werden soll. Der Standard-Name `EDITWORK` kann geändert werden. Das angegebene Source-Objekts wird jedes Mal, wenn die im Feld **Auto Save Numbers** angegebene Anzahl an Änderungen überschritten wird, überschrieben.
Additional Options	Siehe Abschnitt ***Zusätzliche Optionen***.

Zusätzliche Optionen

Wenn Sie im Profil-Feld **Additional Options** ein `Y` eingeben, erscheint ein Fenster. Es enthält folgende Optionen:

- **Editor Defaults**, siehe *Editor-Standardwerte*
- **General Defaults**, siehe *Allgemeine Standardwerte*
- **Color Definitions**, siehe *Farbdefinitionen*

Ein Pluszeichen (+) vor einer Option zeigt an, dass in dem entsprechenden Fenster oder mittels eines entsprechenden Editor-Kommandos schon einige Werte gesetzt worden sind.

Geben Sie ein `Y` ein, um die gewünschte Option zu wählen.

Alle Optionen werden im folgenden Abschnitt erläutert.

Editor-Standardwerte - Editor Defaults

Option	Erläuterung
Escape Character for Line Command	Das Fluchtzeichen, das vor jedem Zeilenkommando angegeben werden muss. Standard-Fluchtzeichen ist ein Punkt (.).
Empty Line Suppression	Diese Option gilt beim Einfügen von Zeilen in den Quellcode eines Objekts (Ausnahme: Text-Objekte) mittels Zeilenkommando `.I`. Weitere Informationen siehe ***Programm-Editor*** und ***Data Area Editor***. Mögliche Einstellungen: `Y` — Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt. Dies ist die Standard-Einstellung. `N` — Leer gelassenen Zeilen werden beim Drücken von Enter *nicht* aus der Source entfernt.
Empty Line Suppression for Text	Diese Option gilt beim Einfügen von Zeilen in den Quellcode eines Text-Objekts mittels Zeilenkommando `.I`. Weitere Informationen siehe ***Programm-Editor***. Mögliche Einstellungen: `Y` — Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt. `N` — Leer gelassenen Zeilen werden beim Drücken von Enter *nicht* aus der Source entfernt. Dies ist die Standard-Einstellung.
Source Size Information	`Y` — Die tatsächliche Größe des in Bearbeitung befindlichen Objekts und der verbleibende, verfügbare Platz werden in der Informationszeile unten im Editor-Bildschirm angezeigt. Im Programm-Editor wird außerdem der Programmiermodus (Reporting oder Structured) in der Informationszeile oben im Editor-Bildschirm angezeigt. `N` — Diese Informationen werden nicht angezeigt.
Source Status Message	`Y` — **Programm-Editor:** Jedes Mal wenn der Quellcode geändert, geprüft (`CHECK`), gespeichert (`SAVE`), katalogisiert (`CATALOG`) oder katalogisiert (kompiliert) und gespeichert (`STOW`) wird, wird in der Informationszeile oben im Editor-Bildschirm eine Transaktionsmeldung angezeigt. Zusätzlich wird in der Editor-Kommandozeile ein Stern (*) angezeigt, falls der Quellcode noch nicht gespeicherte Änderungen enthält. Weitere Informationen siehe ***Änderungsanzeige*** in der *Programm-Editor*-Dokumentation. **Data Area Editor:**

Option	Erläuterung
	In der Informationszeile oben im Editor-Bildschirm wird ein Stern (*) angezeigt, falls der Quellcode noch nicht gespeicherte Änderungen enthält. Weitere Informationen siehe ***Änderungsanzeige*** in der *Data Area Editor*-Dokumentation.
N	Es wird keine Transaktionsmeldung und/oder kein Stern (*) angezeigt.
Absolute Mode for SCAN/CHANGE	Diese Option gilt, wenn das Editor-Kommando SCAN oder CHANGE eingegeben wird.
Y	Entspricht dem Editor-Kommando SET ABS ON.
N	Entspricht dem Editor-Kommando SET ABS OFF.
	Siehe *Editor-Kommandos* in den Abschnitten ***Programm-Editor*** und ***Data Area Editor***.
Range Mode for SCAN/CHANGE	Diese Option gilt, wenn das Editor-Kommando SCAN oder CHANGE eingegeben wird.
Y	Entspricht dem Editor-Kommando SET RANGE ON.
N	Entspricht dem Editor-Kommando SET RANGE OFF.
	Siehe ***Editor-Kommandos*** im Abschnitt *Programm-Editor*.
Ignore Case Mode for SCAN/CHANGE	Diese Option gilt bei der Eingabe des Editor-Kommandos SCAN oder CHANGE. Diese Option ist nur dann relevant, wenn die Option **Editing in Lower Case** auf Y gesetzt ist, damit der gesuchte Wert in Groß- und Kleinbuchstaben eingeben werden kann. Y Entspricht dem Editor-Kommando SET IGNORECASE ON. Das bedeutet, die Suche findet jede beliebige Kombination aus Groß- und Kleinbuchstaben. N Entspricht dem Editor-Kommando SET IGNORECASE ON. Das bedeutet, es wird nach dem exakten Wert gesucht. Siehe ***Editor-Kommandos*** im Abschnitt *Programm-Editor*.
Direction Indicator	Die Richtungsanzeige zeigt die Richtung (+ oder -) an, in der bestimmte Editor-Kommandos funktionieren sollen. Siehe auch *Editor-Kommando Line* in den Abschnitten ***Programm-Editor*** und ***Data Area Editor***.

Option	Erläuterung
Numeric Input as Line Number	Bestimmt, ob eine numerische Eingabe an den ersten vier Stellen einer Quellcodezeile die Zeilennummer ändern, wenn Sie Enter drücken. Die Eingabe wird dann nicht als Text übernommen. `Y` — Eine numerische Eingabe wird als Quellcodezeilennummer übernommen. Entspricht dem Editor-Kommando `SET SEQ OFF`. `N` — Eine numerische Eingabe wird als Text übernommen. Entspricht dem Editor-Kommando `SET SEQ ON`. Siehe ***Editor-Kommandos*** im Abschnitt *Programm-Editor*.

Allgemeine Standardwerte - General Defaults

Option	Erläuterung
Editing in Lower Case	`Y` — Kleinbuchstaben im Quellcode werden *nicht* automatisch in Großbuchstaben umgesetzt. `N` — Kleinbuchstaben im Quellcode werden automatisch in Großbuchstaben umgesetzt. Die automatische Umsetzung erfolgt standardmäßig. **Vorsicht:** Benutzen Sie *nicht* die Terminal-Kommandos `%L` oder `%U` innerhalb des Editors.
Dynamic Conversion of Lower Case	Diese Option ist nur dann relevant, wenn die Option **Editing in Lower Case** auf `Y` gesetzt ist. `Y` — Alle Kleinbuchstaben im Quellcode werden automatisch in Großbuchstaben umgesetzt - mit folgenden Ausnahmen: ■ Textzeichenketten, die durch Apostrophe eingeschlossen sind, ■ Textzeichenketten, die durch Ausrufungszeichen eingeschlossen sind, ■ Kommentare.

Option	Erläuterung
	Diese bleiben so, wie Sie sie eingeben (siehe auch Abschnitt *Programm-Editor*).
N	Der Quellcode bleibt so, wie Sie ihn eingeben.
Position of Message Line	Zeigt die Position der Meldungszeile an. Mögliche Werte sind: TOP (oben), BOT (unten), *nn* und *-nn*.
Cursor Position in Command Line	Y Zeigt an, dass der Cursor in der Editor-Kommandozeile positioniert wird, nachdem der Quellcode geändert worden ist und Sie Enter gedrückt haben.
	N Der Cursor bleibt in der Zeile des Quellcode-Bereichs, nachdem der Quellcode geändert worden ist und Sie Enter gedrückt haben.
Stay on Current Screen	Bestimmt, ob der aktuelle Bildschirm angezeigt bleibt, wenn Sie Enter drücken.
Y	Entspricht dem Editor-Kommando SET STAY ON.
N	Entspricht dem Editor-Kommando SET STAY OFF.
	Siehe *Editor-Kommandos* in den Abschnitten *Programm-Editor* und *Data Area Editor*.
Prompt Window for Exit Function	Y Wenn Sie das Kommando EXIT in der Editor-Kommandozeile absetzen, wird ein Bestätigungsfenster angezeigt, (siehe auch *Exit-Funktion* in den Abschnitten *Programm-Editor* und *Data Area Editor*).
ISPF Editor as Programm-Editor	Y Anstelle des Natural-Programm-Editors wird Natural ISPF aufgerufen (falls installiert).
Leave Editor with Unlock	Y Der Quellcode wird beim Verlassen des Editors entsperrt.
N	Der Editor wird verlassen (Standard-Einstellung).
C	Der Quellcode wird beim Verlassen des Editors entsperrt und der Inhalt des Source-Arbeitsbereich wird komplett gelöscht.
	Weitere Informationen siehe *Sperrmechanismen für Source-Objekte*.

Farbdefinitionen - Color Definitions

In diesem Fenster können Sie Farben angeben, in denen die verschiedenen Elemente des Editor-Arbeitsbereichs und des Split-Screen-Bereichs im Programm-Editor bzw. Data Area Editor angezeigt werden sollen.

Zur Anzeige einer Liste mit Farbcodes geben Sie ein Fragezeichen (?) in ein beliebiges Eingabefeld ein und drücken Sie PF1.

Element	Beschreibung
Command Line	Editor-Kommando-Zeile am oberen Rand des Editor-Arbeitsbereichs.
Label Indicator	Äußerste linke Spalte des Editor-Bildschirms; Verwendung zum Beispiel, um eine Quellcode-Zeile zu markieren, in der ein bestimmtes Kommando ausgeführt worden ist (zum Beispiel die Zeilenkommandos `.X` und `.Y`).
Line Numbers	Spalten der Quellcode-Zeilennummern (nur beim Programm-Editor).
Editor Lines	Quellcode-Zeilen, die sich zurzeit im Editierbereich und/oder im Split-Screen-Bereich befinden.
Scan and Error Line	Alle Zeilen, die als Ergebnis einer Suchoperation mit einem `S` oder, im Split-Screen-Modus, mit einem Größer-als-Zeichen (>) markiert sind, eine Zeile, in der ein Fehler festgestellt wurde (markiert mit `E`, betrifft nur den Programm-Editor), und die Fehlermeldungszeile selbst.
Information Text	Alle Felder in der Informations- oder Kommandozeile mit allgemeinen Informationen, beispielsweise der Objekttyp (zum Beispiel Programm), sich zurzeit im Editierbereich befindet.
Information Value	Alle Felder in der Informations- oder Kommandozeile mit benutzerspezifischen Informationen, zum Beispiel der Name des Objekts, das sich zurzeit im Editierbereich befindet.
Information Line	Obere und untere Informationszeile im Editor-Bildschirm.

Direktkommandos

Im **Editor Profile**-Bildschirm stehen folgende Direktkommandos und PF-Tasten zur Verfügung. Ein Direktkommando wird in der Kommandozeile des Bildschirms eingegeben.

Kommando	Entsprechende PF-Taste	Beschreibung
`RESET`	PF6	Setzt alle Parameters des Editor-Profils auf die Einstellungen zurück, die angezeigt wurden, als der **Editor Profile**-Bildschirm zuletzt in der Datenbank gespeichert wurde. Das Zurücksetzen betrifft nur die Parametereinstellungen des aktuellen Editor-Profils. Es betrifft *nicht* das in der Datenbank gespeicherte Editor-Profil.
`SAVE`	PF5	Speichert alle aktuellen Editor-Profil-Einstellungen sowohl für die aktuelle Editor-Sitzung als auch in der Datenbank. `SAVE` beendet *nicht* die Profil-Wartungsfunktion.

Kommando	Entsprechende PF-Taste	Beschreibung
DELETE	PF11	Löscht das Editor-Profil in der Datenbank. Bevor jedoch das Profil gelöscht wird, erscheint ein Bestätigungsfenster, in dem Sie den Namen des Profils eingeben und Enter drücken müssen, um die Löschung des Profils zu bestätigen. Sie können nur Ihr eigenes Profil löschen.
EXIT	PF3	Beendet die aktuelle Profil-Wartungsfunktion unabhängig von der Einstellung **Prompt Window for Exit Function** im Fenster **General Defaults**. Falls nicht gespeicherte Profiländerungen existieren, werden Sie aufgefordert, diese Änderungen in der Datenbank zu speichern. Siehe auch *Editor-Profil-Verwaltungsbildschirme verlassen*.
CANCEL	PF12	Bewirkt den Abbruch der Profil-Wartungsfunktion, danach Rückkehr zum Bildschirm, aus dem diese Funktion aufgerufen wurde. Änderungen am Profil haben keine Auswirkung auf die aktuelle Editor-Sitzung.

Editor-Profil-Verwaltungsbildschirme verlassen

Nach dem Ändern von Parametereinstellungen im aktuellen Editor-Profil haben Sie folgende Möglichkeiten, um das betreffende Verwaltungsfenster schließen bzw. den **Editor Profile**-Bildschirm zu verlassen:

- Drücken Sie Enter.

  Das aktuelle Verwaltungsfenster schließt sich. Wenn kein Fenster geöffnet ist, kehren Sie zum Eingabeaufforderungszeichen des Programm-Editors bzw. des Data Area Editors zurück.

  Sämtliche Änderungen werden für das aktuelle Editor-Profil gespeichert. Siehe **Exit without Saving** weiter unten.

- Oder:
  Drücken Sie PF3 (Exit) oder geben Sie EXIT in der Kommandozeile des **Editor Profile**-Bildschirms ein (wenn kein Fenster geöffnet ist).

  Wenn an dem aktuellen Editor-Profil keine Änderungen vorgenommen wurden, schließt sich das aktuelle Verwaltungsfenster. Wenn kein Fenster geöffnet ist, kehren Sie zum Eingabeaufforderungszeichen des Programm-Editors bzw. des Data Area Editors zurück.

  Falls Sie Änderungen an dem aktuellen Editor-Profil vorgenommen haben, erscheint das **EXIT Function**-Fenster. Sie können dann eine der folgenden Optionen wählen, indem Sie entweder den Cursor auf die Option positionieren oder neben der Option ein beliebiges Zeichen eingeben und dann Enter drücken:

Funktion	Erläuterung
Save and Exit	**Speichern und Verlassen:** Rückkehr zum Bildschirm, aus dem die aktuelle Profil-Wartungsfunktion aufgerufen wurde. Alle am aktuellen Editor-Profil vorgenommenen Änderungen werden gespeichert. Die Änderungen werden sowohl für die aktuelle Editor-Sitzung und in der Datenbank gespeichert. Wenn Sie jedoch mit dem Editor-Profil eines anderen Benutzers arbeiten, können die an diesem Profil vorgenommenen Änderungen nicht in der Datenbank gespeichert. Sie gelten nur für die aktuelle Editor-Sitzung; es wird eine entsprechende Meldung ausgegeben.
Exit without Saving	**Verlassen ohne Speichern:** Rückkehr zum Bildschirm, aus dem die aktuelle Profil-Wartungsfunktion aufgerufen wurde. Alle am aktuellen Editor-Profil vorgenommenen Änderungen gelten nur für die aktuelle Editor-Sitzung; sie werden *nicht* in der Datenbank gespeichert. Das Verlassen ohne Speichern ist gleichbedeutend mit dem Drücken von Enter.
Resume Function	**Funktion fortsetzen:** Das Aufforderungsfenster schließt sich. Sie können mit der Bearbeitung der aktuellen Wartungsfunktion fortfahren.

IV Programm-Editor

10 Programm-Editor

- Programm-Editor aufrufen 134
- Editor-Kommandozeile 135
- Obere Informationszeile 136
- Editierbereich 136
- Untere Informationszeile 138
- Editor-Kommandos 138
- Editor-Kommandos zum Positionieren 147
- Zeilenkommandos 148
- Spezielle PF-Tastenfunktionen 150
- Cursorsensitive Kommandos 154
- Quellcode speichern und katalogisieren 155
- Exit-Funktion 156

Der Natural-Programm-Editor wird benutzt, um den Quellcode eines Natural-Objekts des Typs Programm, Subprogramm, Subroutine, Helproutine, Copycode, Text, Class oder Function zu erstellen und zu ändern.

Anmerkung: Der Programm-Editor ist standardmäßig in Ihrer Umgebung abgeschaltet (deaktiviert). Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung* und *Abgeschaltete CUI-Editoren*.

Verwandtes Thema:

Informationen zur Unterstützung von Unicode und Codepages bei den Natural-Editoren siehe *Editoren in der SPoD-Umgebung* in der *Unicode- und Codepage-Unterstützung*-Dokumentation.

Programm-Editor aufrufen

Um den Programm-Editor aufzurufen:

- Benutzen Sie das Systemkommando `EDIT`. Weitere Informationen siehe *Systemkommandos*-Dokumentation.

  Wenn der Programm-Editor aufgerufen wird, erscheint der Editor-Bildschirm (Beispiel):

```
>                                      > +  Program     SAGDEMO  Lib SAGTEST
All    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
  0010 ** EXAMPLE 'SAGDEMO': DISPLAY
  0020 ************************************************************************
  0030 DEFINE DATA LOCAL
  0040 1 VIEWEMP VIEW OF EMPLOYEES
  0050   2 PERSONNEL-ID
  0060   2 NAME
  0070   2 BIRTH
  0080   2 JOB-TITLE
  0090 END-DEFINE
  0100 *
  0110 READ (3) VIEWEMP BY BIRTH
  0120   DISPLAY PERSONNEL-ID NAME JOB-TITLE
  0130 END-READ
  0140 END
  ....
  0280
       ....+....1....+....2....+....3....+....4....+....5....+... S 14   L 1
```

  Der Editor-Bildschirm hat folgende Bestandteile (von oben nach unten): die **Editor-Kommandozeile**, die **obere Informationszeile**, den **Editierbereich** und die **untere Informationszeile**. Diese Bestandteile werden in den folgenden Abschnitten erklärt.

Anmerkung: Falls Natural ISPF installiert ist und falls im **Editor-Profil** die Option **ISPF Editor as Programm-Editor** auf `Y` gesetzt ist, dann wird anstelle des Programm-Editors entweder das Natural ISPF Hauptmenü (wenn das `EDIT`-Kommando ohne Objektname eingegeben wurde) oder der Natural ISPF Editor-Bildschirm mit dem angegebenen Objekt aufgerufen.

Editor-Kommandozeile

Die Editor-Kommandozeile wird durch ein Größer-als-Zeichen (>) oben am linken Bildschirmrand angezeigt. In der Kommandozeile können Sie Folgendes eingeben:

- Ein beliebiges Natural-Systemkommando.

  Beispiel: Sie können das Systemkommando `CHECK` benutzen, um die Quellcode-Syntax zu prüfen, und das Systemkommando `SAVE`, um den Quellcode zu speichern (siehe auch ***Quellcode speichern und katalogisieren***).

  Informationen zu weiteren Systemkommandos, die zur Pflege und Benutzung von Source-Objekten dienen, siehe *Anwendungen mit Natural-Programmobjekten pflegen* in der *Systemkommandos*-Dokumentation.
- Ein oder mehrere **Editor-Kommandos**.
- Den Namen eines Natural-Programms, das ausgeführt werden soll.

Außerdem enthält die oberste Zeile folgende Bestandteile (von links nach rechts):

Richtungsanzeige: + oder -	Die Richtungsanzeige kann gesetzt werden, um die Ausführungsrichtung der Editor-Kommandos `ADD` und `SCAN` und der Zeilenkommandos `.C`, `.I` und `.M` zu steuern. Das Pluszeichen (+) steht für *nach* und das Minuszeichen (-) für *vor*. Die genaue Interpretation können Sie der Beschreibung des betreffenden Kommandos entnehmen. Siehe auch die Editor-Profil-Option **Direction Indicator** im Abschnitt ***Editor-Profil***.
Objekttyp	Der Typ des Objekts, das sich zurzeit im Quellcode-Arbeitsbereich befindet. Wenn beim Aufrufen des Programm-Editors kein Objekttyp oder Objektname angegeben wird, dann wird standardmäßig **Program** als Objekttyp angezeigt. Um den Objekttyp zu ändern, können Sie das Editor-Kommando `SET TYPE` benutzen.
Änderungsanzeige: *	Ein Stern (*) zeigt an, ob der Quellcode, der sich zurzeit im Quellcode-Arbeitsbereich befindet, noch nicht gespeicherte Änderungen enthält. Der Stern (*) erscheint auch bei neuem Quellcode, der noch nicht als Source-Objekt gespeichert worden ist. Der Stern (*) ist nur dann sichtbar, wenn die Editor-Profil-Option **Source Status Message** auf `Y` gesetzt ist (siehe ***Editor-Profil***). Der Stern (*) verschwindet, wenn Sie den Quellcode erfolgreich speichern und katalogisieren, indem Sie ein `SAVE`- oder `STOW`-Kommando ausführen.

	Siehe auch ***Exit-Funktion***.
Objektname	Der Name des Objekts, das sich zurzeit im Quellcode-Arbeitsbereich befindet. Wenn der Quellcode-Arbeitsbereich leer ist oder wenn der Quellcode noch nicht als Source-Objekt mit dem Kommando `SAVE`, `CATALOG` oder `STOW` gespeichert worden ist, wird kein Name angezeigt.
Lib	Die Library, bei der Sie zurzeit angemeldet sind.

Obere Informationszeile

Die obere Informationszeile des Editor-Bildschirms enthält ein Lineal und folgende Informationen:

- Eine Meldung, die eine Objekt-Änderung anzeigt. Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Status Message** auf `Y` gesetzt ist (siehe ***Editor-Profil***).
- Der zurzeit wirksame Programmiermodus (Structured oder Reporting). Wenn ein Natural-Objekt in den Quellcode-Arbeitsbereich eingelesen wird, wird der Programmiermodus auf den Modus gesetzt, der eingestellt war, als das Objekt mit dem Kommando `SAVE` oder `STOW` gespeichert wurde. Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Size Information** auf `Y` gesetzt ist (siehe ***Editor-Profil***).

  Informationen zu den Unterschieden zwischen Structured- und Reporting-Modus siehe *Zweck der Programmiermodi* im *Leitfaden zur Programmierung*.

Editierbereich

Der Editierbereich des Editor-Bildschirms enthält nummerierte Zeilen, in denen Sie Quellcode eingeben oder ändern können.

Der Editierbereich ist entweder leer oder er enthält den Quellcode, der zuletzt mit dem Kommando `EDIT` oder `READ` in den Quellcode-Arbeitsbereich eingelesen wurde. Siehe Programm-Beispiel im Abschnitt ***Programm-Editor aufrufen***.

Wenn Sie den Quellcode eines existierenden Objekts einlesen, wird der gesamte Quellcode in den Quellcode-Arbeitsbereich geladen und steht dort zum Bearbeiten zur Verfügung. Je nach Größe des Quellcodes kann es sein, dass nicht alle Zeilen, die zum Quellcode gehören, im Editierbereich angezeigt werden. In diesem Fall müssen Sie im Quellcode weiter nach unten blättern (siehe ***Editor-Kommandos zum Positionieren***), um zu der Zeile zu gelangen, die Sie ansehen oder ändern möchten.

Wenn Sie den Split-Screen-Modus benutzen, zeigt der Editierbereich weniger Quellcode-Zeilen an. Siehe auch ***Split-Screen-Modus***.

Sie haben mehrere Möglichkeiten, um Quellcode zu erstellen oder zu bearbeiten:

- Sie können den Code direkt in der betreffenden Quellcode-Zeile eingeben oder ändern.
- Sie können eines oder mehrere **Editor-Kommandos** (siehe entsprechenden Abschnitt) benutzen.
- Sie können eines oder mehrere **Zeilenkommandos** benutzen (siehe entsprechenden Abschnitt) benutzen.

Beachten Sie bitte folgende Hinweise, wenn Sie mehrere Funktionen ausführen:

- Sie können immer nur ein Einfüge-Zeilenkommando (`.I`) auf einmal ausführen.
- In der Kommandozeile des Editors können Sie mehr als ein Editor-Kommando eingeben, aber nur das letzte in der Editor-Kommandozeile eingegebene Kommando darf ein Systemkommando sein.

  Beispiel:

  ```
  SC 'MOVE',-2,RENUMBER
  ```

  Anmerkung: Natural behandelt das Editor-Kommando `N` wie ein Systemkommando. `N` entspricht dem Systemkommando `RENUMBER`.

- Wenn Sie den Quellcode durch Eingabe einer Änderung oder durch Benutzung eines Editor-Kommandos geändert haben, können Sie erst dann ein Systemkommando eingeben, wenn Sie Enter gedrückt haben.

Dynamische Umsetzung von Kleinbuchstaben in Großbuchstaben

Sie können die dynamische Umsetzung in Großbuchstaben aktivieren oder deaktivieren, indem Sie im **Editor-Profil** die entsprechenden Optionen **Editing in Lower Case** und **Dynamic Conversion of Lower Case** auf `Y` setzen (siehe ***Editor-Profil***). Jeglicher Quellcode, den Sie im Editierbereich eingeben, wird dann in Großbuchstaben umgesetzt.

Ausnahmen:

- Der Inhalt des Natural-Objekttyps Text bleibt so, wie eingegeben.
- Eine Textzeichenkette, die *nicht*
  - eine hexadezimale Konstante oder
  - eine hexadezimale Unicode-Konstante ist

  und die durch Apostrophe und Ausrufungszeichen eingeschlossen ist, bleibt so, wie Sie sie eingeben.
- DBCS-Zeichen (DBCS = Double-Byte Character Set), die durch Shift-out- und Shift-in-Zeichen eingeschlossen sind, bleiben so, wie Sie sie eingeben.
- Ein Kommentar, der durch die Zeichenkette Leerzeichen-Schrägstrich-Stern (/*) angezeigt wird, bleibt so, wie Sie sie eingeben.

Vorsicht: Wenn die Zeichenkette Leerzeichen-Schrägstrich-Stern (/*) einen ausführbaren Teil eines Statements bezeichnet, muss sie *ohne* Leerzeichen (/*) vor der Zeichenkette angegeben werden. Andernfalls wird die Zeichenkette als Kommentar betrachtet.

Untere Informationszeile

Die untere Informationszeile des Editor-Bildschirms enthält ein Lineal und außerdem folgende Informationen:

- **Current Source Size - Aktuelle Quellcode-Größe**

  Die Größe (Anzahl der Zeichen) des aktuellen Quellecodes. Da Quellcode-Zeilen im Quellcode-Arbeitsbereich mit variabler Länge gespeichert werden, werden nachgezogene Leerzeichen innerhalb einer Quellcode-Zeile nicht mit gezählt; führende und eingebettete Leerzeichen werden mit gezählt. Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Size Information** auf `Y` gesetzt ist (siehe ***Editor-Profil***).

- **Char. Free - Zeichen frei**

  Die Anzahl der Zeichen, die im Quellcode-Arbeitsbereich noch verfügbar sind. Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Size Information** auf `Y` gesetzt ist (siehe ***Editor-Profil***).

- **S**

  Die Größe (Anzahl der Zeilen) des Quellcodes, der zurzeit bearbeitet wird.

- **L**

  Die Zeilennummer der Quellcode-Zeile, die zurzeit als oberste Zeile angezeigt wird.

Editor-Kommandos

Editor-Kommandos werden in der Kommandozeile des Programm-Editors eingegeben. Die Kommando-Parameter müssen voneinander abgetrennt werden:

- entweder mit dem Eingabebegrenzungszeichen, das mit dem Natural-Session-Parameter `ID` angegeben ist. Standardmäßig ist das ein Komma (,).
- oder durch Leerzeichen.

Wenn mehrere Kommandos eingegeben werden, müssen diese ebenfalls durch Eingabebegrenzungszeichen oder Leerzeichen voneinander abgetrennt werden.

Die verfügbaren Editor-Kommandos werden in der nachfolgenden Tabelle und im Abschnitt ***Editor-Kommandos zum Positionieren*** beschrieben. Die in den Editor-Kommandos verwendeten Syntax-Symbole werden im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erklärt. Der unterstrichene Teil eines Kommandos kann als gültige Abkürzung verwendet werden.

Editor-Kommando	Funktion
ADD[(*n*)]	Hängt *n* Leerzeilen an. Wenn die **Richtungsanzeige** auf + (Pluszeichen) gesetzt ist, werden die Leerzeilen nach der letzten Zeile des in Bearbeitung befindlichen Objekts hinzugefügt. Wenn die **Richtungsanzeige** auf - (Minuszeichen) gesetzt ist, werden die Leerzeilen vor der ersten Zeile des Objekts hinzugefügt. Der Wert für *n* darf im Bereich von 1 bis 9 liegen. Wenn Sie für *n* keinen oder einen nicht korrekten Wert angeben, werden standardmäßig 9 Zeilen (4 Zeilen im Split-Screen-Modus) hinzugefügt. Wenn Sie das nächste Mal Enter drücken, werden alle Zeilen, die dann noch leer sind, wieder entfernt.
CANCEL oder . (Punkt)	Die Editor-Sitzung wird abgebrochen. Alle seit dem letzten Speichern (SAVE) vorgenommenen Änderungen werden *nicht* gespeichert.
CATALOG [*object-name*]	Führt das Systemkommando CATALOG aus. Der aktuelle Quellcode wird geprüft und katalogisiert. Wenn Sie neu angelegten Quellcode katalogisieren oder wenn Sie den aktuellen Quellcode kopieren wollen, müssen Sie zusammen mit dem Kommando einen Objektnamen angeben. Siehe auch ***Quellcode speichern und katalogisieren***.
CHANGE ['*scan-value*'*replace-value*']	Durchsucht den Quellcode nach der als Suchwert (*scan-value*) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (*replace-value*) eingegebenen Zeichenkette. Jede Zeile, in der eine Zeichenkette ersetzt wurde, wird am linken Zeilenrand mit einem R markiert. Als Eingabebegrenzungszeichen kann jedes Sonderzeichen benutzt werden, dass in einem Natural-Variablennamen nicht gültig ist. Wenn Sie CHANGE ohne Parameter eingeben, wird das **SCAN/REPLACE**-Fenster angezeigt. Zusätzlich zum Suchwert (*scan-value*) und zum Ersetzungswert (*replace-value*) können Sie in diesem Fenster Folgendes angeben:

Editor-Kommando	Funktion
	■ die Nullwert-Option (siehe SET NUL), ■ den absoluten Suchmodus (siehe SET ABS), ■ den Bereichsmodus (siehe SET RANGE).
CHECK	Führt das Systemkommando CHECK aus, das die Syntax des aktuellen Quellcode prüft. Falls ein Fehler gefunden wird, wird die fehlerhafte Zeile mit einem E markiert und eine entsprechende Meldung wird in der Meldungszeile angezeigt. Wenn kein Fehler gefunden wird, wird eine Meldung angezeigt, die die erfolgreiche Beendigung der Prüfung anzeigt.
CLEAR	Führt das Systemkommando CLEAR aus, das den Quellcode-Arbeitsbereich (einschließlich des Objektnamens und der Markierungen X und Y) löscht.
DX oder DY	Löscht die mit X oder Y markierte Zeile. Siehe auch Zeilenkommandos .X und .Y.
DX-Y	Löscht den mit den Markierungen X und Y eingegrenzten Zeilenblock. Siehe auch Zeilenkommandos .X und .Y.
EX oder EY	Löscht alle Zeilen ab dem Anfang des Editierbereichs bis zur mit X markierten Zeile (die jedoch nicht gelöscht wird) oder ab der Zeile, die auf die mit Y markierte Zeile folgt, bis zum Ende des Editierbereichs. Siehe auch Zeilenkommandos .X und .Y.
EX-Y	Löscht alle Zeilen im Editierbereich mit Ausnahme des Blocks, der mit X und Y eingegrenzt ist. Siehe auch Zeilenkommandos .X und .Y.
EXIT	Die Editor-Sitzung wird beendet. Alle am Quellcode vorgenommenen Änderungen werden in Abhängigkeit von der Einstellung im **Editor-Profil** gespeichert. Siehe ***Exit-Funktion***.
LET	Macht alle Änderungen, die Sie im aktuellen Bildschirm vorgenommen haben, nachdem Sie das letzte Mal Enter gedrückt haben, rückgängig. Zusätzlich werden alle Zeilenkommandos, die bereits eingegeben, aber noch nicht ausgeführt wurden, ignoriert.
N [(*nnnn*)]	Dieses Kommando entspricht dem Systemkommando RENUMBER. Es bewirkt, dass die Quellcode-Zeilen des Objekts, das sich gerade im Quellcode-Arbeitsbereich befindet, neu durchnummeriert werden. Wenn Sie nur N eingeben, werden die Zeilen in Zehnerschritten neu nummeriert. Wenn Sie N (*nnnn*) eingeben, werden die Zeilen in Schritten von *nnnn* neu nummeriert.

Editor-Kommando	Funktion	
	Falls der für *n* angegebene Wert zu groß ist, werden die Zeilen in Fünfer-Schritten neu nummeriert. **Anmerkung:** Siehe auch *Quellcode-Zeilennummern umnummerieren* im *Leitfaden zur Programmierung*.	
PROFILE [*name*]	Ruft den **Editor-Profil**-Bildschirm auf. Dort können Sie sich Ihre aktuellen Editor-Profil-Einstellungen ansehen und diese ändern. Weitere Informationen siehe Abschnitt ***Editor-Profil***.	
REN ON|OFF	ON	Jedes Mal, wenn das Kommando CHECK, RUN, SAVE, CATALOG oder STOW am Quellcode ausgeführt wird, erfolgt die Neunummerierung der Quellcode-Zeilen.
	OFF	Zeigt an, dass die automatische Neunummerierung nicht wirksam ist.
	Der Standardwert ist ON. Das Kommando REN entspricht der Editor-Profil-Option **Auto Renumber**. Siehe ***Editor-Profil***. **Anmerkung:** Siehe auch *Quellcode-Zeilennummern umnummerieren* im *Leitfaden zur Programmierung*.	
RESET	Löscht die aktuellen Zeilenmarkierungen X und Y sowie alle zuvor mit dem Zeilenkommando .N gesetzten Markierungen. Siehe auch Zeilenkommandos .X und .Y.	
SAVE [*object-name*]	Führt das Systemkommando SAVE aus, das den aktuellen Quellcode speichert. Wenn Sie neu angelegten Quellcode speichern oder wenn Sie den aktuellen Quellcode kopieren wollen, müssen Sie bei dem Kommando einen Objektnamen angeben. Siehe auch ***Quellcode speichern und katalogisieren***.	
SCAN ['*scan-value*']	Durchsucht den Quellcode nach einer Zeichenkette (*scan-value*). Jede Zeile, in der der Suchwert (*scan-value*) gefunden wird, wird am linken Zeilenrand mit einem S gekennzeichnet. Wird der angegebene Suchwert (*scan-value*) ohne Begrenzungszeichen eingegeben, zum Beispiel: SCAN ABC D, dann wird die gesamte Zeichenkette, die auf das Schlüsselwort SCAN folgt, als Suchwert benutzt. **Anmerkung:** Mit dem Kommando SCAN erfolgt eine exakte Suche nach dem angegebenen Suchwert (*scan-value*). Dies sollte bei der Suche nach DBCS-Zeichen (DBCS = Double-Byte Character Set) berücksichtigt werden.	

Editor-Kommando	Funktion	
	Wenn die **Richtungsanzeige** auf + (Pluszeichen) gesetzt ist, erfolgt die Suche ab der ersten, auf dem Bildschirm angezeigten Zeile bis zur letzten Zeile des Quellcode-Arbeitsbereiches. Wenn die Richtungsanzeige auf - (Minuszeichen) gesetzt ist, erfolgt die Suche ab der letzten, auf dem Bildschirm angezeigten Zeile bis zur ersten Zeile des Quellcode-Arbeitsbereiches. Wenn Sie das SCAN-Kommando ohne Parameter eingeben, wird das **SCAN/REPLACE**-Fenster angezeigt. Sie können dann zusätzlich zum Suchwert (*scan-value*) Folgendes angeben: ■ einen Ersetzungswert (*replace-value*, siehe auch CHANGE), ■ die Nullwert-Option (siehe SET NUL), ■ den absoluten Suchmodus (siehe SET ABS), ■ den Bereichsmodus (siehe SET RANGE). ■ den Modus zum Ignorieren der Groß-/Kleinschreibung (siehe SET IGNORECASE). SCAN ist ein cursorsensitives Kommando, das zusätzliche Optionen bietet, siehe Abschnitt ***Cursorsensitive Kommandos***.	
SCAN=[+|-]	Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts (*scan-value*). Die Suchrichtung für den gegebenen Suchvorgang können Sie explizit angeben, indem Sie SCAN=+ oder SCAN=- eingeben. Die Einstellung in der **Richtungsanzeige** wird dann ignoriert. **Anmerkung:** Das beim SCAN-Kommando verwendete Gleichheitszeichen (=) ist das Standard-Zuweisungszeichen für Eingabe-Parameter. Falls ein anderes Zeichen als Zuweisungszeichen für Eingabe-Parameter angegeben worden ist (siehe Session-Parameter IA in der *Parameter-Referenz-Dokumentation*), muss dieses Zeichen stattdessen benutzt werden.	
SET ABS [ON|OFF]	ON	Die Kommandos SCAN und CHANGE arbeiten im absoluten Modus. Das bedeutet, dass es nicht nötig ist, den Suchwert (*scan-value*) und den Ersetzungswert (*replace-value*) durch Leerzeichen oder Sonderzeichen abzugrenzen.
	OFF	Die Kommandos SCAN und CHANGE arbeiten nicht im absoluten Modus. Das bedeutet, dass Sie den Suchwert (*scan-value*) und den Ersetzungswert (*replace-value*) durch Leerzeichen

Editor-Kommando	Funktion	
		oder Sonderzeichen abgrenzen müssen.
	Der Standardwert ist OFF. Das Kommando SET ABS entspricht der Editor-Profil-Option **Absolute Mode for SCAN/CHANGE**. Siehe Abschnitt ***Editor-Profil***.	
SET IGNORECASE[ON|OFF]	ON	Die Kommandos SCAN und CHANGE arbeiten im Ignore Case-Modus. Das bedeutet, die Suche findet jede beliebige Kombination aus Groß- und Kleinbuchstaben.
	OFF	Die Kommandos SCAN und CHANGE arbeiten nicht im Ignore Case-Modus. Das bedeutet, es wird nach dem exakten Wert gesucht.
	Der Standardwert ist OFF. Das Kommando SET IGNORECASE entspricht der Editor-Profil-Option **Ignore Case Mode for SCAN/CHANGE**. Siehe Abschnitt ***Editor-Profil***.	
SET CAPS [ON|OFF]	Dieses Kommando betrifft die Einstellungen der Editor-Profil-Optionen **Editing in Lower Case** und **Dynamic Conversion of Lower Case**. Siehe Abschnitt ***Editor-Profil***.	
	ON	Konvertiert alle in Kleinschreibung auf dem aktuellen Bildschirm angezeigten Zeichen in Großschreibung. Ausnahme: Zeichen, die sich innerhalb eines Kommentarbereichs befinden oder mit Apostrophen (') oder Anführungszeichen eingeschlossen sind. Setzt die Editor-Profil-Optionen **Editing in Lower Case** auf Y und **Dynamic Conversion of Lower Case** auf Y.
	OFF	Es erfolgt keine Zeichenkonvertierung. Die Zeichen bleiben so wie eingeben. Setzt die Editor-Profil-Optionen **Editing in Lower Case** auf Y und **Dynamic Conversion of Lower Case** auf N.

Editor-Kommando	Funktion	
	ALL	Konvertiert alle Zeichen in Großschreibung. Setzt die Editor-Profil-Optionen **Editing in Lower Case** auf N und **Dynamic Conversion of Lower Case** auf N.
	Der Standardwert ist ON.	
SET ESCAPE *character*	Das Fluchtzeichen, das vor jedem Zeilenkommando vorangestellt werden muss. Das **Standard-Fluchtzeichen** ist ein Punkt (.).	
SET NUL [ON|OFF]	ON	Alle Ausprägungen eines mit dem Kommando SCAN gesuchten Wertes werden gelöscht. Nach dem Löschen des mit SCAN gesuchten Wertes wird das Kommando SET NUL automatisch auf OFF gesetzt.
	Der Standardwert ist OFF.	
SET RANGE [ON|OFF]	ON	Die Kommandos SCAN und CHANGE arbeiten im Bereichsmodus. Das bedeutet, dass der zu suchende/zu ersetzende Wert sich innerhalb des Zeilenbereichs befinden muss, der mit den Zeilenmarkierungen X und Y markiert ist.
	OFF	Die Kommandos SCAN und CHANGE arbeiten nicht im Bereichsmodus. Das bedeutet, dass keine Bereichsgrenzen wirksam sein sollen.
	Der Standardwert ist OFF. Das Kommando SET RANGE entspricht der Editor-Profil-Option **Range Mode for SCAN/CHANGE**. Siehe ***Editor-Profil***.	
SET RELINE [ON|OFF]	ON	Es erfolgt eine automatische Entfernung aller Leerzeilen aus dem Quellcode eines Objekts (Ausnahme: Textobjekte), wenn Sie Enter drücken.
	OFF	Die Leerzeilen bleiben erhalten.
	Die Standard-Einstellung ist ON. Dieses Kommando entspricht der Editor-Profil-Option **Empty Line Suppression**. Siehe Abschnitt ***Editor-Profil***.	
SET SEQ [ON|OFF]	OFF	Bei Eingabe eines numerischen Wertes werden die ersten vier Stellen im Editierbereich als die Zeilennummern betrachtet und werden in die

Editor-Kommando	Funktion	
		Zeilennummer-Position verschoben, sobald Sie Enter drücken. Diese Funktion ist nützlich, wenn zum Beispiel eine Statement-Zeile von einer Quellcode-Zeilennummer in einer anderen Statement-Zeile referenziert werden soll. Wenn Sie den Quellcode neu nummerieren, wird die referenzierende Zeilennummer ebenfalls neu nummeriert.
	ON	Die ersten vier Stellen einer numerischen Werteingabe bleiben so, wie sie eingegeben wurden.
	Außer beim Objekttyp Text ist der Standardwert OFF. Das Kommando SET SEQ entspricht der Editor-Profil-Option **Numeric Input as Line Number**. Siehe ***Editor-Profil***.	
SET SIZE [ON|OFF]	ON	Die Größe des Quellcodes wird in der unteren Informationszeile des Editor-Bildschirms und der Programmiermodus wird in der Lineal-Zeile angezeigt.
	OFF	Diese Informationen werden nicht angezeigt.
	Der Standardwert ist OFF. Das Kommando SET SIZE entspricht der Editor-Profil-Option **Source Size Information**. Siehe ***Editor-Profil***.	
SET STAY [ON|OFF]	ON	Wenn Enter gedrückt wird, wird der aktuelle Bildschirm weiterhin angezeigt. Eine Vorwärts-/Rückwärtspositionierung kann nur mit den Positionierungskommandos bewirkt werden.
	OFF	Wenn Enter gedrückt wird, wird der nächste Bildschirm angezeigt.
	Der Standardwert ist OFF. Das Kommando SET STAY entspricht der Editor-Profil-Option **Stay on Current Screen**. Siehe ***Editor-Profil***.	

Editor-Kommando	Funktion	
SET TYPE *object-type*	Mit diesem Kommando können Sie den (in der Editor-Kommandozeile angezeigten) Objekttyp für die zurzeit im Quellcode-Arbeitsbereich befindliche Source ändern. *object-type* steht für einen Objekttyp, der mit dem Kommando angegeben werden muss. Folgende Werte sind möglich:	
	P	oder PROGRAM
	S	oder SUBROUTINE
	N	oder SUBPROGRAM
	H	oder HELPROUTINE
	T	oder TEXT
	C	oder COPYCODE
	4	oder CL oder CLASS
	7	oder FUNCTION
SHIFT [-*nn*|+*nn*]	Dieses Kommando verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, nach links oder nach rechts. Der Parameter *nn* gibt die Anzahl der Zeichen an, um die die Zeile verschoben werden soll. Kommentarzeilen werden nicht verschoben.	
SHIFT --	Dieses Kommando verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, ganz nach links. Kommentarzeilen werden nicht verschoben.	
SHIFT ++	Dieses Kommando verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, ganz nach rechts (maximal um 99 Stellen). Kommentarzeilen werden nicht verschoben.	
SPLIT [*parameter*]	Dieses Kommando bewirkt, dass der Editor-Bildschirm zwecks gleichzeitiger Anzeige von zwei verschiedenen Objekten in zwei Hälften unterteilt wird. Siehe ***Split-Screen-Modus***. SPLIT ist ein cursorsensitives Kommando, bei den zusätzliche Optionen zur Verfügung stehen. Siehe ***Cursorsensitive Kommandos***. Wenn SPLIT ohne Parameter eingeben wird, wird der Quellcode des zuvor im Split-Screen-Bereich angezeigten Objekts erneut angezeigt. Dies gilt für die Dauer der aktuellen Editor-Sitzung.	
STOW [*object-name*]	Führt das Systemkommando STOW aus, mit dem der aktuelle Quellcode gespeichert und katalogisiert wird. Wenn Sie neuen Quellcode speichern und katalogisieren wollen oder wenn Sie den aktuellen Quellcode kopieren wollen, müssen Sie einen Objektnamen angeben. Andernfalls erscheint eine entsprechende Meldung. Siehe auch ***Quellcode speichern und katalogisieren***.	

Editor-Kommando	Funktion
STRUCT [DISPLAY]	Dieses Kommando dient dazu, die Quellcode-Zeilen eines Programms entsprechend der Programmstruktur einzurücken. Standardmäßig wird schrittweise um zwei Stellen eingerückt. Wird DISPLAY angegeben, wird der Quellcode in komprimierter Form angezeigt. Siehe Systemkommando STRUCT in der *Systemkommandos*-Dokumentation.
*	Dieses Kommando zeigt das zuletzt eingegebene Editor-Kommando an.
*=	Dieses Kommando führt das zuletzt in der Kommandozeile eingegebene Kommando erneut aus.

Editor-Kommandos zum Positionieren

Editor-Kommandos zum Positionieren werden in der Kommandozeile des Programm-Editors eingegeben. Folgende Kommandos stehen zum Positionieren zur Verfügung:

Kommando	Funktion
Enter oder +P oder +	Eine Seite vorwärts positionieren.
-P oder -	Eine Seite rückwärts positionieren.
+H	Eine halbe Seite vorwärts positionieren.
-H	Eine halbe Seite rückwärts positionieren.
T oder -	An den Anfang des Quellcodes positionieren.
B oder ++	An das Ende des Quellcodes positionieren.

Kommando	Funktion
+*nnnn*	Um *nnnn* (maximal 4 Ziffern) Zeilen vorwärts positionieren.
-*nnnn*	Um *nnnn* (maximal 4 Ziffern) Zeilen rückwärts positionieren.
nnnn	Auf Zeile Nummer *nnnn* positionieren.
X	Auf die mit X markierte Zeile positionieren.
Y	Auf die mit Y markierte Zeile positionieren.
POINT	Auf die Zeile positionieren, in der das Zeilenkommando .N eingegeben wurde. Siehe auch Zeilenkommando .P.

Zeilenkommandos

Zeilenkommandos werden an der ersten Stelle einer Quellcode-Zeile eingegeben. Die beim Programm-Editor zur Verfügung stehenden Zeilenkommandos sind nachfolgend aufgeführt. Die Notation (*nn*) oder (*nnnn*) zeigt einen Wiederholungsfaktor an. Der Standard-Wiederholungswert ist 1 (mit Ausnahme beim Kommando .I, siehe unten). Eine Erklärung der verwendeten Syntaxsymbole finden Sie im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation.

Anmerkung: Es wird empfohlen, am Ende eines jeden Zeilenkommandos ein Leerzeichen einzugeben. Dadurch wird verhindert, dass der Editor versucht, die anschließenden Informationen in der Zeile als Bestandteil des Zeilenkommandos zu interpretieren.

Zeilenkommandos	Funktion
.C[(*nnnn*)]	Kopiert die Zeile, in der das Kommando eingegeben wurde.
.CX[(*nnnn*)] oder .CY[(*nnnn*)]	Kopiert die Zeile, die mit X oder mit Y markiert ist. Siehe auch Zeilenkommandos .X und .Y sowie ***Anmerkungen zu den Zeilenkommandos***.
.CX-Y[(*nnnn*)]	Kopiert den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist. (Siehe auch ***Anmerkungen zu den Zeilenkommandos***.)
.D[(*nnnn*)]	Löscht eine oder mehrere Zeilen ab der Zeile, in der Sie das Kommando eingeben, bis zum Ende des Quellcodes (unabhängig von der Einstellung in der **Richtungsanzeige**). Der Standardwert ist 1 Zeile.
.I[(*nn*)]	Fügt *nn* Leerzeilen ein. Dabei kann *nn* im Bereich von 1 bis zur Gesamtzahl minus zwei der im Editierbereich angezeigten Zeilen liegen. Wenn zum Beispiel insgesamt 28 Zeilen im Editierbereich angezeigt werden, können Sie maximal 26 Zeilen einfügen. Wird *nn* nicht (oder nicht korrekt) angegeben, werden standardmäßig 9 Zeilen (4 Zeilen im Split-Screen-Modus) eingefügt. Zeilen, die leer gelassen werden, werden anschließend aus dem Quellcode entfernt. Dies ist abhängig von der Einstellung

Zeilenkommandos	Funktion
	der Editor-Profil-Optionen **Empty Line Suppression** und **Empty Line Suppression for Text**. Siehe ***Editor-Profil***. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
`.I(obj,ssss,nnnn)`	Fügt in den Quellcode ein Objekt ein, das in der aktuellen Library oder in der Steplib (die Standard-Steplib ist SYSTEM) enthalten ist. Je nach Einstellung in der **Richtungsanzeige** wird das Objekt vor oder nach der Zeile eingefügt, in der Sie das Kommando eingeben. Wenn Sie nur einen Teil des Objekts einfügen möchten, müssen Sie für `ssss` die erste einzufügende Zeile angeben (zum Beispiel bedeutet `20`, dass die Einfügung mit der 20. Zeile beginnen wird), und für `nnnn` müssen Sie die Anzahl der einzufügenden Zeilen angeben. Wenn Sie mehrere Kommandos eingeben, wird dieses Kommando stets erst dann ausgeführt, nachdem alle anderen Zeilen- und/oder Editor-Kommandos ausgeführt worden sind. Handelt es sich bei dem Objekt um eine Map (Maske), wird in die aktuelle Zeile automatisch ein `INPUT USING MAP`-Statement (siehe *INPUT-Syntax 2 — Verwendung einer vordefinierten Eingabemaske* in der *Statements*-Dokumentation) mit allen definierten Variablen eingefügt. Handelt es sich bei dem Objekt um eine Data Area, wird die gesamte Data Area mit Ausnahme der Kommentarzeilen eingefügt. Es können jedoch nur Local und Parameter Data Areas, die mit dem `STOW`-Kommando gespeichert und katalogisiert wurden, in den Quellcode-Arbeitsbereich eingefügt werden. Global Data Areas können nicht eingefügt werden. Handelt es sich bei dem Objekt um einen Adapter, wird in die aktuelle Zeile automatisch ein `PROCESS PAGE USING`-Statement (siehe *Syntax 2 - PROCESS PAGE USING* in der *Statements*-Dokumentation) mit allen definierten Variablen eingefügt.
`.J`	Führt die aktuelle Zeile mit der nächsten Zeile zusammen. Ist die resultierende Zeile länger als die Editor-Bildschirmzeile, wird die Zeile mit einem `L` markiert und muss mit dem Zeilenkommando `.S` in zwei Zeilen geteilt werden (siehe unten), bevor sie geändert werden kann.
`.L`	Macht alle seit dem letzten Drücken der Enter-Taste an der Zeile vorgenommenen Änderungen rückgängig.

`.MX`
 	
oder
 	
`.MY`
	Verschiebt die mit `X` oder mit `Y` markierte Zeile. Siehe auch Zeilenkommandos `.X` und `.Y` sowie ***Anmerkungen zu den Zeilenkommandos***.

`.MX-Y`	Verschiebt den Zeilenblock, der mit den Markierungen `X` und `Y` eingegrenzt ist (siehe auch *Anmerkungen zu den Zeilenkommandos*).

Zeilenkommandos	Funktion
.N	Markiert (unsichtbar) eine Zeile, die mit den Editor-Kommando POINT an den Anfang des Quellcode-Arbeitsbereichs positioniert werden soll. Die Markierung wird automatisch gelöscht, wenn ein Fehler bei einem Zeilen- oder Editor-Kommando auftritt oder wenn das RESET-Kommando ausgeführt wird.
.P	Positioniert die mit diesem Kommando markierte Zeile an den Anfang des Bildschirms.
.S	Spaltet die Zeile an der mit dem Cursor markierten Position.
.X	Markiert eine Zeile mit einem X (siehe auch *Anmerkungen zu den Zeilenkommandos*).
.Y	Markiert eine Zeile mit einem Y (siehe auch *Anmerkungen zu den Zeilenkommandos*).

Anmerkungen zu den Zeilenkommandos

- Wenn sowohl das Kommando .X als auch das Kommando .Y bei einer Zeile angewendet werden, wird sie behandelt, als ob sie mit einem X und mit einem Y markiert ist; die tatsächlich angezeigte Zeilenmarkierung, die diesen Status wiedergeben soll, ist ein Z.
- Ist die **Richtungsanzeige** auf + (Pluszeichen) gesetzt, werden die kopierten, eingefügten oder verschobenen Zeilen *nach* der Zeile platziert, in der das entsprechende Kommando eingegeben wurde.

  Ist die **Richtungsanzeige** auf - (Minuszeichen) gesetzt, werden die kopierten, eingefügten oder verschobenen Zeilen *vor* der Zeile platziert, in der das entsprechende Kommando eingegeben wurde.

Spezielle PF-Tastenfunktionen

Die folgenden speziellen Funktionen können auch mittels PF-Tasten bedient werden:

Funktion	Erläuterung
*CURSOR	Eine Funktion zum Aufspalten einer Zeile kann auch mit dem Kommando .I, .CX, .CX-Y, .MX oder .MX-Y kombiniert werden. Dazu muss der Wert *CURSOR einer PF-Taste zugewiesen werden (siehe **PF and PA Keys** im *Editor-Profil*). Wird dann nach Eingabe eines Zeilenkommandos diese PF-Taste anstelle der Enter-Taste gedrückt, wird die Zeile, in der das Kommando eingegeben wurde, zunächst an der Cursorposition aufgespalten und danach wird das Zeilenkommando ausgeführt. Siehe auch *Beispiel für *CURSOR auf PF-Taste*.
*X oder *Y	Wird einer PF-Taste der Wert *X oder *Y im Editor-Profil zugewiesen (siehe **PF- und PA-Tasten** in *Editor-Profil*), wird, jedes Mal wenn diese PF-Taste benutzt wird, die Cursorposition mit X bzw. Y markiert. Diese Positionsmarkierungen werden dann verwendet um zu bestimmen, welcher Teil einer Zeile bei der Kommandoausführung eingefügt werden soll. Siehe auch *Beispiel für *X und *Y auf PF-Tasten*.

Beispiel für *CURSOR auf PF-Taste

Der folgende Abschnitt enthält eine Anleitung, wie Sie eine PF-Taste benutzen können, der der Wert `*CURSOR` zugewiesen ist.

1. Geben Sie im Abschnitt **PF and PA Keys** des **Editor-Profil**-Bildschirms (siehe *Editor-Profil*), neben dem Eintrag PF6 den Wert `*CURSOR` ein.
2. Öffnen Sie den Programm-Editor, geben Sie den folgenden Text ein, und drücken Sie Enter:

```
>                                      > +  Program              Lib SAGTEST
All    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
  0010 MOVE A TO B
  0020 WRITE A B C
  0030 MOVE C TO B MOVE A TO C
  0040
  0050
  0060
  0070
  0080
  0090
  0100
  ....
  0280
       ....+....1....+....2....+....3....+....4....+....5....+... S 3    L 1
```

3. Geben Sie in Zeile `0020` das Zeilenkommando `.X` ein. Drücken Sie Enter.

   Die Zeile wird markiert. Dies wird durch das vorangestellte `X` angezeigt.

4. Geben Sie in Zeile `0030` das Zeilenkommando `.CX` ein. Setzen Sie den Cursor auf das `M` des zweiten `MOVE`-Statements. Drücken Sie PF6.

   Der Bildschirm sieht nun ähnlich wie in der folgenden Abbildung aus:

```
>                                      > +  Program              Lib SAGTEST
All    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
  0010 MOVE A TO B
X 0020 WRITE A B C
  0030 MOVE C TO B
  0020 WRITE A B C
  0030 MOVE A TO C
  0040
  0050
  0060
  0070
  0080
  0090
  0100
  ....
  0260
       ....+....1....+....2....+....3....+....4....+....5....+... S 5    L 1
```

Die Zeile `0030` wird vor der Cursorposition aufgespalten, Zeile `0020` wird in die Zeile kopiert, in der Sie das Kommando `.CX` eingegeben haben, und der zweite Teil der aufgespaltenen Zeile wird in die letzte Zeile verschoben.

Beispiel für *X und *Y auf PF-Tasten

Der folgende Abschnitt enthält eine Anleitung, wie Sie PF-Tasten benutzen können, denen die Werte `*X` und `*Y` zugewiesen sind.

1. Geben Sie im Abschnitt **PF and PA Keys** des **Editor-Profil**-Bildschirms (siehe *Editor-Profil*), neben dem Eintrag PF4 den Wert `*X` und neben PF5 den Wert `*Y` ein.
2. Öffnen Sie den Programm-Editor, geben Sie den folgenden Text ein, und drücken Sie Enter:

```
>                                      > +  Text                 Lib SAGTEST
All    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
  0010 THIS PORTION
  0020 OF TEXT IS
  0030 FOR DEMONSTRATION OF
  0040 PF-KEY ASSIGNMENTS.
  0050
  0060
  0070
  0080
  0090
  0100
  ....
  0280
       ....+....1....+....2....+....3....+....4....+....5....+... S 0    L 1
```

3. Setzen Sie in Zeile `0010` den Cursor auf den Buchstaben `P`, und drücken Sie PF4.

Die Position des `P` wird so, wie im folgenden Beispiel-Bildschirm dargestellt, markiert.

4. Setzen Sie in Zeile `0030` den Cursor auf das Leerzeichen hinter `DEMONSTRATION`, und drücken Sie PF5.

   Der Bildschirm sieht nun ähnlich wie in der folgenden Abbildung aus:

```
>                                      > +  Text                Lib SAGTEST
All    ....+X...1....+..Y.2....+....3....+....4....+....5....+....6....+....7..
X 0010 THIS PORTION
  0020 OF TEXT IS
Y 0030 FOR DEMONSTRATION OF
  0040 PF-KEY ASSIGNMENTS.
  0050
  0060
  0070
  0080
  0090
  0100
  ....
  0280
       ....+X...1....+..Y.2....+....3....+....4....+....5....+... S 4    L 1
```

   Die Positionen der Zeichen (`P` und Leerzeichen) werden, wie durch das `X` bzw. das `Y` angezeigt, markiert. Dabei erscheinen das `X` und das `Y` in der oberen und der unteren Informationszeile und links neben den Quellcode-Zeilen, die die markierten Zeichen enthalten.

5. Geben Sie in Zeile `0040` das Zeilenkommando `.MX-Y` ein, und drücken Sie Enter.

   Der Bildschirm sieht nun ähnlich wie in der folgenden Abbildung aus:

```
>                                      > +  Text                Lib SAGTEST
All    X...+....1....+.Y..2....+....3....+....4....+....5....+....6....+....7..
  0010 THIS
  0030 OF
  0040 PF-KEY ASSIGNMENTS.
X 0010 PORTION
  0020 OF TEXT IS
Y 0030 FOR DEMONSTRATION
  0040
  0050
  0060
  0070
  0080
  0090
  0100
  ....
  0250
       X...+....1....+.Y..2....+....3....+....4....+....5....+... S 6    L 1
```

Der Textblock, der in Zeile `0010` mit `P` markiert ist und in Zeile `0030` bei `N` endet, wird in die darunter befindliche Zeile verschoben, in der Sie das Kommando eingegeben haben. Der verschobene Block und der in Zeile `0010` und `0030` verbliebene Text werden linksbündig ausgerichtet.

Cursorsensitive Kommandos

Cursorsensitive Kommandos werden ohne Parameter eingegeben. Stattdessen wird der Cursor an beliebiger Stelle (Ausnahme: Kommandozeile) auf dem Bildschirm platziert.

Um ein cursorsensitives Kommando zu benutzen:

1 Geben Sie ein cursorsensitives Kommando in der Editor-Kommandozeile ein.

2 Positionieren Sie den Cursor auf einer beliebigen, im Editor-Bildschirm (Ausnahme: Kommandozeile) enthaltenen Zeichenkette.

3 Drücken Sie Enter.

Alternativ können Sie eine PF-Taste benutzen:

1. Positionieren Sie den Cursor auf einer beliebigen, im Editor-Bildschirm (Ausnahme: Kommandozeile) enthaltenen Zeichenkette.
2. Drücken Sie die PF-Taste, die mit dem benötigten cursorsensitiven Kommando belegt ist (siehe **PF and PA Keys** im ***Editor-Profil***).

Nachfolgend werden folgende Themen behandelt:

- SCAN-Kommando
- SPLIT-Kommando
- EDIT- und LIST-Systemkommandos

SCAN-Kommando

Das cursor-sensitive `SCAN`-Kommando durchsucht den aktuellen Quellcode nach der Zeichenkette, auf der der Cursor positioniert ist. (Falls der Cursor jedoch auf einem Leerzeichen positioniert ist, wird das **SCAN/REPLACE**-Fenster aufgerufen.)

Im Split-Screen-Modus kann der Cursor auch auf einer Zeichenkette im Split-Screen-Bereich positioniert werden.

Für die Benutzung des `SPLIT SCAN`-Kommandos gilt das gleiche wie beim `SCAN`-Kommando, jedoch wird die Suchoperation nur im Split-Screen-Bereich durchgeführt (siehe auch Abschnitt ***Split-Screen Kommandos***).

Weitere Informationen zum SCAN-Kommando siehe entsprechende Beschreibung unter ***Editor-Kommandos***.

SPLIT-Kommando

Anstatt objektspezifische SPLIT-Kommandos zu benutzen, um den Quellcode eines Objekts anzuzeigen (siehe Abschnitt ***Split-Screen Kommandos***), können Sie das cursor-sensitive SPLIT-Kommando benutzen.

Wenn Sie das cursor-sensitive SPLIT-Kommando benutzen und den Cursor auf eine Zeichenkette positionieren, die einem Objektnamen oder einem erweiterten Namen (Langnamen) entspricht, wird der Quellcode des entsprechenden Objekts im Split-Screen-Bereich des Editors angezeigt. Das benötigte Objekt muss in der aktuellen Bibliothek enthalten sein.

Diese Funktion ist besonders nützlich, wenn Sie zum Beispiel nicht den Objektnamen kennen, der dem erweiterten Namen (Langnamen) eines Objekts (zum Beispiel eine Subroutine) entspricht, das im aktuellen Quellcode referenziert wird.

EDIT- und LIST-Systemkommandos

Wenn Sie die cursor-sensitiven Kommandos EDIT oder LIST benutzen und den Cursor auf eine Zeichenkette positionieren, die einem Objektnamen oder einem erweiterten Namen (Langnamen) entspricht, wird der Quellcode des entsprechenden Objekts angezeigt:

- Beim EDIT-Kommando wird der Quellcode in den Editierbereich geladen. Falls erforderlich, wird sogar ein anderer Editor aufgerufen.
- Beim LIST-Kommando wird der Quellcode aufgelistet, und zwar auch dann wenn eine View (DDM) referenziert worden ist.

Die zu bearbeitenden bzw. aufzulistenden Objekte müssen in der aktuellen Library enthalten sein.

Weitere Informationen zu den Kommandos EDIT und LIST siehe *Systemkommandos*-Dokumentation.

Quellcode speichern und katalogisieren

Den zurzeit im Quellcode-Arbeitsbereich befindlichen Quellcode können Sie als Source-Objekt und außerdem als katalogisiertes Objekt speichern. Beide Objekte werden in einer Natural-Library in einer Natural-Systemdatei gespeichert.

Um den aktuellen Quellcode zu speichern und/oder zu katalogisieren:

- Benutzen Sie das Systemkommando SAVE, CATALOG oder STOW. Weitere Informationen siehe *Objekte speichern und katalogisieren* in der Dokumentation *Natural benutzen*.

Anmerkung: Wenn Sie den Programm-Editor mit dem Editor-Kommando `EXIT` verlassen, wird der aktuelle Quellcode automatisch gespeichert, wenn die entsprechende **Editor-Profil**-Option demgemäß gesetzt ist. Weitere Informationen siehe *Exit-Funktion*.

Um eine Kopie des aktuelle Quellcodes zu behalten:

- Benutzen Sie die Editor-Optionen **Source Save into** und **Auto Save Numbers** im *Editor-Profil*.

  Eine Kopie des zuletzt mit einem der Natural-Editoren bearbeiteten Quellcodes wird dann automatisch als Source-Objekt in der aktuellen Natural-Umgebung gespeichert.

Exit-Funktion

Mit der Exit-Funktion können Sie die aktuelle Editor-Sitzung beenden, indem Sie das Kommando `EXIT` absetzen.

Die Wirkung des `EXIT`-Kommandos ist abhängig von der Einstellung der **Editor-Profil**-Option **Prompt Window for Exit Funktion:**.

- Ist die Option auf `Y` gesetzt (Standardeinstellung), dann ruft das `EXIT`-Kommando das **EXIT Function**-Fenster immer dann auf, wenn Sie das Kommando bei einer Source eingeben, die nicht gespeicherte Änderungen enthält (siehe auch **Änderungsanzeige**). Wenn an dem Quellcode keine Änderungen vorgenommen wurden, erscheint das Fenster *nicht* und die Editor-Sitzung wird ohne Speicherung des Quellcodes beendet.

  Das **EXIT Function**-Fenster enthält folgende Optionen:

Option	Erläuterung
Save and Exit	Speichert alle am aktuellen Quellcode vorgenommenen Änderungen. Die Editor-Sitzung wird beendet. Siehe auch *SAVE Object-Fenster*.
Exit without Saving	Die Editor-Sitzung wird beendet. Die seit dem letzten Speichern des Quellcode vorgenommenen Änderungen werden *nicht* gespeichert.
Resume Funktion	Die Editor-Sitzung wird *nicht* beendet. Es werden keine Änderungen gespeichert. Das Eingabeaufforderungsfenster wird geschlossen und die aktuelle Funktion wird fortgesetzt.

- Ist die Option auf `N` gesetzt, führt das `EXIT`-Kommando die **Save and Exit**-Funktion aus: Sie speichert alle an der aktuellen Source vorgenommenen Änderungen. Die Editor-Sitzung wird beendet. Siehe auch *SAVE Object-Fenster*.

SAVE Object-Fenster

Die **Save and Exit**-Funktion ruft das **SAVE Object**-Fenster auf, wenn für den aktuellen Quellcode noch kein Objektname definiert wurde (wie in der Editor-Kommandozeile angezeigt).

Sie werden aufgefordert, einen Objektnamen und einen Objekttyp anzugeben. Wenn Sie Ihre Eingaben mit Enter bestätigen, wird der Quellcode unter dem angegebenen Namen als neues Objekt in der aktuellen Library gespeichert und die Sitzung wird beendet. Wenn Sie PF3 drücken, wird das Fenster ohne Maßname geschlossen und die Editor-Sitzung kann fortgesetzt werden.

V Datenbereich-Editor (Data Area Editor)

11 Datenbereich-Editor (Data Area Editor)

- Data Area Editor aufrufen 162
- Obere Informationszeile 164
- Editor-Kommandozeile 165
- Untere Informationszeile 166
- Editierbereich benutzen 166
- Spalten im Editierbereich 167
- Erweiterte Felddefinition bearbeiten 172
- Zeilenkommandos 177
- Editor-Kommandos 183
- Editor-Kommandos zum Positionieren 188
- Data Area speichern und katalogisieren 189
- User Exit für den Data Area Editor 190
- Exit-Funktion 191

Der Datenbereich-Editor *(Data Area Editor)* wird verwendet, um einen Datenbereich (*Data Area*) anzulegen oder zu ändern. Es handelt sich dabei um ein Natural-Objekt des Typs Global Data Area (GDA), Local Data Area (LDA) oder Parameter Data Area (PDA). Informationen zur Verwendung eines Datenbereichs siehe *Datenbereiche (Data Areas)* im *Leitfaden zur Programmierung*.

Ein Datenbereich enthält Datenelement-Definitionen wie zum Beispiel Benutzervariablen, Konstanten und Datenbankfelder, die in einer Datensicht (*Data View*) in einem Datendefinitionsmodul (*Data Definition Module/DDM*) referenziert werden und die von einem oder mehreren Natural-Objekten benutzt werden können. Außerdem können Sie Copycode aus einem Datenbereich erstellen. Beachten Sie, dass es in einer PDA nicht möglich ist, eine Datensicht zu definieren. Sie können jedoch eine Datensicht in einer GDA oder einer LDA in eine Gruppenstruktur umwandeln und dann diese GDA oder LDA als PDA speichern.

Anmerkung: Der Datenbereich-Editor ist standardmäßig in Ihrer Umgebung abgeschaltet (deaktiviert). Weitere Information siehe ***NaturalONE als Standard-Entwicklungsumgebung***.

Verwandtes Thema:

Informationen zur Unterstützung von Unicode und Codepages bei den Natural-Editoren siehe *Editors in the SPoD Environment* in der *Unicode and Code Page Support*-Dokumentation.

Data Area Editor aufrufen

Den Data Area Editor können Sie mit dem Systemkommando `EDIT` aufrufen. Beschreibung siehe *Systemkommandos*-Dokumentation.

Um den Data Area Editor zum Anlegen einer neuen Data Area aufzurufen:

- Setzen Sie das `EDIT`-Kommando ab, indem Sie den Typ der Data Area (`GLOBAL`, `LOCAL` oder `PARAMETER`) angeben, den Sie anlegen wollen.

  Zum Beispiel für eine Local Data Area:

  ```
  EDIT LOCAL
  ```

  Es erscheint ein Editor-Bildschirm mit einem leeren Editierbereich. In der linken oberen Ecke des Bildschirms wird der Typ **Local** angezeigt.

```
Local               Library SAGTEST                       DBID 11177 FNR  1008
Command                                                                 > +
I T L  Name                             F Length     Miscellaneous
All -- -------------------------------- - ---------- ------------------------->
                                                                        S 0    L 1  ↩
  ↩
```

Um den Data Area Editor für eine schon vorhandene Data Area aufzurufen:

- Benutzen Sie das Kommando `EDIT` mit dem Namen einer Data Area, die als Source-Objekt in Ihrer aktuellen Natural-Umgebung gespeichert worden ist.

  Zum Beispiel:

  ```
  EDIT LDA1
  ```

  Es erscheint der Editor-Bildschirm mit dem Quellcode der Local Data Area `LDA1`:

```
Local     LDA1     Library SAGTEST                              DBID  10 FNR  32
  Command                                                                 > +
  I T L  Name                             F Length     Miscellaneous
  All -- -------------------------------- - ---------- ------------------------>
    *    LDA for new application
       1 INCOME                           A         20 (1:3,1:5) INIT ALL<'0'>
       1 PERSON
       2 SEX                              A          6
       2 AGE                              N          3
       1 NAME                             A         24
    R  1 NAME                                        /* REDEF. BEGIN : NAME
       2 FIRST-NAME                       A         10
       2 MIDDLE-INIT                      A          2
       2 LAST-NAME                        A         10
    C  1 DOLLAR                           A          5 CONST<'$US'>
    V  1 FINANCE-VIEW                                FINANCE
       2 PERSONNEL-NUMBER                 N        8.0
    P  2 MAJOR-CREDIT                                (1:1) /* PERIODIC GROUP
       3 CREDIT-CARD                      A         18 (EM=XXX.XXX.XXX.XXX.XXX.XXX)
       3 CREDIT-LIMIT                     N        4.0
       3 CURRENT-BALANCE                  N        4.0
  -- Current Source Size: 1969  Free: 78200 ----------------------- S 12   L 1
```

Der Editor-Bildschirm enthält (von oben nach unten) folgende Bestandteile: Die **obere Informationszeile**, die **Editor-Kommandozeile** , den **Editierbereich** und die **untere Informationszeile**. Diese Bestandteile werden in den folgenden Abschnitten beschrieben.

Obere Informationszeile

Die obere Informationszeile des Editor-Bildschirms kann (von links nach rechts) folgende Bestandteile aufweisen:

Data-Area-Typ	Zeigt den Typ der Data Area an, die sich zurzeit im Editor-Arbeitsbereich befindet: `Local`, `Global` oder `Parameter`. Sie können den Typ mit dem Editor-Kommando `SET TYPE` ändern.
Änderungsanzeige: *	Ein Stern (*) zeigt an, ob der zurzeit im Editor-Arbeitsbereich befindliche Quellcode noch nicht gespeicherte Änderungen enthält. Der Stern (*) erscheint außerdem bei neuem Quellcode, der noch nicht als Source-Objekt gespeichert worden ist. Der Stern (*) ist nur dann vorhanden, wenn die Editor-Profil-Option **Source Status Message** auf `Y` gesetzt ist (siehe ***Editor-Profil***). Der Stern (*) verschwindet, nachdem Sie auf dem Quellcode das Kommando `SAVE` oder `STOW` erfolgreich ausgeführt haben. Siehe auch ***Exit-Funktion***.

Data Area Name	Der Name der Data Area, die sich zurzeit im Editor-Arbeitsbereich befindet. Wenn der Editor-Arbeitsbereich leer ist oder wenn der aktuelle Quellcode noch nicht mit dem Kommando SAVE, CATALOG oder STOW als Source-Objekt gespeichert worden ist, wird kein Name angezeigt.
Lib	Die Library, bei der Sie zurzeit angemeldet sind.
DBID	Die Datenbankkennung der aktuellen Systemdatei.
FNR	Die Dateinummer der aktuellen Systemdatei.

Editor-Kommandozeile

Die Kommandozeile enthält das Wort **Command** als Eingabeaufforderungszeichen. In der Zeile nach diesem Zeichen können Sie Folgendes eingeben:

- Ein beliebiges Systemkommando.

  Beispiel: Das Systemkommando CHECK können Sie benutzen, um die Syntax des Quellcodes zu prüfen, und das Systemkommando SAVE, um den Quellcode zu speichern. Siehe auch ***Data Area speichern und katalogisieren***.

  Weitere Informationen siehe Systemkommandos *Anwendungen mit Natural-Programmobjekten pflegen* in der *Systemkommandos*-Dokumentation.
- Der Name eines Natural-Programms, das ausgeführt werden soll.
- Ein oder mehrere **Editor-Kommandos**.

Anmerkung: Wenn Sie eine Definition geändert haben, indem Sie eine Änderung eingegeben oder ein Editor-Kommando benutzt haben, können Sie ein Systemkommando erst dann eingeben, nachdem Sie Enter drücken.

Richtungsanzeige

Die rechts neben dem Größer-Als-Zeichen > eingegebene Richtungsanzeige bestimmt die Ausführungsrichtung bei bestimmten Editor-Kommandos und Zeilenkommandos:

- **+**

  (Pluszeichen)

  Die Ausführung des Kommandos erfolgt ab der obersten auf dem Bildschirm angezeigten Zeile (bzw. ab der Zeile, in der ein Zeilenkommando eingegeben wird) bis zum Ende des Quellcodes. Dies ist die Standard-Einstellung.
- **-**

  (Minuszeichen)

Die Ausführung des Kommandos erfolgt ab der obersten auf dem Bildschirm angezeigten Zeile (bzw. ab der Zeile, in der ein Zeilenkommando eingegeben wird) bis zum Anfang des Quellcodes.

Die genaue Interpretation können Sie den Beschreibungen der betreffenden Editor-Kommandos und Zeilenkommandos entnehmen. Siehe auch die Editor-Profil-Option **Direction Indicator** im Abschnitt *Editor-Profil*.

Untere Informationszeile

Die untere Informationszeile des Editor-Bildschirms kann folgende Bestandteile haben:

- **Current Source Size - Aktuelle Quellcode-Größe**

  Die Größe des aktuellen Quellcodes (Anzahl der Zeichen, die im Quellcode-Arbeitsbereich noch verfügbar sind). Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Size Information** auf `Y` gesetzt ist (siehe *Editor-Profil*).

- **Free - Zeichen frei**

  Die Anzahl der Zeichen, die im Quellcode-Arbeitsbereich noch verfügbar sind. Diese Information wird nur angezeigt, wenn die Editor-Profil-Option **Source Size Information** auf `Y` gesetzt ist (siehe *Editor-Profil*).

- **S**

  Die Größe (Anzahl der Zeilen) des Quellcodes, der zurzeit bearbeitet wird.

- **L**

  Die Zeilennummer der Quellcode-Zeile, die zurzeit als oberste Zeile angezeigt wird.

Editierbereich benutzen

Der Editierbereich ist entweder leer oder er enthält den Quellcode, der zuletzt mit dem Kommando `EDIT` oder `READ` in den Quellcode-Arbeitsbereich eingelesen wurde. Siehe Beispiel im Abschnitt *Data Area Editor aufrufen*.

Wenn Sie der Quellcode eines existierenden Objekts einlesen, wird der gesamte Quellcode in den Quellcode-Arbeitsbereich geladen und steht dort zum Bearbeiten zur Verfügung. Je nach Größe des Quellcodes kann es sein, dass nicht alle Zeilen, die zum Quellcode gehören, im Editierbereich angezeigt werden. In diesem Fall müssen Sie im Quellcode weiter nach unten blättern (siehe *Editor-Kommandos zum Positionieren*), um zu der Zeile zu gelangen, die Sie ansehen oder ändern möchten.

Wenn Sie den Split-Screen-Modus benutzen, zeigt der Editierbereich weniger Quellcode-Zeilen an. Siehe auch *Split-Screen-Modus*.

Um im Editierbereich zu navigieren:

- Benutzen Sie die Editor-Kommandos. Siehe *Editor-Kommandos zum Positionieren*.

Um Variablen oder Felder anzulegen oder zu ändern:

- Sie können die Definitionen aller Variablen oder Felder in den Spalten der betreffenden Quellcode-Zeile eingeben oder ändern.

  Sie können angeben, ob die Zeichen, die Sie eingeben, automatisch in Großbuchstaben umgesetzt werden sollen, indem Sie die Editor-Profil-Optionen **Editing in Lower Case** und **Dynamic Conversion of Lower Case** entsprechend setzen. Weitere Informationen siehe *Editor-Profil*.

  Oder:

  Sie können eines oder mehrere **Zeilenkommandos** (siehe entsprechenden Abschnitt) benutzen.

  Ein Zeilenkommando wird zum Beispiel benutzt, um eine Zeile einzufügen, die Definition einer Variablen oder eines Feldes aus einem anderen Natural-Objekt zu kopieren oder um Funktion zum Bearbeiten einer erweiterten Feld-Definition aufzurufen (siehe *Erweiterte Felddefinitionen bearbeiten*).

  Oder:

  Sie können eines oder mehrere **Editor-Kommandos** (siehe entsprechenden Abschnitt) benutzen.

  Ein Editor-Kommando wird zum Beispiel benutzt, um einen Zeilenblock zu löschen oder um Präfixe für Namen anzugeben.

Spalten im Editierbereich

Der Editierbereich des Editor-Bildschirms ist in Spalten unterteilt, in denen alle Attributdefinitionen, die zu einer Variablen oder einem Feld gehören, in einer Zeile gepflegt werden können.

Der Editierbereich enthält folgende Spalten:

Spaltenüberschrift	Erklärung
I	Markierungsanzeige. Dies ist ein Informationsfeld, das vom Editor ausgefüllt wird. Diese Spalte ist nicht änderbar. Mögliche Spalteneinträge:
	+ — Zeigt an, dass zu der Variablen oder dem Feld mehr als einer der weiter unten aufgeführten Einträge vorhanden sind.
	E — Zeigt an, dass ein Definitionsfehler festgestellt wurde.
	A — Zeigt an, dass Array-Grenzen mit dem Zeilenkommando `.E` definiert worden sind.
	I — Zeigt an, dass ein Anfangswert mit dem Zeilenkommando `.E` definiert worden ist.
	M — Zeigt an, dass eine Editiermaske und/oder eine Überschrift mit dem Zeilenkommando `.E` definiert worden ist.
	S — Zeigt an, dass sowohl ein Anfangswert als auch eine Editiermaske mit dem Zeilenkommando `.E` definiert worden sind.
	Folgende Anzeigen treffen nur bei PDAs zu:
	blank — Zeigt die Parameterangabe Call-by-Reference an (Standardwert).
	V — Zeigt die Parameterangabe Call-by-Value an.
	R — Zeigt die Parameterangabe Call-by-Value-Result an.
	O — Zeigt einen optionalen Parameter an, der übergeben werden *kann*.
	Weitere Informationen siehe **Funktionscode P** in *Erweiterte Felddefinition bearbeiten*.
T	Typ der Variablen oder des Feldes. Mögliche Typen:

Spaltenüberschrift	Erklärung
	<table><tr><td>B</td><td>Ein Datenblock innerhalb einer GDA.</td></tr><tr><td>C</td><td>Eine benutzerdefinierte Konstante (gilt nicht bei PDAs) oder eine Zähler-Feld (C* Variable). Ein Zähler-Feld wird verwendet, um die Anzahl der Ausprägungen eines multiplen Feldes oder einer Periodengruppe in einer View (DDM) festzustellen Siehe auch *CONSTANT-Option* in der *Statements*-Dokumentation und *Internen Zähler für ein Datenbank-Array referenzieren — C*-Notation* im *Leitfaden zur Programmierung*.</td></tr><tr><td>G</td><td>Eine Gruppe in einer View (DDM).</td></tr><tr><td>M</td><td>Ein multiples Feld in einer View (DDM).</td></tr><tr><td>O</td><td>Handle eines Objekts.</td></tr><tr><td>P</td><td>Eine Periodengruppe in einer View (DDM).</td></tr><tr><td>R</td><td>Die Redefinition einer Variablen oder eines Feldes.</td></tr><tr><td>U</td><td>Der Globally Unique Identifier (GUID).</td></tr><tr><td>V</td><td>Gilt nicht bei PDAs. Eine View-Definition, die aus einer DDM erstellt wurde.</td></tr><tr><td>*blank*</td><td>Eine Benutzervariable oder ein benutzerdefiniertes Feld oder eine Gruppenstruktur (nicht in einer View).</td></tr><tr><td>*</td><td>Ein Kommentarfeld.</td></tr></table>
L	Die Level-Nummer (Strukturebene) der Variablen oder des Feldes (1 - 99). Variablen, die sich nicht in einer hierarchischen Struktur befinden, müssen Sie den Level 1 zuweisen. Level-Nummern können nicht bei Datenblock-Definitionen verwendet werden.
Name	Der Name der Variablen oder des Feldes, des Blocks oder der View. Informationen zu gültigen Namen siehe *Namenskonventionen für Benutzervariablen* in der Dokumentation *Natural benutzen*. Informationen zu Benutzerkonstante siehe CONSTANT in der *Statements*-Dokumentation.

Spaltenüberschrift	Erklärung
	Anstatt der Vergabe eines Variablennamen kann auch die Füllbyte-Notation *n*X benutzt werden. Bei der Füllbyte-Notation können in dem Feld oder der Variablen, das bzw. die neu definiert wird, *n* Füllbytes angegeben werden. Dabei kann *n* für bis zu 10 Stellen (kleiner als 1 GB) stehen. Die Definition von nachgestellten Füllbytes ist optional.
F	Das Natural-Datenformat der Variablen oder des Feldes. Gültige Formate siehe *Format und Länge von Benutzervariablen* und *Spezielle Formate* im *Leitfaden zur Programmierung*. Bei einem Zähler-Feld (C*-Variable) können Sie als Natural-Datenformat/Länge I2 oder I4 angeben. Die Standardeinstellung, bei nicht vorhandener Format/Längenangabe ist N3.
Length	Die Länge der Variablen oder des Feldes. Informationen zu gültigen Längenangaben siehe *Format und Länge von Benutzervariablen* im *Leitfaden zur Programmierung*. Bei den Natural-Datenformaten C, D, T und L ist keine Längenangabe zulässig. Sie können dynamische Variablen definieren, indem Sie DYNAMIC im Feld **Length** angeben. Bei einem Zähler-Feld (C*-Variable) können Sie als Natural-Datenformat/Länge I2 oder I4 angeben. Die Standardeinstellung, bei nicht vorhandener Format/Längenangabe ist N3.
Miscellaneous	Dieses Eingabefeld können Sie benutzen, um die im Abschnitt ***Spalte "Miscellaneous" (Verschiedenes) benutzen*** beschriebenen Definitionen einzugeben.

Spalte "Miscellaneous" (Verschiedenes) benutzen

In diesem Abschnitt werden die Definitionen beschrieben, die in den Feldern der Spalte „Miscellaneous“ (Verschiedenes) eingegeben werden können.

Da das Feld **Miscellaneous** zu kurz sein kann, um alle erforderlichen Angaben zu machen, steht das Zeilenkommando .E für die **erweiterte Bearbeitung der Feld-Definition** zur Verfügung.

Eine Definition kann bis zu 32 Zeichen lang sein, wobei nur 26 Zeichen auf dem Bildschirm angezeigt werden. Mit den Editor-Kommandos M +/- können Sie in dem Feld nach rechts und links blättern. Wenn Sie ein Fragezeichen (?) in der ersten Stelle des Feldes **Miscellaneous** eingeben, öffnet sich ein zusätzliches Fenster, in dem Sie alle 32 Zeichen anzeigen oder zusätzliche Zeichen eingeben können.

Folgendes können Sie definieren:

Array

Geben Sie die obere und die untere Grenze eines Array ein. Weitere Informationen zu Arrays siehe *Arrays* im *Leitfaden zur Programmierung*.

Beispiele:

```
(2,2)   /* 2 dimensions, 2 occurrences
(2,2,2) /* 3 dimensions, 2 occurrences
(1:10,2)
(-1:3,2)
```

Anfangswert

Gilt nicht bei PDAs.

Geben Sie einen Anfangswert gemäß den Natural-Syntax-Definitionen in einem `DEFINE DATA`-Statement ein. Ausführliche Informationen zum Definieren von Anfangswerten siehe *Definition eines Ausgangswerts* und *Ausgangswerte/Konstanten-Werte für ein Array* in der *Statements*-Dokumentation.

Beispiele:

```
INIT<3>
INIT<'ABC'>
INIT<H'F1F2'> /* binary variable (B2)
CONST<12>
INIT ALL<'ABC'>
```

Editiermaske, Überschrift und/oder Druck-/Anzeige-Modus

Editiermasken und Überschriften gelten nicht bei PDAs.

Geben Sie die Definition einer Editiermaske oder einer Überschrift und/oder den Druck-/Anzeige-Modus entsprechend den Regeln ein, die bei den entsprechenden Session-Parametern `EM` oder `EMU`, `HD` und `PM` gelten. Siehe *Parameter-Referenz*-Dokumentation.

Beispiele:

```
(EM=999.99)
(HD='TEXT' EM=XXX.XXX.XX PM=N)
```

Kommentar

Ein Kommentartext, der durch einen Schrägstrich und einen Stern (/*) eingeleitet werden muss.

Name einer DDM

Bei einer View-Definition müssen Sie den Namen der DDM angegeben, von dem die View abgeleitet wird.

Sie können den Namen der DDM ändern, wenn alle Felder der View auch in der DDM mit dem geänderten Namen enthalten sind.

Name übergeordneten „Parent“-Blocks

Bei einer Block-Definition müssen Sie den Namen des entsprechenden, übergeordneten „Parent“-Blocks eingeben.

Erweiterte Felddefinition bearbeiten

Mit der Funktion zur erweiterte Felddefinition können Sie Folgendes definieren:

- Parameter und Arrays in PDAs.
- Arrays, Anfangswerte, Editiermasken und Überschriften in LDAs und GDAs. Dies ist eine Alternative zur Benutzung der Spalte **Miscellaneous**.

Um die Funktion zur erweiterte Felddefinition auszuführen:

1 Geben Sie in der Spalte **T** neben der Variablen oder dem Feld, bei dem Sie erweiterte Attribute definieren möchten, folgendes Zeilenkommando ein:

```
.E
```

Es erscheint das Menü **Extended Field Definition Editing**. Beispiel für eine Benutzervariable in einer LDA:

```
17:54:34                ***** EDIT FIELD *****                      2010-08-23
                  - Extended Field Definition Editing -

Local     *LDA2     Library SAGTEST                        DBID    10 FNR    32



         Code  Function                        Definition
        -----  -------------------------------  ----------
          S     Single Value Initialization    no
          F     Free Mode Initialization       no
          E     Edit Mask Definition           no
          A     Array Index Definition         no
          ?     Help
          .     Exit


        -----  -------------------------------  ----------

   Code   ?  for Field: #USER-VARIABLE-1(A10)
```

Die im Menü **Extended Field Definition Editing** zur Verfügung stehenden Funktionen sind abhängig von der Data Area, dem Typ der Variablen und dem Inhalt des Feldes **Miscellaneous**.

Beispiel: Wenn eine Variable bereits im Feld **Miscellaneous** initialisiert worden ist, dann sind die Funktionen **Single Value Initialization** und **Free Mode Initialization** nicht verfügbar.

Anmerkung: Wird `.E` bei einem DDM-Feld ausgeführt, dann wird der Bildschirm **Define Edit Mask / Header** sofort aufgerufen (siehe folgenden Schritt), weil bei DDM-

Feldern nur Editiermasken und Überschriften definiert werden können. Bei DDM-Feldern können keine Anfangswerte definiert werden.

2 Wählen Sie die benötigte Funktion, indem Sie den entsprechenden Code eingeben. Informationen zu den Funktionen siehe *Funktionen im Menü zur erweiterten Feld-Definition*

Je nach gewählter Funktion erscheint entweder ein weiteres Menü oder ein erweiterter Feld-Editierbereich ähnlich wie im folgenden Beispiel eines **Define Edit Mask / Header**-Bildschirms:

```
17:50:59                ******  EDIT FIELD  ******                    2010-08-23
                      - Define Edit Mask / Header -
Local     *LDA2     Library SAGTEST                       DBID    10 FNR    32
Command

#USER-VARIABLE-1(A10)
-------------------------------------------------------------------------
(EM=                                                                     )
-------------------------------------------------------------------------
Save as unicode edit mask (EMU) .. N  (Y/N)

#USER-VARIABLE-1(A10)
-------------------------------------------------------------------
(HD='                                                           ')
-------------------------------------------------------------------
```

3 Geben Sie eine Definition bzw. einen Funktionscode ein.

Anmerkung: Beim Bearbeiten wird die Definition *nicht* auf Syntaxfehler überprüft. Sie können die Definitionen nach dem Beenden der erweiterten Feld-Definition mit dem Kommando `CHECK` prüfen.

4 Wenn Sie die Definition erstellt haben und zum Menü **Extended Field Definition Editing** zurückkehren, wird die Änderung in der Spalte **Definition** folgendermaßen angezeigt:

```
17:59:00                ***** EDIT FIELD *****                    2010-08-23
                  - Extended Field Definition Editing -

Local     *LDA2     Library SAGTEST                        DBID    10 FNR    32



         Code  Function                        Definition
        -----  ------------------------------  ----------
          S     Single Value Initialization    no
          F     Free Mode Initialization       no
          E     Edit Mask Definition           yes
          A     Array Index Definition         no
          D     Delete all Definitions
          ?     Help
          .     Exit

        -----  ------------------------------  ----------

   Code   ?  for Field: #USER-VARIABLE-1(A10)
```

Wenn Anfangswerte, Editiermasken, Überschriften oder Array-Index-Definitionen definiert worden sind, ändert sich die entsprechende Statusmeldung in der Spalte **Definition** vom `no` nach `yes`. Wenn in einer PDA ein Parametertyp definiert worden ist, wird in der Spalte **Definition** die Abkürzung des Parametertyps (zum Beispiel `Val` für Call-by-Value) angezeigt.

Definitionen, die in den Unterfunktionen **Initial Values** und **Edit Mask / Header** vorgenommen werden, werden sofort in die Data Area übernommen, die zurzeit im Data Area-Editor angezeigt wird, sie werden aber nicht in der Spalte **Miscellaneous** des Editierbereichs angezeigt. Ein entsprechender Eintrag wird nur in der Spalte **I** (Label-Anzeige) angezeigt.

Die Funktionen, die im Menü **Extended Field Definition Editing** zur Verfügung stehen, und die Kommandos, die in einem erweiterten Feld-Bearbeitungsbereich verfügbar sind, werden im folgenden Abschnitt beschrieben:

- Funktionen im Menü zur erweiterten Feld-Definition
- Kommandos im erweiterten Feld-Editierbereich

Funktionen im Menü zur erweiterten Feld-Definition

In der folgenden Tabelle werden alle Funktionen, die im Menü **Extended Field Definition Editing** zur Verfügung stehen können, erklärt:

Bei einer Attribut-Kontrollvariablen sind nur die Funktionscodes **S**, **F**, **P**, **A** und **D** zulässig.

Bei einem Feld, das ein anderes Feld neu definiert, sind nur die Funktionscodes **E**, **A** und **D** zulässig.

Funktionscode	Funktion
S	Definiert einen Anfangswert für die angegebene Variable oder das angegebene Feld im Einzelwertmodus (**S**ingle Value Initialization). Sie brauchen nur den erforderlichen Variablen-/Feldwert einzugeben. Alle sonst noch erforderlichen Angaben (einschließlich Apostrophe bei alphanumerischen Variablen oder Feldern sowie Präfixe wie z.B. `H` für hexadezimal) werden automatisch generiert. Beispiel: Aus einem Anfangswert `F1F2` für eine binäre Variable (B2) erzeugt der Editor `INIT <H'F1F2'>`. Handelt es sich bei der Variablen oder dem Feld um ein Array, kann ein Ausgangswert für jede einzelne Ausprägung definiert werden (muss aber nicht). Bei Arrays mit großen alphanumerischen Variablen oder Feldern (z.B. Arrays dynamischer Variablen) müssen Sie im Feld **Next index (Y/N)** ein `Y` (ja) eingeben, um zum nächsten Index des Arrays zu springen. Bei Arrays kann in der Kommandozeile Stern-Notation (*) eingegeben werden, um den Wert in der letzten Zeile der vorigen Seite bis zum Ende der aktuellen Seite zu wiederholen. Bei Attribut-Kontrollvariablen wird ein Bildschirm angezeigt, in dem Sie Attribute und Farben als Anfangswerte auswählen können. Weitere Informationen siehe Session-Parameter `AD` und `CD` in der *Parameter-Referenz*-Dokumentation. Um anstelle eines Anfangswertes einen Konstanten-Wert zu definieren, müssen Sie im Feld **Define as CONSTANT (Y/N)** ein `Y` eingeben.
F	Definiert einen Anfangswert für das angegebene Feld im Free-Modus (**F**ree Mode Initialization). Dazu ist ein Free-Modus-Editor verfügbar, in dem Sie die Ausgangswerte gemäß den üblichen Natural-Syntax-Definitionen in einem `DEFINE DATA`-Statement eingeben können. Ausführliche Informationen zu Ausgangswerten siehe *Definition eines Ausgangswerts Ausgangswerte/Konstanten-Werte für ein Array* in der *Statements* -Dokumentation. Siehe auch ***Beispiele*** unter ***Ausgangswert***.
E	Definiert eine Editiermaske und/oder eine Überschrift für das angegebene Feld gemäß den Natural-Regeln für die Angabe von Editiermasken und Überschriften. Wenn Sie sowohl eine Editiermaske als auch eine Überschrift angeben, dürfen beide zusammen nicht länger 57 Zeichen sein. Wenn Sie jedoch nur eine Editiermaske angeben, darf diese bis zu 63 Zeichen lang sein. Wenn Sie nur eine Überschrift angeben, darf diese bis zu 58 Zeichen lang sein. Wenn `.E` bei einem DDM-Feld eingeben wird, dann wird diese Funktion sofort aufgerufen, weil bei DDM-Feldern nur Editiermasken und Überschriften definiert werden können.
P	Diese Funktion gilt nur bei PDAs. Mit dieser Funktion können Sie festlegen, auf welche Weise der Wert, der bei einem `CALLNAT`-Statement als Parameter angegeben ist, von einem Programm an ein aufgerufenes

<table>
<tr><th>Funktionscode</th><th>Funktion</th></tr>
<tr><td></td><td>Objekt (z.B. Subprogramm) übergeben werden soll. Im oberen Eingabefeld können Sie einen der folgenden Code-Werte eingeben:

<table><tr><td>D</td><td>Call-by-Reference (Standardwert).</td></tr><tr><td>V</td><td>Call-by-Value.</td></tr><tr><td>R</td><td>Call-by-Value-Result.</td></tr></table>
Weitere Informationen siehe BY VALUE und BY VALUE RESULT in der Beschreibung des DEFINE DATA-Statement in Definition von Parameterdaten und operand2 in der Beschreibung des CALLNAT-Statement in der Statements-Dokumentation.

Zusätzlich können Sie im unteren Eingabefeld angeben, ob ein Parameter übergeben werden muss oder kann:
<table><tr><td>N</td><td>Ein Parameter muss übergeben werden (Standardwert).</td></tr><tr><td>Y</td><td>Ein optionaler Parameter kann übergeben werden.</td></tr></table>
Weitere Informationen siehe entsprechende Option OPTIONAL in der Beschreibung des DEFINE DATA-Statement in Definition von Parameterdaten und operand2 in der Beschreibung des CALLNAT-Statement in der Statements-Dokumentation.</td></tr>
<tr><td>A</td><td>Definiert Array-Grenzen für das angegebene Feld. Dazu ist ein Free-Modus-Editor verfügbar, in dem Sie die Grenzdefinitionen gemäß den üblichen Natural-Syntax-Definitionen eingeben können. Während der Bearbeitung erfolgt jedoch keine Überprüfung der angegebenen Werte (dazu müssen Sie das Kommando CHECK benutzen).</td></tr>
<tr><td>D</td><td>Löscht alle Definitionen, die mit den Funktionscodes S, F, E, P und A vorgenommen wurden. Es erscheint ein zusätzlicher Bildschirm, in dem Sie die zu löschenden Definitionen angeben können.

Standardmäßig sind alle Definitionen mit einem Y (ja) vorbelegt. Wenn Sie eine Definition nicht löschen möchten, müssen Sie hinter der betreffenden Definition das Y entfernen oder es durch ein N (nein) ersetzen.</td></tr>
</table>

Kommandos im erweiterten Feld-Editierbereich

In der folgenden Tabelle werden die Kommandos erläutert, die in der Kommandozeile eines erweiterten Feld-Editierbereichs eingegeben werden können:

Kommando	Funktion
EDIT	Rückkehr zum Editierbereich des Editor-Bildschirms.
I	Zum nächsten Index des Arrays springen.
.	Rückkehr zum vorigen Bildschirm, um die Verarbeitung fortzusetzen.
--	Rückkehr zum Anfang der Anfangswert-Spezifikation(en).
+	Eine Seite nach unten blättern. Beim Erreichen der letzten Seite, oder falls nur eine Seite vorhanden ist, erfolgt die Rückkehr zum Editierbereich des Editor-Bildschirms.
*	Kopiert den Anfangswert der letzten Ausprägung der vorigen Seite in alle leeren Felder der aktuellen Seite. Steht nur bei Arrays im Einzelwertmodus zur Verfügung.

Zeilenkommandos

Zeilenkommandos werden in der Spalte **T** einer Quellcode-Zeile eingegeben.

Anmerkung: Es wird empfohlen, am Ende eines jeden Zeilenkommandos ein Leerzeichen einzugeben. Dadurch wird verhindert, dass der Editor versucht, die anschließenden Informationen in der Zeile als Bestandteil des Zeilenkommandos zu interpretieren.

Das Standard-Fluchtzeichen, das vor jedem Zeilenkommando vorangestellt werden muss, ist ein Punkt (.). Sie können das Standard-Fluchtzeichen mit der Editor-Profil-Option **Escape Character for Line Command** ändern (siehe *Editor-Profil*).

Die beim Data Area Editor zur Verfügung stehenden Zeilenkommandos sind nachfolgend aufgeführt. Die Notation (*n*), (*nnn*) oder (*nnnn*) zeigt einen Wiederholungsfaktor an. Der Standard-Wiederholungswert ist 1 (mit Ausnahme beim Kommando .I). Eine Erklärung der verwendeten Syntaxsymbole finden Sie im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation.

Kommando	Funktion
.C[(*nnnn*)]	Kopiert die Zeile, in der das Kommando eingegeben wurde. Siehe auch *Anmerkungen zu den Zeilenkommandos*.
.CX[(*nnnn*)] or .CY[(*nnnn*)]	Kopiert die Zeile, die mit X oder mit Y markiert ist. Siehe auch Zeilenkommandos .X und .Y sowie *Anmerkungen zu den Zeilenkommandos*.
.CX-Y[(*nnnn*)]	Kopiert den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist. Siehe auch Zeilenkommandos .X und .Y sowie *Anmerkungen zu den Zeilenkommandos*.

Kommando	Funktion
.D	Löscht eine oder mehrere Zeilen ab der Zeile, in der Sie das Kommando eingeben, bis zum Ende des Quellcodes (unabhängig von der Einstellung in der Richtungsanzeige). Wird das Zeilenkommando bei einem einzelnen Feld eingegeben wird, dann wird nur dieses Feld gelöscht. Wird das Zeilenkommando bei einem Teil einer hierarchischen Struktur (View, Gruppe, Redefinition) eingegeben, dann werden alle nachfolgenden Definitionen auf den untergeordneten Ebenen (Level) ebenfalls gelöscht. Wenn Sie zum Beispiel .D bei einer auf Ebene 2 definierten Gruppe eingeben, wird alles, was zu dieser Gruppe gehört und eine Ebenen-Nummer größer als 2 hat ebenfalls gelöscht, und zwar bis zur nächsten Ebene-2-Definition (die es jedoch nicht betrifft). Kommentarzeilen (denen üblicherweise keine Ebene zugewiesen wird) werden ebenso als auf einer untergeordneten Ebene befindlich behandelt. Um ein nicht erwünschtes Löschen eines Kommentars zu vermeiden, müssen Sie ihm eine geeignete Ebene zuweisen.
.D(*nnnn*)	Löscht *nnnn* Zeilen ab der Zeile, in der Sie das Kommando eingeben, bis zum Ende des Quellcodes (unabhängig von der Einstellung in der Richtungsanzeige). Der Standardwert ist 1 Zeile. Im Gegensatz zum Zeilenkommando .D (siehe oben) betrifft .D(*nnnn*) nur die angegebene Anzahl an Zeilen, und zwar unabhängig von einer hierarchischen Struktur.
.E	Ruft den Bildschirm **Extended Field Definition Editing** auf, der dazu dient, Array-Grenzen, Anfangswerte, Editiermasken, Überschriften und Parameter-Attribute zu definieren. Weitere Informationen siehe ***Erweiterte Felddefinition bearbeiten***.
.F(*file-name*)	Fügt eine Predict-Datei ein (gilt bei den Dateitypen „Conceptual“, „Standard“, „Sequential“ und „Other“). Der Name einer Predict-Datei wird in der Spalte **Miscellaneous** angezeigt.
.I[(*n*)]	Fügt *n* Leerzeilen hinzu. Dabei kann *n* im Bereich von 1 bis 9 liegen. Wird *n* nicht (oder nicht korrekt) angegeben, werden standardmäßig 10 Zeilen (5 Zeilen im Split-Screen-Modus) hinzugefügt. Zeilen, die leer gelassen werden, werden anschließend aus dem Quellcode entfernt. Dies ist abhängig von der Einstellung der Editor-Profil-Option **Empty Line Suppression**. Siehe ***Editor-Profil***. **Anmerkung:** Es kein nur jeweils ein .I-Kommando ausgeführt. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
.I(*obj*)	Kopiert Variablen- oder Parameter-Definitionen aus einem anderen Natural-Objekt folgenden Typs: Data Area Programm Subprogramm Subroutine

Kommando	Funktion
	Helproutine Map Function Wenn das bei *obj* angegebene Objekt keine Data Area ist, muss es als katalogisiertes Objekt verfügbar sein. Es erscheint ein Fenster im Data Area Editor-Bildschirm, in dem Sie eine der folgenden Datendefinitionen auswählen können, die in die aktuelle Data Area aufgenommen werden sollen: ■ Alle in dem angegebenen Objekt enthaltenen lokalen Variablen und Parameter (einschließlich der, die aus LDAs und/oder PDAs übernommen wurden). ■ Alle in dem angegebenen Objekt enthaltenen lokalen Variablen (einschließlich der, die aus LDAs übernommen wurden). ■ Nur diejenigen lokalen Variablen, die in dem angegebenen Objekt enthalten sind. ■ Alle Parameter, die in dem angegebenen Objekt enthaltenen sind (einschließlich der, die aus PDAs übernommen wurden). ■ Nur diejenigen Parameter, die in dem angegebenen Objekt enthaltenen sind. Außerdem können Folgendes wählen: ■ Alle nicht benutzten und benutzten Variablen (standardmäßig gewählt). ■ Nur die benutzten lokalen Variablen, die referenziert oder geändert werden. ■ Nur die nicht benutzten Variablen, die definiert, aber nicht referenziert oder geändert werden. Wenn Sie nicht benutzte Variablen übernehmen, kann es sein, dass die Ebenen-Nummern nicht die richtige Reihenfolge aufweisen. Daher sollten Sie die Ebenen aller übernommenen nicht benutzten Variablen auf korrekte Nummerierung prüfen, bevor Sie die Data Area kompilieren. Wenn Sie Variablen-Definitionen aus Objekten übernehmen, die keine `DEFINE DATA`-Statement-Definition enthalten (d.h., aus Objekten, die im Reporting Mode kodiert wurden), kann es vorkommen, dass Variablen-Redefinitionen (siehe `REDEFINE`-Statement in der *Statements*-Dokumentation) an die falsche Stelle, d.h. hinter die falsche Variable, platziert werden. Daher sollten Sie alle Variablen-Definitionen und -Redefinitionen auf korrekte Positionierung prüfen, bevor Sie die neue Data Area kompilieren. Falls bei einer Variablen-Redefinition mehr als eine Variable resultieren, wird jede Variable als einzelne, eigene Redefinition eingebaut, wobei gegebenenfalls Füllbytes verwendet werden. Falls das angegebene Objekt mit dem Natural Optimizer Compiler katalogisiert worden ist, können Anfangswerte und Konstanten nicht übernommen werden.

Kommando	Funktion
	Falls das Objekt, das Sie einfügen möchten, Merkmale aufweist, die der Area Editor nicht unterstützt, erscheint eine entsprechende Meldung und die betroffene Zeile wird als Kommentarzeile markiert. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
`.I(obj,ssss,nnnn)`	Fügt eine GDA, eine LDA oder eine PDA ein. Dieses Merkmal wird nur bei Data Areas unterstützt, die keine Anfangswerte oder Editiermasken enthalten. Der Eintrag *ssss* kann benutzt werden, um anzugeben, bei welcher Zeile die Einfügung beginnen soll. Wenn Sie beispielsweise *ssss* auf `20` setzen, beginnt die Einfügung bei der 20. Zeile der Data Area. Der Eintrag *ssss* kann benutzt werden, um die Anzahl der einzufügenden Zeilen anzugeben. Wird *ssss* und/oder *nnnn* bei einem Objekt angegeben, bei dem es sich nicht um eine Data Area handelt (siehe Kommando `.I(obj)`), wird/werden der/die angegebene(n) Wert(e) nicht beachtet. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
`.L`	Macht alle seit dem letzten Drücken der Enter-Taste an der Zeile vorgenommenen Änderungen rückgängig.

`.MX`
 	
or
 	
`.MY`
	Verschiebt die mit `X` oder mit `Y` markierte Zeile.
	
	Siehe auch Zeilenkommandos `.X` und `.Y` und ***Anmerkungen zu den Zeilenkommandos***.

`.MX-Y`	Verschiebt den Zeilenblock, der mit den Markierungen `X` und `Y` eingegrenzt ist. Siehe auch Zeilenkommandos `.X` und `.Y` und ***Anmerkungen zu den Zeilenkommandos***.
`.N`	Markiert (unsichtbar) eine Zeile, die mit den Editor-Kommando `POINT` an den Anfang des Quellcode-Arbeitsbereichs positioniert werden soll. Siehe *Editor-Kommandos zum Positionieren* . Die Markierung wird automatisch gelöscht, wenn ein Fehler bei einem Zeilen- oder Editor-Kommando auftritt oder wenn das `RESET`-Kommando ausgeführt wird.
`.P`	Positioniert die mit diesem Kommando markierte Zeile an den Anfang des Bildschirms.
`.R`	Redefiniert eine Variable oder ein Feld als eine einzelne Variable oder eine Variablen-Gruppe. Mit der optionalen Füllbyte-Notation (*n*`X`), *n* können Sie in der Variablen oder dem Feld, die/das neu definiert wird, Füllbytes bezeichnen. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.

Kommando	Funktion
.V[(*ddm-name*[,NOFL])]	Gilt nicht bei PDAs. Definiert eine View aus einem DDM. Geben Sie bei *ddm-name* das DDM an, aus dem Sie eine View definieren wollen. Die Felder des DDM werden dann im Editierbereich angezeigt. Markieren Sie die Felder, die in die View übernommen werden sollen, indem Sie ein beliebiges Zeichen in der Spalte **I** neben dem (den) benötigten Feld(ern) ein. Wenn Sie Enter drücken, werden diese Felder in eine View-Definition in der aktuellen Data Area kopiert, wobei der Name der View (standardmäßig ist das der Name des DDM) der Ebene 1 zugeordnet wird. Im **Split-Screen-Modus** wird das DDM, das sich zurzeit im geteilten Bildschirm befindet, im Editierbereich angezeigt, wenn Sie .V ohne *ddm-name* eingeben. Wird .V(*ddm-name*) in einer View angegeben, die denselben Namen hat wie bei *ddm-name* angegeben, werden die gewählten Felder in diese View eingefügt, und es wird keine neue View definiert. Wird NOFL angegeben, werden die gewählten Felder ohne Format/Längen-Angaben eingefügt. Wenn eine Periodengruppe oder ein multiples Feld, die/das in einem mit Predict als PC bzw. MC erzeugten DDM definiert ist, in eine Data Area eingefügt wird, dann wird automatisch ein Zähler-Feld (C*-Variable) generiert und vor die Gruppe bzw. das Feld platziert. Der Index für eine solche Periodengruppe oder ein solches multiples Feld wird mit der in Predict definierten Anzahl an Ausprägungen definiert. Ist die Anzahl der Ausprägungen in Predict nicht definiert worden, wird der Wert 191 verwendet. Wenn Predict aktiv ist, werden auch Redefinitionen und Kommentare eingefügt. Bei VSAM-Views wird immer die aktuelle Anzahl der Ausprägungen angezeigt. Zusätzlich können VSAM-Views noch Informationen zu Subdeskriptoren und Superdeskriptoren enthalten. Weitere Informationen siehe *Natural for VSAM*-Dokumentation.
.VG	Gilt nur bei Views. Konvertiert alle Felder einer View-Definition in Felder einer Gruppenstruktur. Geben Sie .VG neben der Zeile ein, die in der Spalte **T** ein V enthält. Wurde die View-Definition mit der NOFL-Option eingefügt (siehe Zeilenkommando .V), wird die Natural-Format/Längendefinition aus dem DDM gelesen und bei den Feldern der View hinzugefügt. Falls die View-Definition ein Zähler-Feld enthält (C*-Variable), ändert sich der Feldname von C*NAME in C_NAME und die Natural-Format/Längendefinition wird auf N3 gesetzt.

Kommando	Funktion
	Anmerkung: .VG gilt nicht bei Masken- oder Überschriftendefinitionen. Wenn Sie eine Data Area in eine PDA konvertieren möchten, müssen Sie diese Definitionen in der LDA oder GDA entfernen.
.X	Gilt nicht bei Periodengruppen, multiplen Feldern oder View-Definitionen. Markiert eine Zeile mit einem X. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
.Y	Gilt nicht bei Views, Periodengruppen, multiplen Feldern oder Redefinitionen. Markiert eine Zeile mit einem Y. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
.*	Generiert ein Zähler-Feld (C*-Variable) für multiple Felder oder Felder in einer Periodengruppe. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.
number [(*nnn*[,*m*])]	Dieses Kommando ist nur im Split-Screen-Modus verfügbar, und wenn sich ein DDM im Split-Screen-Bereich befindet. Um Felder und Gruppen aus dem Split-Screen-Bereich zu übernehmen, muss die Zeilennummer des Feldes oder der Gruppe in der ersten Spalte ohne punkt (.) angegeben werden. Felder und Gruppen aus dem Split-Screen-Bereich können als Felder einer View (wenn *number* in einer View eingegeben wird) oder als Benutzervariablen eingefügt werden. Wenn das gewählte Feld denselben Namen hat wie das Feld, bei dem das Kommando eingeben wurde, dann wird es nicht eingefügt, sondern ersetzt. Wenn Sie die Notation *nnn* benutzen, können Sie mehrere Zeilen aus dem geteilten Bildschirm übernehmen. Dabei steht *nnn* für die Anzahl der einzufügenden Zeilen. Die Notation *m* kann benutzt werden, um dem Feld oder der Gruppe, das bzw. die eingefügt werden soll, eine Ebenen-Nummer zuzuweisen. Die Ebenen-Nummer in der Data Area kann geändert werden. Siehe auch ***Anmerkungen zu den Zeilenkommandos***.

Anmerkungen zu den Zeilenkommandos:

- Die Kommandos .I(*obj*), .R und .* stehen im Full-Screen-Modus (Vollbild), aber nicht Split-Screen-Modus (geteilter Bildschirm) zur Verfügung.
- Wenn sowohl das Kommando .X als auch das Kommando .Y bei einer Zeile angewendet werden, wird sie behandelt, als ob sie mit einem X und mit einem Y markiert ist; die tatsächlich angezeigte Zeilenmarkierung, die diesen Status wiedergeben soll, ist ein Z.

- Ist die **Richtungsanzeige** auf + (Pluszeichen) gesetzt, werden die kopierten, eingefügten oder verschobenen Zeilen *nach* der Zeile platziert, in der das entsprechende Kommando eingegeben wurde.

  Ist die **Richtungsanzeige** auf - (Minuszeichen) gesetzt, werden die kopierten, eingefügten oder verschobenen Zeilen *vor* der Zeile platziert, in der das entsprechende Kommando eingegeben wurde.

Editor-Kommandos

Editor-Kommandos werden in der Kommandozeile des Data Area Editors eingegeben. Die verfügbaren Editor-Kommandos werden in der nachfolgenden Tabelle beschrieben. Die in den Editor-Kommandos verwendeten Syntax-Symbole werden im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erklärt.

Editor-Kommando	Funktion
ADD[(*n*)]	Fügt *n* Leerzeilen hinzu. Wenn die **Richtungsanzeige** auf + (Pluszeichen) gesetzt ist, werden die Leerzeilen nach der letzten Zeile des in Bearbeitung befindlichen Objekts hinzugefügt. Wenn die **Richtungsanzeige** auf - (Minuszeichen) gesetzt ist, werden die Leerzeilen vor der ersten Zeile des Objekts hinzugefügt. Der Wert für *n* darf im Bereich von 1 bis 9 liegen. Wenn Sie für *n* keinen oder einen nicht korrekten Wert angeben, werden standardmäßig 9 Zeilen (4 Zeilen im Split-Screen-Modus) hinzugefügt. Wenn Sie das nächste Mal Enter drücken, werden alle Zeilen, die dann noch leer sind, wieder entfernt.
CANCEL oder . (Punkt)	Die Editor-Sitzung wird abgebrochen. Alle seit dem letzten Speichern (SAVE) vorgenommenen Änderungen werden *nicht* gespeichert.
CATALOG [*object-name*]	Führt das Systemkommando CATALOG aus. Die aktuelle Data Area-Definition wird geprüft und katalogisiert. Wenn Sie eine neu angelegte Data Area-Definition katalogisieren oder wenn Sie die aktuelle Data Area-Definition kopieren wollen, müssen Sie zusammen dem Kommando einen Objektnamen angeben. Siehe auch ***Data Area speichern und katalogisieren***.
CHANGE ['*scan-value*'*replace-value*']	Durchsucht die Data Area nach der als Suchwert (*scan-value*) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (*replace-value*) eingegebenen Zeichenkette. Als Eingabebegrenzungszeichen kann jedes

Editor-Kommando	Funktion
	Sonderzeichen benutzt werden, dass in einem Natural-Variablennamen nicht gültig ist. Jede Zeile, in der eine Zeichenkette ersetzt wurde, wird am linken Zeilenrand mit einem `R` markiert. Weitere Informationen siehe `SCAN`-Kommando.
`CHECK`	Führt das Systemkommando `CHECK` aus, das die Syntax der aktuellen Data Area prüft. Falls ein Fehler gefunden wird, wird die fehlerhafte Zeile mit einem `E` markiert, und eine entsprechende Meldung wird in der Meldungszeile angezeigt. Wird kein Fehler gefunden, wird eine Meldung angezeigt, die die erfolgreiche Beendigung der Prüfung anzeigt. Bei Ausführung von `CHECK` werden die Einträge in der Spalte **Miscellaneous** (Verschiedenes) in folgender Reihenfolge sortiert: Array-Definition Anfangswert Editiermaske, Überschrift und/oder PM - Druck-/Anzeige-Modus Name eines DDM oder eines „Parent“-Blocks Kommentar
`CLEAR`	Führt das Systemkommando `CLEAR` aus, das den Arbeitsbereich löscht. Änderungen, die an der Data Area vorgenommen wurden, die sich zurzeit im Arbeitsbereich befindet, gehen dabei verloren, wenn sie nicht vorher gespeichert wurden.

`DX`
 	
or
 	
`DY`
	Löscht die mit `X` oder `Y` markierte Zeile.
	
	Siehe auch Zeilenkommandos `.X` und `.Y`.

`DX-Y`	Löscht den mit den Markierungen `X` und `Y` eingegrenzten Zeilenblock. Siehe auch Zeilenkommandos `.X` und `.Y`.

`EX`
 	
or
 	
`EY`
	Löscht alle Zeilen ab dem Anfang des Editierbereichs bis zur mit `X` markierten Zeile (die jedoch nicht gelöscht wird) oder ab der Zeile, die auf die mit `Y` markierte Zeile folgt, bis zum Ende des Editierbereichs.
	
	Siehe auch Zeilenkommandos `.X` und `.Y`.

`EX-Y`	Löscht alle Zeilen im Editierbereich mit Ausnahme des Blocks, der mit `X` und `Y` eingegrenzt ist. Siehe auch Zeilenkommandos `.X` und `.Y`.
`EXIT`	Die Editor-Sitzung wird beendet. Alle am Quellcode vorgenommenen Änderungen werden in Abhängigkeit von der Einstellung im **Editor-Profil** gespeichert. Siehe ***Exit-Funktion***.

Editor-Kommando	Funktion
GENERATE [*object-name*]	Generiert ein Natural-Objekt des Typs Copycode aus der Data Area-Definition, die sich zurzeit im Arbeitsbereich befindet. Der Programm-Editor-Bildschirm erscheint. Er enthält den generierten Copycode im Editierbereich. Der Quellcode enthält ein DEFINE DATA LOCAL-Statement und das entsprechende END-DEFINE-Statement. Wird ein Objektname (*object-name*) angegeben, dann wird der generierte Copycode unter diesem Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert.
M +|-	Blätterkommandos für die Spalte **Miscellaneous** (Verschiedenes). + — Blättert nach rechts. - — Blättert nach links.
PROFILE [*name*]	Ruft den **Editor-Profil**-Bildschirm auf. Dort können Sie sich Ihre aktuellen Editor-Profil-Einstellungen ansehen und diese ändern. Weitere Informationen siehe Abschnitt ***Editor-Profil***.
READ *object-name*	Führt das Systemkommando READ aus. Damit wird eine existierende Data Area-Definition in den Arbeitsbereich eingelesen. Die vollständige Syntax des Kommandos finden Sie bei READ in der *Systemkommandos*-Dokumentation.
RESET	Löscht die aktuellen Zeilenmarkierungen X und Y sowie alle zuvor mit dem Zeilenkommando .N gesetzten Markierungen. Siehe auch Zeilenkommandos .X und .Y.
SAVE [*object-name*]	Führt das Systemkommando SAVE aus, das die aktuelle Data Area-Definition speichert. Wenn Sie eine neu angelegte Data Area-Definition speichern oder wenn Sie die aktuelle Data Area kopieren wollen, müssen Sie bei dem Kommando einen Objektnamen angeben. Siehe auch ***Data Area speichern und katalogisieren***.
SCAN *scan-value*	Durchsucht die Data Area nach einer Zeichenkette (*scan-value*) in der Spalte **Name** (standardmäßig) und/oder in der Spalte **Miscellaneous** des Editor-Bildschirms (abhängig davon, ob zuvor das Kommando SET SCAN ausgeführt wurde). Jede Zeile, in der der Suchwert (*scan-value*) gefunden wird, wird am linken Zeilenrand mit einem S gekennzeichnet. Die erste Zeile, die den Suchwert (*scan-value*) enthält wird, abhängig von der aktuellen Einstellung der **Richtungsanzeige**, in die oberste oder in die unterste Zeile positioniert. **Anmerkung:** Mit dem Kommando SCAN erfolgt eine exakte Suche nach dem angegebenen Suchwert (*scan-value*). Dies sollte bei der

Editor-Kommando	Funktion
	Suche nach DBCS-Zeichen (DBCS = Double-Byte Character Set) berücksichtigt werden. Wenn die **Richtungsanzeige** auf + (Pluszeichen) gesetzt ist, erfolgt die Suche ab der ersten, auf dem Bildschirm angezeigten Zeile bis zur letzten Zeile des Arbeitsbereiches. Wenn die Richtungsanzeige auf - (Minuszeichen) gesetzt ist, erfolgt die Suche ab der letzten, auf dem Bildschirm angezeigten Zeile bis zur ersten Zeile des Arbeitsbereiches.
SCAN =[+|-]	Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts (*scan-value*). Die Suchrichtung für den gegebenen Suchvorgang können Sie explizit angeben, indem Sie SCAN=+ oder SCAN=- eingeben. Die Einstellung in der **Richtungsanzeige** wird dann ignoriert. **Anmerkung:** Das beim SCAN-Kommando verwendete Gleichheitszeichen (=) ist das Standard-Zuweisungszeichen für Eingabe-Parameter. Falls ein anderes Zeichen als Zuweisungszeichen für Eingabe-Parameter angegeben worden ist (siehe Session-Parameter IA in der *Parameter-Referenz-Dokumentation*), muss dieses Zeichen stattdessen benutzt werden.
SET ABS [ON|OFF]	Legt fest, ob das SCAN-Kommando im absoluten oder nicht im absoluten Modus arbeitet. ON: Das SCAN-Kommando arbeitet im absoluten Modus. Das bedeutet, dass es nicht nötig ist, den Wert, nach dem gesucht werden soll, durch Leerzeichen oder Sonderzeichen abzugrenzen. OFF: Das SCAN-Kommando arbeitet nicht im absoluten Modus. Das bedeutet, dass Sie den Wert, nach dem gesucht werden soll, durch Leerzeichen oder Sonderzeichen abgrenzen müssen. Der Standardwert ist OFF. Das Kommando SET ABS entspricht der Editor-Profil-Option **Absolute Mode for SCAN/CHANGE**. Siehe Abschnitt ***Editor-Profil***.
SET PREFIX *prefix*|OFF	Dient zur Angabe eines Präfixes für Variablen- oder Feldnamen. Dieser Präfix wird bei jeder eingegebenen oder geänderten Zeile automatisch dem in der Spalte **Name** eingegebenen Wert vorangestellt, falls der Name noch nicht mit diesem Präfix beginnt.

<table>
<tr><th>Editor-Kommando</th><th>Funktion</th></tr>
<tr><td></td><td>Ist die verkettete Variable oder das verkettete Feld länger als 32 Bytes, erscheint eine entsprechende Meldung und der Wert in der Spalte Name kann verkürzt werden. Wird keine Verkürzung vorgenommen, dann wird der Präfix nicht eingefügt.</td></tr>
<tr><td>SET SAVEFORMAT V31|V41

or
SET SF V31|V41</td><td>Gibt das Standard-Quellcode-Format für Data Areas an.

Einstellung V31: Data Area-Quellcodes werden im Format gespeichert, das kompatibel zu Natural Version 3.1 ist.

Einstellung V41: Data Area-Quellcodes werden im erweiterten Quellcode-Format gespeichert (Standardeinstellung).

Siehe auch Quellcode-Format zum Speichern der Data Area.</td></tr>
<tr><td>SET SCAN COMMENT|NAME|ALL</td><td>Legt die Spalte(n) fest, in denen das SCAN-Kommando nach einem Suchwert (scan-value) suchen soll:

<table><tr><td>COMMENT</td><td>Die Spalte Miscellaneous wird durchsucht.</td></tr><tr><td>NAME</td><td>Die Spalte Name wird durchsucht.</td></tr><tr><td>ALL</td><td>Die Spalten Name und Miscellaneous werden durchsucht.</td></tr></table>
Die Standardeinstellung ist NAME.</td></tr>
<tr><td>SET SIZE ON|OFF</td><td>Wenn SET SIZE auf ON gesetzt ist, wird die Größe der Data Area in der unteren Informationszeile des Editor-Bildschirms angezeigt.

Das Kommando SET SIZE entspricht der Editor-Profil-Option Source Size Information. Siehe Editor-Profil.</td></tr>
<tr><td>SET STAY ON|OFF</td><td>Wenn STAY auf ON gesetzt ist, wird der aktuelle Bildschirm weiterhin angezeigt. Eine Vorwärts-/Rückwärtspositionierung kann nur mit den Positionierungskommandos bewirkt werden.

Wenn STAY auf OFF gesetzt ist, wird nach Drücken von Enter der nächste Bildschirm angezeigt, falls in dem aktuellen Bildschirm keine Änderungen vorgenommen worden sind.

Das Kommando SET STAY entspricht der Editor-Profil-Option Stay on Current Screen. Siehe Editor-Profil.</td></tr>
<tr><td>SET TYPE G|L|A</td><td>Mit diesem Kommando können Sie den Typ der aktuellen Data Area ändern:

<table><tr><td>G</td><td>Global Data Area</td></tr><tr><td>L</td><td>Local Data Area</td></tr><tr><td>A</td><td>Parameter Data Area</td></tr></table></td></tr>
</table>

Editor-Kommando	Funktion
SPLIT *parameter*	Dieses Kommando bewirkt, dass der Editor-Bildschirm zwecks gleichzeitiger Anzeige eines anderen Objekts in zwei Hälften unterteilt wird. Siehe ***Split-Screen-Modus***. *parameter* steht für einen Parameter, den Sie mit dem Kommando eingeben müssen. Siehe ***Split-Screen-Kommandos***.
STOW [*object-name*]	Führt das Systemkommando STOW aus, mit dem die aktuelle Data Area-Definition gespeichert und katalogisiert wird. Wenn Sie eine neue Data Area-Definition speichern und katalogisieren wollen oder wenn Sie die aktuelle Data Area kopieren wollen, müssen Sie einen Objektnamen angeben. Sonst erscheint eine entsprechende Meldung. Siehe auch ***Data Area speichern und katalogisieren***.

Editor-Kommandos zum Positionieren

In diesem Abschnitt werden die Editor-Kommandos beschrieben, die zum Navigieren in der aktuellen Data Area benutzt werden können. Diese Editor-Kommandos können Sie in der Kommandozeile des Data Area Editors eingeben.

Kommando	Funktion
Enter oder +P oder +	Eine Seite vorwärts positionieren.
-P oder -	Eine Seite rückwärts positionieren.
+H	Eine halbe Seite vorwärts positionieren.
-H	Eine halbe Seite rückwärts positionieren.
T oder --	An den Anfang des Quellcodes positionieren.
B	An das Ende des Quellcodes positionieren.

Kommando	Funktion
oder ++	
+nnnn	Um *nnnn* (maximal 4 Ziffern) Zeilen vorwärts positionieren.
-nnnn	Um *nnnn* (maximal 4 Ziffern) Zeilen rückwärts positionieren.
X	Auf die mit X markierte Zeile positionieren.
Y	Auf die mit Y markierte Zeile positionieren.
POINT	Auf die Zeile positionieren, in der das Zeilenkommando .N eingegeben wurde. Siehe auch Zeilenkommando .P.

Data Area speichern und katalogisieren

Bevor eine Data Area in einem Natural-Programm (oder einem anderen Objekt) benutzt werden kann, muss sie als Source-Objekt und/oder als katalogisiertes Objekt in einer Natural Library in der aktuellen Systemdatei gespeichert und katalogisiert werden.

Um die aktuelle Data Area zu speichern und zu katalogisieren:

- Benutzen Sie das Systemkommando SAVE, CATALOG oder STOW. Siehe *Objekte speichern und katalogisieren* in *Natural benutzen*.

Anmerkung: Wenn Sie den Data Area Editor mit dem Editor-Kommando EXIT verlassen, wird der aktuelle Quellcode automatisch gespeichert, wenn die entsprechende Option im **Editor-Profil** entsprechend gesetzt ist. Weitere Informationen siehe ***Exit-Funktion***.

Um eine Kopie des aktuellen Quellcodes zu behalten:

- Benutzen Sie die Editor-Optionen **Source Save into** und **Auto Save Numbers**. Weitere Informationen siehe ***Editor-Profil***.

  Eine Kopie des zuletzt mit einem der Natural-Editoren bearbeiteten Quellcodes wird dann automatisch als Source-Objekt in der aktuellen Natural-Umgebung gespeichert.

Quellcode-Format zum Speichern der Data Area

Beim Data Area Editor wird ein internes Quellcode-Format zum Speichern der Data Areas in der Systemdatei FUSER verwendet. Ab Natural Version 4.1 stehen neue Merkmale und Definitionen zur Verfügung. Deshalb muss die Data Area jetzt unter Verwendung eines erweiterten Quellcode-Formats in der Systemdatei FUSER gespeichert werden.

Der Platzbedarf zum Speichern des erweiterten Quellcode-Formats unter Natural Version 4.1 (und höher) ist größer als beim alten Quellcode-Format.

Data Areas, die im erweiterten Quellcode-Format gespeichert werden, können nicht mit Natural Version 3.1, bei der ein anderes Quellcode-Format verwendet wurde, benutzt oder bearbeitet werden. Der Data Area Editor in Natural Version 4.1 (und höher) unterstützt sowohl das Format der Version 3.1 als auch das erweitere Quellcode-Format. Der Editor kann beide Formate lesen. Er konvertiert das Natural Version 3.1-Format in das erweitere Quellcode-Format.

Standardmäßig werden Data Areas Data im Natural Version 4.1-Quellcode-Format gespeichert.

Solange keine Merkmale und Definitionen der Natural Version 4.1 (und höher) verwendet werden, können Data Areas auch im Format der Natural Version 3.1 gespeichert werden.

Das Quellcode-Format, das standardmäßig zum Speichern von Data Area benutzt werden soll, kann mit der User Exit Routine GDA-EX01 (siehe ***User Exit für den Data Area Editor***) oder, während einer Editor-Sitzung, mit den Editor-Kommandos `SET SAVEFORMAT V31` oder `SET SAVEFORMAT V41` angegeben werden.

User Exit für den Data Area Editor

Für den Data Area Editor steht eine User Exit Routine zur Angabe von Standard-Einstellungen vorhanden. Der Quellcode der User Exit Routine ist in der Library SYSEXT unter dem Namen GDA-ES01 zur Verfügung. Um den Exit zu aktivieren, müssen Sie das Source-Objekt als GDA-EX01 mit katalogisieren (`CATALOG`) oder speichern und katalogisieren (`STOW`) und dann GDA-EX01 in die Library SYSLIB kopieren. Eine ausführliche Beschreibung finden Sie beim Source-Objekt GDA-ES01 in der Library SYSEXT.

Exit-Funktion

Mit der Exit-Funktion können Sie die aktuelle Editor-Sitzung beenden, indem Sie das Kommando `EXIT` absetzen.

Die Wirkung des `EXIT`-Kommandos ist abhängig von der Einstellung der **Editor-Profil**-Option **Prompt Window for Exit Function:**.

- Ist die Option auf `Y` gesetzt (Standardeinstellung), dann ruft das `EXIT`-Kommando das **EXIT Function**-Fenster immer dann auf, wenn Sie das Kommando bei einer Source eingeben, die nicht gespeicherte Änderungen enthält (siehe auch **Änderungsanzeige**). Wenn an der Source keine Änderungen vorgenommen wurden, erscheint das Fenster *nicht* und die Editor-Sitzung wird ohne Speicherung des Quellcodes beendet.

  Das **EXIT Function**-Fenster enthält folgende Optionen:

Option	Explanation
Save and Exit	Speichert alle am aktuellen Quellcode vorgenommenen Änderungen. Die Editor-Sitzung wird beendet. Siehe auch *SAVE Object Window*.
Exit without Saving	Die Editor-Sitzung wird beendet. Die seit dem letzten Speichern des Quellcode vorgenommenen Änderungen werden *nicht* gespeichert.
Resume Function	Die Editor-Sitzung wird *nicht* beendet. Es werden keine Änderungen gespeichert. Das Eingabeaufforderungsfenster wird geschlossen und die aktuelle Funktion wird fortgesetzt.

- Ist die Option auf `N` gesetzt, führt das `EXIT`-Kommando die **Save and Exit**-Funktion aus: Sie speichert alle an der aktuellen Source vorgenommenen Änderungen. Die Editor-Sitzung wird beendet. Siehe auch *SAVE Object-Fenster*.

SAVE Object-Fenster

Die **Save and Exit**-Funktion ruft das **SAVE Object**-Fenster auf, wenn für den aktuellen Quellcode noch kein Objektname definiert wurde (wie in der Editor-Kommandozeile angezeigt).

Sie werden aufgefordert, einen Objektnamen und einen Objekttyp anzugeben. Wenn Sie Ihre Eingaben mit Enter bestätigen, wird der Quellcode unter dem angegebenen Namen als neues Objekt in der aktuellen Library gespeichert und die Sitzung wird beendet. Wenn Sie PF3 drücken, wird das Fenster ohne Maßnahme geschlossen und die Editor-Sitzung kann fortgesetzt werden.

VI Masken-Editor (Map Editor)

Der Natural-Masken-Editor (englisch: „Map Editor“) wird verwendet, um ein Natural-Objekt des Typs „Map“ anzulegen, das die Definition einer Natural-„Map“ enthält (in der Natural-Anwendungsentwicklung verwendete Bezeichnung: „Maske“ oder „Bildschirmmaske“). Ein solches Map-Objekt kann in einem anderen Natural-Objekt (zum Beispiel einem Natural-Programm) referenziert werden. Dazu wird entweder ein `INPUT USING MAP`-Statement (bei Eingabe-Masken) oder ein `WRITE USING MAP` (bei Ausgabe-Masken) benutzt.

Ein Map-Objekt enthält Textfelder und Datenfelder. Text-Felder sind Literal-Zeichenketten und Datenfelder sind Variablen. Datenfelder können entweder Benutzervariablen oder Natural-Systemvariablen sein.

Nachdem ein Map-Objekt erstellt worden ist, kann es als Source-Objekt und als katalogisiertes Objekt in einer Library in einer Natural-Systemdatei gespeichert werden.

Anmerkungen:

1. Wenn Sie Natural Studio in einer Windows-Umgebung benutzen, unterstützt der Natural-Masken-Editor Felder mit Unicode-Format und Unicode-Zeichenketten. Wenn Sie jedoch den Quellcode einer Unicode Map in den Editierbereich eines Natural-Masken-Editors in einer lokalen Großrechner- oder UNIX-Umgebung einlesen, werden alle Unicode-Zeichenketten aus dem Quellcode entfernt.
2. Der Masken-Editor (Map Editor) ist standardmäßig in Ihrer Umgebung abgeschaltet (deaktiviert). Weitere Information siehe *NaturalONE als Standard-Entwicklungsumgebung*.

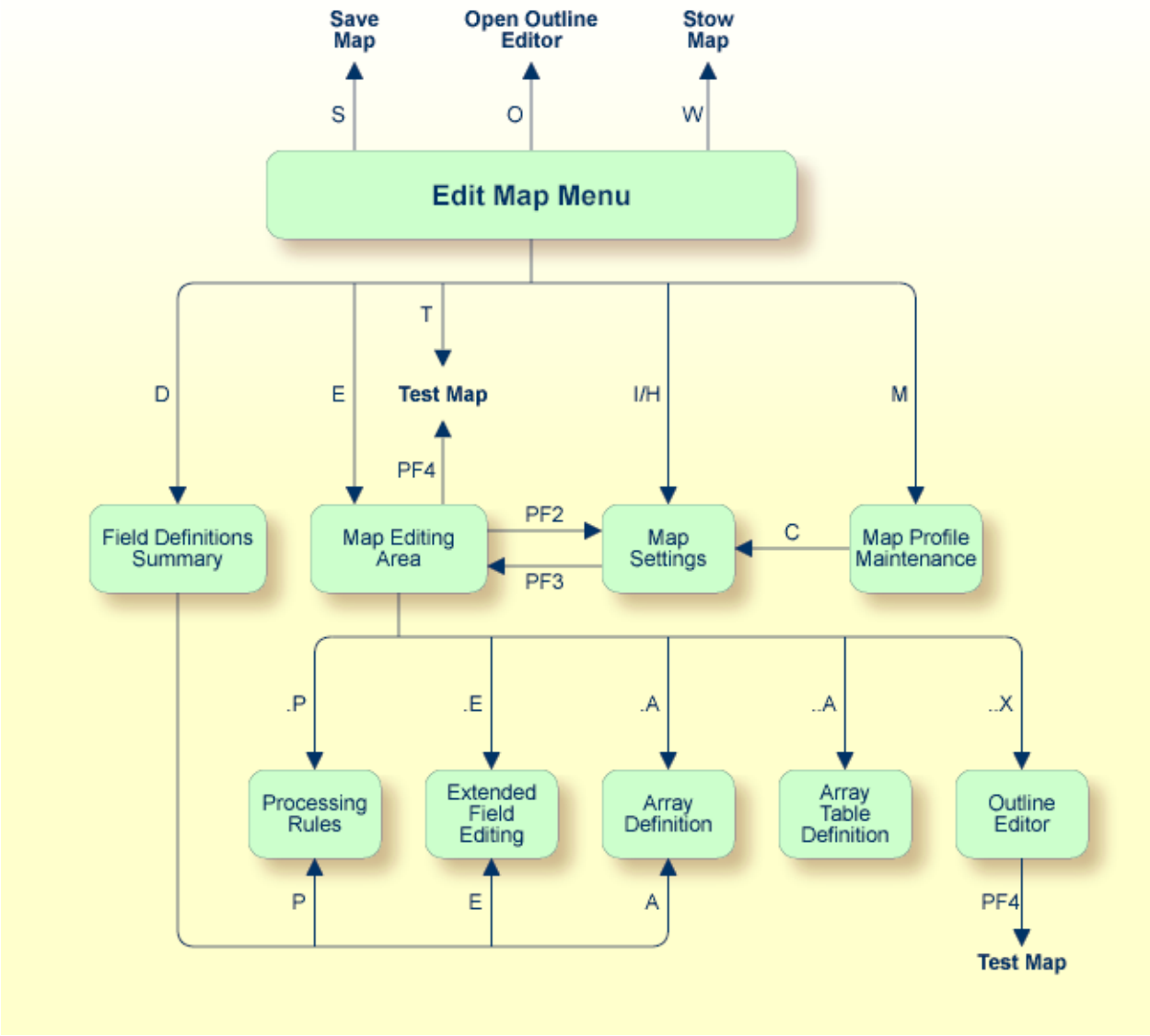
Verwandte Themen:

- *Map Editor Tutorial*
- *Editors in the SPoD Environment* in der *Unicode and Code Page Support*-Dokumentation

Komponenten des Masken-Editors
Die Vier Schritte zur Masken-Erstellung
Masken-Editor aufrufen und beenden
Funktionen im Menü 'Edit Map'
Maske erstellen — Funktion 'Initialize New Map'
Maske bearbeiten — Funktion 'Edit Map'
Maskenfelder definieren
Erweiterte Feld-Bearbeitung
Feldumrahmungen erstellen — mit dem Outline-Editor
Nachträgliches Zuweisen von Feldern
Array- und Tabellen-Definitionen
Verarbeitungsregeln

12 Komponenten des Masken-Editors

Die folgende Abbildung vermittelt einen Überblick über die verschiedenen Komponenten des Masken-Editors und zeigt, wie Sie aus einer Komponente in eine andere gelangen können. Die Buchstaben neben den Verbindungslinien sind die Funktionscodes bzw. Kommandos, die Sie zum Aufrufen eines Bildschirms oder zum Ausführen eines Kommandos benutzen können.



Anmerkungen:

1. Den Outline Editor können Sie nur dann benutzen, wenn Ihr Terminal bzw. Ihre Terminal-Emulation das Anzeigen von Feldumrahmungen gestattet.
2. „Outlining“ oder „Boxing“ bezeichnet die Möglichkeit, bestimmte Felder auf dem Bildschirm mit Kästchen eingerahmt anzuzeigen. Diese Form der Anzeige ist eine zusätzliche Möglichkeit, dem Benutzer Länge und Position von Feldern auf dem Schirm deutlich zu machen. Die Outline-Funktion steht nur bei bestimmten Terminal-Typen zur Verfügung

13 In vier Schritten zur fertigen Maske

- Schritt 1 - Masken-Profil definieren 198
- Schritt 2 - Maske definieren 198
- Schritt 3 - Maskenfelder definieren 198
- Schritt 4 - Masken-Definition speichern 199

Schritt 1 - Masken-Profil definieren

Um das Profil einer Maske („Map“) zu definieren, müssen Sie die erforderlichen Einstellungen aus einem Menü auswählen.

Das Masken-Profil umfasst die Feld-Trennzeichen, Format-Einstellungen, Kontext-Einstellungen und Füllzeichen, die benutzt werden sollen.

Schritt 2 - Maske definieren

Sie können die Maske („Map“) definieren, bevor oder nachdem Sie die entsprechenden Datendefinitionen in dem Objekt (evtl. mehrere) vorgenommen haben, das die Map referenziert (z.B. Programm oder Data Area). Die beiden Methoden zum Definieren einer Maske werden weiter nachfolgend beschrieben:

- Definieren Sie zunächst eine Prototyp-Maske. Nehmen Sie dann die entsprechenden Datendefinitionen in dem Objekt vor, das die Maske referenziert, anschließend integrieren Sie die Maske in die Anwendung.

  Felder können Sie direkt im Masken-Editierbereich definieren. Jedem Feld wird ein Standard-Name zugewiesen. Anschließend, nachdem Sie die entsprechenden Datendefinitionen in dem betreffenden Objekt vorgenommen haben, können Sie diese Datendefinitionen den Maskenfeldern zuweisen (nachträgliches Zuweisen).
- Definieren Sie eine Maske, indem Sie bereits existierende Datendefinitionen benutzen.

  Falls die Datendefinitionen in einem Objekt, das die Maske referenziert, schon existieren, können Sie die Maskenfelder erstellen, indem Sie die in diesem Objekt enthaltenen Datendefinitionen benutzen. In diesem Fall werden alle Eigenschaften der Datendefinitionen in die Map kopiert.

Schritt 3 - Maskenfelder definieren

Maskenfelder können Sie erstellen, indem Sie entweder die Felddefinitionen direkt in den Masken-Editierbereich eingeben oder indem Sie Datendefinitionen aus anderen Natural-Objekten auswählen (siehe *Datendefinitionen auswählen*).

Der Masken-Editor bietet folgende Funktionalität:

- Bearbeitung im Vollbild- oder Split-Screen-Editier-Modus.

  Der Vollbild-Editier-Modus zeigt den Masken-Editierbereich an, in dem Sie die Maske tatsächlich entwickeln und die Felddefinitionen eingeben.

Im Split-Screen-Editier-Modus (Standardeinstellung) können Sie die obere Hälfte des Editor-Bildschirms benutzen, um Datendefinitionen aus anderen Natural-Objekten anzuzeigen. In der unteren Bildschirmhälfte wird der Masken-Editierbereich angezeigt.

- Editor-Kommandos zum Positionieren.
- **Zeilenkommandos**, die Sie benutzen können, um Tabellen zu definieren und Zeilen zu beeinflussen.
- **Feld-Kommandos**, die Sie benutzen können, um Arrays zu definieren und Felder zu beeinflussen.
- Editor-Funktionalität, mit der Sie Verarbeitungsregeln (Validierungsregeln) bearbeiten können.
- Einen Outline Editor, den Sie benutzen können, um **Feldumrahmungen** (Box-Definitionen) für Maskenfelder zu erstellen, wenn Ihr Terminal bzw. Ihre Terminal-Emulation das Anzeigen von Feldumrahmungen gestattet.

Schritt 4 - Masken-Definition speichern

Nachdem Sie die Maske so wie in den vorangegangenen Schritten beschrieben definiert haben, kann die Definition (Natural-Objekttyp Map) als Source-Objekt und/oder katalogisiertes Objekt in der aktuellen Library in der aktuellen Natural-Systemdatei gespeichert werden. Nachdem die Definition als Source-Objekt gespeichert worden ist, können Sie die Definition in einer späteren Masken-Editor-Sitzung einlesen und ändern. Nachdem die Definition als katalogisiertes Objekt gespeichert worden ist, kann die Maske aus einem Natural-Programm aufgerufen werden.

Anmerkung: Beim Masken-Editor wird die **Auto Save Numbers**-Funktion benutzt, die beim Programm-Editor und beim Datenbereich-Editor (Data Area Editor) Verwendung findet. Siehe *Editoren - Allgemeine Informationen*.

14 Masken-Editor aufrufen und beenden

- Masken-Editor aufrufen und neue Maske erstellen 202
- Masken-Editor aufrufen und vorhandene Maske bearbeiten 203
- Masken-Editor beenden 204

Masken-Editor aufrufen und neue Maske erstellen

Um den Masken-Editor aufzurufen und eine neue Maske zu erstellen:

- Geben Sie folgendes Kommando ein:

```
EDIT MAP
```

Oder:

```
E M
```

(abgekürztes Kommando)

Falls sich im Quellcode-Bereich bereits ein Map-Objekt befindet, wird die Masken-Definition dieser Maske im Editierbereich angezeigt (siehe auch *Maske bearbeiten — Funktion "Edit Map"*). Sie können den Inhalt des Quellcode-Bereichs löschen, indem Sie das Systemkommando `CLEAR` absetzen, bevor Sie den Masken-Editor aufrufen.

Wenn der Quellcode-Bereich leer ist, erscheint folgendes Menü (Beispiel):

```
10:10:22               ***** NATURAL MAP EDITOR *****              2014-06-12
User SAG                        - Edit Map -                  Library SYSTEM

                  Code    Function
                  ----    ---------------------------------
                    D     Field and Variable Definitions
                    E     Edit Map
                    O     Outline Editor
                    I     Initialize new Map
                    H     Initialize a new Help Map
                    M     Maintenance of Profiles & Devices
                    S     Save Map
                    T     Test Map
                    W     Stow Map
                    ?     Help
                    .     Exit

             Code .. I     Name .. ________       Profile .. SYSPROF_

Command ===>



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

Anmerkung: Die Menü-Funktion **Outline Editor** erscheint nur, wenn Ihr Terminal oder Ihre Terminal-Emulation das Anzeigen von Feldumrahmungen gestattet.

Das Menü **Edit Map** ist das Hauptmenü des Masken-Editors. Beschreibung der Funktionen siehe *Funktionen im Edit Map-Menü*. Die im oberen und unteren Bereich des Bildschirms enthaltenen Felder sind in der folgenden Tabelle beschrieben:

Feld	Erläuterung
User	Die Natural-Benutzerkennung des aktuellen Benutzers.
Library	Der Name der zurzeit gültigen Natural-Library.
Code	Der Code, mit dem die auszuführende Funktion aufgerufen werden kann (siehe *Funktionen im Edit Map-Menü*).
Name	Der Name des Source-Objekts, das die Map oder Help Map enthält. Bei mehrsprachig verfügbaren Maps sollte eine Stelle des Objektnamens für den Sprachschlüssel reserviert werden. Zum Beispiel: `USERMAP1` (der Sprachcode für Englisch ist `1`, für Deutsch `2`) Aufgerufen wird die Map aus dem Programm mittels folgendem Statement: `INPUT USING MAP 'USERMAP&'` Dabei wird `&` zur Ausführungszeit durch den Inhalt der Systemvariablen `*LANGUAGE` ersetzt. **Anmerkung:** Eine Liste der in Natural verwendeten Sprachcodes finden Sie bei der Beschreibung der Systemvariablen `*LANGUAGE` in der *Systemvariablen*-Dokumentation.
Profile	Das zurzeit gültige Sitzungsprofil. Als Name des Profils wird der Name der aktuellen Library gesetzt. Falls diese Library nicht verfügbar ist, wird die aktuelle Benutzerkennung eingesetzt. Falls diese Profilkennung nicht vorhanden ist, wird der Name auf `SYSPROF` gesetzt.

Masken-Editor aufrufen und vorhandene Maske bearbeiten

Um den Masken-Editor aufzurufen und eine vorhandene Maske zu bearbeiten:

- Setzen Sie folgendes Systemkommando ab:

```
EDIT map-name
```

oder benutzen Sie die Abkürzung:

```
E map-name
```

Dabei ist *map-name* der Name der zu bearbeitenden Maske („Map“).

Wenn die Maske mit dem angegebenen Namen als Source-Objekt in der aktuellen Library existiert und nicht durch einen anderen Benutzer gesperrt worden ist, wird die Masken-Definition dieser Maske im Editierbereich des Masken-Editor angezeigt (siehe auch *Maske bearbeiten — Funktion "Edit Map"*).

Durch das Sperren einer Masken-Definition werden konkurrierende Änderungen vermieden. Das Sperren ist abhängig von der aktuellen Einstellung des Profilparameters `SLOCK`. Weitere Informationen siehe *SLOCK - Source-Sperrung* in der *Parameter-Referenz*-Dokumentation und *Sperrmechanismen für Source-Objekte* in der *Editoren*-Dokumentation.

Oder:

Wenn die Maske, die Sie bearbeiten möchten, noch infolge einer früheren Editor-Sitzung im Quellcode-Bereich enthalten ist, können Sie auch folgendes Kommando benutzen:

```
E M
```

Masken-Editor beenden

Um den Masken-Editor zu beenden:

- Im **Edit Map**-Menü: Drücken Sie `PF3` oder geben Sie im Feld **Code** einen Punkt (.) ein.

  Oder:

  In der Kommandozeile: Geben Sie einen Punkt (.) oder das Kommando `EXIT` ein.

Anmerkung: Beim Masken-Editor kommt die aktuelle Einstellung der Editor-Profil-Option **Leave Editor with Unlock** zur Anwendung. Mit dieser Option wird festgelegt, ob ein Quellcode entsperrt werden soll, wenn der Masken-Editor beendet wird. Weitere Informationen siehe *Allgemeine Standardwerte* in *Editor-Profil* im Abschnitt *Editoren - Allgemeine Informationen*.

15 Funktionen im Menü 'Edit Map'

- Feld- und Variablen-Definitionen bearbeiten — Funktion "Field and Variable Definitions" 206
- Maske editieren — Funktion "Edit Map" 207
- Feldumrahmungen editieren — Funktion "Outline Editor" 208
- Neue Maske erstellen — Funktion "Initialize New Map" 208
- Neue Hilfe-Maske erstellen — Funktion "Initialize a New Help Map" 208
- Profile und Geräte pflegen— Funktion "Maintenance of Profiles & Devices" 209
- Maske speichern — Funktion "Save Map" 209
- Maske testen — Funktion "Test Map" 209
- Maske speichern und katalogisieren — Funktion "Stow Map" 210
- Hilfe zum Masken-Editor aufrufen — Funktion "Help" 210

Feld- und Variablen-Definitionen bearbeiten — Funktion "Field and Variable Definitions"

Mit dieser Funktion können Sie alle Maskenfelder, Parameter und lokalen Variablen, die von der Maske („Map“) benutzt werden, anzeigen und bearbeiten.

Die Funktion **Field and Variable Definitions** ist äquivalent zum Zeilenkommando `..E`, das Sie im Editierbereich des Masken-Editors eingeben können (siehe auch ***Zeilenkommandos*** in *Maske bearbeiten — Funktion "Edit Map"*).

Nach Aufrufen der Funktion **Field and Variable Definitions** erscheint der Bildschirm **Field and Variable Definitions - Summary**. Dort werden zu jedem Maskenfeld folgende Informationen angezeigt:

Spalte	Erläuterung	
Name	Feldname.	
Mod	Feld-Modus (Feldtyp):	
	`D`	Datenbereichsfeld. Das Feld ist von einer Global Data Area (GDA), Local Data Area (LDA) oder Parameter Data Area (PDA).
	`S`	Systemvariable.
	`U`	Benutzerfeld.
	`V`	View-(DDM)-Feld.
	leer	Nicht definiertes Feld.
Format	Natural-Datenformat und -Länge des Feldes.	
Ar	Das Feld ist ein Array, wenn ein `A` eingegeben wird; andernfalls ist diese Spalte leer.	
Ru	Anzahl der beigefügten Verarbeitungsregeln.	
Lin	Zeilenposition.	
Col	Spaltenposition.	

Im Bildschirm **Field and Variable Definitions - Summary** stehen Ihnen folgende **Zeilenkommandos** und PF-Tasten zur Verfügung:

Spalte/PF-Taste	Erläuterung
A	Ein Array definieren.
D	Ein Feld löschen.
E	Bearbeitet ein Feld.
P*rr*	Eine Verarbeitungsregel bearbeiten.
- -	An den Anfang des Bildschirms springen.
.	Bildschirm verlassen.
PF9	Aufruf des Fensters **PARAMETER DEFINITIONS**. Dort können Sie Parameter, die nicht als Maskenfelder im Editierbereich des Masken-Editors erscheinen, aber dennoch mit der Maske („Map“) als Attribut-Kontrollvariablen, Startwerte oder Hilfe-Parameter verknüpft sind, anzeigen, hinzufügen oder ändern.
PF10	Aufruf des Fensters **LOCAL DATA DEFINITIONS**. Dort können Sie lokale Variablen, die zur Übergabe von Werten von einer Verarbeitungsregel an eine andere benutzt werden können, anzeigen, hinzufügen oder ändern.

Im Fenster **PARAMETER DEFINITIONS** bzw. **LOCAL DATA DEFINITIONS** stehen Ihnen folgende Kommandos zur Verfügung:

Kommando	Erläuterung
A	Ein Array definieren.
D	Eine Variable löschen. **Anmerkung:** Das Kommando D löscht einen Parameter nicht, wenn dieser Parameter noch bei einem Maskenfeld als Attribut-Kontrollvariablen, Startwerte oder Hilfe-Parameter angewendet wird.
- -	An den Anfang des Fensters springen.
.	Fenster verlassen.

Maske editieren — Funktion "Edit Map"

Ruft den Bildschirm auf, in dem Sie eine vorhandene Masken-Definition oder Hilfe-Masken-Definition bearbeiten können.

Bei einer Masken-Definition erfolgt die Editor-Sitzung im Split-Screen-Modus, bei einer Hilfe-Masken-Definition im Vollbild-Modus.

Feldumrahmungen editieren — Funktion "Outline Editor"

Anmerkung: Die Menü-Funktion **Outline Editor** wird nur dann im **Edit Map**-Menü angezeigt, wenn Ihr Terminal oder Ihre Terminal-Emulation das Anzeigen von Feldumrahmungen gestattet.

Diese Funktion ruft den Outline Editor auf, mit dem Sie Umrahmungen für Maskenfelder erstellen können.

Die mit dem Outline Editor erstellten Umrahmungen werden als `BX`-Parameter im Quellcode der Map abgelegt. Sie können mit der Funktion zur **erweiterten Feldbearbeitung** (siehe entsprechenden Abschnitt) bearbeitet werden.

Siehe auch *Feldumrahmungen erstellen — mit dem Outline-Editor*.

Neue Maske erstellen — Funktion "Initialize New Map"

Diese Funktion kann nur dann ausgeführt werden, wenn in der Natural-Systemdatei kein Objekt mit demselben Namen gespeichert ist.

Siehe auch *Maske erstellen — Funktion "Initialize New Map"*

Neue Hilfe-Maske erstellen — Funktion "Initialize a New Help Map"

Diese Funktion sollten Sie zum Erstellen einer Hilfe-Maske benutzen, denn sie bietet Ihnen maximale Flexibilität bei der Eingabe und Bearbeitung von Text (führende Leerzeichen müssen eingegeben werden). Außerdem bietet sie Prüffunktionen, die sicherstellen, dass eine gültige Hilfe-Masken erstellt wird.

Die Funktion kann nur dann ausgeführt werden, wenn in der Natural-Systemdatei kein Source-Objekt und kein katalogisiertes Objekt mit demselben Namen vorhanden ist.

Eine Hilfe-Maske wird als „Help Map“ gespeichert und kann mit dem Parameter `HE` in der Masken-Definition referenziert werden. (Hilfe-Masken sind im Prinzip „Maps“ wie andere Masken-Typen, aber wenn sie als Hilfe zugewiesen werden, werden zusätzliche Prüfungen vorgenommen, um ihre Verwendbarkeit für Hilfe-Zwecke zu gewährleisten.)

Beim Erstellen oder Bearbeiten einer Hilfe-Maske können Sie in den Map-Einstellungen angeben, an welcher Stelle des Bildschirms die Hilfe-Maske zur Ausführungszeit erscheinen soll.

Siehe auch *Maske erstellen — Funktion "Initialize New Map"*.

Profile und Geräte pflegen— Funktion "Maintenance of Profiles & Devices"

In dieser Funktion können Sie folgende Profile anlegen, ändern oder löschen.

- **Sitzungsprofil**
  Dient dazu, Standard-Maskeneinstellungen zuzuweisen, die beim Erstellen einer Maske oder einer Hilfe-Maske verwendet werden sollen.
- **Maskenprofil**
  Dient dazu, die Maskeneinstellungen zu definieren, die beim Erstellen und beim Ausführen einer Masken-Definition gelten sollen.
- **Geräteprofil**
  Definiert die Standard-Eigenschaften und -Einstellungen für ein Gerät. Dieses Profil kann verwendet werden, um die Kompatibilität zwischen der Masken-Definition und dem zu verwendenden Gerät sicherzustellen.

Siehe auch ***Device Check*** (Geräteprüfung) in *Kontext-Einstellungen* im Abschnitt ***Maske erstellen — Funktion "Initialize New Map"***.

Maske speichern — Funktion "Save Map"

Die Funktion speichert die Masken-Definition als Source-Objekt in der aktuellen Natural-Library in der aktuellen Systemdatei.

Maske testen — Funktion "Test Map"

Die Funktion teste die aktuelle Masken-Definition um sicherzustellen, dass sie erfolgreich ausgeführt werden kann. Dazu gehört auch die Prüfung aller Verarbeitungsregeln (einschließlich Predict-Regeln) und der zugehörigen Hilfe-Funktionen.

Beim Testen der Maske werden alle zusätzlich erstellten Maskenparameter mit dem Ausgangswert `1` vorbelegt.

Maske speichern und katalogisieren — Funktion "Stow Map"

Die Funktion speichert die Masken-Definition als Source-Objekt und als katalogisiertes Objekt in der aktuellen Natural-Library in der aktuellen Systemdatei.

Hilfe zum Masken-Editor aufrufen — Funktion "Help"

Die Hilfe-Funktion des Masken-Editors bietet themenspezifische und feldspezifische Informationen zu den Funktionen, die im Masken-Editor zur Verfügung stehen.

Um die themenspezifische Hilfe des Masken-Editors aufzurufen:

1 Im Menü **Edit Map**: Drücken Sie PF1.

Oder:

In der Kommandozeile des Menüs **Edit Map**: Geben Sie ein Fragezeichen (?) ein.

Das Hauptmenü **Help Main Menu** erscheint.

2 Im Feld **Select chapter**: Geben Sie die Ziffer oder den Buchstaben ein, der dem benötigten Hilfethema entspricht, und drücken Sie Enter.

Es erscheint ein Hilfe-Bildschirm zum gewählten Thema oder es erscheint ein Untermenü mit weiteren Hilfe-Themen, das Ihnen hilft, die Suche weiter einzugrenzen.

Zusätzlich zur Themen-Hilfe bietet der Masken-Editor Ihnen spezifische Informationen zu jedem Eingabefeld auf den einzelnen Bildschirmen des Masken-Editors.

Um die Feld-Hilfe aufzurufen:

- Platzieren Sie den Cursor in das Feld, zu dem Sie Hilfe benötigen, und drücken Sie PF1.

  Oder:

  Geben Sie in dem Feld, zu dem Sie Hilfe benötigen, ein Fragezeichen (?) ein und drücken Sie Enter.

  Es erscheint ein Bildschirm mit entsprechenden Hilfe-Informationen oder ein Auswahlfenster, aus dem Sie einen gültigen Eingabewert auswählen können.

16 Maske erstellen — Funktion 'Initialize New Map'

- Trennzeichen — "Delimiters" 213
- Masken-Format-Einstellungen — "Format" 215
- Kontext-Einstellungen — "Context" 219
- Füllzeichen — "Filler Characters" 221

Dieser Abschnitt beschreibt, wie Sie die Einstellungen (Profil) für eine neue Maske oder Hilfe-Maske definieren.

Wenn Sie die Funktion **Initialize New Map** oder **Initialize a New Help Map** aufrufen, erscheint als erster Bildschirm der **Define Map Settings**-Bildschirm, der in mehrere Bereiche unterteilt ist (Beispiel):

```
10:41:16              Define Map Settings for MAP                  2007-08-13

 Delimiters           Format                          Context
 -----------------    ----------------------------    -------------------------
 Cls Att CD  Del      Page Size ...... 23             Device Check .... ________
  T   D      BLANK    Line Size ...... 79             WRITE Statement   _
  T   I      ?        Column Shift ... 0 (0/1)        INPUT Statement   X
  A   D      _        Layout ......... ________       Help  ____________________
  A   I      )         dynamic ....... N (Y/N)         as field default N (Y/N)
  A   N      ¬        Zero Print ..... N (Y/N)
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)        Automatic Rule Rank  1
  O   D      +        Decimal Char ... .              Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
  D   D      $        Justification .. L (L/R)        Filler Characters
  D   I      /        Print Mode ..... __             ------------------------
                                                      Optional, Partial ....
                      Control Var .... ________       Required, Partial ....
                                                      Optional, Complete ...
                                                      Required, Complete ...

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

Anmerkung: Die Trennzeichen-(Delimiter-)Klasse `D` wird nur dann angezeigt, wenn Ihr Terminal bzw. Ihrer Terminal-Emulation die Darstellung von Feldumrahmungen (Box) unterstützt.

Die einzelnen Bereiche des **Define Map Settings**-Bildschirms werden in den folgenden Abschnitten beschrieben.

Trennzeichen — "Delimiters"

Unter der Überschrift **Delimiters** im Bildschirm **Define Map Settings** werden die Standard-Trennzeichen angezeigt, die für die aktuelle Maske gelten.

```
Delimiters
-----------------
Cls Att CD  Del
 T   D      BLANK
 T   I      ?
 A   D      _
 A   I      )
 A   N      ¬
 M   D      &
 M   I      :
 O   D      +
 O   I      (
 D   D      $
 D   I      /
```

Ein Trennzeichen wird hier verwendet, um einem Feld bestimmte Eigenschaften zuzuweisen: Klasse (**Cls**, z.B. Ein-/Ausgabefeld), Attribut (**Att**, z.B. Schriftart) und Farbdefinitionen (**CD**).

Anmerkung: Die Verwendung von Feldattributen, Feldfarben und Feld-Druck- oder Anzeige-Modi setzt entsprechende Hardware-Merkmale voraus. Sie werden zur Laufzeit ignoriert, wenn diese Merkmale nicht vorhanden sind. Siehe auch die entsprechenden Session-Parameter `AD`, `CD` und `PM` in der *Parameter-Referenz*-Dokumentation.

Jede Einstellung für eine Feldklasse, ein Attribut oder eine Farbe besteht aus einem ein- oder zweistelligen Buchstabencode. So kennzeichnet zum Beispiel der Buchstabe `A` ein Eingabefeld, der Buchstabe `I` bedeutet „intensiviert“ (hell leuchtend). Klasse, Attribut und Farbe können zu einem Trennzeichen (Delimiter) kombiniert werden. Ein Trennzeichen gibt zum Beispiel an, dass ein Feld ein Eingabefeld (Buchstabencode `A`) ist *und* „intensiviert“ (Buchstabencode `I`) dargestellt wird. Im oben abgebildeten Beispiel-Bildschirm ist das Trennzeichen für diese Kombination (Buchstabencodes `A` und `I`) die rechte, runde Klammer `)`.

Ein Trennzeichen ist ein nicht-alphabetisches Zeichen, das dem Feld im Masken-Editierbereich vorangestellt wird. Beispiele für die Verwendung von Trennzeichen siehe ***Maskenfelder definieren***. Um die Klassen-, das Attribut- und Farbeinstellungen, die einem Feld über ein Trennzeichen zugewiesen wurden, anzuzeigen oder zu ändern, müssen Sie die Funktion zur **erweiterten Feldbearbeitung** benutzen (siehe entsprechenden Abschnitt). Als Trennzeichen kann jedes beliebige nicht-alphabetisches Zeichen verwendet werden. Ausnahmen: das Steuerzeichen für Terminal-Kommandos, das Steuerzeichen für Masken-Kommandos und das Dezimalstellenzeichen.

Buchstabencodes und Trennzeichen können Sie in den S**Cls** (Klasse), **Att** (Attribut), **CD** (Farbdefinition) und **Del** (Delimiter) oder im Editierbereich palten Funktion zur **erweiterten Feldbearbeitung** eingegeben werden.

Um die Standard-Trennzeichen-Einstellungen für die aktuelle Maske zu ändern:

- Überschreiben Sie im Bereich **Delimiters** in den Spalten **Cls**, **Att**, **CD** und/oder **Del** den/die zu ändernde(n) Werte mit dem benötigten Wert bzw. geben Sie in einer leeren Spalte einen Wert ein.

Um die Standard-Trennzeichen-Einstellungen für die aktuelle Editor-Sitzung zu ändern:

- Bevor Sie mit dem Anlegen einer Maske beginnen, ersetzen Sie im Bildschirm **Edit Map** im Feld **Profiles** das Standard-Masken-Profil `SYSPROF` durch den Namen des Profils, das Sie zuvor mit der Funktion **Maintenance of Profiles & Devices** (siehe *Profile und Geräte pflegen — Funktion "Maintenance of Profiles & Devices"*) erstellt haben.

  Anmerkung: Das Profil `SYSPROF` kann nur der Natural-System-Administrator ändern.

Gültige Buchstabencodes für Klassen, Attribute und Farben:

Die folgende Tabelle erläutert die gültigen Buchstabencodes, die in den Spalten **Cls** (Class), **Att** (Attribute) und **CD** oder in den Feldern der Funktion zur **erweiterten Feldbearbeitung** eingegeben werden können.

Trennzeichen-Klasse (Cls)		Attribut-Definition (Att)		Farbdefinition (CD)	
A	Eingabefeld	**B**	blinkend	**BL**	blau
M	Änderbares Feld	**C**	kursiv	**GR**	grün
O	Ausgabefeld, geschützt	**D**	Standardwert (z.B. nicht intensiviert, blinkend)	**NE**	neutral
T	Text-Feld	**I**	intensiviert	**PI**	rosa
D	Erweitertes Text-Feld Siehe auch *Trennzeichen-Klasse D*.	**N**	nicht sichtbar	**RE**	rot
		U	unterstrichen	**TU**	türkis
		V	invers	**YE**	gelb
		Y	dynamisch (Attribute sollen durch ein Programm dynamisch zugewiesen werden)		

Trennzeichen-Klasse D

Anmerkung: Die Trennzeichen-Klasse `D` gilt nur, wenn Ihr Terminal bzw. Ihre Terminal-Emulation das Anzeigen von Feldumrahmungen gestattet.

Die Trennzeichen-Klasse `D` muss in den Masken-Einstellungen definiert werden, um mit dem Outline Editor Feldumrahmungen für Maskenfelder zu erstellen. Die Trennzeichen-Klasse `D` gestattet die Definition von Umrahmungen bei Text-Feldern und leeren Text-Feldern in einer Maske.

Bei einer bereits existierenden, mit Natural erstellten Maske können Sie die Trennzeichen-Klasse `D` wie im Abschnitt *Trennzeichen-Klasse D definieren* beschrieben hinzufügen.

Erweiterte Text-Felder

Ein Text-Feld, für das Sie eine Umrahmung erstellen möchten, muss mit dem Trennzeichen beginnen, das Sie der Trennzeichen-Klasse D zugewiesen haben. Ein Feld, dem dieses Trennzeichen vorangestellt ist, wird dann als erweitertes Text-Feld behandelt, das mit der Funktion zur **erweiterten Feldbearbeitung** des Masken-Editors bearbeitet werden kann.

Ein erweitertes Text-Feld, das als natives DBCS-Feld (DBCS = Double Byte Character Set) definiert worden ist, wird durch eine Reihe von Ks im Masken-Editierbereich repräsentiert. Sie auch Beschreibung des **PM**-Feldes im Abschnitt *Felder im erweiterten Text-Feld-Bearbeitungsbereich*.

Leere Text-Felder

Leerzeichen, bei denen Sie eine Umrahmung (Box) definieren, werden als leeres Text-Feld behandelt. Wenn Sie die Funktionen des Outline Editor benutzen, wird ein leeres Text-Feld automatisch zu der Maske mit dem Trennzeichen, das die Klasse `D` repräsentiert, hinzugefügt. Im Masken-Editierbereich folgt, abhängig von der Länge des Feldes, auf das Trennzeichen ein oder mehrere Punkte (.), wobei jeder Punkt für ein Leerzeichen steht. Ein leeres Feld mit einer Länge größer als Null (0) wird als erweitertes Text-Feld betrachtet, das mit der Funktion zur **erweiterten Feldbearbeitung** bearbeitet (oder erstellt) werden kann. Siehe entsprechenden Abschnitt.

Masken-Format-Einstellungen — "Format"

Unter der Überschrift **Format** im Bildschirm **Define Map Settings** werden die Masken-Format-Einstellungen angezeigt, die für die aktuelle Maske gelten.

```
Format
--------------------------
Page Size ...... 23
Line Size ...... 79
Column Shift ... 0 (0/1)
Layout ......... ________
 dynamic ....... N (Y/N)
Zero Print ..... N (Y/N)
Case Default ... UC (UC/LC)
Manual Skip .... N (Y/N)
Decimal Char ... .
Standard Keys .. N (Y/N)
Justification .. L (L/R)
Print Mode ..... __

Control Var .... ________
                       ↩
```

Folgende Maskenformat-Einstellungen stehen zur Verfügung:

Feld	Erläuterung
Page Size	Die Anzahl der zu bearbeitenden Maskenzeilen (1 - 250); wenn **Standard Keys** (siehe weiter unten) auf `Y` gesetzt ist, dann ist die Anzahl der Zeilen auf 3 - 250 eingeschränkt. Bei einer Maske, die mit einem `WRITE`-Statement ausgegeben wird, geben Sie die Anzahl der Zeilen für die logische Seitenausgabe beim `WRITE`-Statement an, d.h., nicht die Maskengröße. Auf diese Weise kann die Maske mehrmals auf einer Seite ausgegeben werden.
Line Size	Die Anzahl der zu bearbeitenden Maskenspalten (5 - 249).
Column Shift	Spaltenverschiebung (0 oder 1), die bei der Maske angewendet werden soll. Dieses Merkmal kann benutzt werden, um auf einem Bildschirm mit 80 Spalten alle 80 Spalten anzusprechen (**Column Shift** = 1, **Line Size** = 80). Um alle Maskenpositionen zu bearbeiten, müssen Sie Kommandos zum Positionieren (PF10, PF11) benutzen. Das längste Feld, das Sie auf einem Bildschirm der Größe 24 x 80 anzeigen können, darf nicht länger als 79 Zeichen sein. Wenn Sie ein Feld mit 80 Zeichen oder mehr anzeigen möchten, müssen Sie die Zeilenlänge (**Line Size**) entsprechend anpassen. Beispiel: Um ein Feld mit Format/Länge A80 anzuzeigen, müssen Sie **Line Size** = 81 setzen.
Layout	Der Name einer Masken-Quellcode-Definition, die ein vordefiniertes Layout enthält.
dynamic	`Y` Gibt an, dass das Layout dynamisch sein soll. Das dynamisch verwendete Layout wird nicht bei der Kompilierung zu einem festen Bestandteil der Maske, sondern wird erst zur Laufzeit ausgeführt. Auf diese Weise wird ermöglicht, dass nachträgliche Änderungen am Layout einer Maske bei allen Masken wirksam werden, bei denen die Layout-Maske benutzt wird. Falls die Layout-Maske Benutzervariablen enthält, müssen Sie diese Parameter in der Maske, bei der die Layout-Maske benutzt wird, definieren. Eingabefelder und änderbare

Feld	Erläuterung	
		Felder in der Layout-Maske sind zur Laufzeit nicht eingabefähig. Zum Hinzufügen von Parametern muss PF9 in der Funktion **Field and Variable Definitions** gedrückt werden.
	N	Gibt an, dass das Layout statisch sein soll. Das statische Layout wird in den Quellcode-Bereich kopiert, wenn eine Maske initialisiert wird. Füllzeichen werden nicht mit übertragen. N ist die Standard-Einstellung.
Zero Print	Y	Ein Feldwert, der nur aus Nullen besteht, wird nur als eine einzige Null angezeigt.
	N	Ein Nullwert wird als Leerzeichen angezeigt. N ist die Standard-Einstellung.
	Dieser Wert wird in die Feld-Definition kopiert, wenn ein neues Feld angelegt wird. Er kann bei einzelnen Feldern mit der Funktion zur **erweiterten Feldbearbeitung** bearbeitet werden.	
Case Default	UC	Gibt an, dass alle bei Feldern zur Ausführungszeit gemachten Eingaben in Großbuchstaben umgesetzt werden sollen, d.h. der Session-Parameter AD=T wird als Feld-Standardeinstellung verwendet. Siehe auch *AD - Attribut-Definition* in der *Parameter-Referenz*-Dokumentation.
	LC	Gibt an, dass keine Umsetzung von Klein- in Großbuchstaben erfolgen soll, d.h., der Session-Parameter AD=W wird als Feld-Standardeinstellung verwendet. Um den LC-Wert wirksam zu machen, müssen Sie den Natural-Profilparameter LC auf ON setzen. Siehe auch *AD - Attribut-Definition* und *LC - Umsetzung von Klein- in Großbuchstaben* in der *Parameter-Referenz*-Dokumentation.
	Wenn ein neues Feld angelegt wird, wird dieser Wert in die Feld-Definition kopiert. Er bei einzelnen Feldern mit der Funktion zur **erweiterten Feldbearbeitung** geändert werden.	
Manual Skip	Y	Der Cursor wird zur Ausführungszeit *nicht* automatisch in das nächste Feld in der Maske verschoben, und zwar selbst dann nicht, wenn das aktuelle Feld vollständig ausgefüllt ist.
	N	Der Cursor wird zur Ausführungszeit automatisch in das nächste Feld in der Maske verschoben, wenn das aktuelle Feld vollständig ausgefüllt ist. N ist die Standard-Einstellung.
Decimal Char	Das Zeichen, das als Dezimalstellenzeichen verwendet werden soll. Dieses Zeichen kann nur mit dem Kommando GLOBALS DC='c' geändert werden, wobei *c* das als	

Feld	Erläuterung
	Dezimalstellenzeichen zu benutzende Zeichen (Dezimalkomma bzw. Dezimalpunkt) darstellt. Siehe *DC - Dezimalstellenzeichen* in der *Parameter-Referenz*-Dokumentation.
Standard Keys	`Y` Die letzten beiden Zeilen der Maske werden leer gelassen, damit zur Ausführungszeit Funktionstastenangaben eingegeben werden können. `N` Bewirkt, dass alle Zeilen für die Maske genutzt werden können. `N` ist die Standard-Einstellung.
Justification	Die Art der Ausrichtung, die bei numerischen und alphanumerischen Feldern verwendet werden soll, wenn diese aus den Datendefinitionen in einem anderen Natural-Objekt übernommen werden sollen. `L` Linksbündig. `R` Rechtsbündig. Dieser Wert wird beim Anlegen eines neuen Feldes in die Felddefinition kopiert.
Print Mode	Der Standard-Druck-/Anzeige-Modus für Variablen: `C` Gibt an, dass ein alternativer Zeichensatz verwendet werden soll (spezielle, vom Natural-Administrator angelegte Zeichentabelle). `D` Gibt an, dass der DBCS-Modus (DBCS = Double Byte Character Set) verwendet werden soll. `I` Gibt an, dass invertierte Richtung verwendet werden soll. `N` Gibt an, dass keine Hardcopy gemacht werden kann. Dieser Wert wird beim Anlegen eines neuen Feldes in die Felddefinition kopiert.
Control Var	Der Name einer Attribut-Kontrollvariablen, deren Inhalt die Attribut-Eigenschaften von Feldern und Texten festlegt, die die Attribut-Definition `AD=Y` oder `Y` haben. Die in der Maske referenzierte Attribut-Kontrollvariable muss in dem Programm definiert werden, das diese Maske verwendet. Wenn eine Attribut-Kontrollvariable aus der Format-Masken-Einstellung entfernt wird, hat das zur Folge, dass die Attribut-Kontrollvariable auch aus der Maske entfernt wird, falls sie nicht mit einem anderen Maskenfeld verbunden ist.

Kontext-Einstellungen — "Context"

Unter der Überschrift **Context** sind im Bildschirm folgende Kontext-Einstellmöglichkeiten vorhanden:

```
Context
-------------------------
Device Check .... ________
WRITE Statement   _
INPUT Statement   X
Help  ____________________
 as field default N (Y/N)


Automatic Rule Rank  1
Profile Name .... SYSPROF
```

Folgende Kontext-Einstellmöglichkeiten stehen zur Verfügung:

Feld	Erläuterung	
Device Check	Wird in dieses Feld ein Gerätename eingegeben, dann werden die Maskeneinstellungen auf Kompatibilität mit dem Geräte-Profil des angegebenen Geräts geprüft. Falls eine Einstellung nicht kompatibel ist, wird eine Warnmeldung ausgegeben (siehe auch ***Profile und Geräte pflegen— Funktion "Maintenance of Profiles & Devices"*** im Abschnitt *Funktionen im Menü "Edit Map"*.	
WRITE Statement	Wird dieses Feld mit einem beliebigen Wert (kein Leerzeichen) markiert, dann wird am Ende des Masken-Definitionsvorgangs ein `WRITE`-Statement erzeugt. Die resultierende Maske kann dann mit einem `WRITE USING MAP`-Statement aus einem Natural-Programm heraus aufgerufen werden. Leerzeilen am Ende der Maske werden automatisch gelöscht, damit die Maske auf einer Seite mehrmals ausgegeben werden kann.	
INPUT Statement	Wird dieses Feld mit einem beliebigen Wert (kein Leerzeichen) markiert, dann wird am Ende des Masken-Definitionsvorgangs ein `INPUT`-Statement erzeugt. Die resultierende Maske kann dann mit einem `INPUT USING MAP`-Statement aus einem Natural-Programm heraus aufgerufen werden.	
Help	Der Name einer Helproutine oder Hilfe-Maske, die zur Ausführungszeit aufgerufen wird, wenn die Masken-Hilfe-Funktion (nicht die Feld-Hilfe) zu dieser Maske aufgerufen wird. Die gültige Syntax für die Eingabe von Werten in das **Help**-Feld entspricht der Syntax, die beim Session-Parameter `HE` gilt. Weitere Informationen siehe *HE-Parameter-Syntax* in der *Parameter-Referenz*-Dokumentation.	
as field default	`Y`	Gibt an, dass die im **Help**-Feld eingegebene Helproutine oder Hilfe-Maske als Standard-Einstellung bei jedem einzelnen Feld auf der Maske gelten soll, was bedeutet, das

Feld	Erläuterung	
		der Name eines jeden Feldes individuell an die Helproutine übergeben wird.
	N	Gibt an, dass der Name der Maske an die Helproutine oder Hilfe-Maske übergeben werden soll. Dies ist die Standard-Einstellung.
	Anmerkung: Wenn Sie im Bildschirm **Define Map Settings** die Masken-Einstellungen für eine Hilfe-Maske definieren, werden die Felder **Help** und **as field default** durch das Feld **Position Line Col** ersetzt.	
Position Line Col	Die Position, an der die Hilfe-Maske zur Ausführungszeit auf dem Bildschirm erscheinen soll. Dieses Feld erscheint nur dann, wenn Sie Masken-Einstellungen für eine Hilfe-Maske definieren, die mit der Funktion **Initialize a new Help Map** erstellt wurde. Dieses Feld ersetzt die Felder **Help** und **as field default** im Bildschirm **Define Map Settings for HELPMAP**.	
Automatic Rule Rank	Der Rang (Priorität), der automatischen Predict-Regel zugewiesen wird, wenn sie während der Maskenfeld-Definition mit der Maske verknüpft werden. Der Standardwert ist 1.	
Profile Name	Der Name des Profils, das bei der Initialisierung der Maske aktiv war. Falls ENFORCED angezeigt wird, dann sind die folgenden Masken-Einstellungen geschützt: ■ Alle Masken-Trennzeichen (Delimiters) ■ Statisches und dynamisches Layout ■ Geräte-Prüfung ■ WRITE und INPUT-Statements ■ Alle Füllzeichen ■ Automatische Regel-Rangordnung ■ Positionierung von Hilfe-Masken Der Name des Profils, das bei der Initialisierung der Maske aktiv ist, wird im Map-Objekt gespeichert. Wenn ein anderes Profil aktiv ist, wenn die Maske später bearbeitet wird, wird eine Warnmeldung ausgegeben, es wird jedoch eine Bearbeitung gestattet.	

Füllzeichen — "Filler Characters"

Füllzeichen können zugewiesen werden, um anzuzeigen, ob die Informationen bei einem Feld erforderlich sind (Pflichtfeld) und ob das Feld vollständig ausgefüllt werden muss:

```
Filler Characters
-------------------------
Optional, Partial ....
Required, Partial ....
Optional, Complete ...
Required, Complete ...
```

Folgende Optionen stehen zur Verfügung:

Feldtyp	Erläuterung
Optional, Partial	Kein Pflichtfeld, Feld braucht nicht vollständig ausgefüllt zu werden.
Required, Partial	Pflichtfeld, Feld braucht nicht vollständig ausgefüllt zu werden. (`AD=E`).
Optional, Complete	Kein Pflichtfeld; falls das Feld ausgefüllt wird, muss es vollständig ausgefüllt werden (`AD=G`).
Required, Complete	Pflichtfeld, Feld muss vollständig ausgefüllt werden (`AD=EG`).

Mit der Funktion zur **erweiterten Feldbearbeitung** können außerdem Füllzeichen für einzelne Felder definiert werden. Zur Definition von Feldtypen siehe auch Session-Parameter `AD` in der *Parameter-Referenz*-Dokumentation.

17 Maske bearbeiten — Funktion 'Edit Map'

- Bildschirm-Modi 224
- PF-Tasten und Kommandos zum Positionieren 225
- Zeilenkommandos 226
- Feld-Kommandos 228

Bildschirm-Modi

Zu Beginn einer Masken-Editor-Sitzung wird zunächst immer der Split-Screen-Modus aufgerufen. Dabei enthält die obere Hälfte des Masken-Editor-Bildschirms Datendefinitionen anderer Natural-Objekte (siehe *Definitionen aus anderen Objekten übernehmen*) und/oder die aktuellen Trennwert-(Delimiter-)Einstellungen. Die untere Hälfte des Bildschirms enthält den Editierbereich. Siehe folgendes Beispiel:

```
Ob _                                  Ob D CLS ATR DEL     CLS ATR DEL
 .                                     .     T  D    Blnk    T  I    ?
 .                                     .     A  D    _       A  I    )
 .                                     .     A  N    ¬       M  D    &
 .                                     .     M  I    :       O  D    +
 .                                     .     O  I    (
 .                                     .
 001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+-




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

Mit PF9 können Sie zwischen Vollbild- und Split-Screen-Modus hin- und herschalten.

Sie können den Masken-Editierbereich verlassen, indem Sie PF3 drücken oder einen Punkt (.) in den Eingabefeldern **Ob** im oberen Bereich des Bildschirms eingeben.

PF-Tasten und Kommandos zum Positionieren

Mit den PF-Tasten und Kommandos, die in der folgenden Liste enthalten sind, können Sie eine Editor-Funktion aufrufen oder im Masken-Editierbereich navigieren. Die Kommandos geben Sie am Anfang einer Maskenzeile ein:

PF-Taste	Kommando	Erläuterung
PF1		Ruft die Hilfe-Funktion des Masken-Editors auf.
PF2		Ermöglicht die Anzeige und das Ändern der aktuellen Masken-Einstellungen.
PF3	.Q	Beendet die Bearbeitung der Maske. Es erscheint wieder das Menü **Edit Map**.
PF4		Testet die Masken-Definition.
PF5		Ruft die Funktion zur **erweiterten Feldbearbeitung** zu dem Feld auf, in dem sich zurzeit der Cursor befindet..
PF6	.--	An den Anfang der Maske positionieren.
PF7	.-	Eine halbe Fensterseite nach oben positionieren.
	.-*nnn*	Um *nnn* Zeilen nach oben positionieren.
PF8	.+	Eine halbe Fensterseite nach unten positionieren.
	.+*nnn*	Um *nnn* Zeilen nach unten positionieren.
	.++	An das Ende der Masken positionieren.
PF9	./	Umschalten zwischen Split-Screen- und Vollbild-Modus.
PF10	.<	Eine halbe Fensterseite nach links positionieren.
	.<*nnn*	Um *nnn* Spalten nach links positionieren.
	.<<	An den linken Rand der Maske positionieren.
PF11	.>	Eine halbe Fensterseite nach rechts positionieren.
	.>*nnn*	Um *nnn* Spalten nach rechts positionieren.
	.>>	An den rechten Rand der Maske positionieren.
PF12		Abbruch der Bearbeitung. Alle seit dem letzten Drücken der Enter-Taste vorgenommenen Änderungen werden verworfen.
	.*	Linke obere Ecke zur Cursor-Position verschieben.
PF13	..X	Nur verfügbar, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Anzeige von Feldumrahmungen gestattet. Umschalten zwischen dem Editierbereich des Masken-Editors und dem Outline Editor.

Zeilenkommandos

Zeilenkommandos müssen in folgender Form eingegeben werden:

```
..line-command
```

wobei die zwei Punkte (..) zwei Ausprägungen der für die Maskendefinition wirksamen Steuerzeichen darstellen.

Anmerkung: Es wird empfohlen, am Ende eines jeden Zeilenkommandos ein Leerzeichen einzugeben. Dadurch wird verhindert, dass der Editor versucht, die anschließenden Informationen in der Zeile als Bestandteil des Zeilenkommandos zu interpretieren.

Folgende Zeilenkommandos stehen beim Masken-Editor zur Verfügung:

Kommando	Erläuterung
`..A`	Array-Tabellen-Definition (siehe Abschnitt ***Array- und Tabellen-Definitionen***).
`..An`	Array-Tabellen-Definition mit *n* Ausprägungen. Dieses Kommando können Sie benutzen, um eine Tabelle mit vertikal *n* Ausprägungen für alle in der aktuellen Zeile angegebenen Felder zu erstellen.
`..C`	Zentriert eine einzelne Zeile (d.h. die Zeile, in der das Kommando eingegeben wurde). Zwei Kommandos `..C`, die auf demselben Bildschirm eingegeben werden, zentrieren die erste Zeile und passen den Rest der selektierten Zeilen an.
`..Cn`	Zentriert die Zeile und verschiebt die *n*-1 Zeilen darunter entsprechend.
`..C*`	Zentriert die Zeile und verschiebt alle Zeilen darunter entsprechend.
`..D`	Löscht eine einzelne Zeile (d.h. die Zeile, in der das Kommando eingegeben wurde). Zwei Kommandos `..D`, die auf demselben Bildschirm eingegeben werden, löschen den Zeilenblock, der durch diese Kommandos eingegrenzt wird.
`..Dn`	Löscht die Zeile und die *n*-1 Zeilen darunter.
`..D*`	Löscht die Zeile und alle Zeilen darunter. Falls der Löschvorgang Array-Elemente betrifft, wird das gesamte Array gelöscht.
`..E`	Dient zum Aufrufen des Bildschirms **Field and Variable Definitions - Summary** der Funktion **Field and Variable Definitions** (siehe *Funktionen im Menü "Edit Map"*) für alle Felder, die in der betreffenden Zeile enthalten sind. Zwei Kommandos `..E`, die auf demselben Bildschirm eingegeben werden, bringen alle Felder zur Anzeige, die sich innerhalb des Zeilenbereiches befinden, der durch diese Kommandos eingegrenzt wird.

Kommando	Erläuterung
..E*n*	Dient zum Aufrufen des Bildschirms **Field and Variable Definitions - Summary** der Funktion **Field and Variable Definitions** (siehe *Funktionen im Menü "Edit Map"*) für die Zeile und die *n*-1 Zeilen darunter.
..E*	Dient zum Aufrufen des Bildschirms **Field and Variable Definitions - Summary** der Funktion **Field and Variable Definitions** (siehe *Funktionen im Menü "Edit Map"*) für die Zeile und alle Zeilen darunter. Die Kommandos ..E bringen einen Bildschirm mit dem Namen und der Natural-Daten-Format/Länge der angeforderten Felder zur Anzeige. Die gezeigten Feldnamen können geändert werden. In der Spalte **Cmd** können Sie die gewünschte Funktion auswählen: **Erweiterte Feld-Bearbeitung**, Bearbeitung von **Array-Definitionen** und **Verarbeitungsregeln**.
..F*c*	Füllt die Leerräume in einer Zeile mit dem Buchstaben *c* auf.
..I	Fügt eine einzelne Zeile ein. Um das Einfügen der Zeile zu ermöglichen, wird die letzte Leerzeile auf dem Bildschirm gelöscht,.
..I*n*	Fügt *n* Zeilen unterhalb der Zeile ein, in der das Kommando eingegeben wurde.
..I*	Fügt so viele Zeilen wie möglich unterhalb der Kommandozeile ein.
..J	Führt die Zeile, in der das Kommando eingegeben wurde, mit der unter ihr befindlichen Zeile zusammen. Zwei Kommandos ..J, die auf demselben Bildschirm eingegeben werden, bewirken, dass der mit diesen Kommandos eingegrenzte Zeilenbereich zusammengeführt wird.
..J*n*	Führt die Zeile, in der das Kommando eingegeben wurde, mit den *n*-1 unter ihr befindlichen Zeilen zusammen.
..J*	Führt die Zeile mit allen Zeilen zusammen, die sich unter ihr befinden. Wenn beim Zusammenführen von Zeilen eine zu lange Zeile entsteht, wird die untere Zeile an einer soweit wie möglich rechts gelegenen Stelle aufgespalten und der linke Teil wird dann mit der vorhergehenden Zeile zusammengeführt. Der rechte Teil der aufgespaltenen Zeile wird dann so weit nach links verschoben, bis er in einer senkrechten Linie mit der Zeile abschließt, in der das Kommando eingegeben wurde.
..L	Dient zum Aufrufen des Bildschirms **Modify INCDIR Statements of Map**, in dem Sie die INCDIR-Statements auflisten und aktualisieren können, die für Maskenfelder generiert wurden, welche aus DDMs kopiert wurden (siehe ***Referenzen auf DDMs prüfen und korrigieren***.
..M	Verschiebt die Zeile, in der das Kommando eingegeben wird, unter die mit dem Cursor markierte Zeile. Zwei Kommandos ..M, die auf demselben Bildschirm eingegeben werden, bewirken, dass der mit diesen Kommandos eingegrenzte Zeilenblock unter die mit dem Cursor markierte Zeile verschoben wird.
..M*n*	Verschiebt die Zeile, in der das Kommando eingegeben wird, und die *n*-1 unter ihr befindlichen Zeilen unter die mit dem Cursor markierte Zeile.

Kommando	Erläuterung
..M*	Verschiebt die Zeile, in der das Kommando eingegeben wird, und alle unter ihr befindlichen Zeilen unter die mit dem Cursor markierte Zeile. Dieses Kommando ist nur zweckmäßig, wenn sich die mit dem Cursor markierte Zeile oberhalb der Zeile befindet, in der Das Kommando eingegeben wird.
..P	Ruft die Bearbeitung der PF-Tasten-Verarbeitungsregeln auf. Bei den PF-Tasten-Verarbeitungsregeln handelt es sich um spezielle Verarbeitungsregeln zum Definieren von Aktivitäten, die programmsensitiven PF-Tasten zugewiesen werden.
..P*n*	Ruft die Bearbeitung der PF-Tasten-Verarbeitungsregeln für Rangebene *n* auf.
..P*	Listet alle Verarbeitungsregeln auf, die für die PF-Tasten in dieser Maske definiert sind.
..Q	Beendet die Bearbeitung der Maske. Es erscheint wieder das Menü **Edit Map**.
..R	Wiederholt alle Literal-Zeichenketten einmal in der Zeile, in der das Kommando eingegeben wurde. Die Cursor-Position dient zur Anzeige der Zielzeile. Zwei Kommandos ..R, die auf demselben Bildschirm eingegeben werden, bewirken, dass die Literal-Zeichenketten innerhalb des mit diesen Kommandos eingegrenzten Zeilenblocks wiederholt werden.
..R*n*	Wiederholt alle Literal-Zeichenketten in dieser und den *n*-1 darauf folgenden Zeilen. Falls sich der Cursor unterhalb der Kommandozeile befindet, wird derselbe Text *n* Mal wiederholt.
..S	Spaltet die Zeile an der Cursor-Position auf. Zwei Kommandos ..S, die auf demselben Bildschirm eingegeben werden, bewirken, dass der mit diesen Kommandos eingegrenzte Zeilenblock aufgespalten wird.
..S*n*	Spaltet die Zeile, in der das Kommando eingegeben wird, und die *n*-1 unter ihr an der Cursor-Position auf.
..X	Nur verfügbar, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Anzeige von Feldumrahmungen gestattet. Umschalten zwischen dem Editierbereich des Masken-Editors und dem Outline Editor. Als Alternative zu diesem Zeilenkommando können Sie PF13 benutzen.

Feld-Kommandos

Feld-Kommandos müssen Sie in folgender Form eingeben:

```
.field-command
```

wobei der Punkt (.) das für die Maskendefinition wirksame Steuerzeichen darstellt.

Jedes Kommando muss in der ersten Position eines Texts oder Datenfeldes beginnen.

Ein Feld-Kommando können Sie auf einen Bereich von Feldern oder Konstanten anwenden. Zur Festlegung eines Bereichs haben Sie folgende Möglichkeiten:

- Sie können zwei oder noch mehr Feld-Kommandos im selben Bildschirm benutzen. Der Spaltenbereich (horizontaler Bereich) und der Zeilenbereich (vertikaler Bereich) werden durch die Positionen der Kommandos bestimmt. (Im ***Map Editor Tutorial*** wird dies anhand einiger Beispiele veranschaulicht.)
- Sie können einen Wiederholungsfaktor *n* benutzen. Das Kommando wird bei dem vorgesehenen Feld und außerdem in den Feldern in den *n*-1 Zeilen unter diesem Feld angewendet. Ein Stern (*) als Wiederholungsfaktor bewirkt, dass das Kommandos solange wiederholt wird, bis das untere Ende der Maske erreicht ist.

Anmerkung: Es wird empfohlen, am Ende eines jeden Feld-Kommandos ein Leerzeichen einzugeben. Dadurch wird verhindert, dass der Editor versucht, Teile des Feldes als Bestandteil des Feld-Kommandos zu interpretieren.

Folgende Feld-Kommandos stehen beim Masken-Editor zur Verfügung:

Kommando	Erläuterung
.A	Definiert ein Array. Dieses Kommando können Sie nur bei einem einzelnen Feld (nicht bei einem Bereich von Feldern) anwenden. Die **Array-Definition** (siehe *Array- und Tabellen-Definitionen*) wird auf dem dafür vorgesehenen Bildschirm angegeben. Das resultierende Array wird mit seiner linken oberen Ecke an der Stelle positioniert, an der das Kommando eingegeben wurde. Sie können ein Array redefinieren, indem Sie das Kommando .A bei einem seiner Elemente anwenden.
.A*n*	Beim Kommando .A können Sie einen Wiederholungsfaktor *n* angeben, um so ein eindimensionales Array (kein Abstand, kein Versatz) zu definieren, ohne dafür einen separaten Funktionsbildschirm benutzen zu müssen.
.C	Zentriert ein Feld oder einen Bereich von Feldern zwischen angrenzenden Feldern. Um ein einzelnes Feld zu zentrieren, müssen Sie das Kommando .C in das zu zentrierende Feld eingeben. Um einen Bereich von Feldern zu zentrieren, müssen Sie das Kommando .C in das erste und das letzte zu zentrierende Feld eingeben. Alternativ können Sie Sie das Kommando .C in das erste zu zentrierenden Feld eingeben und den Cursor in das letzte zu zentrierenden Feld setzen. Falls kein angrenzendes Feld oder keine angrenzenden Felder vorhanden sind, werden stattdessen die für die Masken-Definition wirksamen Spaltenbegrenzungen benutzt.
.D	Löscht ein Feld oder einen Bereich von Feldern. Um ein einzelnes Feld zu löschen, müssen Sie das Kommando .D in das zu löschende Feld eingeben. Um einen Bereich von Feldern zu löschen, müssen Sie das Kommando .D in das erste und das letzte zu löschende Feld eingeben. Der zu löschende Bereich von Feldern kann länger als eine Zeile sein. Wenn Sie ein Array-Element löschen, wird das ganze Array gelöscht.

Kommando	Erläuterung
.E	Aufruf der Funktion zur **erweiterten Feld-Bearbeitung** (siehe entsprechenden Abschnitt) für das Feld, in dem sich zurzeit der Cursor befindet. Dieses Kommando können Sie nur bei einem einzelnen Feld (nicht bei einem Bereich von Feldern) anwenden. Zum Aufrufen der Funktion zur **erweiterten Feld-Bearbeitung** können Sie auch den Cursor in das ausgewählte Feld stellen und PF5 drücken.
.J	Fügt Felder zusammen, die sich in aufeinander folgenden Zeilen befinden. Die linke Begrenzung für die Zusammenfügung entspricht der Stelle, an der das Kommando .J eingegeben wird, und die rechte Begrenzung entspricht der Position des Cursors.
.M	Verschiebt ein Feld oder einen Bereich von Feldern. Um ein einzelnes Feld zu verschieben, müssen Sie das Kommando .M in das zu verschiebende Feld eingeben und den Cursor in die Zielposition stellen. Um einen Bereich von Feldern zu verschieben, müssen Sie das Kommando .M in das erste und das letzte zu verschiebende Feld eingeben und den Cursor in die Zielposition stellen.
.P[*n*]	Ermöglicht die Bearbeitung von **Verarbeitungsregeln** (siehe entsprechenden Abschnitt) für ein Feld. Um die Priorität (den Rang) der zu bearbeitenden Verarbeitungsregel anzugeben, können Sie beim Kommando .P einen Parameter *n* mit angeben. Falls nötig, können Sie den Wert für *n* in runden Klammern () einschließen.
.R	Wiederholt (kopiert) ein Feld oder einen Bereich von Feldern. Um ein einzelnes Feld zu kopieren, müssen Sie das Kommando .R in das zu kopierende Feld eingeben und den Cursor in die Zielposition stellen. Um einen Bereich von Feldern zu kopieren, müssen Sie das Kommando .R in das erste und das letzte zu kopierende Feld eingeben und den Cursor in die Zielposition stellen. Die Wiederholung erfolgt immer abwärts und von links nach rechts. Den Feldern, die mit diesem Kommando erzeugt werden, wird ein funktionsloser Blindname zugewiesen. Um einem solchen Feld einen gültigen Namen zuzuweisen, müssen Sie die Vorgehensweise zum **nachträglichen Zuweisen von Feldern** oder Funktion zur **erweiterten Feld-Bearbeitung** anwenden (siehe entsprechenden Abschnitt). **Anmerkung:** Arrays können nicht kopiert werden.
.S	Spaltet (verschiebt) eine Zeile oder einen Bereich von Zeilen. Geben Sie .S in das Feld ein, in dem die Aufspaltung beginnen soll, und setzen Sie den Cursor an die Zielposition. Die Zeile wird an der Stelle geteilt, an der das Kommando .S eingegeben wurde. Der rechte Teil wird dann an die Cursor-Position verschoben.
.T	Verkürzt (löscht) ein Feld oder einen Bereich von Feldern in einer Zeile. Geben Sie .T in das Feld ein, bei dem die Verkürzung beginnen soll. Wenn Sie dieses Kommando benutzen, um ein Array-Element zu verkürzen (zu löschen), wird das ganze Array gelöscht.

18 Maskenfelder definieren

- Maskenfelder direkt im Bildschirm definieren 232
- Definitionen aus anderen Objekten übernehmen 233
- Systemvariablen in einer Masken-Definition benutzen 238

Maskenfelder direkt im Bildschirm definieren

Sie können die Maskenfelder direkt im Masken-Editierbereich definieren, indem Sie ein Trennzeichen (Delimiter) und danach die Anzahl der Stellen angeben, die dem Feld zugeordnet werden sollen. Sie können dazu die folgenden Zeichen verwenden:

Zeichen	Bedeutung
9	Numerische Stelle.
0	Numerische Stelle, rechtsbündig.
.	Dezimal-Notation (nur bei einem numerischen Feld).
S	Vorzeichen-Stelle (nur bei einem numerischen Feld).
HH	Hexadezimale Stelle (binär) (muss in Zweiergruppen eingegeben werden).
X	Alphanumerische Stelle.

Außerdem kann ein Wiederholungsfaktor in der Form `(n)` angegeben werden. Beispiel: `X(5)` ist gleichbedeutend mit `XXXXX`.

Nachfolgend einige Beispiele für Feld-Definitionen (das Trennzeichen kann entsprechend den Erfordernissen geändert werden):

:999	3 Stellen, numerisch
:000	3 Stellen, numerisch, rechtsbündig
:99.9	3 Stellen, numerisch, mit Dezimalzeichen
:S9(6)	6 Stellen, numerisch, mit Vorzeichen
:HHHH	4 Stellen, hexadezimal
:X	1 Stelle, alphanumerisch
:X(7)	7 Stellen, alphanumerisch

Feldern, die wie oben dargestellt eingegeben werden, wird vom Masken-Editor ein funktionsloser Blindname zugewiesen. Vor der Ausführung der Maske muss jedem Feld ein gültiger Name zugewiesen werden. Dazu können Sie entweder die Funktion zur **erweiterten Feld-Bearbeitung** oder die Vorgehensweise beim **nachträglichen Zuweisen von Feldern** anwenden. Andere Feldformate können Sie mit Hilfe der Funktion zur erweiterten Feld-Bearbeitung angeben.

Definitionen aus anderen Objekten übernehmen

Um eine Maske zu definieren, können Sie auch Definitionen aus der lokalen Datendefinition eines anderen Natural-Source-Objekts auswählen. Bei den Datendefinitionen handelt es sich entweder um Felddefinitionen in einem DDM oder um Variablen-Definitionen.

Eine Liste von Objekttypen, die Sie für die Auswahl von Datendefinitionen benutzen können, finden Sie im Abschnitt ***Objekttypen und Typen-Codes***.

Einschränkungen bei der Auswahl:

- Programme, Subroutinen, Subprogrammme, Helproutinen und Functions können Sie nur dann benutzen, wenn diese ein `DEFINE DATA LOCAL`-Statement enthalten. Die Definitionen bei einem `DEFINE DATA USING`-Statement stehen nicht für die Auswahl zur Verfügung.
- Felder mit dem Natural-Datenformat U (Unicode), C (Attribute Control) oder Handle (Object Handles) können Sie nicht auswählen. Diese stehen nicht für die Auswahl zur Verfügung.
- Wenn Sie Konstanten für die Begrenzungen eines Array benutzen und wenn diese Konstanten nicht in der lokalen Datendefinition des gewählten Natural-Source-Objekts definiert sind, gehen die Begrenzungen verloren, wenn Sie das Array wählen.
- Sie können kein Element wählen, dem ein Punkt (.) vorangestellt ist (z.B. eine Gruppe).
- Eingabefelder und änderbare Felder (Eingabe/Ausgabe) können im Masken-Editierbereich nur einmal angegeben werden.

Dieser Abschnitt behandelt folgende Themen:

- Datendefinitionen auflisten und kopieren
- Referenzen auf DDMs prüfen und korrigieren
- Objekttypen und Typen-Codes

Datendefinitionen auflisten und kopieren

Um Datendefinitionen aus anderen Natural-Objekten aufzulisten und zu kopieren:

1 Benutzen Sie den Split-Screen-Modus des Masken-Editors.

```
Ob _                                  Ob D CLS ATR DEL     CLS ATR DEL
 .                                     .     T  D    Blnk    T  I    ?    ↩
 .                                     .     A  D    _       A  I    )    ↩
 .                                     .     A  N    ¬       M  D    &    ↩
 .                                     .     M  I    :       O  D    +    ↩
 .                                     .     O  I    (                    ↩
 .                                     .                                  ↩
 001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+-
```

2 Geben Sie in die Eingabefelder **Ob** (Object) links oben im Bildschirm einen gültigen Objektcode (siehe auch *Objekttypen und Typen-Codes*) und danach einen Objektnamen ein (z.B. `P TEST` wie im **Beispiel** in Schritt 4). Löschen Sie, soweit vorhanden, alle übrigen Zeichen in dieser Zeile.

Oder:

Lassen Sie das erste (durch einen Unterstrich gekennzeichnete) Eingabefeld der Eingabefelder **Ob** (Object) links oben im Bildschirm leer und geben Sie im zweiten Eingabefeld ein Fragezeichen (?) ein oder drücken Sie PF1 (Help). Löschen Sie, falls vorhanden, alle übrigen Zeichen in dieser Zeile.

Wenn Sie Enter drücken, erscheint der Bildschirm **Select an Object** (Objekt auswählen) mit einer Liste aller Objekte, die in der aktuellen Library enthalten sind. Durch Drücken von Enter können Sie in der Liste blättern. Geben Sie im Feld **Select** die Nummer ein, die dem Objekt entspricht, das Sie auswählen möchten.

3 Drücken Sie Enter.

Links oben im Masken-Editierbildschirm werden jetzt die Datendefinitionen angezeigt, die im `DEFINE DATA`-Abschnitt des Natural-Programms enthalten sind (Programm `TEST` im folgenden **Beispiel**).

Um in der Liste der Datendefinitionen in den **Ob**-Eingabefeldern zu blättern, können Sie die vorhandenen Einträge (`P TEST` im folgenden **Beispiel**) mit einem der folgenden Positionierungskommandos überschreiben. Lassen Sie die übrig gebliebenen Zeichen bestehen.

Kommando	Erläuterung
+	In der Liste eine Seite nach unten blättern.
++	An das Ende der Liste blättern.
-	In der Liste eine Seite nach oben blättern.
-	An den Anfang der Liste blättern.
+*n*	Um *n* Zeilen nach unten blättern.
-*n*	Um *n* Zeilen nach oben blättern.

4 Um die Datendefinitionen eines anderen Objekts in den **Ob**-Eingabefeldern links oben im Bildschirm anzuzeigen, geben Sie einen gültigen Objektcode (siehe auch ***Objekttypen und Typen-Codes***) und danach einen Objektnamen ein. Löschen Sie, soweit vorhanden, alle übrig gebliebenen Zeichen in dieser Zeile.

Beispiel:

```
Ob P TEST                                Ob L LDA01
1 #NAME-START              A20           A #NAME-START              A20
2 #NAME-END                A20           B #NAME-END                A20
3 #MARK                    A1            C #PERS-ID                 A8
. EMPLOYEES-VIEW           *V1           D #MAKE                    A20
4 PERSONNEL-ID             A8            E #MODEL                   A20
5 NAME                     A20           .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Oder:

Wählen Sie, wie zuvor in **Schritt 2** beschrieben, ein Objekt aus der Liste.

5 Kopieren Sie die gewünschten Datendefinitionen in den Masken-Editierbereich:

- Geben Sie an beliebiger Stelle einer Leerzeile im Masken-Editierbereich ein Trennzeichen und danach die neben der gewünschten Datendefinition angezeigte Folgenummer oder den Buchstaben ein, zum Beispiel:

  ```
  (3
  ```

  Diese Eingabe wählt die Datendefinition aus, die der Nummer 3 (im obigen Beispiel #MARK) zugewiesen ist.

```
:C
```

Diese Eingabe wählt die Datendefinition aus, die dem Buchstaben C (im obigen Beispiel #PERS-ID) zugewiesen ist.

- Drücken Sie Enter.

  Die Eigenschaften der Datendefinitionen (Name, Format, Länge oder Array-Definitionen) werden in die Maske kopiert.

  Mittels der Feldbearbeitungsfunktionen des Masken-Editors können Sie die kopierten Werte anzeigen oder ändern. Siehe auch ***Erweiterte Feld-Bearbeitung*** und ***Array- und Tabellen-Definitionen*** .

Referenzen auf DDMs prüfen und korrigieren

Zu jedem aus einem DDM kopierten Maskenfeld wird automatisch ein INCDIR-Statement in den Quellcode der Maske generiert. Das INCDIR-Statement enthält den Namen der DDM-Datei und des Feldes, das von dem Maskenfeld referenziert wird. Sie können prüfen, ob ein INCDIR-Statement eine ungültige DDM-Referenz enthält, indem Sie die CHKRULE-Option des Systemkommandos COMPOPT einschalten. Natural gibt dann, wenn Sie die Maskendefinition katalogisieren, eine entsprechende Fehlermeldung aus. Weitere Informationen siehe *CHKRULE - Validierung von INCDIR-Statements in Masken* in der *Systemkommandos*-Dokumentation.

Im Masken-Editierbereich können Sie alle INCDIR-Statements auflisten und, falls erforderlich, den Namen eines DDM ändern.

Um ein einzelnes oder mehrere INCDIR-Statements aufzulisten und zu aktualisieren:

1 Geben Sie das folgende Zeilenkommando in einer beliebigen Zeile des Editierbereichs ein:

```
..L
```

Der Bildschirm **Modify INCDIR Statements of Map** erscheint (Beispiel):

```
16:24:23        - Modify INCDIR Statements of Map  MAPTEST  -       2008-05-14
                                                                   Top of List

      DDM Name                          Field Name
      AUTOMOBILES                       MAKE
      AUTOMOBILES                       OWNER-PERSONNEL-NUMBER
      FINANCE                           BANK
      AUTOMOBILES                       MODEL
      EMPLOYEES                         PERSONNEL-ID
      AUTOMOBILES                       BODY-TYPE
      EMPLOYEES                         FIRST-NAME
      AUTOMOBILES                       NUMBER-OF-CYLINDERS
      EMPLOYEES                         MIDDLE-I
      AUTOMOBILES                       HORSEPOWER
      EMPLOYEES                         NAME
      AUTOMOBILES                       PISTON-DISPLACEMENT
      EMPLOYEES                         MIDDLE-NAME
      AUTOMOBILES                       WEIGHT
      EMPLOYEES                         MAR-STAT
      AUTOMOBILES                       COLOR









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

Der Bildschirm enthält eine Liste aller INCDIR-Statements, die in den Quellcode der aktuellen Masken-Definition generiert worden sind. Die Liste enthält außerdem die Namen aller DDM-Dateien und DDM-Felder, aus denen Sie ein Maskenfeld kopiert haben.

2 Falls erforderlich, können Sie den DDR-Namen für ein INCDIR-Statement ändern, indem Sie den Namen oder die Namen in der Spalte **DDM Name** ersetzen.

Oder:

Drücken Sie PF5, um das Fenster **Generic Change** aufzurufen. Geben Sie im Feld **New DDM Name** den Namen des neuen DDM ein, der von allen INCDIR-Statements benutzt werden soll, die diesen DDM referenzieren.

Der DDM-Name bzw. die DDM-Namen in dem (den) entsprechenden INCDIR-Statement(s) werden aktualisiert, wenn Sie die Masken-Definition speichern (Funktion **Save Map** oder **Stow Map**).

Objekttypen und Typen-Codes

Die folgende Liste enthält eine Aufstellung der Objekttypen und der entsprechenden Typen-Codes, die zum Auswählen von Datendefinitionen verwendet werden können:

Objekttyp	Typen-Code	Bemerkung
Parameter Data Area	A	
Predict Conceptual Files	C	Der Objekttyp „Predict Conceptual Files“ ist nur verfügbar, wenn Predict installiert ist.
Global Data Area	G	
Helproutine	H	
Local Data Area	L	
Map	M	
Subprogram	N	
General Object	O	Der Objekttyp „General Object“ (O) ermöglicht es Ihnen, Daten aus einem Quellcode zu importieren, ohne den korrekten Objekttyp anzugeben, wenn Sie zum Beispiel den Objektnamen, aber nicht den Objekttyp kennen. Im Feld **Ob** können Sie in einem solchen Fall `O objectname` eingeben. Dabei stellt *objectname* den Objektnamen dar. Der Masken-Editor ändert dann den Objekttyp von O in den korrekten Objekttyp und importiert die Daten.
Program	P	
Subroutine	S	
View (DDM)	V	
Function	7	

Systemvariablen in einer Masken-Definition benutzen

Natural-Systemvariablen können in einer Masken-Definition angegeben werden. Weitere Informationen zu Systemvariablen siehe *Systemvariablen* im *Leitfaden zur Programmierung* und die ausführlichen Einzelbeschreibungen in der *Systemvariablen*-Dokumentation.

Einer Systemvariablen muss ein Ausgabetrennzeichen (Delimiter) vorangestellt werden. Siehe folgendes Beispiel:

```
(*TIME
(*DATE
(*APPLIC-ID
```

19 Erweiterte Feld-Bearbeitung

- Funktion zur erweiterten Feld-Bearbeitung aufrufen und beenden 242
- Felder im erweiterten Feld-Bearbeitungsbereich 244
- Felder im erweiterten Text-Feld-Bearbeitungsbereich 248

Mit der Funktion zur erweiterten Feld-Bearbeitung können Sie zusätzliche Attribute für Felder definieren.

Funktion zur erweiterten Feld-Bearbeitung aufrufen und beenden

Um die Funktion zur erweiterten Feld-Bearbeitung aufzurufen:

- Innerhalb des Masken-Editierbereichs:

  Geben Sie das folgende Zeilenkommando in der Zeile oder in den Zeilen ein, die Felder enthält bzw. enthalten, für die Sie zusätzliche Attribute definieren möchten:

  ```
  ..E
  ```

  Weitere Optionen siehe ***Zeilenkommandos*** im Abschnitt *Masken bearbeiten - Funktion "Edit Map"*.

  Oder:

  Innerhalb des Masken-Editierbereichs:

  Geben Sie das folgende Zeilenkommando in das Feld ein, für das Sie zusätzliche Attribute definieren möchten:

  ```
  .E
  ```

  Oder:

  Im Menü **Edit Map**:

  Führen Sie die Funktion `D` (**Field and Variable Definitions**) für die Maske aus, die das betreffende Feld bzw. die betreffenden Felder enthält.

  Es erscheint der Bildschirm **Field and Variable Definitions - Summary**.

  Geben Sie dort neben dem Feld, für das Sie zusätzliche Attribute definieren möchten, folgendes Zeilenkommando ein:

  ```
  .E
  ```

  Bei einem Datenfeld wird ein erweiterter Feld-Editierbereich im oberen Bereich des Masken-Editor-Bildschirms angezeigt (Beispiel):

```
Fld #001                                                                Fmt A6
--------------------------------------------------------------------------------
AD= MIT_______     ZP=          SG=        HE= _____________________   Rls 0
AL= ______         CD= __       CV= _____________________________   Mod Undef
PM= __  DF=        BX= ____     DY= _____________________________
EM= _____________________________ SB= _____________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----

.EXXXXX




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  <---  --->  --    -     +           <     >     Let
```

Anmerkung: Das Feld **BX** wird nur dann angezeigt, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Anzeige von Feldumrahmungen gestattet.

Bei einem Text-Feld, das mit der **Trennzeichen-Klasse** `D` erstellt wurde, sieht der angezeigte erweiterte Feld-Editierbereich folgendermaßen aus (Beispiel):

```
Extended Text Field
--------------------------------------------------------------------------------
AD= I_________     CD= __    PM= __    BX= ____    Hex Editing .. N
Value Field1_____________________________________________________________________
      _____________________________________________________________________
      _____________________________________________________________________
      _____________________________________________________________________
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----

   .Field1




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  <---  --->  --    -     +           <     >     Let
```

Sie können die Funktion zur erweiterten Feld-Bearbeitung für das nächste oder das vorhergehende Feld im Masken-Editierbereich direkt aufrufen, indem Sie PF4 bzw. PF5 drücken.

Um die Funktion zur erweiterten Feld-Bearbeitung zu beenden:

- Drücken Sie PF3.

  Oder:

  Drücken Sie Enter.

Felder im erweiterten Feld-Bearbeitungsbereich

Die folgende Tabelle enthält die Beschreibungen zu den Feldern, die im erweiterten Feld-Bearbeitungsbereich des Editor-Bildschirms vorhanden sind:

Feld	Erläuterung
Fld **Arr**	Der Feld- bzw. Array-Name. Falls der Name länger als der vorhandene Platz ist, geben Sie, um zusätzlichen Platz zu schaffen, am Zeilenanfang das Kommando `.E` ein. Der im Feld **Fld** bzw. **Arr** eingegebene Name ist abhängig davon, welche Methode Sie beim Anlegen des Feldes angewendet haben: ■ Wenn das Feld aus einer Variablen in einem anderen Natural-Objekt kopiert wurde, dann wird der in diesem Objekt verwendete Name eingegeben. ■ Wenn das Feld von einem DDM-Feld kopiert wurde, dann wird der Name des DDM, gefolgt von dem im DDM benutzten Namen, eingegeben. Beispiel: `EMPLOYEES.PERSONNEL-ID` wobei `EMPLOYEES` den Namen der DDM und `PERSONNEL-ID` den Namen des in diesem DDM definierten Feldes bezeichnet. ■ Wenn das Feld als eine Natural-Systemvariable angegeben wurde, wird der Name der angegebenen Systemvariablen eingegeben. ■ Wenn das Feld keiner der vorgenannten Methoden entspricht, wird ihm ein funktionsloser Blindname zugewiesen. Sie müssen dem Feld einen Namen zuweisen, bevor Sie die Maske ausführen. Der Name eines Feldes kann geändert werden. Es ist dabei jedoch nicht möglich, ein Präfix bei einem Feld zu benutzen, dem zuvor kein Präfix zugewiesen worden war. Um einen Feldnamen mit Präfix zu erhalten, müssen Sie das Feld aus einer Datendefinition in einem anderen Natural-Objekt wählen. **Anmerkung:** Feldnamen-Dubletten sind nur bei Feldern zulässig, die als reine Ausgabefelder definiert sind. Weitere Informationen siehe ***Maskenfelder definieren***.
Fmt	Das Natural-Datenformat und die Länge des Feldes. Diese Angaben können Sie durch Überschreiben des aktuellen Eintrags ändern. Die Datenformate U (Unicode), C (Attribute Control) und Handle (Object Handle) sind *nicht* zulässig.

Feld	Erläuterung
	Um eine Referenz auf eine dynamische alphanumerische Variable zu definieren, müssen Sie hinter dem Eintrag ein (D) oder DYNAMIC angeben. Der Parameter AL wird automatisch gesetzt, und zwar entweder auf den angegebenen Wert oder auf die maximale auf dem Bildschirm verfügbare Länge.
AL oder **FL** oder **NL**	Die Länge, die beim Anzeigen des Feldes verwendet werden soll. Bei dynamischen und langen Variablen wird die Länge automatisch gesetzt, kann aber geändert werden. Diese Felder entsprechen den Session-Parametern AL, FL bzw. NL. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *AL - Alphanumerische Länge der Ausgabe*, *FL - Gleitkomma-Mantissenlänge*, und *NL - Numerische Länge der Ausgabe* in der *Parameter-Referenz*-Dokumentation.
Rls	Die Anzahl der zurzeit für das Feld definierten Verarbeitungsregeln.
ZP	Anzeige von Nullwerten. In das Feld **ZP** können Sie nur dann einen Wert eingeben, wenn das Feld numerisch ist oder wenn es sich um eine Zeit-Systemvariable handelt. Dieses Feld entspricht dem Session-Parameter ZP. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *ZP - Anzeige von Nullwerten* in der *Parameter-Referenz*-Dokumentation.
SG	Vorzeichen-Stelle. In das Feld **SG** können Sie nur dann einen Wert eingeben, wenn das Feld numerisch ist oder wenn es sich um eine Zeit-Systemvariable handelt. Dieses Feld entspricht dem Session-Parameter SG. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *SG - Vorzeichen-Stelle* in der *Parameter-Referenz*-Dokumentation.
PM	Druck-/Anzeige-Modus. Dieses Feld entspricht dem Session-Parameter PM. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *PM - Druck-/Anzeige-Modus* in der *Parameter-Referenz*-Dokumentation.
DF	Datumsformat (gilt nur bei Datumsfeldern). Dieses Feld entspricht dem Session-Parameter DF. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *DF - Datumsformat* in der *Parameter-Referenz*-Dokumentation.
DY	Dynamische Attribute. Dieser Parameter wird benutzt, um bestimmte, in der Zeichenkette einer alphanumerischen Variable enthaltene Zeichen zu definieren, die die Attributeinstellung steuern sollen. Dieses Feld entspricht dem Session-Parameter DY. Ausführliche Informationen zum Definieren dynamischer Attribute siehe *DY - Dynamische Attribute* in der *Parameter-Referenz*-Dokumentation. Wenn Sie für Ihre DY-Definition mehr als 30 Zeichen benötigen, geben Sie an der ersten Stelle des **DY**-Feldes ein Pluszeichen (+) ein und drücken Sie Enter. Das Fenster **DY Extension** wird angezeigt. Dort können Sie maximal 59 Zeichen eingeben. Wenn eine DY-Definition länger ist als das **DY**-Feld,

Feld	Erläuterung
	wird dies durch ein Pluszeichen (+) in dem **DY**-Feld angezeigt. Um das Fenster **DY Extension** aufzurufen und die erweiterte Definition anzuzeigen, zu bearbeiten oder zu löschen, brauchen Sie nur das Pluszeichen mit einem beliebigen Zeichen zu überschreiben. **Wichtig:** Eine erweiterte DY-Definition können Sie nur dann eingeben, wenn noch keine Kontrollvariable für diese Maske definiert worden ist (siehe auch **CV**-Feld).
HE	Helproutine oder Hilfemaske. Die Option **HE** können Sie benutzen, um dem Maskenfeld eine Helproutine oder eine Hilfe-Maske zuzuweisen. Eine Helproutine oder eine Hilfe-Maske wird zur Ausführungszeit aufgerufen, wenn die Feldhilfe für das Feld aufgerufen wird. Ausführliche Informationen siehe Beschreibung des Session-Parameters HE im Abschnitt *HE - Helproutine* in der *Parameter-Referenz*-Dokumentation. In das Feld **HE** können Sie den Namen einer Helproutine oder einer Hilfe-Maske und die Parameter eingeben, die an diese Helproutine oder Hilfe-Maske übergeben werden sollen. Falls Sie für die Eingabe zusätzlichen Platz benötigen, benutzen Sie in dem Feld das Feld-Kommando .E oder geben Sie ein Pluszeichen (+) ein und drücken Sie Enter. Es erscheint ein Fenster mit zusätzlichen Eingabezeilen. Die Syntax, die beim Angaben von Namen und Parametern im Feld **HE** gilt, entspricht der Syntax, die beim Session-Parameter HE beschrieben ist. Weitere Informationen siehe *HE-Parameter-Syntax* in der *Parameter-Referenz*-Dokumentation. Zusätzlich zu den dort vorhandenen Syntax-Erläuterungen gilt bei der Benutzung des Masken-Editors Folgendes: *operand1*: ■ Wird ein Variablenname angegeben, der dem Namen eines Maskenfeldes entspricht, muss das Feld Natural-Datenformat/Länge A8 haben. ■ Wird ein Variablenname angegeben, zu dem noch kein Maskenfeld existiert, wird automatisch ein Maskenparameter mit diesem Namen im Natural-Datenformat/Länge A8 definiert. *operand2*: ■ Wird ein Variablenname angegeben, zu dem noch kein Maskenfeld existiert, wird automatisch ein Maskenparameter mit diesem Namen im Natural-Datenformat/Länge N7 definiert. Wenn Sie einen Parameter aus dem **HE**-Feld entfernen, müssen Sie diesen Parameter außerdem aus der Maske entfernen, wenn es sich bei dem Parameter nicht um ein Maskenfeld handelt oder wenn der Parameter nicht mit einem anderen Maskenfeld als Hilfe-Parameter oder **Starting from**-Wert verbunden ist (siehe *Array-Definition*.
AD	Feldattribute. Dieses Feld entspricht dem Session-Parameter AD. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *AD - Attribut-Definition* in der *Parameter-Referenz*-Dokumentation. Um den Quellcode zu optimieren, werden die Standardwerte D, H, F und W akzeptiert aber nicht gespeichert (siehe auch Session-Parameter AD).

Feld	Erläuterung
CD	Farbattribute. Dieses Feld entspricht dem Session-Parameter CD. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *CD - Farbdefinition* in der *Parameter-Referenz*-Dokumentation.
BX	Feldumrahmungen. Dieses Feld wird nur dann angezeigt, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Darstellung von Feldumrahmungen unterstützt. Dieses Feld entspricht dem Session-Parameter BX, mit dem eine Umrahmung für ein Maskenfeld definiert werden kann. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *BX - Feldumrahmung (Box-Definition)* in der *Parameter-Referenz*-Dokumentation. **Anmerkung:** Eine komfortable Alternative für die manuelle Eingabe von Parameterwerten in das Feld **BX** bietet der Outline-Editor. Damit können Sie eine Feldumrahmung direkt an Ort und Stelle im Editierbereich um das Feld zeichnen, indem Sie Box-Codes angeben, für die automatisch die entsprechenden BX-Parameterwerte in den Quellcode der Maske generiert werden. Weitere Informationen siehe ***Feldumrahmungen erstellen — mit dem Outline-Editor***.
CV	Kontrollvariable für dynamische Feldattribute. Dieses Feld entspricht dem Session-Parameter CV. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *CV - Kontrollvariable* in der *Parameter-Referenz*-Dokumentation. **Anmerkung:** Wenn Sie eine Attribut-Kontrollvariable aus einem Feld entfernen, müssen Sie diese Attribut-Kontrollvariable außerdem aus der Maske entfernen, wenn sie nicht mit einem anderen Maskenfeld verbunden ist.
EM oder **EMU**	Editiermaske (**EM**) oder Unicode-Editiermaske (**EMU**), die bei dem Feld angewendet werden soll. Um zwischen diesen beiden Möglichkeiten umzuschalten, können Sie das Kommando .U in das Feld eingeben. Die Standardeinstellung ist **EM**. Dieses Feld entspricht dem Session-Parameter EM bzw. EMU. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *EM - Editiermaske* und *EMU - Unicode-Editiermaske* (*Parameter-Referenz*-Dokumentation) und *EMU, ICU, LCU, TCU versus EM, IC, LC, TC* (*Unicode and Code Page Support*-Dokumentation). Falls Sie für die Eingabe zusätzlichen Platz benötigen, geben Sie an der ersten Stelle in dem Feld das Feld-Kommando .E ein und drücken Sie Enter. Es erscheint ein Fenster mit zusätzlichen Platz für die Eingabe. Ein Feld, bei dem eine Editiermaske zur Anwendung kommt, wird im Masken-Editierbereich durch ein M gekennzeichnet.
SB	Der Name eines Array, in dem die Werte für eine Auswahlbox zur Verfügung gestellt werden. Um zu zeigen, dass eine Auswahlbox verfügbar ist, wird das Kennzeichen V angezeigt. Es gilt das Format des Quellfeldes. Sie können es im Fenster **Parameter Definitions** ändern.

Feld	Erläuterung	
	Siehe auch *SB - Auswahlfeld* in der *Parameter-Referenz*-Dokumentation.	
Mod	Das Feld **Mode** zeigt an, auf welche Weise das Feld erstellt wurde:	
	`Data`	Das Feld wurde durch Auswählen eines Feldes aus einer `DEFINE DATA`-Statement-Definition erstellt.
	`Sys`	Das Feld ist eine Systemvariable.
	`Undef`	Das Feld wurde direkt im Bildschirm erstellt und hat einen funktionslosen Blindnamen.
	`User`	Der Name des Feldes wurde mittels der Funktion zur **erweiterten Feld-Bearbeitung** erstellt.
	`View`	Das Feld wurde durch Auswählen eines Feldes aus einer View (Datei) erstellt.

Felder im erweiterten Text-Feld-Bearbeitungsbereich

Dieser Abschnitt gilt nur, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Darstellung von Feldumrahmungen unterstützt.

Die folgende Tabelle enthält die Beschreibungen zu den Feldern, die im erweiterten Feld-Bearbeitungsbereich des Editor-Bildschirms vorhanden sind:

Feld	Erläuterung
AD	Feldattribute. Dieses Feld entspricht dem Session-Parameter `AD`. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *AD - Attribut-Definition* in der *Parameter-Referenz*-Dokumentation. Standardmäßig ist der Wert `D` zugewiesen.
CD	Farbattribute. Dieses Feld entspricht dem Session-Parameter `CD`. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *CD - Farbdefinition* in der *Parameter-Referenz*-Dokumentation.

Feld	Erläuterung
PM	Druck-/Anzeige-Modus. Dieses Feld entspricht dem Session-Parameter PM. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *PM - Druck-/Anzeige-Modus* in der *Parameter-Referenz*-Dokumentation. Wenn Sie D in dieses Feld eingeben (PM=D). um ein erweitertes Text-Feld als natives DBCS-Feld zu definieren, werden im Masken-Editierbereich die Zeichen des erweiterten Feldes durch ein oder mehrere Paare des Buchstabens K dargestellt. Siehe auch Feld **Value** (weiter unten).
BX	Feldumrahmung, die mit dem Parameter BX angegeben werden. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe *BX - Feldumrahmung (Box-Definition)* in der *Parameter-Referenz*-Dokumentation. Die Angaben zum BX-Parameter werden gewöhnlich automatisch mittels der Funktionalität des Outline-Editors vorgenommen.
Hex Editing	Wenn Sie Y (Yes) eingeben, wird das Fenster **Text Field Hex Editing** angezeigt. Dort können Sie das hexadezimale Äquivalent des Nicht-Leerzeichen- oder Leerzeichen-Wertes ändern, der im Feld **Value** eingegeben wurde. Ein Leerzeichen-Wert wird durch eine Reihe von hexadezimalen Werten 00 dargestellt. Ein leeres Text-Feld können Sie mit dem Masken-Editor erstellen, indem Sie die hexadezimalen Werte eines existierenden Feldes durch 00 ersetzen. Beispiel: Für ein Text-Feld mit dem Wert TEST (siehe Feld **Value** weiter unten) ersetzen Sie seine hexadezimale Entsprechung der darin enthaltenen Zeichenkette A385A2A3 durch 00000000. Die Standardeinstellung beim **Hex Editing**-Feld ist N (No).
Value	Die alphanumerische Zeichenkette eines erweiterten Text-Feldes oder ein Leerzeichen-Wert für ein erweitertes leeres Text-Feld, das durch eine Anzahl von Punkten (.) im Masken-Editierbereich dargestellt wird. Dieses Feld kann auch ein Doppel-Byte-Zeichen enthalten, wenn im Feld **PM** ein D eingegeben wird. Falls die Terminal-Emulation Doppel-Byte-Zeichensätze (Double Byte Character Sets/DBCS) unterstützt, wird im Feld **Value** die dekodierte Grafik eines solchen Zeichens angezeigt. Den aktuellen Wert können Sie ändern, indem Sie entweder die Zeichenkette im Feld **Value** oder die im Fenster **Text Field Hex Editing** eingegebenen äquivalenten hexadezimalen Werte ersetzen. Das Feld **Value** darf *keine* Leerzeichen enthalten.

20 Feldumrahmungen erstellen — mit dem Outline-Editor

- Voraussetzungen 252
- Editor für Feldumrahmungen (Outline-Editor) aufrufen 254
- Box-Codes für Feldumrahmungen benutzen 255
- Ende-Markierungen für Feldumrahmungen benutzen 257
- Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen 258
- Beispiel für Feldumrahmungen — Box-Coding 259
- Zeilenkommandos und PF-Tasten 262

Mit dem Outline-Editor können Sie Feldumrahmungen in Masken definieren, die mit dem Natural-Masken-Editor erstellt wurden. Dazu setzen Sie Box-Codes bei den Daten- und Text-Feldern, die Sie mit einem Rahmen versehen möchten. Sie können auf diese Weise auch Leerzeichenbereiche mit Rahmen versehen.

Anmerkungen:

1. Bei den Box-Codes handelt es sich um Werte des Parameters `BX`. Siehe Beschreibung des Parameters `BX`.
2. Prinzipiell ist der Outline-Editor nur dafür vorgesehen, *Feldumrahmungen* zu erstellen. Er dient nicht zum *Definieren* von Feldern. Die Felddefinitionen werden im Editierbereich des Masken-Editors angelegt (siehe ***Maskenfelder definieren***).

Dieser Abschnitt behandelt folgende Themen:

Voraussetzungen

Um den Outline Editor benutzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Geeignetes Terminal
- Trennzeichen-Klasse D definieren

Geeignetes Terminal

Das verwendete Terminal bzw. die Terminal-Emulation muss die Anzeige von Feldumrahmungen unterstützen. Dies ist gewöhnlich bei Terminals bzw. Terminal-Emulation der Fall, die Doppel-Byte-Zeichensätze (Double Byte Character Sets/DBCS) unterstützen. Andernfalls werden die für eine Maske erzeugten Feldumrahmungen zur Ausführungszeit nicht angezeigt.

Verwandtes Thema:

Informationen zu den in Natural implementierten Merkmalen zur Unterstützung von Doppel-Byte-Zeichensätzen siehe *Double-Byte Character Sets* in der *Operations*-Dokumentation.

Trennzeichen-Klasse D definieren

Bei jeder Maske, die mit dem Outline-Editor bearbeitet werden soll, muss die Trennzeichen-Klasse `D` in den Maskeneinstellungen gesetzt sein. Falls diese Trennzeichen-Klasse nicht gesetzt ist, können Sie den Outline-Editor nicht öffnen. Stattdessen wird eine entsprechende Fehlermeldung angezeigt.

Für gewöhnlich wird die Trennzeichen-Klasse `D` automatisch hinzugefügt, wenn Sie eine neue Maske in Natural erstellen. Siehe Beschreibung im Abschnitt *Trennzeichen-Klasse D*. Bei einer schon existierenden, mit Natural erstellten Maske müssen Sie jedoch die Trennzeichen-Klasse wie in der folgenden Anleitung beschrieben hinzufügen.

Um die Trennzeichen-Klasse D zu definieren:

1 Rufen Sie den Masken-Editor unter Angabe einer Maske auf, die als Source-Objekt in einer Library in einer Natural-Systemdatei gespeichert ist.

Der Editierbereich des Masken-Editors erscheint. Er enthält die angegebene Maske.

2 Drücken Sie PF2 (Mset), um die Standardeinstellungen für die Maske zu ändern.

Der Bildschirm **Define Map Settings** wird angezeigt.

3 Fügen Sie im Bereich **Delimiters**, die Trennzeichen-Klasse `D` hinzu (oder ersetzen Sie das bereits vorhandene Trennzeichen).

Beispiel:

```
Delimiters
----------------
Cls Att CD  Del
 T   D      BLANK
 T   I      ?
 A   D      _
 A   I      )
 A   N      ^
 M   D      &
 M   I      :
 O   D      +
 O   I      (
 D   I      #
```

4 Rufen Sie das Menü **Edit Map** des Masken-Editors auf und speichern Sie die Maske.

Weitere Informationen siehe Abschnitt *Trennzeichen — Delimiters*.

Editor für Feldumrahmungen (Outline-Editor) aufrufen

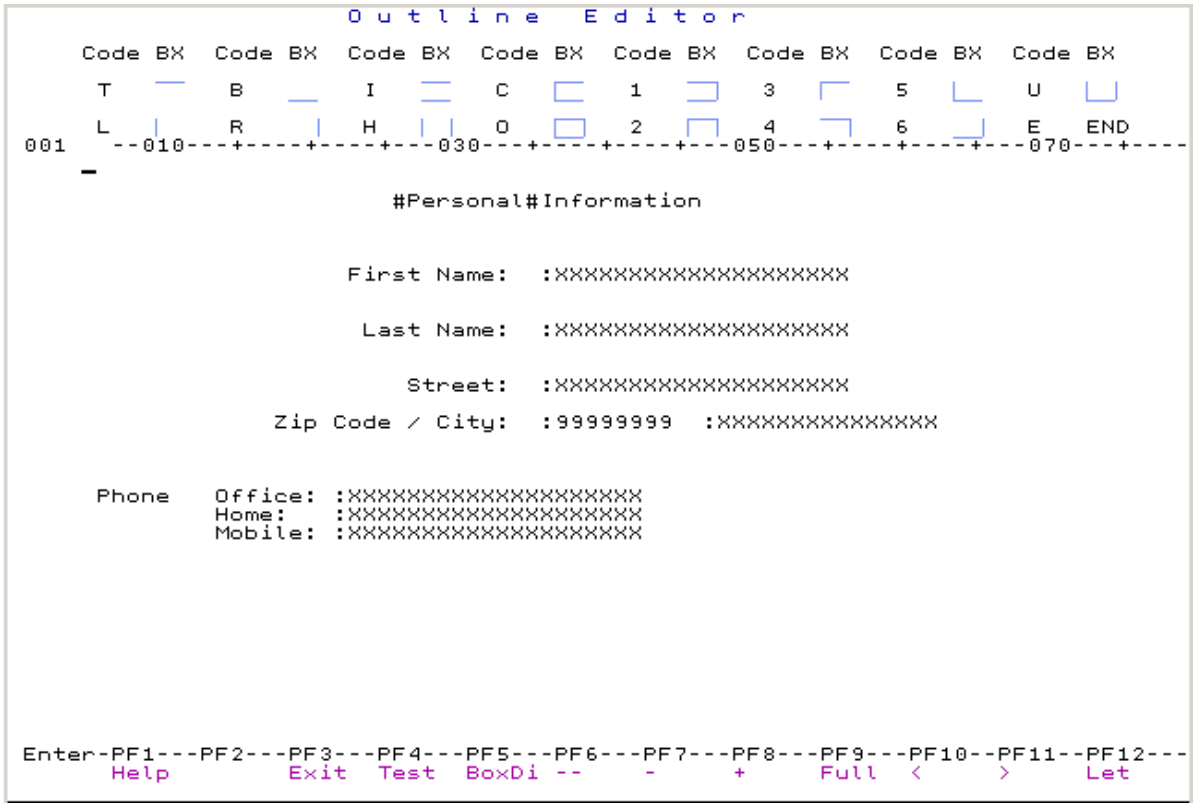
Dieser Abschnitt beschreibt, wie Sie den Editor zum Erstellen von Feldumrahmungen aufrufen.

Um den Editor aufzurufen:

1 Rufen Sie das Menü **Edit Map** des Masken-Editors auf.

2 Geben Sie den Funktionscode `0` für **Outline Editor** und den Namen der zu bearbeitenden Maske ein. Die Masken-Definition muss entweder im aktuellen Quellcode-Editierbereich enthalten oder als Source-Objekt in einer Library in einer Natural-Systemdatei enthalten sein.

Der **Outline Editor**-Bildschirm wird im Split-Screen-Modus angezeigt. Er enthält die angegebene Maske (Beispiel):



Der obere Bereich des Outline-Editor-Bildschirms zeigt alle gültigen Box-Codes (z.B. **O**), Symbole für die damit erzeugbaren Feldumrahmungen (z.B. ein Rechteck für den Box-Code **O**) und die Ende-Markierung **E** (**END**). Im Vollbild-Modus wird diese Übersicht nicht angezeigt.

Der untere Bereich des Outline-Editor-Bildschirms enthält den Editierbereich mit der Masken-Definition der angegebenen Maske. Die Zeichen, die zur Darstellung der Datenfelder (z.B. `X` oder `9`) dienen, sind identisch mit denen, die im Editierbereich des Masken-Editors verwendet werden. Weitere Informationen siehe *Maskenfelder definieren*.

Box-Codes für Feldumrahmungen benutzen

Ein Box-Code ist ein einzelnes Zeichen, das einen Einzelwert des Session-Parameters `BX` oder eine Kombination von Werten repräsentiert. Beispiel: Der Box-Code `O` steht für die `BX`-Parametereinstellung `TBLR`. Weitere Informationen zu möglichen Parameterwerten siehe ***Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen***.

Im Folgenden werden die Regeln und Bedingungen beschrieben, die bei der Benutzung von Box-Codes gelten. Die erwähnten Beispiele beziehen sich auf den Abschnitt ***Beispiel für Feldumrahmungen — Box-Coding***.

Sie können Box-Codes (beliebig als Groß- oder als Kleinbuchstaben) in folgenden Fällen eingeben:

- Bei einem Datenfeld.
- Bei einem Text-Feld, das mit Trennzeichen-Klasse `D` angelegt wurde (Trennzeichen-Klasse `T` ist *nicht* zulässig).
- Bei einem Leerzeichen oder einer Folge von Leerzeichen.

Dieser Abschnitt behandelt folgende Themen:

- Box-Codes bei Daten-Feldern und Text-Feldern eingeben
- Box-Codes bei Leerzeichen eingeben
- Automatische Fortsetzung der Umrahmung
- Box-Codes ändern oder löschen

Box-Codes bei Daten-Feldern und Text-Feldern eingeben

Bei einem Datenfeld oder einem Text-Feld, das mit Trennzeichen-Klasse `D` angelegt wurde, können Sie einen Box-Code bei der Attribut-Position des Feldes eingeben, um das Sie eine Umrandung erzeugen möchten. Die Attribut-Position befindet sich unmittelbar vor dem Feld und ist gekennzeichnet durch eines der folgenden Merkmale:

- Das Trennzeichen („Delimiter“), das dem Feld mit dem Masken-Editor zugewiesen wurde.

  Beispiel: Das Text-Feld `Information`.
- Ein Box-Code, der schon für das Feld eingegeben worden ist. Der Outline-Editor verbirgt das Trennzeichen und zeigt stattdessen den Box-Code.

  Beispiel: Das Text-Feld `Personal`.

Bei einem Datenfeld, das als Array definiert worden ist, können Sie einen Box-Code zu jeder einzelnen Ausprägung dieses Array eingeben.

Beispiel:

`Office:`, `Home:` und `Mobile:` sind drei Ausprägungen eines eindimensionalen Array `(A20/1:3)`. Wenn Sie möchten, dass der Box-Code für alle Ausprägungen dieses Array gilt, können Sie den `BX`-Parameter mit der Funktion zur erweiterten Feld-Bearbeitung des Masken-Editors ändern.

Box-Codes bei Leerzeichen eingeben

Eine Umrahmung können Sie auch um ein Leerzeichen oder eine Folge von Leerzeichen erzeugen. Dadurch haben Sie die Möglichkeit, leere Maskenbereiche zu nutzen, um Umrahmungen zu entwerfen.

Bei einem Leerzeichen oder einer Folge von Leerzeichen geben Sie den Box-Code an einer leeren Stelle einer Zeile ein.

Die Länge der Umrahmung wird durch die Anzahl der Leerzeichen bestimmt, für die der verwendete Box-Code gilt, und durch die Ende-Markierung (siehe ***Ende-Markierungen für Feldumrahmungen benutzen***):

- Ist die Ende-Markierung gesetzt, dann wird die Umrahmung um die Anzahl an Zeichen ab dem Box-Code bis zur Ende-Markierung gezeichnet.

  Beispiele: Die Box-Codes **2** und **U** im leeren Bereich neben dem Datenfeld `Last Name:`.

- Ist keine Ende-Markierung gesetzt, dann wird die Umrahmung entweder um die Anzahl an Zeichen ab dem Box-Code bis zum nächsten Feld oder bis zum Zeilenende gezeichnet.

  Ausnahmen: Die Box-Codes **L** und **R** erzeugen immer nur Umrahmungen um ein einzelnes Leerzeichen, egal ob die Ende-Markierung gesetzt oder nicht gesetzt ist.

  Beispiel: Der Box-Code **L** zwischen den Text-Feldern `Phone` und `Office:`.

Eine Umrahmung, die bei einer Folge von Leerzeichen erzeugt wird, wird durch eine Anzahl von Punkten (.) angezeigt. Siehe ***Beispiel für Feldumrahmungen (Box-Coding) nach Drücken von Enter***. Bei einer Umrahmung, bei der unmittelbar auf den Box-Code die Ende-Markierung folgt (Beispiel: `OE`) oder bei den Box-Codes **L** und **R** werden keine Punkte angezeigt.

Eine Umrahmung, die bei einem Leerzeichen oder einer Folge von Leerzeichen erzeugt wird, wird als leeres Text-Feld mit der Trennzeichenklasse `D` in die Masken-Definition hinein generiert.

Um die Länge einer Umrahmung zu verringern, können Sie einen der Punkte (.) durch die Ende-Markierung oder einen Box-Code ersetzen (siehe auch ***Um die Ende-Markierung zu platzieren***). Das Feld endet dann an dem Punkt, den Sie ersetzt haben.

Um ein leeres Text-Feld zu bearbeiten, können Sie die Funktion zur erweiterten Feld-Bearbeitung des Masken-Editors benutzen.

Automatische Fortsetzung der Umrahmung

Ein Box-Code, der eine obere und/oder untere Linie, aber keine rechte senkrechte Linie aufweist (die Box-Codes **T**, **B**, **I**, **C**, **3** und **5**) bleibt für die anschließenden Felder in dieser Zeile gültig. Die Umrahmung endet dann entweder beim nächsten Box-Code, dem nächsten Text-Feld mit Trennzeichenklasse `T` oder am Zeilenende.

Beispiele:

Die Text-Felder `Personal` und `Information` und die Datenfelder neben dem Feld `City:`.

Wenn Sie einen der oben genannten Box-Codes eingeben, platziert der Outline-Editor automatisch generierte Box-Codes vor alle anschließenden Felder, für die dieser Box-Code gilt. Die Reihenfolge der automatisch generierten Box-Codes ist aufgeführt in der Tabellenspalte ***Automatisch generierte Box-Codes*** im Abschnitt *List der Box-Codes und BX-Parameterwerte für Feldumrahmungen*. Die automatische Box-Code-Generierung erfolgt, wenn Sie zu Beginn einen Box-Code eingegeben haben. Automatisch generierte Box Code können Sie ersetzen.

Box-Codes ändern oder löschen

Sie können einzelne oder mehrere Box-Codes ändern, indem Sie jeden Box-Code einzeln ersetzen.

Sie können einzelne oder mehrere Box-Codes löschen, indem Sie entweder jeden Box-Code einzeln durch ein Leerzeichen ersetzen oder indem Sie eines der Zeilenkommandos benutzen, die im Abschnitt ***Zeilenkommandos und PF-Tasten*** beschrieben werden.

Grundsätzlich gilt, dass Sie nur den Box-Code bei einem Feld und *nicht* das Feld selbst löschen können. Eine Feld-Definition können Sie nur mit dem Masken-Editor löschen. Ausnahme: Wenn Sie den Box-Code bei einem leeren Feld löschen, dann löschen Sie außerdem auch das entsprechende Feld in der Masken-Definition.

Ende-Markierungen für Feldumrahmungen benutzen

Die Ende-Markierung **E** (**END**) bestimmt die Länge einer Umrahmung, die bei einem einzelnen Leerzeichen oder einer Folge von Leerzeichen erzeugt wird.

Ausnahmen: Die Box-Codes **L** und **R** erzeugen immer nur Umrahmungen bei einem einzelnen Leerzeichen. Sie können sie ohne die Ende-Markierung eingeben. Beispiel: Der Box-Code **L** zwischen den Text-Feldern `Phone` und `Office:`.

Bei der Eingabe der Ende-Markierung wird *nicht* zwischen Groß- und Kleinschreibung unterschieden.

Die Ende-Markierung wird *nicht* als `BX`-Parameterwert in den Quellcode der Maske hinein generiert.

Die Ende-Markierung gilt für den Box-Code und die Zeile, wo sie platziert wird. Es ist nicht zulässig, einen Box-Code in der einen und die Ende-Markierung in einer anderen Zeile zu platzieren, um eine Umrahmung über mehrere Zeilen hinweg zu erweitern.

Um die Ende-Markierung zu platzieren:

1 Geben Sie den Box-Code an einer leeren Stelle in der Zeile ein, wo die Umrahmung beginnen soll, und die Ende-Markierung `E` dort, wo die Umrahmung enden soll.

Beispiel: Die Spalten **035** und **059** in der leeren Zeile über dem Feld `Last Name:` im ***Beispiel für Feldumrahmungen — Box-Code-Einstellungen***.

2 Drücken Sie Enter, um die Eingaben zu bestätigen.

Die von dem Box-Code und der Ende-Markierung eingegrenzten Leerzeichen (im o.g. Beispiel zwischen den Spalten **035** und **059**) werden durch Punkte (.) ersetzt.

Die Ende-Markierung wird nicht mehr im Bildschirm angezeigt.

3 Sie können die Länge einer Umrahmung verringern, indem Sie einen der Punkte (.) mit der Ende-Markierung überschreiben. Beispiel: `U.........E.....`

Die Umrahmung endet dann an dem Punkt, den Sie überschrieben haben.

Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen

In der folgenden Tabelle werden die Box-Codes erläutert, die im Editierbereich des Outline-Editors aufgelistet werden. Außerdem enthält die Tabelle die Abfolge der automatisch generierten Box-Codes (falls relevant bei einem Box-Code) und die äquivalenten Werte des `BX`-Parameters.

Box	Ränder	Box-Codes	BX-Werte	Automatisch generierte Box-Codes	BX-Werte für automatisch generierte Box-Codes
	Oberer waagerechter Rand.	T	T	T	T
	Unterer waagerechter Rand.	B	B	B	B
	Oberer und unterer waagerechter Rand.	I	TB	I	TB
	Oberer und unterer waagerechter und linker senkrechter Rand.	C	LTB	I	TB
	Oberer und unterer waagerechter und rechter senkrechter Rand.	1	TRB		

Box	Ränder	Box-Codes	BX-Werte	Automatisch generierte Box-Codes	BX-Werte für automatisch generierte Box-Codes
	Oberer waagerechter und linker senkrechter Rand.	3	LT	T	T
	Oberer waagerechter und rechter senkrechter Rand.	4	TL		
	Unterer waagerechter und linker senkrechter Rand.	5	LB	B	B
	Unterer waagerechter und linker und rechter senkrechter Rand.	U	LBR		
	Linker senkrechter Rand.	L	L		
	Rechter senkrechter Rand.	R	R		
	Linker und rechter senkrechter Rand.	H	LR		
	Oberer und unterer waagerechter und linker und rechter senkrechter Rand.	O	TBLR		

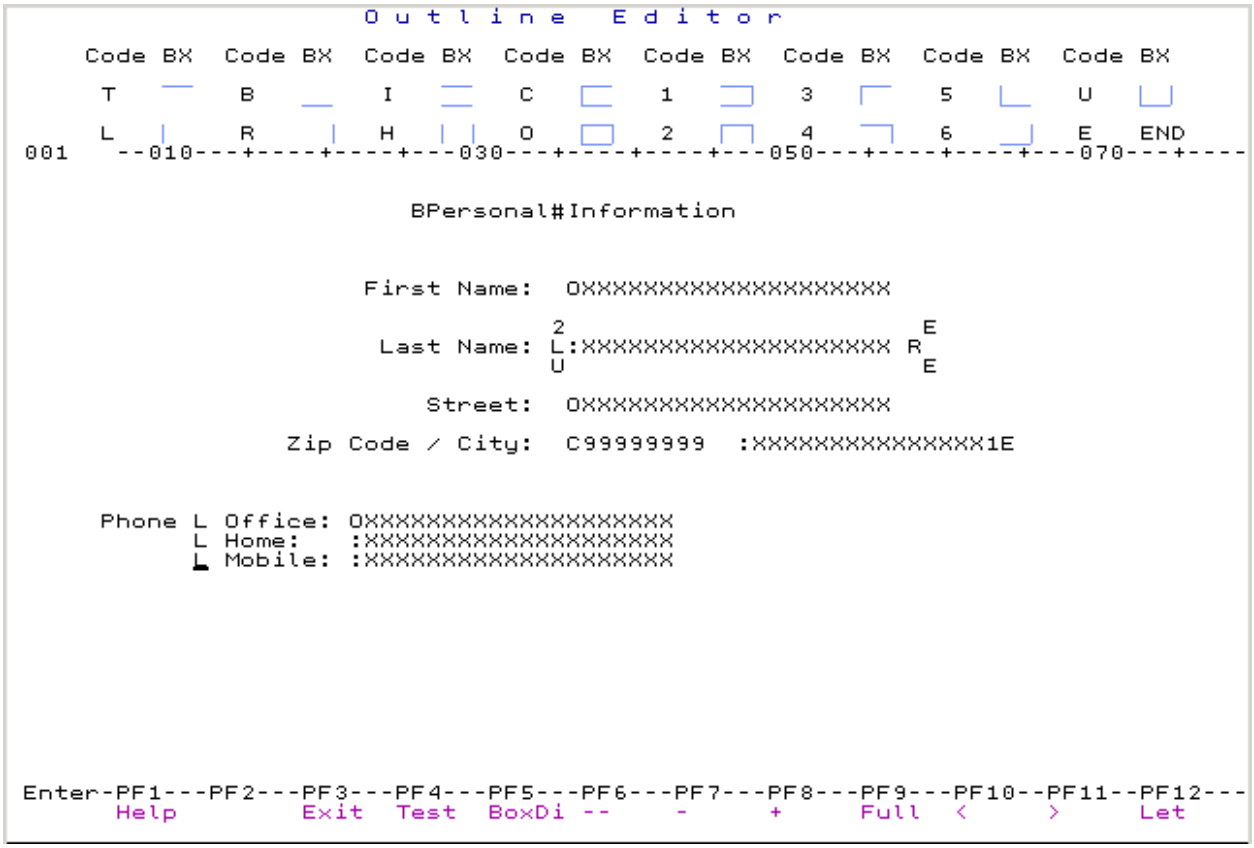
Beispiel für Feldumrahmungen — Box-Coding

Dieser Abschnitt enthält ein Beispiel für Feldumrahmungen, das auf der **Maskendefinition** basiert, die im Abschnitt *Editor für Feldumrahmungen (Outline-Editor) aufrufen* gezeigt wird. Dieses Beispiel ist nicht als Anleitung zum Anlegen von Masken mit Feldumrahmungen gedacht, sondern soll nur veranschaulichen, wie Sie die Box-Codes verwenden können.

- Beispiel für Feldumrahmungen — Box-Code-Einstellungen
- Beispiel für Feldumrahmungen (Box-Coding) nach Drücken von Enter
- Beispiel für Feldumrahmungen — Box-Layout

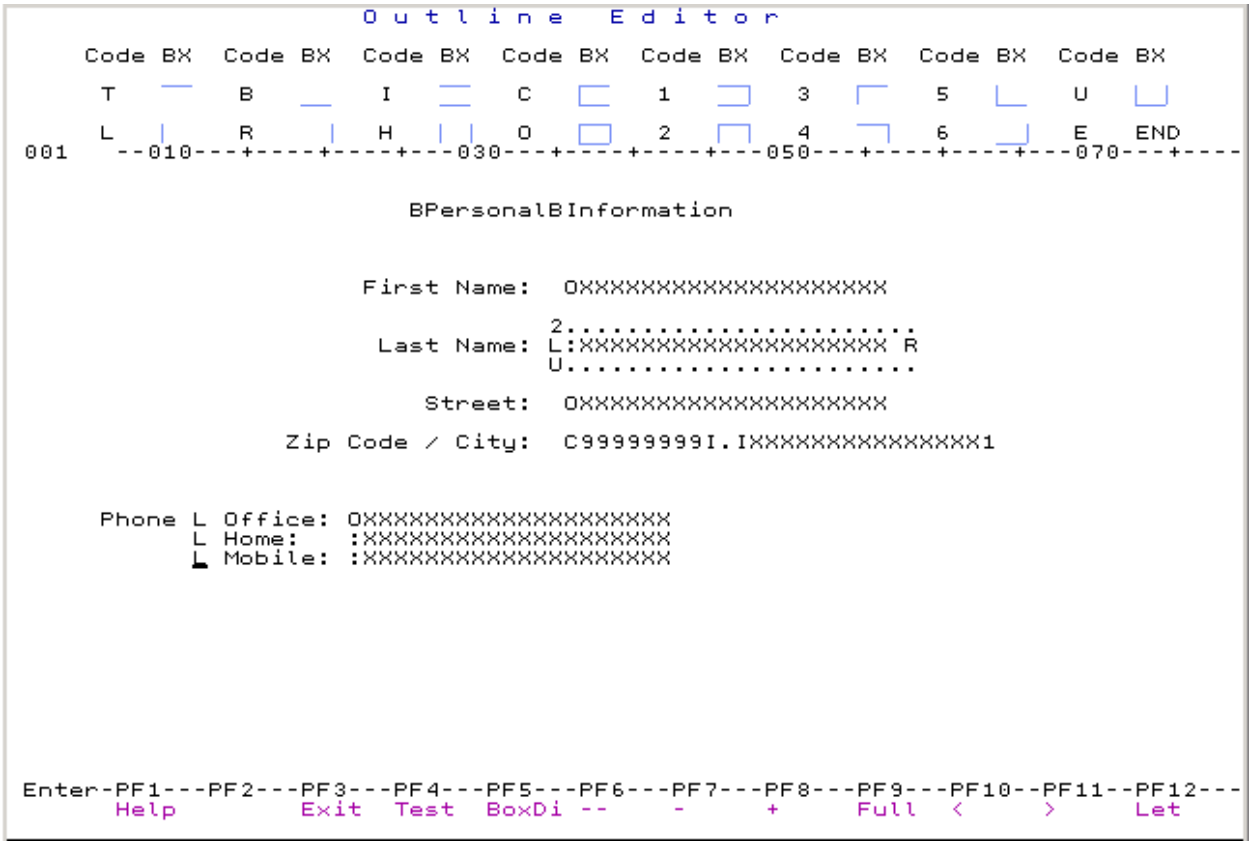
Beispiel für Feldumrahmungen — Box-Code-Einstellungen

Das folgende Beispiel zeigt, wie die Box-Codes im Outline-Editor eingegeben werden können:



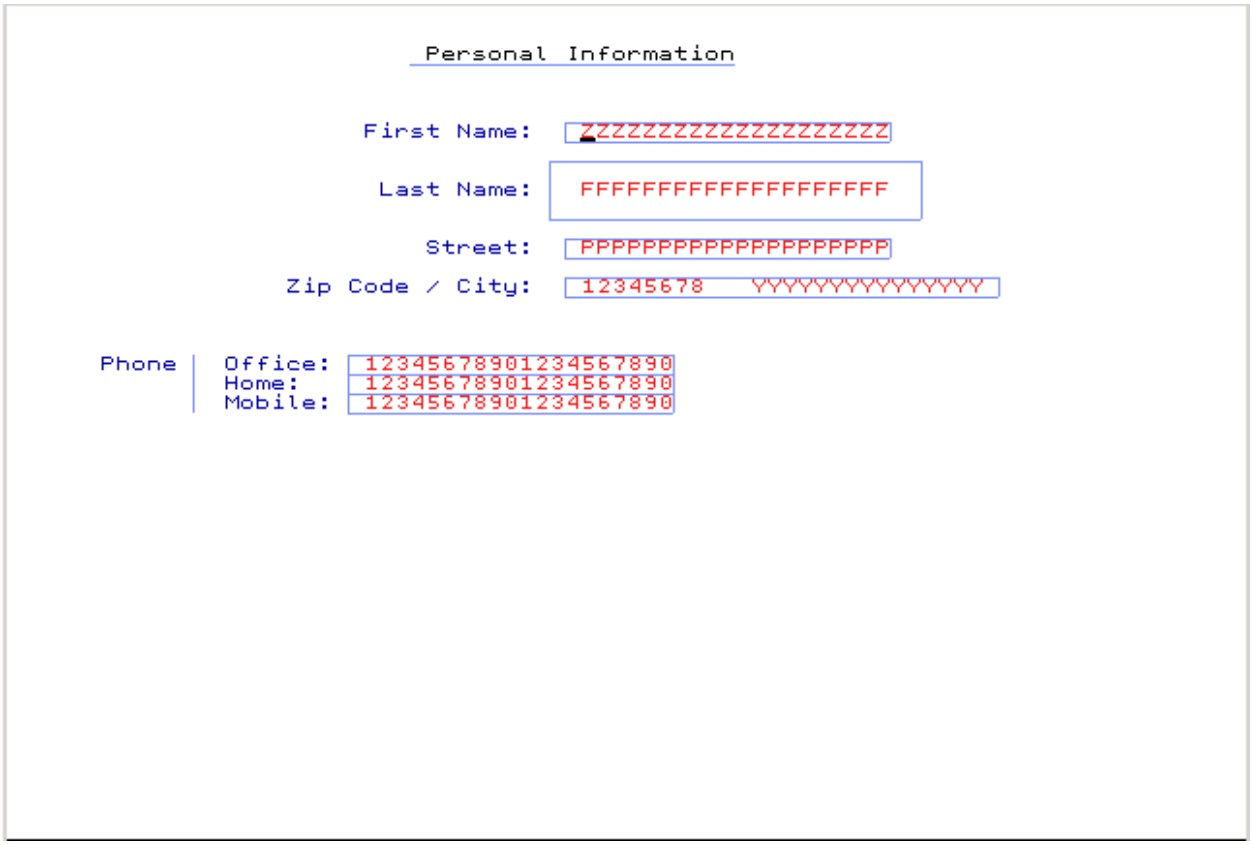
Beispiel für Feldumrahmungen (Box-Coding) nach Drücken von Enter

Das folgende Beispiel zeigt, wie das Box-Coding aussehen kann, nachdem Sie Ihre Box-Code-Eingaben mit Enter bestätigt haben:



Beispiel für Feldumrahmungen — Box-Layout

Das folgende Beispiel zeigt das Layout der Feldumrahmungen in einer Maske, wenn diese durch ein Programm oder mittels der Test-Funktion (PF4) aufgerufen wird:



Zeilenkommandos und PF-Tasten

Die **Zeilenkommandos** und PF-Tasten, die beim Outline-Editor zur Verfügung stehen, können Sie benutzen, um Box-Codes und leere Text-Felder zu löschen, im Editerbereich des Outline-Editors zu navigieren, zwischen den Editerbereichen (Outline-Editor/Masken-Editor) hin und her zu schalten, die Anzeige-Modi zu ändern und den Outline-Editor zu beenden.

Die Eingabe eines Zeilenkommandos erfolgt an der ersten Stelle einer Zeile. Bei der Eingabe der Ende-Markierung wird *nicht* zwischen Groß- und Kleinschreibung unterschieden.

Zeilenkommando	PF-Taste	Erläuterung
`..D`		Löscht alle Box-Codes und leeren Text-Felder in der Zeile, in der das Kommando eingegeben wurde.
`..D`*nn*		Löscht alle Box-Codes und leeren Text-Felder in der Zeile, in der das Kommando eingegeben wurde, sowie *nn* Zeilen unterhalb dieser Zeile. Dabei steht *nn* für einen Zeilenbereich von 1 bis 99.
`..D*`		Löscht alle Box-Codes und leeren Text-Felder in der Zeile, in der das Kommando eingegeben wurde, sowie *alle* Zeilen unterhalb dieser Zeile.

Zeilenkommando	PF-Taste	Erläuterung
. . X	PF13	Umschalten zwischen dem Editierbereich des Masken-Editors und dem Outline-Editor.
	PF1	**Help**/Hilfe — Anzeige von Hilfe-Informationen zum Outline-Editor und zum Masken-Editor.
. . Q	PF3 **Exit**	Beendet den Outline-Editor. Rückkehr zum **Edit Map**-Menü. Wird PF3 im **Box-Code Display**-Bildschirm oder im Test-Bildschirm gedrückt, erfolgt die Rückkehr zum Editier-Bereich des Outline-Editors.
	PF4 **Test**	Test der Masken-Definition mit dem Box-Layout. Diese Funktion kann auch aus dem Masken-Editor ausgeführt werden. Drücken Sie PF3 oder Enter, um zum Editier-Bereich zurückzukehren.
	PF5 **Box Di**	Umrahmungsanzeige — Zeigt die Box-Code-Einstellungen einer Maske an. Siehe *Bildschirm "Box-Code Display"* weiter unten. Drücken Sie PF3 oder Enter, um zum Editier-Bereich zurückzukehren.
	PF7 **-**	Eine halbe Seite nach oben blättern.
	PF8 **+**	Eine halbe Seite nach unten blättern.
	PF9 **Full**	Umschalten zwischen Vollbild- und Split-Screen-Modus.
	PF10 **<**	Eine halbe Bildschirmseite nach links verschieben, wenn die Maske breiter ist als der Bildschirm.
	PF11 **>**	Eine halbe Bildschirmseite nach rechts verschieben, wenn die Maske breiter ist als der Bildschirm.
	PF12 **Let**	Macht die letzte Maßnahme rückgängig, die im Editier-Bereich *vor* dem Drücken von Enter vorgenommen wurde.

Bildschirm "Box-Code Display"

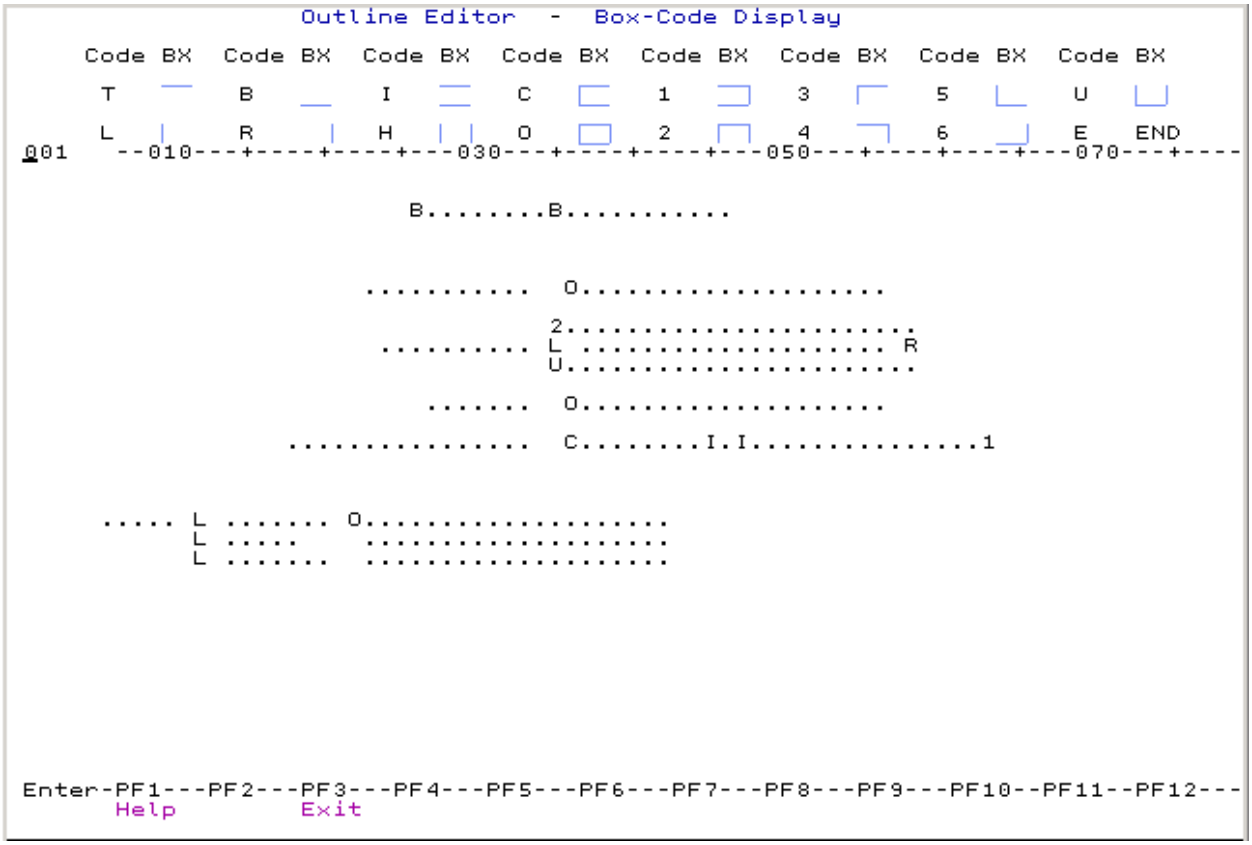
Die Funktion zum Anzeigen der Umrahmung (PF4 Test) ruft den Bildschirm **Box-Code Display** auf, der Ihnen die Orientierung erleichtern und dabei helfen soll, die die Box-Codes von Text mit identischen Zeichen zu unterscheiden.

Der **Box-Code Display**-Bildschirm zeigt die gesetzten Box-Codes. Die entsprechenden Feldpositionen werden durch ein Anzahl von Punkten (.) angegeben. Das als Ende-Markierung eingegebene E wird auf diesem Bildschirm nicht gezeigt.

Im **Box-Code Display**-Bildschirm sind keine Änderungen möglich.

Beispiel — Box-Code Display:

Das folgende Beispiel zeigt die Box-Code-Einstellungen einer Maske nach Drücken von PF4. Es basiert auf dem *Beispiel für Feldumrahmungen — Box-Code-Einstellungen*.


Outline Editor - Box-Code Display
Code BX Code BX Code BX Code BX Code BX Code BX Code BX Code BX
T B I C 1 3 5 U
L R H O 2 4 6 E END
001 --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
B.....B.....
..... O.....
2.....
..... L ..... R
U.....
..... O.....
..... C.....I.I.....1
..... L ..... O.....
L ....
L ....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Exit

21 Nachträgliches Zuweisen von Feldern

Einem Feld, das zuvor im Bildschirm-Layout einer Maske definiert worden ist, können Sie den Feldnamen und Feldattribute einer DDM-Felddefinition oder einer `DEFINE DATA`-Statement-Definition zuweisen.

Anmerkung: Die doppelte Vergabe von Feldnamen ist nur bei Feldern zulässig, die als reine Ausgabefelder definiert sind.

Ein Maskenfeld, das aus einem DDM erstellt wurde, kann unter Verwendung der entsprechenden DDM-Felddefinition aus diesem DDM oder aus einer `DEFINE DATA`-Statement-Definition redefiniert werden.

Die nachträgliche Zuweisung erfolgt durch Eingabe der Nummer (oder des Buchstabens), der einer DDM-Felddefinition zugewiesen wurde. Weitere Informationen siehe *Datendefinitionen auswählen*.

Eine nachträgliche Zuweisung kann nur dann erfolgen, wenn das Format des Layouts mit der Felddefinition übereinstimmt. Die Formate `N` und `P` werden dabei als identische numerische Formate betrachtet.

Bei Arrays einer View (in einer DDM-Felddefinition oder einer `DEFINE DATA`-Statement-Definition) kann keine nachträgliche Zuweisung erfolgen, wenn eine oder mehrere Dimensionen dieses Arrays kleiner ist oder sind als die Dimensionen des Array in dem Layout.

Falls ein Längenkonflikt auftritt, wird ein **AL**-/**FL**-/**NL**-Attribut generiert, um die Felddefinition per Verkürzung oder Erweiterung auf die Layout-Definition abzubilden. Die Daten werden verkürzt, wenn **AL**/**FL**/**NL** angegeben wird.

22 Array- und Tabellen-Definitionen

- Array-Definition 268
- Array-Tabellen-Definition 272

Die Funktion zur Array-Definition wird verwendet, um die Ausprägungen und das Layout eines Array zu definieren (X-Arrays werden nicht unterstützt).

Eingeleitet wird die Array-Definition durch das Feld-Kommando `.A` oder durch Absetzen des Zeilenkommandos `..E` und durch anschließendes Markieren des betreffenden Feldes mit dem Funktionscode `A`.

Die Funktion zur Tabellendefinition wird verwendet, um die Ausprägungen und das Layout von mehr als einem Array gleichzeitig zu definieren. Die Arrays müssen in derselben Maskenzeile beginnen.

Aufgerufen wird die Tabellendefinition mit dem Zeilenkommando `..A`.

Array-Definition

Im oberen Bereich des folgenden Beispiel-Bildschirms kann die Array-Definition vorgenommen werden:

```
Name #001                          Top Dim 1__________ 1__________ 1__________
------------------------------------------------------------------------------
Dimensions                      Occurrences    Starting from     Spacing
0 . Index vertical              1__            ___________       0   Lines
0 . Index horizontal            1__            ___________       1   Columns
0 . Index (h/v) V               1__            ___________       0   Cls/Ls

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
   .AXXXXX


                Please enter starting name .AXXXXXXXXXXXXXXXXXXX









 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help  Mset  Exit              --    -     +           <     >     Let
```

Folgende Angaben sind möglich:

Feld	Erläuterung
Top Dim	Zeigt die obere Dimension eines Array, d.h. die höchste Ausprägung (von links nach rechts) in der ersten, zweiten und dritten Dimension. Wenn zum Definieren des Masken-Array eine Felddefinition in einem Programm verwendet wird, werden, so wie im Programm definiert, die oberen Begrenzungen dieses Feldes (Benutzervariable oder Datenbankfeld) verwendet. Diese können im Array-Definitionsbildschirm nicht überschrieben werden. Siehe auch Hinweis in *Top Dim ändern* weiter unten. Wenn Sie eine Masken-Array-Definition aus einer Datendefinition in einem anderen Natural-Objekt wählen, dürfen die Dimensionen des Masken-Array nicht die Dimensionen überschreiten, die in diesem Feld gezeigt werden. Wenn Sie keine Masken-Array-Definition aus einer Datendefinition wählen, dürfen die Dimensionen des Masken-Array nicht die Dimensionen überschreiten, die in dem Natural-Programm definiert sind.
Dimensions	Ein Array kann bis zu drei Dimensionen haben. Die Reihenfolge, in der die Dimensionen das Array im Masken-Layout abgebildet werden, wird durch die Werte bestimmt, die links von den **Index**-Operanden eingegeben werden.
Occurrences	Die Anzahl der zu einer Dimension zu definierenden Ausprägungen.
Starting from	Der Start-Indexwert für eine Dimension. Sie können einen numerischen Wert benutzen, oder Sie können einen Variablennamen verwenden, um anzugeben, dass der tatsächliche Wert in dem Natural-Programm geliefert wird, das die Masken-Definition aufruft. Falls die Variable nicht anderweitig als ein Feld in der Maske definiert ist, wird davon ausgegangen, dass es Natural-Datenformat/Länge N7 hat. Falls dem so ist, können Sie sie nach Drücken von PF9 im Bildschirm **Field and Parameter Definition** bearbeiten. **Anmerkung:** Wenn Sie einen **Starting from**-Wert aus einem Array entfernen, hat dies zur Folge, dass auch die Variable aus der Maske entfernt wird, es sei denn, sie ist ein Maskenfeld oder sie ist mit einem anderen Maskenfeld als **Starting from**-Wert oder als Hilfe-Parameter verbunden. Um **Starting from**-Werte zu bearbeiten, müssen Sie PF9 im Bildschirm **Field and Variable Definitions - Summary** drücken.
Spacing	Die Anzahl der Leerzeilen (bei vertikalen Dimensionen) oder Leerspalten (bei horizontalen Dimensionen), die zwischen jeder Ausprägung einer Dimension eingefügt werden soll.

Top Dim ändern

Wenn Sie die obere Dimension eines Array ändern müssen, beispielsweise um mehr Zeilen auf einer Maske zuzulassen, können Sie folgendermaßen vorgehen:

1. Benennen Sie jedes Feld, dessen Größe geändert werden soll, so um, dass es in keinem der Quellcodes in der aktuellen Library eine solche Deklaration gibt. Überschreiben Sie z.B. das erste Zeichen mit X.
2. Aktualisieren Sie nun jede obere Dimension: Da es keine Beziehung mehr zu einer Definition gibt, sollte dieses Feld für Änderungen zugänglich sein.

3. Benennen Sie jedes Array-Feld wieder so um, wie es vor Schritt 1 war. D.h. machen Sie die Umbenennung jedes Array-Feldes rückgängig.
4. Vergessen Sie nicht, die Programme anzupassen, die die Größen dieser Felder definieren.

Beispiele für Array-Definitionen

Beispiel 1:

Ein eindimensionales Array, bestehend aus 10 vertikalen Ausprägungen mit zwei Leerzeilen, die zwischen den benachbarten Ausprägungen eingefügt werden sollen.

```
Name #001                          Top Dim 10_________ 1__________ 1__________
-------------------------------------------------------------------------------
Dimensions                      Occurrences    Starting from     Spacing
1 . Index vertical              10_            _________         2   Lines
0 . Index horizontal            1__            _________         1   Columns
0 . Index (h/v) V               1__            _________         0   Cls/Ls
```

Beispiel 2:

Wie bei *Beispiel 1*, jedoch soll das Array horizontal ausgerichtet sein.

```
Name #001                          Top Dim 10_________ 1__________ 1__________
-------------------------------------------------------------------------------
Dimensions                      Occurrences    Starting from     Spacing
0 . Index vertical              1__            _________         0   Lines
1 . Index horizontal            10_            _________         1   Columns
0 . Index (h/v) V               1__            _________         0   Cls/Ls
```

Beispiel 3:

Ein zweidimensionales Array. Die erste Dimension besteht aus 10 vertikalen Ausprägungen mit einer Leerzeile zwischen den benachbarten Ausprägungen. Die zweite Dimension besteht aus 5 horizontalen Ausprägungen mit zwei Leerspalten zwischen den benachbarten Ausprägungen.

```
Name #001                          Top Dim 10_________ 5__________ 1__________
-------------------------------------------------------------------------------
Dimensions                      Occurrences    Starting from     Spacing
1 . Index vertical              10_            _________         1   Lines
2 . Index horizontal            5__            _________         2   Columns
0 . Index (h/v) V               1__            _________         0   Cls/Ls
```

Beispiel 4:

Wie bei *Beispiel 3*, jedoch ist die Reihenfolge der Dimensionen umgekehrt.

```
Name #001                          Top Dim 5_________ 10_________ 1_________
-------------------------------------------------------------------------------
Dimensions                       Occurrences    Starting from     Spacing
2 . Index vertical               10_            _________         1   Lines
1 . Index horizontal             5__            _________         2   Columns
0 . Index (h/v) V                1__            _________         0   Cls/Ls
```

Beispiel 5:

Ein dreidimensionales Array. Die erste Dimension besteht aus 3 vertikalen Ausprägungen mit einer Leerzeile zwischen den benachbarten Ausprägungen. Die zweite Dimension besteht aus 5 horizontalen Ausprägungen mit zwei Leerspalten zwischen den benachbarten Ausprägungen. Die dritte Dimension besteht aus zwei Ausprägungen, die innerhalb jeder Ausprägung der ersten Dimension vertikal erweitert sind.

```
Name #001                          Top Dim 3_________ 5_________ 2_________
-------------------------------------------------------------------------------
Dimensions                       Occurrences    Starting from     Spacing
1 . Index vertical               3__            _________         1   Lines
2 . Index horizontal             5__            _________         2   Columns
3 . Index (h/v) V                2__            _________         0   Cls/Ls
```

Beispiel 6:

Ein Beispiel für die Verwendung von **Starting from**. Die erste Dimension besteht aus 10 vertikalen Ausprägungen, die ab dem Index `I` beginnen. `I` ist im Masken-Editor standardmäßig mit Natural-Datenformat/Länge N7 definiert. Die zweite Dimension besteht aus 5 horizontalen Ausprägungen, die ab dem Index `I` beginnen.

```
Name #001                          Top Dim 10________ 5_________ 1_________
-------------------------------------------------------------------------------
Dimensions                       Occurrences    Starting from     Spacing
1 . Index vertical               10_            I________         1   Lines
2 . Index horizontal             5__            3________         2   Columns
0 . Index (h/v) V                1__            _________         0   Cls/Ls
```

Beispiel 7:

Ein Beispiel, wie man aus einem eindimensionalen ein zweidimensionales Array macht. Das Array besteht aus 40 Elementen. Angezeigt wird es in zwei Spalten mit jeweils 20 Zeilen. Das wird dadurch erreicht, dass man `0` als horizontalen Index angibt.

```
Name #001                          Top Dim 40_________ 1__________ 1__________
------------------------------------------------------------------------------
Dimensions                      Occurrences    Starting from     Spacing
1 . Index vertical              20_            _________         0   Lines
0 . Index horizontal            2__            _________         10  Columns
0 . Index (h/v) V               1__            _________         0   Cls/Ls
```

Array-Tabellen-Definition

Eine Tabelle mit einem oder mehreren Arrays, die alle in derselben Maskenzeile beginnen, können Sie erstellen, indem Sie das Zeilenkommando `..A` benutzen. Nach Eingabe des Kommandos wird folgender Bildschirm angezeigt (Beispiel):

```
14:41:47               ***** NATURAL MAP EDITOR *****               2007-10-22
                         - Array Table Definition -

Main   Index:  Vert. Occur.   1    Starting from ____________ Spacing 0   Lines
Second Index:  Direction(H/V) V                  ____________         0   Cls/Ls
Third  Index:  Direction(H/V) V                  ____________         0   Cls/Ls
------------------------------------------------------------------------------
Name of Variable         Col           Dimension Size            Order 2.  3.
(truncated)              Pos Ind1        Ind2        Ind3        M S T Occ Occ
------------------------------------------------------------------------------
#001                       2 1           1           1
#002                      25 1           1           1









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

Vorsicht: Wenn Sie das Zeilenkommando `..A` bei Arrays anwenden, die nicht mit einem `..A`-Kommando, sondern mit einem `.A`-Kommando definiert wurden, kann das zu einer Veränderung oder sogar zu einer Zerstörung dieser Arrays führen.

Der obige Beispiel-Bildschirm enthält folgende Felder:

Feld	Erläuterung
Main Index	Die Anzahl der vertikalen Ausprägungen, die Start-Position und die Anzahl an Zeilen, die zwischen den Ausprägungen der Dimensionen übersprungen werden sollen.
Second Index	Die Richtung (horizontal oder vertikal), die Start-Position und die Anzahl an Zeilen/Spalten, die zwischen den Ausprägungen der Dimensionen übersprungen werden sollen. Die zweite Dimension gilt nur, wenn eines der Arrays mehr als eine Dimension hat. In diesem Fall kann die zweite Dimension entweder horizontal oder vertikal angezeigt werden. Bei horizontaler Anzeige muss in der Zeile genügend Platz für alle gewählten Ausprägungen vorhanden sein. Bei vertikaler Anzeige müssen in der Maske genügend Zeilen vorhanden sein, um die Anzeige der Hauptdimension mal der zweiten Dimension, einschließlich der Zeilenabstände, zu ermöglichen.
Third Index	Die Richtung (horizontal oder vertikal), die Start-Position und die Anzahl an Zeilen/Spalten, die zwischen den Ausprägungen der Dimensionen übersprungen werden sollen. Die dritte Dimension gilt nur, wenn eines der Arrays mehr als zwei Dimensionen hat. In diesem Fall kann die dritte Dimension entweder horizontal oder vertikal angezeigt werden. Bei horizontaler Anzeige muss in der Zeile genügend Platz für alle gewählten Ausprägungen vorhanden sein. Bei vertikaler Anzeige müssen in der Maske genügend Zeilen vorhanden sein, um die Anzeige der Hauptdimension mal der dritten Dimension, einschließlich der Zeilenabstände, zu ermöglichen.
Name of Variable	Alle Namen von Feld-Arrays, die in der Tabelle enthalten sind, werden hier angezeigt.
Col Pos	Die Spaltenposition, in der sich das Feld befindet. Wird nur zur Information angezeigt.
Dimension Size	Die oberen Begrenzungen **Ind1**, **Ind2** und **Ind3** eines Array. Die Dimensionen des in der Maske definierten Array dürfen nicht größer sein als die Dimensionen des entsprechenden Array, das in dem Natural-Objekt definiert ist, das die Maske aufruft.
Order	Die Reihenfolge, in der die Dimensionen definiert werden sollen: **M**, **S** und **T** entsprechen Main (Hauptdimension), Second (zweite) und Third (dritte Dimension).
2. Occ.	Die Anzahl der Ausprägungen, die für den zweiten Index definiert werden sollen.
3. Occ.	Die Anzahl der Ausprägungen, die für den dritten Index definiert werden sollen.

Beispiel einer Tabellen-Definition

Dies ist ein Beispiel für die Definition von Maskenfeldern, die der folgenden Programmdefinition entsprechen:

```
DEFINE DATA
  1 ARRAY1 (A3/1:10)
  1 ARRAY2 (A5/1:10,1:2)
  1 ARRAY3 (A7/1:10,1:2,1:3)
END-DEFINE
```

Tabellen-Definition:

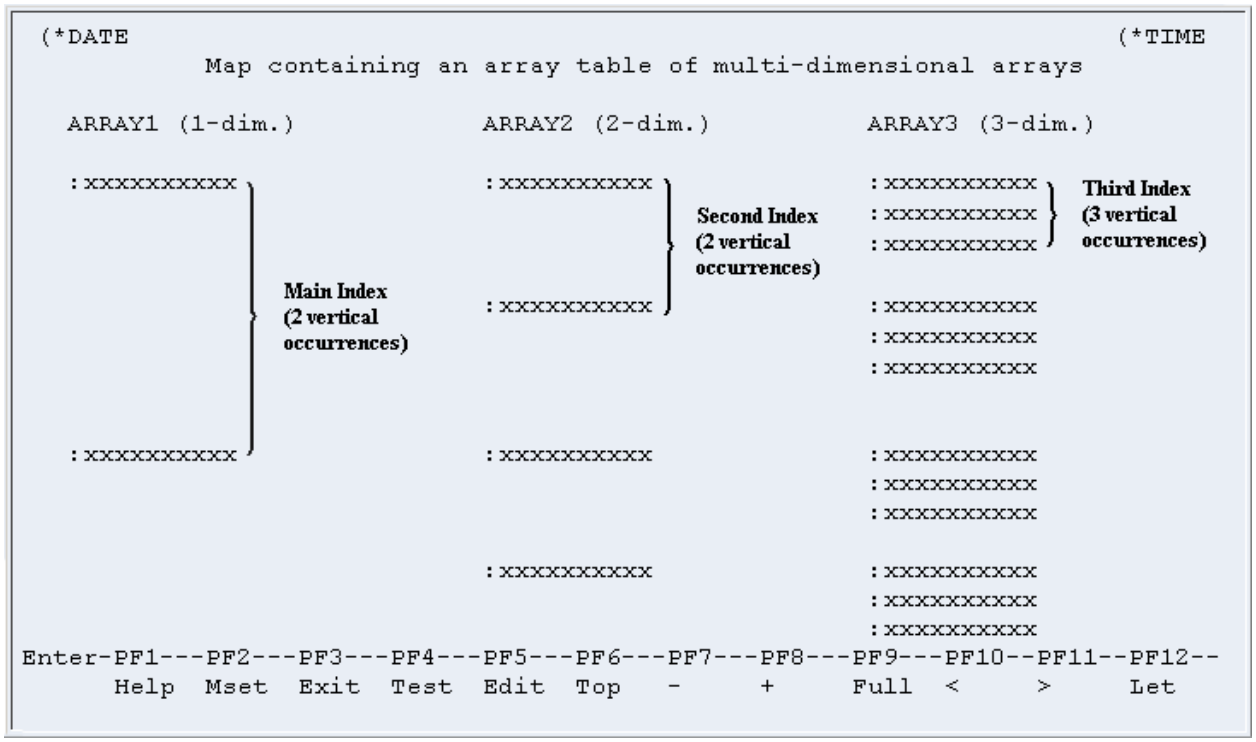
```
14:41:47               ***** NATURAL MAP EDITOR *****               2006-07-24
                         - Array Table Definition -

Main   Index:  Vert. Occur.   1    Starting from ____________ Spacing 0   Lines
Second Index:  Direction(H/V) V                 ____________         0   Cls/Ls
Third  Index:  Direction(H/V) V                 ____________         0   Cls/Ls
-------------------------------------------------------------------------------
Name of Variable         Col           Dimension Size            Order 2.  3.
(truncated)              Pos Ind1        Ind2        Ind3        M S T Occ Occ
-------------------------------------------------------------------------------
  ARRAY1                  3  10          1           1           1
  ARRAY2                 32  10          2           1           1 2    2
  ARRAY3                 58  10          2           3           1 2 3  2  3









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

ARRAY1 ist ein eindimensionales Array mit 10 Ausprägungen. Die ersten zwei Ausprägungen sind in der Tabelle erweitert.

ARRAY2 ist zweidimensionales Array. Der erste Index besteht aus 10 Ausprägungen, der zweite Index aus zwei Ausprägungen. Die ersten zwei Ausprägungen des ersten Index und beide Ausprägungen des zweiten Index sind in der Tabelle erweitert.

ARRAY3 ist dreidimensionales Array. Der erste Index besteht aus 10 Ausprägungen, der zweite Index aus zwei Ausprägungen, der dritte Index aus drei Ausprägungen. Die ersten zwei Ausprägungen des ersten Index und beide Ausprägungen des zweiten Index und alle drei Ausprägungen des dritten Index sind in der Tabelle erweitert.

Tabellen-Layout:



Die Tabelle ist als eine Zusammenstellung von Arrays definiert, die die folgenden Eigenschaften gemeinsam haben:

- Die Anzahl der Ausprägungen für den Hauptindex muss bei jedem Array der Tabelle gleich sein. Der Hauptindex wird immer vertikal erweitert.
- Alle Elemente eines bestimmten Index müssen in derselben Zeile platziert werden. Dadurch hängt der Abstand zwischen den Elementen eines bestimmten Index von dem Array mit der größten Dimension ab.

23 Verarbeitungsregeln

- Feldbezogene Verarbeitungsregeln 278
- Funktionstastenbezogene Verarbeitungsregeln 279
- Verarbeitungsregel bearbeiten 280

Feldbezogene Verarbeitungsregeln

Es können drei Arten von feldbezogenen Verarbeitungsregeln definiert werden.

- **Inline-Verarbeitungsregeln**
  Diese Regeln werden innerhalb eines Map Source-Objekts definiert. Ihnen wird kein Name zugewiesen. Predict braucht nicht vorhanden sein.
- **Predict Free Rules**
  Diese Regeln haben einen zugewiesenen Namen. Gespeichert werden sie in Predict.

  Um eine Predict Free Rule zu bearbeiten, müssen Sie die Regel beim Anlegen der Maske eingeben und ihr einen Namen zuweisen.
- **Predict Automatic Rules**
  Diese Regeln betreffen Datenbankfelder. Definiert werden Sie vom Predict-Administrator. Wird ein Feld von den Datendefinitionen in einem anderen Natural-Objekt ausgewählt, werden alle automatischen Regeln für dieses Feld mit der Maskendefinition verknüpft. Alle automatischen Regeln werden miteinander verkettet und als eine einzige Maskenregel behandelt.

  Der Rang der automatischen Regeln wird in den Maskeneinstellungen definiert (die Standardeinstellung ist `1`).

  Im Masken-Editor können Sie die automatischen Regeln nicht ändern. Sie können ihnen jedoch einen anderen Rang zuweisen. Dazu können Sie entweder das Kommando `P=`*n* benutzen oder einfach den alten Rang überschreiben.

Anmerkungen:

1. Sie können eine Inline-Verarbeitungsregel zu einer Predict Free Rule machen (und umgekehrt), indem Sie den Namen der Regel zuweisen (bzw. ihn entfernen).
2. Wird ein Feld mit verlinkten Predict-Verarbeitungsregeln umbenannt, gehen die Regeln verloren, und Sie müssen sie erneut verlinken.

Ein Und-Zeichen (&) im Quellcode einer Verarbeitungsregel wird dynamisch durch den vollständigen Namen des Feldes ersetzt, für das die Regel definiert ist. Array-Indices sind von der Ersetzung nicht betroffen. Sie müssen dazu nach dem Und-Zeichen (&) ausdrücklich die Index-Notation angeben, siehe folgendes Beispiel:

Beispiele:

```
IF &    = ' ' THEN REINPUT 'Enter NAME' MARK *&   /* For a scalar field
IF &(1) = ' ' THEN MOVE 'X' TO &(*)               /* For an array field
```

Die Feldnamen-Notation `&.`*`field-name`* im Quellcode einer Verarbeitungsregel ermöglicht es, DDM-spezifische Regeln zu haben, die die Integrität von Werten zwischen Datenbankfeldern gegenprüfen, ohne dazu die Felder ausdrücklich mit einem View-Namen vollständig benennen zu müssen. Anstelle von *`field-name`* geben Sie den Namen des Datenbankfeldes an, so wie er in dem DDM definiert ist. Beim Kompilieren qualifiziert Natural das Feld, indem es das Und-Zeichen (&) durch den entsprechenden View-Namen ersetzt. Dadurch haben Sie die Möglichkeit, dieselbe Verarbeitungsregel für bestimmte Felder zu benutzen, ohne berücksichtigen zu müssen, aus welcher View die Felder genommen werden.

Funktionstastenbezogene Verarbeitungsregeln

Es können zwei Arten von funktionstastenbezogenen Verarbeitungsregeln definiert werden.

- Inline-Verarbeitungsregeln
- Predict Free Rules

Funktionstastenbezogene Verarbeitungsregeln können Sie benutzen, um programmsensitiven Funktionstasten (PF-Tasten) bei der Maskenverarbeitung bestimmte Aktivitäten zuzuweisen. Bei PF-Tasten, denen bereits ein Kommando durch das Programm zugewiesen ist, wird dieses Kommando ohne Regelverarbeitung ausgeführt.

Beispiel:

```
IF *PF-KEY = 'PF3'
   ESCAPE ROUTINE
END-IF
```

Wird diese Regel ausgeführt, dann wird die Maskenverarbeitung ohne weitere Regelverarbeitung beendet.

Verarbeitungsregel bearbeiten

Um eine Verarbeitungsregel zu bearbeiten:

- Geben Sie das Feld-Kommando `.P` ein.

  Oder:

  Setzen Sie das Zeilenkommando `..E` ab und platzieren Sie dann den Funktionscode `P` neben dem Feld, für das eine Verarbeitungsregel-Bearbeitung erfolgen soll.

Um eine funktionstastenbezogene Verarbeitungsregel zu bearbeiten:

- Setzen Sie das Zeilenkommando `..P` ab.

Beim Feld-Kommando `.P` können Sie einen Parameter (`.P`*rr*) angeben, der den Rang (Priorität) der zu definierenden oder zu bearbeitenden Verarbeitungsregel angibt. Ein Feld kann bis zu 100 Verarbeitungsregeln (Rang 0 bis 99) haben. Bei der Ausführung der Maske werden die Verarbeitungsregeln in aufsteigender Reihenfolge oder nach Rang und Position auf dem Bildschirm ausgeführt.

Bei den funktionstastenbezogenen Verarbeitungsregeln wird immer angenommen, dass sie die erste Position auf dem Bildschirm haben.

Bei der Zuweisung von Rängen für Verarbeitungsregeln werden folgende Zuweisungen empfohlen, um eine optimale Performance zu erzielen:

Rang	Verarbeitungsregel
0	Beendigungsregel
1 - 4	Automatische Regel
5 - 24	Formatprüfung
25 - 44	Wertprüfung bei einzelnen Feldern
45 - 64	Gegenkontrolle des Wertes zwischen Feldern
65 - 84	Datenbankzugriff
85 - 99	Für einen besonderen Zweck

Anmerkung: Denken Sie bitte beim Ändern oder Hinzufügen einer Verarbeitungsregel daran, dass Sie alle Map-Objekte, die diese Regel referenzieren, neu katalogisieren müssen.

Dieser Abschnitt behandelt Informationen zu folgenden Themen:

- Verarbeitungsregel zur Bearbeitung auswählen

- Direktkommandos
- Editor-Kommandos zum Positionieren
- Zeilenkommandos

Verarbeitungsregel zur Bearbeitung auswählen

Wenn Sie das Feld-Kommando `.P*` in ein Maskenfeld eingeben, wird eine Liste aller zu diesem Feld definierten Verarbeitungsregeln angezeigt.

Wenn Sie das Zeilenkommando `..P*` in einer beliebigen Maskenzeile eingeben, wird eine Liste aller Verarbeitungsregeln angezeigt, die zu den in dieser Maske verwendeten PF-Tasten definiert sind.

In jeder dieser Listen sind die Predict-Regeln durch ihre Namen, die Inline-Regeln durch ihre ersten drei Quellcode-Zeilen identifiziert. Sie können aus der Liste eine Regel zum Bearbeiten auswählen, indem Sie ihren Rang eingeben.

Der Bildschirm zum Bearbeiten einer Regel (mit Beispiel einer Verarbeitungsregel) sieht wie folgt aus:

```
Variables used in current map                                          MOD
MODTXT(A3)                                                               U
FVAR(A75/1:6)                                                            U
FTYP(A1/1:6)                                                             U
RULEMODE(A6)                                                             U
RULE-NAME(A32)                                                           D
FIELDAN(A5)                                                              D

Rule _____________________________ Field FULCB3.CBCOM
>                                   > + Rank 0     S 1   L 1    Struct Mode
ALL   ....+....10...+....20...+....30...+....40...+....50...+....60...+....70..
  0010 *
  0020 IF & EQ MASK('?')
  0030  REINPUT USING HELP
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

Während der Bearbeitung einer Verarbeitungsregel können Sie zwischen Split-Screen- und Vollbild-Modus hin und her schalten., indem Sie PF9 drücken oder das Kommando `SPLIT/SPLIT E` absetzen. Beim Split-Screen-Modus zeigt der obere Bildschirmbereich die Definitionen aller Maskenfelder

(außer Systemvariablen) an. Sie können die Position der Anzeige mittels **Editor-Kommandos zum Positionieren** verändern.

Die Eingabe bzw. Bearbeitung des Quellcodes zum Definieren der Verarbeitungsregel erfolgt so wie beim Natural-Programm-Editor.

Direktkommandos

Wenn Sie eine Verarbeitungsregel bearbeiten, können Sie eines der folgenden Direktkommandos in der Kommandozeile (>) eingeben, um bestimmte Funktionen auszuführen. Ein unterstrichener Teil eines Schlüsselworts stellt die minimal mögliche Abkürzung dar.

Weitere Informationen siehe *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation.

Kommando	Erläuterung
ADD[(*n*)]	Fügt *n* Leerzeilen zum Quellcode hinzu. Weitere Informationen siehe ADD-Kommando in ***Editor-Kommandos*** in der *Programm-Editor*-Dokumentation.
CHANGE '*string1*'*string2*'	Durchsucht den Quellcode nach der als Suchwert (*string1*) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (*string2*) eingegebenen Zeichenkette.
CHECK	Prüft die Regel.
CLEAR	Löscht den Inhalt des Editierbereichs (auch die Markierungen X und Y).
DX oder DY oder DX-Y	Löscht die mit X markierte Zeile oder die mit Y markierte Zeile oder den mit den Markierungen X und Y eingegrenzten Zeilenblock.
EX oder EY oder EX-Y	Löscht alle Zeilen ab dem Anfang des Editierbereichs bis zur mit X markierten Zeile (die jedoch nicht gelöscht wird) oder ab der Zeile, die auf die mit Y markierte Zeile folgt, bis zum Ende des Editierbereichs oder alle Quellcodezeilen im Quellcodebereich mit Ausnahme des Zeilenblocks, der mit den Markierungen X und Y eingegrenzt ist. .
EXIT .	Beendet die Bearbeitung der Verarbeitungsregel. Anschließend Rückkehr zur Maskenbearbeitung.
P	Positioniert vorwärts zur nächsten für das Feld definierten Regel.
P*	Auswahl einer Regel aus dem Auswahlmenü.
P*rr*	Auswahl der Regel mit Rang *rr*.

Kommando	Erläuterung
P=*rr*	Ändert den Rang einer Verarbeitungsregel in den Rang *rr*.
POINT	Positioniert die Zeile, in der das Zeilenkommando .N eingegeben wurde, an den Anfang des aktuellen Bildschirms.
RESET	Löscht die aktuellen Markierungen X und/oder Y sowie jede zuvor mit dem Zeilenkommando .N gesetzte Markierung.
SAVE *name*	Speichert die Verarbeitungsregel als Copycode unter dem Namen *name*.
SCAN ['*scan-value*']	Durchsucht den Quellcode-Arbeitsbereich nach Daten. Wenn Sie das SCAN-Kommando ohne Parameter eingeben, wird das **SCAN/REPLACE**-Fenster angezeigt. Sie können dann zusätzlich zum Suchwert (*scan-value*) Folgendes eingeben: Wenn Sie SCAN '*scan-value*' eingeben, erfolgt die Suche nach dem Suchwert *scan-value*.
SCAN = [+|-]	Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts in der angegebenen Richtung. Weitere Informationen siehe SCAN=[+|-] unter *Editor-Kommandos* beim Programm-Editor.
SHIFT [-|+*nn*]	Verschiebt jede Quellcode-Zeile, die mit den Markierungen X und Y begrenzt ist, nach links oder nach rechts. Der Parameter *nn* gibt die Anzahl der Zeichen an, um die die Zeile verschoben werden soll. Kommentarzeilen werden nicht verschoben.
SHIFT --	Verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, ganz nach links. Kommentarzeilen werden nicht verschoben.
SHIFT ++	Verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, ganz nach rechts (maximal um 99 Stellen). Kommentarzeilen werden nicht verschoben.
SPLIT [E]	Umschalten zwischen ***Split-Screen-Modus*** und Vollbild-Modus. Siehe auch SPLIT in ***Editor-Kommandos zum Positionieren***.
TEST	Testet eine Maskendefinition.
UNLINK	Trennt die Verknüpfung zwischen einer Inline-Regel (Inline Rule) oder einer Predict Free Rule und dem Feld.

Anmerkung: Um aus den vorhanden Predict Free Rules eine Regel auszuwählen, müssen Sie ein Fragezeichen (?) in das Regelnamensfeld des Editier-Bildschirms für die Verarbeitungsregel eingeben.

Editor-Kommandos zum Positionieren

Editor-Kommandos zum Positionieren werden in der Kommandozeile (>) des Regel-Editors eingegeben. Die folgenden Kommandos stehen zum Positionieren zur Verfügung. Ein unterstrichener Teil eines Schlüsselworts stellt die minimal mögliche Abkürzung dar.

Kommando	Erläuterung
+P	Eine Seite vorwärts positionieren.
+	
-P	Eine Seite rückwärts positionieren.
-	
+H	Eine halbe Seite vorwärts positionieren.
-H	Eine halbe Seite rückwärts positionieren.
TOP	An den Anfang der Regel positionieren.
-	
BOTTOM	An das Ende der Regel positionieren.
++	
+*nnnn*	Um *nnnn* (maximal 4 Ziffern) Zeilen vorwärts positionieren.
-*nnnn*	Um *nnnn* (maximal 4 Ziffern) Zeilen rückwärts positionieren.
nnnn	Auf Zeile Nummer *nnnn* positionieren.
X	Auf die mit X markierte Zeile positionieren.
Y	Auf die mit Y markierte Zeile positionieren.
SPLIT[-|--|+|++]	Im Split-Screen rückwärts (- oder --) oder vorwärts positionieren (+ oder ++).

Zeilenkommandos

Zusätzlich zu den Editor-Kommandos stehen beim Bearbeiten einer Verarbeitungsregel die folgenden Zeilenkommandos zur Verfügung:

Kommando	Erläuterung
.C(*nnnn*)	Kopiert die Zeile, in der das Kommando eingegeben wurde.
.CX(*nnnn*) oder .CY(*nnnn*)	Kopiert die Zeile, die mit X oder mit Y markiert ist. Siehe auch Zeilenkommandos .X und .Y in den folgenden Abschnitten.
.CX-Y(*nnnn*)	Kopiert den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist. Ist die Richtungsanzeige auf + (Pluszeichen) gesetzt, werden die kopierten Zeilen nach der Zeile platziert, in der das entsprechende Kommando eingegeben wurde.

Kommando	Erläuterung
	Ist die Richtungsanzeige auf - (Minuszeichen) gesetzt, werden die kopierten Zeilen vor die Zeile platziert, in der das entsprechende Kommando eingegeben wurde.
.D(*nnnn*)	Löscht eine oder mehrere Zeilen. Der Standardwert ist 1 Zeile.
.I(*n*)	Fügt *n* Leerzeilen ein. Zeilen, die leer gelassen werden, werden beim nächsten Drücken von Enter wieder entfernt.
.I(*obj*,*ssss*,*nnnn*)	Fügt in den Quellcode ein Objekt ein, das in der aktuellen Library oder in der Steplib enthalten ist. Der Eintrag *ssss* kann benutzt werden, um die Zeile anzugeben, ab der die Einfügung beginnen soll. Der Eintrag *nnnn* kann benutzt werden, um die Anzahl der einzufügenden Zeilen anzugeben. Weitere Informationen zu den Zeilenkommandos .I siehe ***Zeilenkommandos*** im Abschnitt *Maske bearbeiten*.
.J	Führt die aktuelle Zeile mit der nächsten Zeile zusammen. Ist die resultierende Zeile länger als die Editor-Bildschirmzeile, wird die Zeile mit einem L markiert und muss mit dem Zeilenkommando .S in zwei Zeilen geteilt werden (siehe unten), bevor sie geändert werden kann.
.L	Macht alle seit dem letzten Drücken der Enter-Taste an der Zeile vorgenommenen Änderungen rückgängig.
.MX oder .MY	Verschiebt die mit X oder mit Y markierte Zeile. Siehe auch Zeilenkommandos .X und .Y (weiter unten).
.MX-Y	Verschiebt den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist. Ist die Richtungsanzeige auf + (Pluszeichen) gesetzt, werden die verschobenen Zeilen nach der Zeile platziert, in der das entsprechende Kommando eingegeben wurde. Ist die Richtungsanzeige auf - (Minuszeichen) gesetzt, werden die verschobenen Zeilen vor die Zeile platziert, in der das entsprechende Kommando eingegeben wurde.
.P	Positioniert die mit diesem Kommando markierte Zeile an den Anfang des Bildschirms.
.S	Spaltet die Zeile an der mit dem Cursor markierten Position.
.W	Fügt *n* Leerzeilen ein. Zeilen, die leer gelassen werden, werden beim nächsten Drücken von Enter wieder entfernt.
.X	Markiert eine Zeile oder den Anfang eines Zeilenblocks, die bzw. der bearbeitet werden soll.

Kommando	Erläuterung
.Y	Markiert eine Zeile oder das Ende eines Zeilenblocks, die bzw. der bearbeitet werden soll. **Anmerkung:** Wenn sowohl das Kommando .X als auch das Kommando .Y bei einer Zeile angewendet werden, wird sie behandelt, als ob sie mit einem X und mit einem Y markiert ist; die tatsächlich angezeigte Zeilenmarkierung, die diesen Status wiedergeben soll, ist ein Z.

VII Map Editor Tutorial

Dieses Tutorial bietet eine allgemeine Einführung in die Benutzung des Natural-Masken-Editors (*Natural Map Editor*). Es werden nur minimale Erläuterung gegeben. Eine umfassende Beschreibung aller Funktionen des Masken-Editors finden Sie im Abschnitt ***Masken-Editor***.

Das Layout (24x80) der in diesem Tutorial enthaltenen Beispiel-Bildschirme und das hier beschriebene Verhalten von Natural kann von Ihren Ergebnissen abweichen. Beispielsweise können die Kommandozeile und die Nachrichtenzeile an einer anderen Stelle im Bildschirm erscheinen oder die Ausführung eines Natural-Kommandos kann wegen entsprechender Security-Einstellung nicht erlaubt sein. Die Standard-Einstellungen in Ihrer Umgebung sind von den Systemparametern abhängig, die Ihr Natural-Systemadministrator gesetzt hat.

Wichtig: Wichtig ist, dass Sie bei der Abarbeitung der Übungen die unten angegebene Reihenfolge einhalten. Andernfalls kann es sein, dass Sie nicht die in den Übungen beabsichtigten Ergebnisse erzielen.

Anmerkung: Der Masken-Editor (Map Editor) ist standardmäßig in Ihrer Umgebung abgeschaltet (deaktiviert). Weitere Information siehe ***NaturalONE als Standard-Entwicklungsumgebung***.

Masken-Editor öffnen	Wie Sie das Menü des Masken-Editors aufrufen und eine Maske anlegen.
Maskenfelder anlegen, positionieren und löschen	Wie Sie Maskenfelder anlegen und diese an die gewünschte Stelle in der Maske positionieren.
Maske testen und speichern	Wie Sie eine Maske (Map) als Objektmodul testen und speichern.
Verarbeitungsregeln definieren	Wie Sie Regeln für die Verarbeitung einer Maske definieren.
Felder benennen und Maske speichern/katalogisieren	Wie Sie die Maskenfelder mit Namen versehen und eine Maske als Objektmodul speichern/katalogisieren.
Feld-Eigenschaften definieren	Wie Sie die Funktion zur erweiterten Feld-Bearbeitung benutzen, um Feld-Eigenschaften zu definieren.
Hilfe-Maske erstellen und testen	Wie Sie eine Hilfe-Maske für ein Maskenfeld anlegen und testen.

Maske mit INPUT USING MAP aufrufen	Wie Sie ein Programm erstellen und ausführen, das eine Maske mit dem `INPUT USING MAP`-Statement aufruft.
Maske für WRITE erstellen und Feld-Definitionen kopieren	Wie Sie eine Maske erstellen, indem Sie Felder aus einem anderen Natural-Objekt kopieren. Wie Sie die Masken-Einstellungen ändern, um das `WRITE`-Statement zu benutzen.
Layout einer Maske wiederverwenden	Wie Sie eine Maske unter Verwendung des Layouts einer anderen Maske anlegen.
Maske mit WRITE USING MAP aufrufen	Wie Sie ein Programm erstellen und ausführen, das eine Maske mit dem `WRITE USING MAP`-Statement aufruft.

24 Masken-Editor öffnen

In dieser Sitzung werden Sie den Masken-Editor aufrufen, Maskeneinstellungen angeben und den Editor-Bildschirm öffnen.

Um den Masken-Editor aufzurufen:

1 Wählen Sie im Natural-Hauptmenü (**Main Menu**) die Entwicklungsfunktionen (**Development Functions**) und drücken Sie Enter.

Das Menü **Development Functions** erscheint.

2 Falls Sie im Reporting-Modus arbeiten, müssen Sie in den Structured-Modus umschalten:

Geben Sie dazu an der ersten Stelle des Eingabefeldes **Mode** ein `S` ein und drücken Sie Enter.

Im **Mode**-Feld wird jetzt der Modus `Structured` angezeigt.

3 Geben Sie im **Code**-Feld ein `E` (für **Edit Object**) ein und im **Type**-Feld ein `M` (Map). Ignorieren Sie das Feld **Name**.

4 Drücken Sie Enter.

Das Menü **Edit Map** erscheint:

```
14:08:08               ***** NATURAL MAP EDITOR *****              2007-12-14
User SAG                        - Edit Map -                   Library SAGTEST

                   Code    Function
                   ----    --------------------------------
                     D     Field and Variable Definitions
                     E     Edit Map
                     I     Initialize new Map
                     H     Initialize a new Help Map
                     M     Maintenance of Profiles & Devices
                     S     Save Map
                     T     Test Map
                     W     Stow Map
                     ?     Help
                     .     Exit



             Code .. I     Name .. ________       Profile .. SYSPROF_

Command ===>

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

Das Menü **Edit Map** ist das Hauptmenü des Masken-Editors.

Tipp: Der Masken-Editors bietet ein umfangreiches Hilfesystem. Sie können jederzeit ein Fragezeichen (?) in einem Feld eingeben, zu dem Sie weitere Informationen erhalten möchten. Es wird die Online-Hilfe zu diesem Feld aufgerufen. Falls einem Feld keine spezifische Hilfe zugewiesen worden ist, wird ein Hilfe-Menü angezeigt, in dem Sie die benötigte Hilfe-Information auswählen können.

5 Geben Sie im **Code**-Feld ein `I` (für **Initialize new Map**) und im Feld **Name** den Namen `MAP001` ein.

6 Drücken Sie Enter.

Der Bildschirm **Define Map Settings for MAP** erscheint:

```
14:10:19              Define Map Settings for MAP                  2007-12-14

 Delimiters           Format                          Context
 -----------------    ---------------------------     --------------------------
 Cls Att CD  Del      Page Size ...... 31             Device Check .... ________
  T   D      BLANK    Line Size ...... 79             WRITE Statement   _
  T   I      ?        Column Shift ... 0 (0/1)        INPUT Statement   X
  A   D      _        Layout ......... ________       Help  ____________________
  A   I      )         dynamic ....... N (Y/N)         as field default N (Y/N)
  A   N      ¬        Zero Print ..... N (Y/N)
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)        Automatic Rule Rank  1
  O   D      +        Decimal Char ... .              Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)        Filler Characters
                      Print Mode ..... __             ------------------------
                                                      Optional, Partial ....
                      Control Var .... ________       Required, Partial ....
                                                      Optional, Complete ...
                                                      Required, Complete ...

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

7 Platzieren Sie den Cursor in den Abschnitt **Filler Characters** (Füllzeichen) und geben Sie so wie unten gezeigt hinter jeder der vier Optionen einen Unterstrich (_) ein:

```
14:10:19              Define Map Settings for MAP                 2007-12-14

 Delimiters           Format                        Context
 -----------------    ---------------------------   --------------------------
 Cls Att CD  Del      Page Size ...... 31           Device Check .... ________
  T   D      BLANK    Line Size ...... 79           WRITE Statement   _
  T   I      ?        Column Shift ... 0 (0/1)      INPUT Statement   X
  A   D      _        Layout ......... ________     Help  ____________________
  A   I      )         dynamic ....... N (Y/N)       as field default N (Y/N)
  A   N      ¬        Zero Print ..... N (Y/N)
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)      Automatic Rule Rank  1
  O   D      +        Decimal Char ... .            Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)      Filler Characters
                      Print Mode ..... __           -----------------------
                                                    Optional, Partial .... _
                      Control Var .... ________     Required, Partial .... _
                                                    Optional, Complete ... _
                                                    Required, Complete ... _









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

Das hat zur Folge, dass alle leeren Stellen in einem Eingabefeld auf der Maske mit einem Unterstrich (_) gefüllt werden. Dadurch können Sie dann die exakte Position und die Länge eines Feldes erkennen, was die Eingabe erleichtert.

8 Ignorieren Sie die übrigen Masken-Einstellungen und drücken Sie Enter *zweimal*.

Der folgende Editor-Bildschirm erscheint:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

Der Bildschirm wird im Split-Screen-Modus angezeigt: Im oberen Bereich werden die Begrenzungszeichen angezeigt, die für die zu erstellende Maske gelten, und der untere Bereich ist der Editier-Bereich, in dem Sie eine Maske entwerfen können.

Sie können nun fortfahren mit *Maskenfelder anlegen, positionieren und löschen*.

25 Maskenfelder anlegen, positionieren und löschen

- Felder anlegen und zentrieren 296
- Felder verschieben 301
- Felder löschen und Zeilen einfügen 316

In dieser Sitzung werden Sie eine Maske entwerfen.

Felder anlegen und zentrieren

Um Text-Felder anzulegen:

1 Geben Sie in der ersten Zeile des Editier-Bereichs für `MAP001` das Zeilenkommando `..F*` ein.

Geben Sie in der zweiten Zeile den Text `PERSONNEL INFORMATION` so wie unten gezeigt ein:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
..F*
PERSONNEL INFORMATION




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

2 Drücken Sie Enter.

Der Bildschirm sieht nun folgendermaßen aus:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
PERSONNEL INFORMATION









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

3 Drücken Sie PF9.

Der Masken-Editier-Bildschirm wechselt nun in den Vollbild-Anzeigemodus: Die Begrenzungszeichen sind nun ausgeblendet.

4 Geben Sie in der unteren Zeile das Zeilenkommando `..F*` ein.

5 Drücken Sie Enter

Der Bildschirm sieht nun folgendermaßen aus:

```
********************************************************************************
PERSONNEL INFORMATION









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Um alle in einer Zeile enthaltenen Felder zu zentrieren:

1 Geben Sie in den ersten drei Stellen im Text das Zeilenkommando `..C` so wie unten gezeigt ein:

```
********************************************************************************
..CSONNEL INFORMATION




                                                                                
                                                                                
                                                                                
```

```
********************************************************************************
(*DATX                       PERSONNEL INFORMATION
(*TIMX

PLEASE Enter CITY::X(20)
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

`*DATX` und `*TIMX` sind Natural-Systemvariablen, die das aktuelle Datum bzw. die aktuelle Zeit anzeigen. Die öffnende Klammer (`(`) ist das Begrenzungszeichen für hervorgehobene Ausgabefelder. Der Doppelpunkt (`:`) ist das Begrenzungszeichen für hervorgehobene, änderbare Felder. Die Anzahl der X-Zeichen zeigt die Länge des Feldes an.

2 Drücken Sie Enter.

Der Bildschirm sieht nun folgendermaßen aus:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Felder verschieben

Um ein einzelnes Feld oder eine ganze Zeile zu verschieben:

1 Geben Sie im Editier-Bereich das Zeilenkommando `.M` so wie unten gezeigt ein und platzieren Sie den Cursor an die mit `[]` markierte Stelle. Die beiden eckigen Klammern werden in den folgenden Beispiel-Bildschirmen lediglich dazu verwendet, die Position des Cursors zu kennzeichnen. Diese Klammern sollen nicht eingegeben werden.

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

.MEASE Enter CITY::XXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

2 Drücken Sie Enter.

Das Text-Feld, in das das Kommando eingegeben wurde, wird zur Position des Cursors verschoben:

```
********************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

       Enter CITY::XXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

3 Geben Sie das Zeilenkommando `..M` so wie unten gezeigt ein und platzieren Sie den Cursor an die mit `[]` markierte Stelle:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

..M    Enter CITY::XXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]LEASE









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

4 Drücken Sie Enter.

Die Zeile, in die das Kommando eingegeben wurde, wird in die Zeile nach der Zeile verschoben, in der der Cursor positioniert ist:

```
********************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE
       Enter CITY::XXXXXXXXXXXXXXXXXX









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Um Zeilen zusammenzuführen:

1 Geben Sie das Zeilenkommando `..J` wie unten gezeigt ein:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
..JASE
       Enter CITY::XXXXXXXXXXXXXXXXXXX









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

2 Drücken Sie Enter.

Die Zeile, in der das Kommando eingegeben wurde, und die Zeile darunter werden zusammengeführt:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX









*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Um Felder oder Zeilen blockweise zu verschieben:

1 Geben Sie zusätzlichen Text in der gleichen Reihenfolge und an der gleichen Stelle wie unten gezeigt ein:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

THIS PORTION OF TEXT IS
FOR FURTHER DEMONSTRATION
OF THE MOVE
COMMANDS









*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

2 Drücken Sie Enter.

3 Geben Sie das Feld-Kommando `.M` zweimal ein und platzieren Sie den Cursor an die mit `[]` markierte Stelle:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX

.MIS PORTION OF TEXT IS
FOR FURTHER DEMONSTRATION
OF THE MOVE
.MMMANDS




                                    []
*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Der Block mit den zu verschiebenden Feldern wird durch die `.M`-Kommandos und die Breite der Felder, für die diese Kommandos gelten, begrenzt. In diesem Beispiel beginnt der Block links oben beim Feld `THIS` und endet beim letzten Feld `COMMANDS`. Das breiteste Feld `COMMANDS` (das sich über zwei oder drei der in der vorangehenden Zeile enthaltenen Felder erstreckt) bestimmt, welche Felder innerhalb des markierten Blocks verschoben werden.

4 Drücken Sie Enter.

Der Block mit den markierten Feldern wird an die Position des Cursors verschoben und das Feld oben links wird auf dem Cursor platziert.

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

            OF TEXT IS
           DEMONSTRATION




                                    THIS PORTION
                                    FOR FURTHER
                                    OF THE MOVE
                                    COMMANDS









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

5 Geben Sie das Feld-Kommando `.M` zweimal ein (siehe unten) und positionieren Sie den Cursor an die mit `[]` markierte Stelle.

```
*******************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX

             .M TEXT IS
            .MMONSTRATION




                                      THIS PORTION[]
                                     FOR FURTHER
                                     OF THE MOVE
                                     COMMANDS









*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

6 Drücken Sie Enter.

Der Block mit den markierten Feldern wird an die Position des Cursors verschoben.

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

                                    THIS PORTION OF TEXT IS
                                    FOR FURTHER DEMONSTRATION
                                    OF THE MOVE
                                    COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

7 Geben Sie das Feld-Kommando `.M` dreimal ein (siehe unten) und positionieren Sie den Cursor an die mit `[]` markierte Stelle:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

                            []

                                        .MIS PORTION OF TEXT IS
                                        FOR FURTHER .MMONSTRATION
                                       OF THE MOVE
                                        .MMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

8 Drücken Sie Enter.

Der gesamte mit den Kommandos eingegrenzte Felder-Block wird an die Cursor-Position verschoben:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

                           THIS PORTION OF TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

9 Geben Sie das Zeilenkommando `..M` zweimal ein (siehe unten) und platzieren Sie den Cursor an die mit `[]` markierte Stelle:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

..M                          THIS PORTION OF TEXT IS
                             FOR FURTHER DEMONSTRATION
                             OF THE MOVE
..M                          COMMANDS


[]









********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

10 Drücken Sie Enter.

Der mit den Kommandos eingegrenzte Zeilen-Block wird unterhalb der Zeile eingefügt, in der der Cursor steht. (Der alte Zeilen-Block wird an der früheren Stelle gelöscht.)

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX



                          THIS PORTION OF TEXT IS
                          FOR FURTHER DEMONSTRATION
                          OF THE MOVE
                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Felder löschen und Zeilen einfügen

Um Felder zu löschen:

1 Geben Sie im Editier-Bereich das Feld-Kommando `.T` so wie unten gezeigt ein:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX

                           THIS PORTION .T TEXT IS
                           FOR FURTHER DEMONSTRATION
                           OF THE MOVE
                           COMMANDS

*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

2 Drücken Sie Enter.

Das Feld, in das das Kommando eingegeben wurde, und die Felder in der restlichen Zeile werden gelöscht:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX




                          THIS PORTION
                          FOR FURTHER DEMONSTRATION
                          OF THE MOVE
                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

3 Geben Sie das Feld-Kommando `.D` so wie unten gezeigt ein:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX



                          THIS PORTION
                          FOR .DRTHER DEMONSTRATION
                          OF THE MOVE
                          COMMANDS

*******************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

4 Drücken Sie Enter.

Das Feld, in das das Kommando eingegeben wurde, wird gelöscht:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX




                          THIS PORTION
                          FOR         DEMONSTRATION
                          OF THE MOVE
                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

5 Geben Sie das Zeilenkommando `.M` ein (siehe unten) und positionieren Sie den Cursor an die mit `[]` markierte Stelle:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX




                          THIS PORTION
                          FOR []      .MMONSTRATION
                          OF THE MOVE
                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

6 Drücken Sie Enter.

Das Feld, in das das Kommando eingegeben wurde, wird an die Cursor-Position verschoben:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX




                          THIS PORTION
                          FOR DEMONSTRATION
                          OF THE MOVE
                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

7 Geben Sie das Zeilenkommando `..D` zweimal so wie unten gezeigt ein:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX




..D                          THIS PORTION
                             FOR DEMONSTRATION
                             OF THE MOVE
..D                          COMMANDS

********************************************************************************
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

8 Drücken Sie Enter.

Der mit den Kommandos eingegrenzte Zeilen-Block wird gelöscht:

```
********************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

********************************************************************************

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

Um Zeilen einzufügen:

1 Geben Sie das Zeilenkommando `..I4` so wie unten gezeigt ein:

```
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX
..I4









*******************************************************************************



001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

2 Drücken Sie Enter.

Es werden vier Leerzeilen eingefügt und die untere Linie mit den Sternen wird um vier Zeilen nach unten verschoben.

Sie können nun fortfahren mit ***Maske testen und speichern***.

26 Maske testen und speichern

In dieser Sitzung werden Sie das Layout einer Maske testen, den Masken-Editier-Bereich verlassen und eine Maske als Source-Objekt speichern.

Um eine Maske zu testen und sie als Source-Objekt zu speichern:

1 Drücken Sie im Editier-Bereich die Taste PF4.

Die Maske wird jetzt in der Layout-Ansicht angezeigt. Sie erscheint dabei so, wie sie beim Ausführen des Programms, das diese Maske referenziert, dargestellt wird:

```
********************************************************************************
 07-12-14                    PERSONNEL INFORMATION
 15:26:05

PLEASE Enter NAME: ____________________________________________
PLEASE Enter CITY: ____________________




********************************************************************************
```

2 Drücken Sie PF3.

Der Test wird beendet und der Masken-Editier-Bereich erscheint.

3 Drücken Sie PF3.

Der Bildschirm **Field and Variable Definitions - Summary** erscheint. Dieser Bildschirm wir in einer späteren Sitzung dieses Tutorials behandelt.

4 Drücken Sie Enter.

Das Menü **Edit Map** erscheint. Das Feld **Name** ist mit dem Maskennamen MAP001 belegt.

5 Geben Sie im Feld **Code** ein S ein und drücken Sie Enter.

Die Maske wird als Source-Objekt mit dem Namen MAP001 in der aktuellen Natural-Library in der aktuellen Systemdatei gespeichert.

Sie können nun fortfahren mit *Verarbeitungsregeln definieren*.

27 Verarbeitungsregeln definieren

In dieser Sitzung werden Sie Verarbeitungsregeln für ein Maskenfeld definieren.

Um Verarbeitungsregeln zu definieren:

1 Geben Sie im Feld **Code** des Menüs **Edit Map** ein `E` ein und geben Sie im Feld **Name** den Maskennamen `MAP001` ein.

Masken-Editier-Bildschirm erscheint jetzt im Split-Screen-Modus und die Maske `MAP001` wird im Editier-Bereich angezeigt.

2 Geben Sie das Feld-Kommando `.P` so wie unten gezeigt ein:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME:.PXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

3 Drücken Sie Enter.

Der Verarbeitungsregel-Editor erscheint für das Feld, in dem das Kommando eingegeben wurde:

```
Variables used in current map                                          Mod
#001(A40)
#002(A20)




Rule _____________________________________ Field #001
>                                          > + Rank 0    S      L 1     Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010
  0020
  0030
  0040
  0050
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

4 Geben Sie die folgende Verarbeitungsregel ein:

```
Rule ____________________________________ Field #001
>                                        > + Rank 0    S      L 1     Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010 *
  0020 IF & = ' ' REINPUT 'PLEASE TYPE IN A NAME'
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

Das kaufmännische Und-Zeichen (&) in der Verarbeitungsregel wird dynamisch durch den Namen des Feldes ersetzt, mit dem die Verarbeitungsregel verknüpft ist.

5 Drücken Sie Enter und dann PF3.

Der Masken-Editier-Bildschirm erscheint.

6 Drücken Sie PF4, um die Maske zu testen.

Der Test-Bildschirm erscheint.

7 Drücken Sie Enter, um die Verarbeitungsregel zu testen.

Die Verarbeitungsregel wird ausgeführt und der in der Regel angegebene Text erscheint:

```
********************************************************************************
 07-12-14                    PERSONNEL INFORMATION
 16:04:15

PLEASE ENTER NAME: _____________________________________________
PLEASE ENTER CITY: ____________________


********************************************************************************
PLEASE TYPE IN A NAME
```

Anmerkung: Der Text `PLEASE TYPE IN A NAME` muss nicht unbedingt (so wie oben gezeigt) am unteren Bildschirmrand erscheinen. Je nachdem, wie der Natural-Administrator die Nachrichtenzeile eingestellt hat, kann er auch in einer anderen Zeile erscheinen.

8 Geben Sie an der ersten Stelle des Eingabefeldes neben `PLEASE ENTER NAME:` ein beliebiges Zeichen ein und drücken Sie Enter.

Der Test wird beendet und der Masken-Editier-Bildschirm erscheint.

9 Geben Sie das Feld-Kommando `.P` an der gleichen Stelle wie zuvor ein und drücken Sie Enter.

Die Verarbeitungsregel für den Rang (Priorität) 0 des Feldes, in dem das Kommando eingegeben wurde, wird wieder angezeigt.

10 Geben Sie das Kommando `P=5` so wie unten gezeigt ein:

```
Rule _____________________________________ Field #001
> P=5                                      > + Rank 0    S 5    L 1     Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010 *
  0020 IF & = ' ' REINPUT 'PLEASE TYPE IN A NAME'
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

11 Drücken Sie Enter.

Die Verarbeitungsregel, der vorher der Rang 0 zugewiesen worden war, erhält jetzt den Rang 5 (die Verarbeitung der Verarbeitungsregeln erfolgt in aufsteigender Reihenfolge und beginnt bei Rang 0).

12 Geben Sie das Kommando `P0` so wie unten gezeigt ein:

```
Rule _____________________________________ Field #001
> P0                                    > + Rank 5     S 5   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = ' ' REINPUT 'PLEASE TYPE IN A NAME'
  0030            MARK *&
  0040 END-IF
  0050 *
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

13 Drücken Sie Enter.

Es erscheint jetzt ein leerer Verarbeitungsregel-Editor-Bildschirm, weil zu Rang 0 keine Verarbeitungsregel mehr zugewiesen ist.

```
Rule _____________________________________ Field #001
>                                          > + Rank 0    S      L 1     Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010
  0020
  0030
  0040
  0050
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

14 Geben Sie die folgende Verarbeitungsregel ein:

```
Rule _____________________________________ Field #001
>                                          > + Rank 0     S 0   L 1     Struct Mode
ALL   ....+....10...+....+....+....30...+....+....+....50...+....+....+....70..
  0010 *
  0020 IF & = MASK ('.') STOP
  0030 END-IF
  0040 *
  0050
  0060
  0070
  0080
  0090
  0100
  0110
  0120
  0130
  0140
  0150
  0160
  0170
  0180
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test        --    -     +     Full  Sc=         Let
```

15 Drücken Sie Enter und dann PF3.

Der Masken-Editier-Bildschirm erscheint.

Sie können nun fortfahren mit ***Felder benennen und Maske speichern/katalogisieren***.

28 Felder benennen und Maske speichern/katalogisieren

In dieser Sitzung werden Sie benutzerdefinierten Felder in einer Maske mit Namen versehen und eine Maske speichern und katalogisieren.

Um Felder zu benennen und eine Maske zu speichern/zu katalogisieren:

1 Drücken Sie im Masken-Editier-Bildschirm der `MAP001` die Taste PF3.

Der Bildschirm **Field and Variable Definitions - Summary** erscheint:

```
16:10:49          Field and Variable Definitions - Summary          2007-12-14

Cmd Name (truncated)                                Mod Format       Ar Ru Lin Col
___ *DATX_____________________________________________ S   D                    2   2
___ *TIMX_____________________________________________ S   T                    3   2
___ #001______________________________________________     A40               2   5  20
___ #002______________________________________________     A20                   6  20

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit              --                Parm  Local       Let
```

Die in dem *Map*-Objekt enthaltenen Felder werden in der Reihenfolge gelistet, in der sie auf der Maske erscheinen. Den beiden benutzerdefinierten Felder werden Platzhalternamen zugewiesen, vor denen ein Doppelkreuz (#) steht. Diese Platzhalternamen müssen Sie ersetzen, um die Maske katalogisieren zu können.

2 Ersetzen Sie die Platzhalternamen so wie unten gezeigt:

```
16:10:49          Field and Variable Definitions - Summary          2007-12-14

Cmd Name (truncated)                                Mod Format       Ar Ru Lin Col
___ *DATX_____________________________________________ S   D                    2   2
___ *TIMX_____________________________________________ S   T                    3   2
___ #NAME_____________________________________________     A40                2   5  20
___ #CITY_____________________________________________     A20                    6  20
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit              --                Parm  Local       Let
```

3 Drücken Sie Enter *zweimal*.

Das Menü **Edit Map** erscheint. Im Feld **Name** steht `MAP001`.

4 Geben Sie im Feld **Code** ein `W` (für **Stow Map**) ein und drücken Sie Enter.

Die Funktion **Stow Map** führt das Kommando `STOW` aus, das die Syntax der Maske `MAP001` prüft und die Maskendefinition als Source-Objekt *und* als katalogisiertes Objekt in der aktuellen Natural-Library in der aktuellen Systemdatei speichert.

Sie können nun fortfahren mit ***Feld-Eigenschaften definieren***.

29 Feld-Eigenschaften definieren

In dieser Sitzung definieren Sie die Eigenschaften eines Feldes.

Um die Eigenschaften eines Feldes zu definieren:

1 Geben Sie im Feld **Code** im Menü **Edit Map** ein `E` und im Feld **Name** den Maskennamen `MAP001` ein (falls der nicht schon eingegeben ist).

Der Masken-Editier-Bildschirm erscheint und die Maske `MAP001` wird im Editier-Bereich angezeigt.

2 Geben Sie zusätzlichen Text so wie unten gezeigt ein:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

Das fett markierte Fragezeichen (?) ist das Begrenzungszeichen (Delimiter) für hervorgehobene Textfelder.

3 Geben Sie das Feld-Kommando `.E` so wie unten gezeigt ein:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME:.EXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

4 Drücken Sie Enter.

Der erweiterte Feld-Editier-Bereich zu dem Feld, in dem das Kommando eingegeben wurde, erscheint:

```
Fld #NAME                                                          Fmt A40
--------------------------------------------------------------------------------
AD= MIT'_'____     ZP=         SG=        HE= _____________________   Rls 2
AL= _____          CD= __      CV= _____________________________   Mod User
PM= __  DF=                    DY= _____________________________
EM= _____________________ SB= _____________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
********************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME:.EXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  <---  --->  --    -     +           <     >     Let
```

5 Ersetzen Sie im Feld **Fmt** den aktuellen Wert durch `A20`, und geben Sie im Feld **HE=** den Wert `'HELP001'` (in Hochkommas) so wie unten gezeigt ein:

```
Fld #NAME                                                          Fmt A20
------------------------------------------------------------------------------
AD= MIT'_'____     ZP=         SG=        HE= 'HELP001'_____________   Rls 2
AL= _____          CD= __      CV= ____________________________________   Mod User
PM= __  DF=                    DY= ___________________________________
EM= ____________________________ SB= ___________________________________

001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                    PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME:.EXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  <---  --->  --    -     +           <     >     Let
```

6 Drücken Sie Enter.

Der erweiterte Feld-Editier-Bereich verschwindet. HELP001 (die noch zu erstellen ist) wird als Helproutine/Hilfe-Maske zu dem Feld zugewiesen und die Feldlänge wird auf 20 verringert:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .     A  D    _       A  I    )
.                                       .     A  N    ¬       M  D    &
.                                       .     M  I    :       O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
*******************************************************************************
(XXXXXXXX                   PERSONNEL INFORMATION
(XXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

7 Drücken Sie PF3 *zweimal*.

Das Menü **Edit Map** erscheint.

8 Katalogisieren und speichern Sie die Maske `MAP001`, indem Sie die Funktion **Stow Map** ausführen.

Sie können nun fortfahren mit ***Hilfe-Maske erstellen und testen***.

30 Hilfe-Maske erstellen und testen

In dieser Sitzung erstellen und testen Sie eine Helproutine/Hilfe-Maske für ein Masken-Feld.

Um eine Hilfe-Maske zu erstellen:

1 Geben Sie im Menü **Edit Map** ein `H` in das Feld **Code** und `HELP001` in das Feld **Name** ein.

2 Drücken Sie Enter.

Der Bildschirm **Define Map Settings for HELPMAP** erscheint.

3 Geben Sie im Feld **Page Size** den Wert `15` und im Feld **Line Size** den Wert `25` so wie unten gezeigt ein:

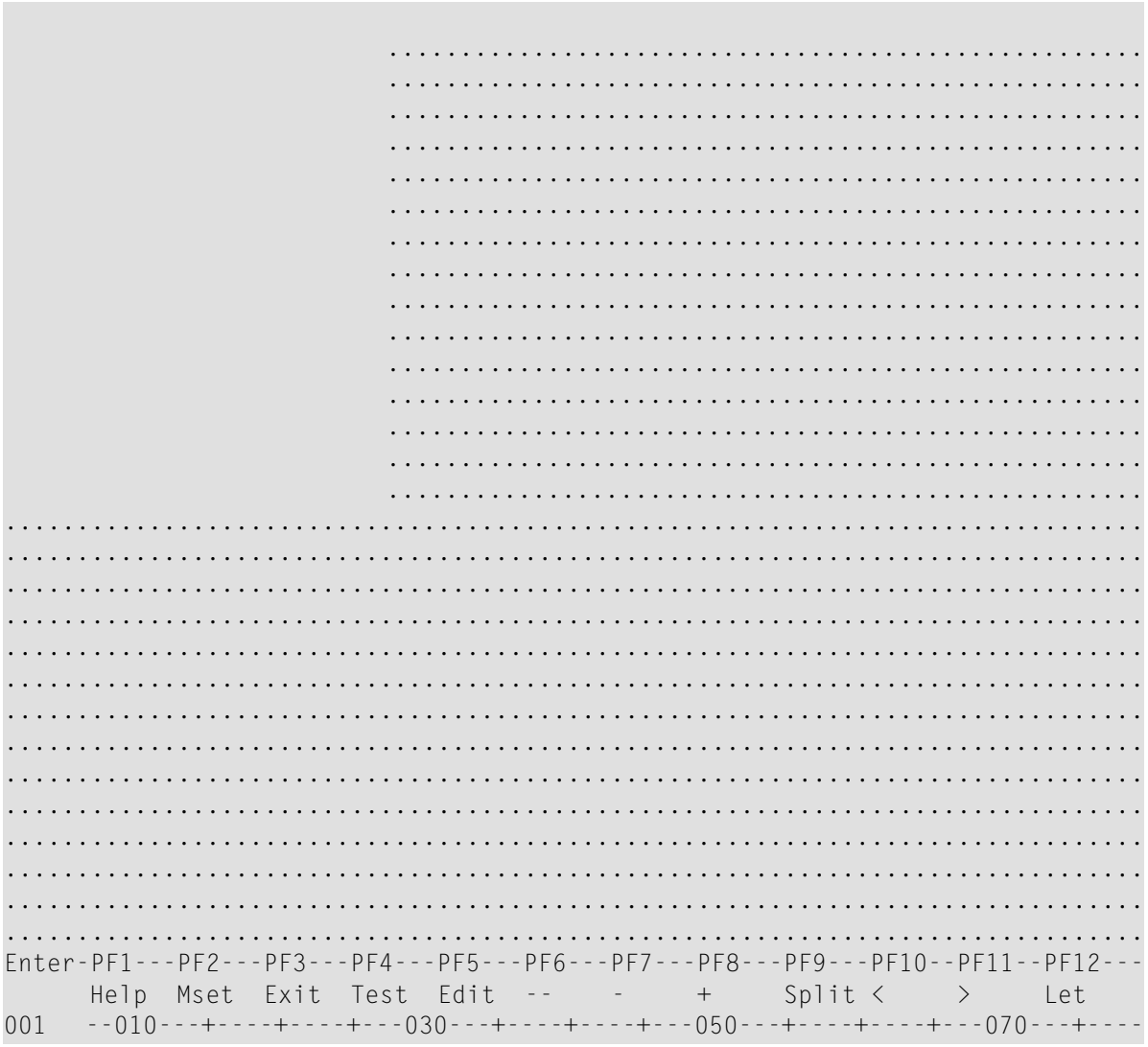
```
11:36:53              Define Map Settings for HELPMAP              2007-12-14

 Delimiters           Format                          Context
 -----------------    ---------------------------     --------------------------
 Cls Att CD  Del      Page Size ...... 15             Device Check .... ________
  T   D      BLANK    Line Size ...... 25             WRITE Statement   _
  T   I      ?        Column Shift ... 0 (0/1)        INPUT Statement   X
  A   D      _        Layout ......... ________
  A   I      )         dynamic ....... N (Y/N)
  A   N      ¬        Zero Print ..... N (Y/N)        Position Line    Col
  M   D      &        Case Default ... UC (UC/LC)
  M   I      :        Manual Skip .... N (Y/N)        Automatic Rule Rank  1
  O   D      +        Decimal Char ... .              Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)        Filler Characters
                      Print Mode ..... __             ------------------------
                                                      Optional, Partial ....
                      Control Var .... ________       Required, Partial ....
                                                      Optional, Complete ...
                                                      Required, Complete ...

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

4 Drücken Sie Enter *zweimal*.

Der Masken-Editier-Bildschirm erscheint.

5 Drücken Sie PF9.

Der Bildschirm wird im Vollbild-Modus angezeigt:



Der Teil des Bildschirms, der nicht benutzt werden soll, ist mit Punkten (.) gefüllt.

6 Geben Sie Text so wie unten gezeigt ein:

```
                          ..................................................
Type in the name of an    ..................................................
employee in the first     ..................................................
field and press Enter.    ..................................................
You will then receive     ..................................................
a list of all employees   ..................................................
of that name.             ..................................................
                          ..................................................
For a list of employees   ..................................................
of a certain name who     ..................................................
live in a certain city,   ..................................................
type in a name in the     ..................................................
first field and a city    ..................................................
in the second field       ..................................................
and press Enter.          ..................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
..........................................................................
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Split <     >     Let
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

7 Drücken Sie PF3.

Das Menü **Edit Map** erscheint. Im Feld **Name** steht `HELP001`.

8 Katalogisieren und speichern Sie die Hilfe-Maske `HELP001`, indem Sie die Funktion **Stow Map** ausführen.

9 Geben Sie im Feld **Code** ein `T` und im Feld **Name** den Maskennamen `MAP001` ein.

10 Drücken Sie Enter.

Der Test-Bildschirm für die Maske `MAP001` erscheint.

11 Geben Sie an der ersten Stelle des Eingabefeldes neben `PLEASE ENTER NAME:` ein Fragezeichen (?) ein und drücken Sie Enter.

Die Hilfe-Maske `HELP001` erscheint:

```
********************************************************************************
 07-12-14                    PERSONNEL INFORMATION
 16:58:37

PLEASE ENTER NAME: ?____________________
PLEASE ENTER CITY: +--------------------------+
                   |                          |
TYPE IN . TO STOP  | Type in the name of an   |
                   | employee in the first    |
                   | field and press Enter.   |
                   | You will then receive    |
                   | a list of all employees  |
                   | of that name.            |
                   |                          |
                   | For a list of employees  |
                   | of a certain name who    |
                   | live in a certain city,  |
                   | type in a name in the    |
                   | first field and a city   |
                   | in the second field      |
                   | and press Enter.         |
                   |                          |
                   +--------------------------+
********************************************************************************
```

12 Drücken Sie Enter *zweimal*.

Die Verarbeitungsregel für das erste Feld (#NAME) wird getestet und die folgende Meldung wird angezeigt: PLEASE TYPE IN A NAME.

13 Geben Sie an der ersten Stelle des ersten Feldes ein beliebiges Zeichen ein und drücken Sie Enter.

Der Test wird beendet und das Menü **Edit Map** erscheint.

Sie können nun fortfahren mit ***Maske mit INPUT USING MAP aufrufen***.

31 Maske mit INPUT USING MAP aufrufen

In dieser Sitzung werden Sie das Beispiel-Programm PROG001 erstellen und ausführen, um die Wirkung des Statements INPUT USING MAP zu testen, das die Maske MAP001 aufruft.

Das Programm PROG001 ist in der Natural-System-Library SYSEXPG vorhanden. Wenden Sie sich wegen weiterer Einzelheiten an Ihren Natural-System-Administrator.

Um das Programm PROG001 zu erstellen und auszuführen:

1 Wenn Sie Zugriff auf eine Kopie des Programms PROG001 haben, geben Sie in der Kommandozeile des Menüs **Edit Map** Folgendes ein:

```
EDIT PROG001
```

Der Programm-Editor wird aufgerufen und der Quellcode des Programms PROG001 wird im Editier-Bereich angezeigt. Vergewissern Sie sich, dass das Programm mit dem weiter unten gezeigten Programm identisch ist.

Oder:

Falls Sie keinen Zugriff auf eine Kopie des Programms PROG001 haben, geben Sie in der Kommandozeile des Menüs **Edit Map** Folgendes ein:

```
EDIT PROGRAM
```

Der Programm-Editor wird aufgerufen. Falls erforderlich, löschen Sie den Inhalt des Editier-Bereichs durch Eingabe des Kommandos CLEAR am Eingabeaufforderungszeichen (>) des Programm-Editors.

Geben Sie dann das folgende Programm ein:

PROG001:

```
** Example 'PROG001': Example program for the Map Tutorial
************************************************************************
 DEFINE DATA LOCAL
 01 PERS-VIEW VIEW OF EMPLOYEES
   02 NAME
   02 FIRST-NAME
   02 CITY
 *
 01 #NAME (A20)
 01 #CITY (A20)
 END-DEFINE
 *
 REPEAT
   /*
   INPUT USING MAP 'MAP001'
   /*
   IF #CITY NE ' ' AND #NAME NE ' '
     FIND PERS-VIEW WITH NAME = #NAME AND CITY = #CITY
       IF NO RECORDS FOUND
         REINPUT 'NO ONE BY THIS NAME LIVING IN THIS CITY.'
                 MARK *#CITY
       END-NOREC
       /*
       DISPLAY NOTITLE NAME FIRST-NAME CITY
       /*
     END-FIND
   ELSE
     IF #NAME NE ' '
       FIND PERS-VIEW WITH NAME = #NAME
         IF NO RECORDS FOUND
           REINPUT 'PLEASE TRY ANOTHER NAME.'
         END-NOREC
         /*
         DISPLAY NOTITLE NAME FIRST-NAME CITY
         /*
       END-FIND
     END-IF
   END-IF
   /*
 END-REPEAT
 END
```

2 Falls in der obersten Zeile des Editors kein Programmname angezeigt wird, setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das Kommando `STOW PROG001` ab.

Oder:

Wenn in der obersten Zeile des Editors der Programmname `PROG001` angezeigt wird, setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das Kommando `STOW` ab.

Falls erforderlich: Korrigieren Sie eventuell vorhandene Syntaxfehler und wiederholen Sie das Kommando STOW.

Nachdem das Kommando STOW erfolgreich ausgeführt worden ist, wird das Programm als Source-Objekt und als katalogisiertes Objekt in der aktuellen Library in der aktuellen Systemdatei gespeichert.

3 Setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das folgende Kommando ab:

```
RUN
```

Das Programm PROG001 wird ausgeführt und die Maske MAP001 erscheint.

Um zu prüfen, ob die Maske MAP001 wie beabsichtigt funktioniert:

1 Drücken Sie Enter, ohne etwas einzutippen.

Die folgende Meldung wird angezeigt: PLEASE TYPE IN A NAME.

2 Geben Sie im ersten Eingabefeld ein Fragezeichen (?) ein und drücken Sie Enter.

Die Hilfe-Maske HELP001 erscheint.

3 Drücken Sie Enter.

Die Hilfe-Maske verschwindet.

4 Geben Sie im ersten Eingabefeld den Namen MCKENNA ein und drücken Sie Enter.

Die folgende Meldung wird angezeigt: PLEASE TRY ANOTHER NAME (*Versuchen Sie es bitte mit einem anderen Namen.*).

5 Ersetzen Sie MCKENNA durch JONES und drücken Sie Enter.

Das Programm erzeugt die folgende Liste:

```
        NAME             FIRST-NAME             CITY
-------------------- -------------------- --------------------

JONES                VIRGINIA             TULSA
JONES                MARSHA               MOBILE
JONES                ROBERT               MILWAUKEE
JONES                LILLY                BEVERLEY HILLS
JONES                EDWARD               CAMDEN
JONES                MARTHA               KALAMAZOO
JONES                LAUREL               BALTIMORE
JONES                KEVIN                DERBY
JONES                GREGORY              NOTTINGHAM
```

6 Drücken Sie Enter.

Die Maske `MAP001` erscheint

7 Geben Sie im ersten Eingabefeld der Maske den Namen `JONES` und im zweiten Eingabefeld den Namen `DUNFERMLINE` ein.

8 Drücken Sie Enter.

Die folgende Meldung wird angezeigt: `NO ONE BY THIS NAME LIVING IN THIS CITY` (*Es existiert niemand mit diesem Namen in dieser Stadt.).*

9 Geben Sie im ersten Eingabefeld der Maske den Namen `JONES` und im zweiten Eingabefeld den Namen `TULSA` ein.

10 Drücken Sie Enter.

Das Programm erzeugt die folgende Liste:

```
        NAME             FIRST-NAME             CITY
-------------------- -------------------- --------------------

JONES                VIRGINIA             TULSA
```

11 Drücken Sie Enter.

Die Maske `MAP001` erscheint

12 Geben Sie im ersten Eingabefeld einen Punkt (.) ein und drücken Sie Enter.

Der Editier-Bereich des Programm-Editors erscheint. Er enthält den Quellcode des Programms `PROG001`.

Sie können nun fortfahren mit *Maske für WRITE erstellen und Feld-Definitionen kopieren*.

32 Maske für WRITE erstellen und Feld-Definitionen kopieren

In dieser Sitzung werden Sie eine Maske erstellen, indem Sie Felddefinitionen aus einem Datendefinitionsmodul (DDM) kopieren. Außerdem werden Sie die Maskeneinstellungen so ändern, dass Sie das Statement `WRITE USING MAP` benutzen können.

Um eine Maske aus DDM-Feldern zu erstellen:

1 Setzen Sie am Eingabeaufforderungszeichen (`>`) des Programm-Editors das folgende Kommando ab:

```
EDIT MAP
```

Das Menü **Edit Map** erscheint:

2 Führen Sie die Funktion **Initialize new Map** aus, um eine Maske mit dem Namen `MAP002` zu erstellen.

Der Bildschirm **Define Map Settings For MAP** erscheint.

3 Ändern Sie die Seitenlänge (**Page Size**) auf `60`. Markieren Sie **WRITE Statement**, indem Sie ein `X` in das daneben befindliche Eingabefeld eingeben. Entfernen Sie die Markierung bei **INPUT Statement**, indem Sie das daneben befindliche `X` durch Drücken der Leertaste überschreiben.

4 Drücken Sie Enter.

Die **WRITE Statement**-Option ist markiert, was bedeutet, dass diese Maske nur mit einem `WRITE USING MAP`-Statement aus einem Programm aufgerufen weren kann.

5 Drücken Sie Enter *zweimal*.

Der Masken-Editier-Bildschirm erscheint:

6 Geben Sie neben den **Ob**-Eingabefeld im oberen Bereich des Bildschirms Folgendes ein:

```
V EMPLOYEES
```

7 Drücken Sie Enter.

Die Felddefinitionen des Datendefinitionsmoduls (DDM) EMPLOYEES werden aufgelistet:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID              A8          .     T  D    Blnk    T  I    ?
. FULL-NAME                 *G1         .     A  D    _       A  I    )
2 FIRST-NAME                A20         .     A  N    ¬       M  D    &
3 MIDDLE-I                  A1          .     M  I    :       O  D    +
4 NAME                      A20         .     O  I    (
5 MIDDLE-NAME               A20         .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

8 Geben Sie im Editier-Bildschirm Folgendes ein:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID                  A8      .     T  D    Blnk    T  I    ?
. FULL-NAME                     *G1     .
2 FIRST-NAME                    A20     .
3 MIDDLE-I                      A1      .                      O  D    +
4 NAME                          A20     .     O  I    (
5 MIDDLE-NAME                   A20     .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(4
(2
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

Dadurch werden zwei Felder für die Maske definiert, deren Definitionen aus dem DDM kopiert werden: Die eingegebene Folgenummer 4 bewirkt das Kopieren der Definition des entsprechenden Feldes `NAME` und die Nummer 2 bewirkt das Kopieren der Definition des entsprechenden Feldes `FIRST-NAME`.

9 Drücken Sie Enter.

Die Felddefinitionen werden aus dem DDM in die Maske kopiert:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID                   A8     .     T  D    Blnk    T  I    ?
. FULL-NAME                      *G1    .
2 FIRST-NAME                     A20    .
3 MIDDLE-I                       A1     .                      O  D    +
4 NAME                           A20    .     O  I    (
5 MIDDLE-NAME                    A20    .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX
(XXXXXXXXXXXXXXXXXXX




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

10 Benutzen Sie das Kommando `.M`, um das in der zweiten Zeile des Editier-Bereichs befindliche Feld an die unten gezeigte Position zu verschieben:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
1 PERSONNEL-ID             A8           .     T  D    Blnk    T  I    ?
. FULL-NAME                *G1          .
2 FIRST-NAME               A20          .
3 MIDDLE-I                 A1           .                      O  D    +
4 NAME                     A20          .     O  I    (
5 MIDDLE-NAME              A20          .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXX


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

11 Ersetzen Sie im **Ob**-Eingabefeld das `V` durch eine Pluszeichen (+) und drücken Sie Enter.

Die nächste Seite mit Felddefinitionen wird angezeigt.

12 Wiederholen Sie den vorigen Schritt, bis das Feld `CITY` in der Liste erscheint. Ersetzen Sie das `V` durch ein Minuszeichen (-), wenn Sie eine Seite in der Liste nach oben blättern müssen.

13 Geben Sie `CITY:(2` so wie unten gezeigt ein:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
. FULL-ADDRESS              *G1         .     T  D    Blnk    T  I    ?
1 ADDRESS-LINE              A20         .     A  D    _       A  I    )
2 CITY                      A20         .     A  N    ¬       M  D    &
3 ZIP                       A10         .     M  I    :       O  D    +
4 POST-CODE                 A10         .     O  I    (
5 COUNTRY                   A3          .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXX CITY:(2
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

14 Drücken Sie Enter.

Der Bildschirm sieht jetzt folgendermaßen aus:

```
Ob V EMPLOYEES                          Ob D CLS ATT  DEL    CLS ATT  DEL
. FULL-ADDRESS              *G1         .     T  D    Blnk    T  I    ?
1 ADDRESS-LINE              A20         .     A  D    _       A  I    )
2 CITY                      A20         .     A  N    ¬       M  D    &
3 ZIP                       A10         .     M  I    :       O  D    +
4 POST-CODE                 A10         .     O  I    (
5 COUNTRY                   A3          .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXX CITY:(XXXXXXXXXXXXXXXXXXX




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

15 Drücken Sie PF3.

Das Menü **Edit Map** erscheint.

16 Katalogisieren und speichern Sie die Maske `MAP002`, indem Sie die Funktion **Stow Map** ausführen.

Sie können nun fortfahren mit *Layout einer Maske wiederverwenden*.

33 Layout einer Maske wiederverwenden

In dieser Sitzung erstellen Sie eine Maske, indem Sie das Layout einer anderen Maske benutzen.

Um eine Maske auf der Basis eines Masken-Layouts zu erstellen:

1 Führen Sie im Menü **Edit Map** die Funktion **Initialize new Map** aus, um eine Maske mit dem Namen `MAP003` zu erstellen.

2 Ändern Sie im Bildschirm **Define Map Settings for MAP** die Seitenlänge (**Page Size**) auf `60`, markieren Sie **WRITE Statement**, und entfernen Sie die Markierung bei **INPUT Statement**, und geben Sie `MAP002` neben **Layout** ein.

3 Drücken Sie Enter.

Die Masken-Einstellungen sehen nun folgendermaßen aus:

```
16:57:39              Define Map Settings for MAP                 2007-12-14

 Delimiters           Format                         Context
 -----------------    ----------------------------   --------------------------
 Cls Att CD  Del      Page Size ...... 60            Device Check .... ________
  T   D      BLANK    Line Size ...... 79            WRITE Statement   X
  T   I      ?        Column Shift ... 0 (0/1)       INPUT Statement   _
                      Layout ......... MAP002__      Help  ____________________
                       dynamic ....... N (Y/N)        as field default N (Y/N)
                      Zero Print ..... N (Y/N)
                      Case Default ... UC (UC/LC)
                      Manual Skip .... N (Y/N)       Automatic Rule Rank  1
  O   D      +        Decimal Char ... .             Profile Name .... SYSPROF
  O   I      (        Standard Keys .. N (Y/N)
                      Justification .. L (L/R)       Filler Characters
                      Print Mode ..... __            -----------------------
                                                     Optional, Partial ....
                      Control Var .... ________      Required, Partial ....
                                                     Optional, Complete ...
                                                      Required, Complete ...

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

4 Drücken Sie Enter.

Der Masken-Editier-Bildschirm erscheint mit dem Layout der Maske MAP002.

5 Löschen Sie CITY:(XXXXXXXXXXXXXXXXXXX, indem Sie das Feld-Kommando .T benutzen.

6 Verschieben Sie das zweite der verbleibenden Ausgabefelder in die unten gezeigte Position, indem Sie das Feld-Kommando .M benutzen..

Der Bildschirm sieht dann folgendermaßen aus:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .
.                                       .
.                                       .                     O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX             (XXXXXXXXXXXXXXXXXXX




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

7 Fügen Sie den Text `FIRST NAME:` so wie unten gezeigt in die Zeile ein:

```
Ob _                                    Ob D CLS ATT  DEL    CLS ATT  DEL
.                                       .     T  D    Blnk    T  I    ?
.                                       .
.                                       .
.                                       .                      O  D    +
.                                       .     O  I    (
.                                       .
001   --010---+----+----+---030---+----+----+---050---+----+----+---070---+----
NAME:(XXXXXXXXXXXXXXXXXXX FIRST NAME: (XXXXXXXXXXXXXXXXXXX




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test  Edit  --    -     +     Full  <     >     Let
```

8 Drücken Sie PF3.

Das Menü **Edit Map** erscheint.

9 Katalogisieren und speichern Sie die Maske `MAP003`, indem Sie die Funktion **Stow Map** ausführen.

Sie können nun fortfahren mit *Maske mit WRITE USING MAP aufrufen*.

34 Maske mit WRITE USING MAP aufrufen

In dieser Sitzung erstellen Sie und führen Sie das Beispiel-Programm `PROG002` aus, um die Wirkung des Statements `WRITE USING MAP` zu testen, das die Masken `MAP002` und `MAP003` aufruft.

Um das Programm PROG002 zu erstellen:

1 Geben Sie in der Kommandozeile des Menüs **Edit Map** Folgendes ein:

```
EDIT PROG002
```

Der Programm-Editor wird aufgerufen und der Quellcode des Programms `PROG002` wird im Editier-Bereich angezeigt.

2 Setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das folgende Kommando ab:

```
SAVE PROG002
```

Eine Kopie des Programms `PROG001` wird als Source-Objekt unter dem Namen `PROG002` in der aktuellen Natural-Library in der aktuellen Systemdatei gespeichert.

3 Setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das folgende Kommando ab:

```
READ PROG002
```

Der Quellcode des Programms `PROG002` wird im Editier-Bereich des Programm-Editors angezeigt.

4 Ersetzen Sie die `DISPLAY`-Statements durch die unten fett hervorgehoben gezeigten Zeilen:

PROG002:

```
** Example 'PROG002': Example program for the Map Tutorial
************************************************************************
DEFINE DATA LOCAL
01 PERS-VIEW VIEW OF EMPLOYEES
  02 NAME
  02 FIRST-NAME
  02 CITY
*
01 #NAME (A20)
01 #CITY (A20)
END-DEFINE
*
REPEAT
  /*
  INPUT USING MAP 'MAP001'
  /*
  IF #CITY NE ' ' AND #NAME NE ' '
    FIND PERS-VIEW WITH NAME = #NAME AND CITY = #CITY
      IF NO RECORDS FOUND
        REINPUT 'NO ONE BY THIS NAME LIVING IN THIS CITY.'
                MARK *#CITY
      END-NOREC
      /*
      AT START OF DATA
        WRITE 'THE FOLLOWING EMPLOYEES LIVE IN' CITY
      END-START
      WRITE USING MAP 'MAP003'
      /*
    END-FIND
  ELSE
    IF #NAME NE ' '
      FIND PERS-VIEW WITH NAME = #NAME
        IF NO RECORDS FOUND
          REINPUT 'PLEASE TRY ANOTHER NAME.'
        END-NOREC
        /*
        WRITE USING MAP 'MAP002'
        /*
      END-FIND
    END-IF
  END-IF
  /*
END-REPEAT
END
```

5 Wenn Sie alle Änderungen durchgeführt haben, setzen Sie am Eingabeaufforderungszeichen (>) des Programm-Editors das folgende Kommando ab:

```
STOW
```

Das Source-Objekt PROG002 wird aktualisiert und ein katalogisiertes Objekt wird in der aktuellen Natural-Library in der aktuellen Systemdatei gespeichert.

Um das Programm PROG002 auszuführen:

1 Geben Sie am Eingabeaufforderungszeichen (>) des Programm-Editors Folgendes ein:

```
PROG002
```

2 Die Ausführung des Programms beginnt und die Maske MAP001 erscheint.

3 Geben Sie den Namen JONES ein und lassen Sie das zweite Eingabefeld leer:

4 Drücken Sie Enter.

Für die vom Programm erzeugte Liste wird nun die Maske MAP002 benutzt:

```
Page      1                                              07-12-14  17:12:41

NAME: JONES               VIRGINIA            CITY: TULSA
NAME: JONES               MARSHA              CITY: MOBILE
NAME: JONES               ROBERT              CITY: MILWAUKEE
NAME: JONES               LILLY               CITY: BEVERLEY HILLS
NAME: JONES               EDWARD              CITY: CAMDEN
NAME: JONES               MARTHA              CITY: KALAMAZOO
NAME: JONES               LAUREL              CITY: BALTIMORE
NAME: JONES               KEVIN               CITY: DERBY
NAME: JONES               GREGORY             CITY: NOTTINGHAM
```

5 Drücken Sie Enter.

Die Maske MAP001 erscheint:

6 Lassen Sie den Namen JONES stehen und geben Sie die Stadt DERBY ein.

7 Drücken Sie Enter.

Für die vom Programm erzeugte Liste wird nun die Maske MAP003 benutzt:

```
Page      2                                                07-12-14  17:15:18

THE FOLLOWING EMPLOYEES LIVE IN DERBY
NAME: JONES               FIRST NAME:  KEVIN
```

8 Drücken Sie Enter.

Die Maske MAP001 erscheint:

9 Geben Sie an der ersten Stelle des Eingabefeldes NAME einen Punkt ein.

Das Menü **Edit Map** erscheint.

Sie haben das Tutorial erfolgreich abgeschlossen.

VIII DDM-Editor (SYSDDM Utility)

Dieses Dokument beschreibt, wie Sie Natural-Datendefinitionsmodule (*Data Definition Modules*/DDMs) erstellen, pflegen und löschen können.

Anmerkung: Bei Natural für Großrechner ist der Editor, mit dem die Datendefinitionsmodule (DDMs) erstellt, gepflegt und gelöscht werden können, als Utility SYSDDM implementiert.

Funktionsbeschreibung

SYSDDM aufrufen und beenden

SYSDDM Pflege- und Service-Funktionen benutzen

DDMs erstellen

DDM-Editor aufrufen und beenden

DDM-Editor benutzen

DDM katalogisieren

DDMs auflisten

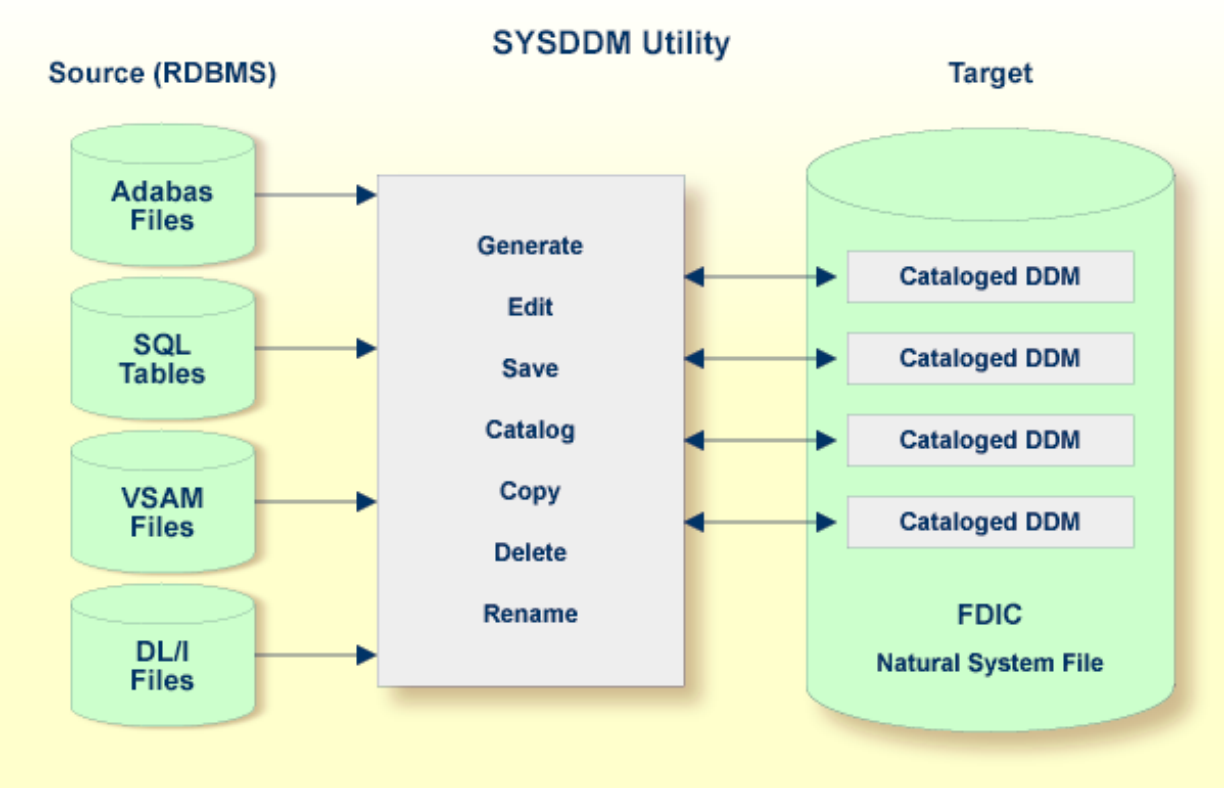
DDMs in unterschiedlichen Umgebungen pflegen

35 Funktionsbeschreibung

- DDMs speichern 382
- Einschränkungen bei der Benutzung 383

Die SYSDDM Utility wird verwendet, um ein Natural-Datendefinitionsmodul (Data Definition Modules/DDM) aus einer Datenbankdatei oder aus einem anderen DDM zu erstellen. Ein Natural-Objekt (z.B. ein Programm oder ein Datenbereich (Data Area) kann auf eine Datenbankdatei nur dann zugreifen, wenn für diese Datei ein entsprechendes DDM erstellt und als katalogisiertes Objekt gespeichert worden ist. Weitere Informationen zu DDMs siehe *Datendefinitionsmodul (DDM)* im *Leitfaden zur Programmierung*.

Die folgende Abbildung veranschaulicht die wesentlichen Merkmale und grundlegenden Funktionsprinzipien bei der Verarbeitung von DDMs mit der SYSDDM Utility:



DDMs speichern

DDMs werden in der Natural-Systemdatei FDIC gespeichert. In der Systemdatei FDIC befinden sich alle DDMs, die in Ihrer Natural-Umgebung vorhanden sind. Das Speichern einer DMM und der Zugriff darauf sind nur in der Systemdatei FDIC möglich.

DDMs können nicht nur in Datei in einer Adabas-Datenbank, sondern auch in einem anderen Datenbankverwaltungssystem gespeichert werden. Einige in der *SYSDDM Utility* beschriebenen Optionen gelten nur bei Adabas. Bei Verwendung eines anderen Datenbankverwaltungssystems können diese Optionen ignoriert werden.

Einschränkungen bei der Benutzung

Dieser Abschnitt beschreibt Einschränkungen, die gelten können, wenn Funktionen der SYSDDM Utility mit einem der folgenden Produkte benutzt wird:

- Predict
- Natural Security

Predict

Um die Integrität der Daten bei DDMs, die in Predict definiert werden, zu garantieren, kann der Predict-Administrator die Benutzung von Funktionen der SYSDDM Utility zum Bearbeiten, Kopieren, Erstellen, Löschen, Umbenennen und Katalogisieren, für die es in Predict gleichwerte Funktionen gibt, einschränken.

Grundsätzlich empfehlen wir, *keine* Funktionen der SYSDDM Utility zu benutzen, die alternativ von Predict ausgeführt werden können.

Wenden Sie sich bitte an Ihren Predict-Administrator, wenn Sie weitere Informationen benötigen.

Natural Security

Wenn Natural Security installiert worden ist, kann der Zugriff auf DDMs eingeschränkt werden. Im DDM Security-Profil kann eine Definition vorhanden sein, die festlegt, ob ein DDM nur durch bestimmte Benutzer („DDM Modifiers“) oder durch den Eigentümer des Security-Profils geändert werden kann.

Informationen bezüglich des Schutzes von DDMs auf Großrechnern siehe *Protecting DDMs On Mainframes* in der *Natural Security*-Dokumentation.

36 SYSDDM aufrufen und beenden

Dieser Abschnitt erklärt, wie Sie die SYSDDM Utility aufrufen und beenden.

Um die SYSDDM Utility aufzurufen:

- Setzen Sie das folgende Natural-Systemkommando in der Kommandozeile ab:

  ```
  SYSDDM
  ```

  Oder:

  1. Wählen Sie im Natural-Hauptmenü die Option **Maintenance and Transfer Utilities** (Pflege- und Übertragungs-Utilities) aus.

     Das Menü **Maintenance and Transfer Utilities** wird angezeigt.
  2. Wählen Sie im Menü **Maintenance and Transfer Utilities** die Funktion **Maintain DDMs** (DDMs pflegen).

  Das Menü der SYSDDM Utility erscheint (Beispiel):

```
User SAG                          - Menu -                      FDIC (10,460)
                                                              Work area empty
             DDM Maintenance         List/Copy Services

             E  Edit DDM             L  List DDMs
             R  Read DDM             X  List DDMs with Additional Information
             C  Catalog DDM          S  Show Defined DBIDs and Used FNRs
             U  Delete DDM           M  Copy DDM to Another FDIC File
             ?  Help
             .  Exit                 Other Services

                                     G  Generate DDM from Adabas FDT
                                      B  SQL Services (NDB)
                                     D  DL/I Services
                                     Z  SQL Services (NSB)
 Code ...... _                                FDIC Type ........ A
 DDM Name .. ____________________________      DDM Type ......... _
 FNR ....... 0     DBID .. 0                  Adabas Password ..
 Replace ... N                                DBID Type ........ 7

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

Informationen zu den Feldern und Funktionen im Menü der SYSDDM Utility siehe *SYSDDM Pflege- und Service-Funktionen benutzen*.

Um die SYSDDM Utility zu beenden:

- Drücken Sie im Menü der SYSDDM-Utility die Taste PF3 (Exit).

  Oder:

  Geben Sie in der Kommandozeile einen Punkt (.) ein und drücken Sie Enter.

37 SYSDDM Pflege- und Service-Funktionen benutzen

- Hilfe zu Funktionen 388
- Funktionen ausführen 389
- Beschreibung der Funktionen 389
- DDM-Parameter angeben 393

Die im Menü der SYSDDM Utility verfügbaren Funktionen ermöglichen das Erstellen, Anzeigen, Bearbeiten, Umbenennen oder Löschen eines Datendefinitionsmoduls (DDM).

Dieser Abschnitt beschreibt die Felder und Funktionen, die im Menü der SYSDDM Utility aufgerufen werden können, und erklärt, wie DDMs in unterschiedlichen Umgebungen gepflegt werden.

Hilfe zu Funktionen

Dieser Abschnitt erklärt, wie Sie Informationen zu den Feldern und Funktionen erhalten, die im Menü der SYSDDM Utility vorhanden sind.

Um SYSDDM-Hilfe-Informationen anzuzeigen:

1 Drücken Sie im SYSDDM Utility-Bildschirm die Taste PF1.

Oder:

Geben Sie in der Kommandozeile bzw. in einem Eingabefeld ein Fragezeichen (?) ein und drücken Sie Enter.

Es erscheint das **Help Main Menu** der SYSDDM Utility.

2 Platzieren Sie den Cursor neben das Größer-als-Zeichen (>) des gewünschten Hilfethemas und drücken Sie PF1.

Der SYSDDM-Hilfe-Bildschirm zu dem gewählten Thema wird angezeigt.

- Um vorwärts zu blättern, drücken Sie Enter.
- Um zum **Help Main Menu** zurückzukehren, drücken Sie PF3.
- Um die Hilfefunktion zu beenden und zum SYSDDM Utility-Menü zurückzukehren, drücken Sie PF12.

3 Um das **Help Main Menu** zu verlassen, geben Sie in der Kommandozeile einen Punkt (.) ein und drücken Sie Enter oder drücken Sie PF3.

Das SYSDDM Utility-Menü wird angezeigt.

Funktionen ausführen

Dieser Abschnitt beschreibt, wie Sie eine Funktion aus dem SYSDDM Utility-Menü ausführen.

Um eine Funktion auszuführen:

- Im SYSDDM Utility-Menü:
  - Wählen Sie im Abschnitt **DDM Maintenance**, **List/Copy Services** oder **Other Services** den einstelligen Code, der neben dem Namen der Funktion angezeigt wird, geben Sie ihn im Feld **Code** ein und drücken Sie Enter.

    Um zum Beispiel ein DDM zu bearbeiten, geben Sie den Funktionscode `E` ein.

    Erklärung der Funktionen siehe *Beschreibung der Funktionen*.
  - Geben Sie in den Eingabefeldern neben dem Feld **Code** einen gültigen Wert ein, um das (oder die) DDM(s) anzugeben, das (die) bearbeitet werden soll(en). Siehe Abschnitt *DDM-Parameter angeben*.

Beschreibung der Funktionen

Dieser Abschnitt beschreibt die Funktionen, die SYSDDM Utility-Menü zur Verfügung stehen, und listet die entsprechenden Funktionscodes.

Funktionscode	Feld	Funktion
E	**Edit DDM**	Ruft den DDM-Editor auf und liest eine DDM-Source aus der Systemdatei FDIC in den Source-Bereich des Editors.
R	**Read DDM**	Liest eine DDM-Source aus der Systemdatei FDIC in den Source-Bereich des Editors, ruft aber *nicht* den Editor auf.
C	**Catalog DDM**	Speichert das zurzeit im Source-Bereich des Editors enthaltene DDM als katalogisiertes Objekt in der aktuellen Systemdatei FDIC. Informationen zu den Namenskonventionen, die beim Katalogisieren eines DDM gelten, siehe *Namenskonventionen für Objekte* in der *Natural benutzen*-Dokumentation. Falls der Source-Bereich des Editors leer ist, benutzen Sie die Funktion **Generate DDM from Adabas FDT** oder **Edit DDM**, um eine DDM-Source in den Source-Bereich des Editors zu laden.

<table>
<tr><th>Funktionscode</th><th>Feld</th><th>Funktion</th></tr>
<tr><td></td><td></td><td>Bei einem DDM aus VSAM (DDM Type auf <code>V</code> gesetzt - siehe auch DDM Type in DDM-Parameter angeben) werden Sie aufgefordert, zusätzliche Angaben zu machen.

Weitere Informationen siehe Natural File Access in der Database Management System Interfaces-Dokumentation.</td></tr>
<tr><td>U</td><td>Delete DDM</td><td>Löscht ein (oder mehrere) katalogisierte(s) DDM(s) aus der Systemdatei FDIC.

Der Inhalt des Editor-Source-Bereichs ist von der Löschung nicht betroffen.

Beim Löschen eines DDM mit der Utility SYSDDM wird außerdem auch das entsprechende Natural Security-Dateiprofil gelöscht.</td></tr>
<tr><td>L</td><td>List DDMs</td><td>Zeigt eine einzelne DDM-Source an (dazu wird der DDM-Editor nicht aufgerufen) oder eine Liste von DDMs, die in der angegebenen FDIC-Systemdatei gespeichert sind - ähnlich wie beim Natural-Systemkommando <code>LIST DDM</code> (siehe Systemkommandos-Dokumentation). Anders als beim Systemkommando <code>LIST DDM</code> werden jedoch bei der Funktion List DDMs zusätzlich alle DDMs angezeigt, für die in Natural Security (falls installiert) keine Lese- oder Änderungsberechtigung definiert ist. Weitere Informationen siehe Protecting DDMs On Mainframes in der Natural Security-Dokumentation.

In der Liste der DDMs, die im Bildschirm LIST DDMs angezeigt wird, können Sie ein DDM zur Weiterverarbeitung auswählen, indem Sie das für die gewünschte Aktion erforderliche Zeilenkommando in der Spalte Cmd absetzen. Informationen zu den möglichen Zeilenkommandos erhalten Sie, indem Sie ein Fragezeichen (?) in der Spalte Cmd eingeben.</td></tr>
<tr><td>X</td><td>List DDMs with Additional Information</td><td>Zeigt eine Liste von DDMs, die in der angegebenen FDIC-Systemdatei gespeichert sind. Die Liste enthält folgende Informationsfelder:
<table>
<tr><td></td><td></td></tr>
<tr><td>DBID</td><td>Die Datenbankkennung: siehe DBID in DDM-Parameter angeben.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>FNR</td><td>Die Dateinummer: siehe FNR in DDM-Parameter angeben.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>DDM Typ</td><td>Der DDM-Typ: siehe DDM Type in DDM-Parameter angeben.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Bytes</td><td>Die Größe des DDM in Bytes.</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
</table>

Funktionscode	Feld	Funktion	
		Secur. Type	Gilt nur, wenn Natural Security installiert ist. Der Security-Typ: ■ `Public` ■ `Private` ■ `Access` ■ `Undef` Siehe auch Status eines DDM im Abschnitt *Components of a File Profile* in der *Natural Security*-Dokumentation).
		File Type	Dateityp - Gilt nur bei DDMs, die aus VSAM erstellt wurden oder eine PersonalDB-Datei aus Super Natural. Der VSAM-Dateityp: `Log.View` (logische Datensicht), `Phy.File` (physische Datensicht) oder `Log.File` (logische Datei). Super Natural: `Userfile` zeigt an, dass das DDM in Super Natural wurde.
		File Name	Dateiname - Gilt nur bei DDMs aus VSAM-Dateien. Der Name einer logischen oder physischen Datei.
		Remark	Eine Bemerkung, z.B. `SupNat` (bei einer Benutzerdatei `Userfile` aus Super Natural) oder die VSAM-Dateiorganisation (`KSDS`, `RRDS`, `ESDS` oder `VRDS`).
		In der Liste, die im Bildschirm **List DDMs with Additional Information** angezeigt wird, können Sie ein DDM zur Weiterverarbeitung auswählen, indem Sie das für die gewünschte Aktion erforderliche Zeilenkommando in der Spalte **Cmd** absetzen. Informationen zu den möglichen Zeilenkommandos erhalten Sie, indem Sie ein Fragezeichen (?) in der Spalte **Cmd** eingeben.	
S	**Show Defined DBIDs and Used FNRs**	See *Definierte Datenbankkennungen und verwendete Dateinummern anzeigen*.	

Funktionscode	Feld	Funktion
M	**Copy DDM to Another FDIC File**	Kopiert ein oder mehrere DDMs von einer Systemdatei FDIC in eine andere. Dies kann beispielsweise erforderlich sein, wenn eine Natural-Anwendung vom Test- in den Produktionsstatus überführt wird. Diese Funktion ruft das Fenster **Copy DDMs** des Natural-Dienstprogramms (Utility) SYSMAIN auf, in dem Sie die Quell- und die Zielumgebung des oder der zu kopierenden DDMs angeben können. Weitere Informationen siehe *DDMs bearbeiten* in der *SYSMAIN Utility*-Dokumentation.
G	**Generate DDM from Adabas FDT**	Generiert ein DDM aus einer Adabas Feld Definition Table (FDT) und stellt es zur Weiterverarbeitung in den Source-Bereich. Wird als DBID eine 0 (Null) angegeben, dann wird die Standard-DBID benutzt, die im Profilparameter `UDB` im Natural-Parametermodul angegeben ist.
B	**SQL Services (NDB)**	Nur verfügbar, wenn Natural for Db2 (NDB) installiert ist. Die **SQL Services (NDB)** werden benutzt, um DDMs aus Db2-Tabellen zu generieren. Weitere Informationen siehe *SQL Services (NDB)* (Natural for Db2) in der *Database Management System Interfaces*-Dokumentation.
D	**DL/I Services**	Nur verfügbar, wenn Natural for DL/I installiert ist. Die **DL/I Services** werden zur Pflege einer Natural for DL/I Umgebung benutzt. Sie bieten Funktionen zur Suche innerhalb von und für Änderungen an Strukturen wie z.B. DL/I Datenbankbeschreibungen (DBDs), Program Specification Blocks (PSBs), Program Communication Blocks (PCBs), DDMs und Segment-Layouts. Weitere Informationen zu *DL/I Services* siehe *Database Management System Interfaces*-Dokumentation.
Z	**SQL Services (NSB)**	Diese Funktion wird nicht mehr unterstützt.

Definierte Datenbankkennungen und verwendete Dateinummern anzeigen

Die Funktion **Show Defined DBIDs and Used FNRs** zeigt an, welche Datenbankkennungen (DBIDs) definiert sind. Außerdem zeigt sie alle Dateinummern (FNRs) zu einer gegebenen DBID, für die DDMs definiert worden sind.

Wenn Sie diese Funktion aufrufen, erscheint ein Menü, in dem Sie die nachfolgend beschriebenen Unterfunktionen auswählen können.

- In Natural definierte Datenbankkennungen

- Dateinummern von existierenden DDMs für eine Datenbank

In Natural definierte Datenbankkennungen

Diese Funktion listet alle Datenbankkennungen (DBIDs) und Datenbanktypen, die in dem Natural-Profilparameter `DB` angegeben sind (siehe auch *DB - Datenbanktypen und -optionen* in der *Parameter-Referenz*-Dokumentation). Die Liste enthält nicht die Datenbankkennung des Standard-Datenbanktyps. Diese wird im oberen Bereich des Bildschirms angezeigt.

Dateinummern von existierenden DDMs für eine Datenbank

Diese Funktion listet zu einer gegebenen Datenbankkennung (DBID) die Nummern aller Dateien, für die DDMs definiert worden sind.

Um die Funktion aufzurufen:

- Geben Sie im Menü **Show Defined DBIDs and Used FNRs** im Feld **Code** ein `F` ein und ändern Sie, falls erforderlich, die standardmäßig eingetragene Datenbankkennung (DBID).

  Oder:

  Geben Sie im Bildschirm **Database IDs Defined in Natural** in der Kommandozeile eine Datenbankkennung ein (gültige DBID-Werte sind `1 - 65535`) und drücken Sie PF5.

DDM-Parameter angeben

Zu jeder im Menü der SYSDDM Utility vorhandenen Funktion können Sie einen bzw. mehrere Parameter angegeben, durch die bestimmt wird, welches oder welche DDMs für welche Datenbank verarbeitet werden soll. In diesem Abschnitt werden die Felder beschrieben, in denen Sie diese Parameter eingeben können.

Feld	Erläuterung
DDM Name	Der Name des Datendefinitionsmoduls, das verarbeitet werden soll. Sie können auch einen Namensbereich angeben, indem Sie Stern-Notation (*) benutzen, um alle DDMs oder nur bestimmte DDMs zu verarbeiten. Beispiel: Mit der Eingabe `EMP*` wählen Sie alle DDMs aus, deren Name mit `EMP` beginnt. Bei Bereichsangaben in den Funktionen **Delete DDMs** oder **Copy DDM to Another FDIC File** wird das Dienstprgramm (Utility) SYSMAIN aufgerufen, in der Sie die DDMs weiterverarbeiten können. Weitere Informationen siehe *DDMs bearbeiten* in der *SYSMAIN Utility* -Dokumentation.

Feld	Erläuterung
FNR	Die Dateinummer (FNR) der Datenbankdatei, für die das Datendefinitionsmodul definiert wird bzw. werden soll. Die Dateinummer entspricht dem Typ der Datenbank. Wenn z.B. eine Adabas-Datei benutzt wird, müssen Sie die Adabas-Dateinummer eingeben. Wenn Sie einen DL/I-Segmenttyp benutzen, erfolgt die Angabe der Dateinummer intern durch Natural für DL/I. Bezüglich VSAM-Dateien siehe *Natural for VSAM* in der *Database Management System Interfaces*-Dokumentation. Gültige Werte: `0` - `5000`.
DBID	Die Kennung (DBID) der Datenbank, in der die Datenbankdatei enthalten ist, die von dem Datendefinitionsmodul referenziert wird. Gültige Werte: `1` - `65535`, außer `255`. Wenn `0` (Null) angegeben wird, dann wird die Datenbankkennung benutzt, die im Natural Profilparameter `UDB` im Natural-Parametermodul angegeben ist. Weitere Informationen siehe `UDB` in der *Parameter-Referenz*-Dokumentation. Wenn Sie keine DBID eingeben, wird diese dynamisch zur Ausführungszeit auf der Basis der DBID der Systemdatei FUSER generiert.
Replace	Gibt an, ob ein DDM ersetzt werden soll: `Y` — Ja. Ein DDM, das momentan kopiert oder katalogisiert wird, ersetzt ein DDM mit dem gleichen Namen. `N` — Nein. Existierende DDMs mit dem gleichen Namen werden nicht ersetzt. Dies ist die Standard-Einstellung. Diese Funktion entspricht der `REPLACE`-Option des Kommandos `CATALOG`. Siehe ***Editor- und Systemkommandos***.
FDIC Type	Der Datenbanktyp der Systemdatei. Mögliche Typen wie bei **DDM Type** (siehe unten). Dieses Feld ist ein Ausgabefeld.
DDM Type	Der Typ des Datendefinitionsmoduls, z.B.:

Feld	Erläuterung
	A — Adabas Dies ist der Standardtyp.. V — VSAM 2 — Db2 D — DL/I P — PROCESS (Entire System Server) C — Kommandoprozessor S — Super Natural E — Entire DB Engine Der Typ des Datendefinitionsmoduls entspricht dem Typ der Datenbank, die von dem DDM referenziert wird.
Adabas Password	Das von Adabas (falls installiert) geforderte Passwort.
DBID Type	Der Datenbanktyp der Datenbank, die im Feld **DBID** angegeben ist. Mögliche Typen wie bei **DDM Type** (siehe oben). Ausnahme: Bei einer Adabas-Datenbank wird die Adabas-Version (zum Beispiel 8) angezeigt. Dieses Feld ist ein Ausgabefeld.

38 DDMs erstellen

Dieser Abschnitt beschreibt, wie Sie ein Datendefinitionsmodul anlegen, indem Sie entweder ein vorhandenes DDM kopieren oder indem Sie es direkt aus den Felddefinitionen in einer Datenbank erstellen.

Um ein DDM aus einer Adabas-Datenbank zu erstellen:

1 Geben Sie im Menü der SYSDDM Utility Folgendes ein:

Geben Sie im Feld **Code** entweder den Funktionscode `E` (**Edit DDM**) oder den Funktionscode `G` (**Generate DDM from Adabas FDT**).
Lassen Sie das Feld **Name** leer.
Geben Sie in den Feldern **DBID** und **FNR** die Kennung und die Dateinummer der Datenbank ein, auf die das DDM, das Sie anlegen möchten, zugreifen soll (siehe auch *DDM-Parameter angeben*).

Bei Funktionscode `E`: Der leere Bildschirm **Edit DDM** wird angezeigt. Sie können hier Felddefinitionen eingeben. Weitere Informationen siehe *DDM Editor benutzen*.

Bei Funktionscode `G`: Das Datendefinitionsmodul wird automatisch aus der FDT (Feld Definition Table) generiert, die für die angegebene Datenbank definiert ist. Der Bildschirm **Edit DDM** erscheint. Er enthält alle relevanten Felddefinitionen.

2 Je nachdem, welchen Funktionscode Sie im Bildschirm **Edit DDM** benutzt haben, können Sie jetzt entweder die erforderlichen Felddefinitionen eingeben (siehe *DDM Editor benutzen*) oder die aus der FDT generierten Felddefinitionen ändern.

Wenn Sie mit der Bearbeitung des Datendefinitionsmoduls fertig sind, prüfen Sie die Source und drücken Sie `PF3`.

Das Menü der SYSDDM Utility wird angezeigt.

3 Geben Sie im Menü der SYSDDM Utility Folgendes ein:

Geben Sie im Feld **Name** einen gültigen DDM-Namen ein (siehe auch *Namenskonventionen für Objekte* in *Natural benutzen*). Bei Funktionscode `G` ist das Feld **Name** vorbelegt mit dem Standardnamen `FILE-file-number`. Dabei entspricht *file-number* der im Feld **FNR** angegebenen Dateinummer.
Geben Sie im Feld **Code** den Funktionscode `C` (**Catalog DDM**) ein.

Das DDM wird als katalogisiertes Objekt in der angegebenen Datenbank und Systemdatei gespeichert. Siehe auch *DDM katalogisieren*.

Um ein DDM aus einer VSAM-Datei zu erstellen:

1 Geben Sie im Menü der SYSDDM Utility Folgendes ein:

Geben Sie im Feld **Code** den Funktionscode `E` ein (siehe auch *Beschreibung der Funktionen*).
Geben Sie im Feld **DDM Type** ein `V` ein.
Geben Sie in den Feldern **DBID** und **FNR** die Kennung und die Dateinummer der zu VSAM zugewiesenen Datenbank ein, auf die das DDM, das Sie anlegen möchten, zugreifen soll.
Gültige Feldeingaben siehe *DDM-Parameter angeben*.

Ein leerer Bildschirm **Edit DDM** wird angezeigt.

2 Geben Sie die Felddefinitionen ein. Siehe *DDM-Editor benutzen*.

3 Katalogisieren Sie das DDM. Weitere Informationen siehe *Catalog DDM* in *Natural for VSAM* in der *Database Management System Interfaces*-Dokumentation.

Um ein DDM aus einer Db2-Tabelle zu erstellen:

1 Vorgehensweise bei Db2 siehe *Generate DDM from an SQL Table* in der *Database Management System Interfaces*-Dokumentation.

2 Katalogisieren Sie das DDM. Weitere Informationen siehe *DDM katalogisieren*.

Um ein einzelnes DDM oder mehrere DDMs zu kopieren:

1 Geben Sie im Menü der SYSDDM Utility Folgendes ein:

Geben Sie im Feld **Code** den Funktionscode `M` ein (siehe auch *Beschreibung der Funktionen*).
Geben Sie im Feld **Name** den Namen des zu kopierenden Datendefinitionsmoduls (oder einen Namensbereich) ein.
Geben Sie in den Feldern **DBID** und **FNR** die Datenbankkennung und die Dateinummer der Systemdatei FDIC ein, die das gewünschte DDM (oder die DDMs) enthält.
Gültige Feldeingaben siehe *DDM-Parameter angeben*.

Das Fenster **Copy DDMs (via SYSMAIN)** wird angezeigt.

2 Geben Sie die Quell- und Zielumgebung für das (oder die) zu kopierenden DDM(s) ein. Siehe *DDMs bearbeiten* in der Dokumentation des Dienstprgramms (Utility) SYSMAIN.

Informationen, wie Sie Datendefinitionsmodule zwischen verschiedenen Systemdateien und/oder Hardware-Plattformen kopieren möchten, finden Sie im Abschnitt ***DDMs in unterschiedlichen Umgebungen pflegen***.

39 DDM-Editor aufrufen und beenden

- Editor aufrufen 402
- Editor-Sitzung beenden 403

Mit dem DDM-Editor können Sie die Source eines Datendefinitionsmoduls bearbeiten.

Dieser Abschnitt beschreibt, wie Sie den DDM-Editor im Menü der SYSDDM Utility aufrufen und beenden.

Editor aufrufen

Dieser Abschnitt beschreibt, wie Sie den DDM-Editor im Menü der SYSDDM Utility aufrufen, um ein existierendes DDM zu ändern oder um ein neues DDM zu erstellen.

Um den DDM-Editor aufzurufen und ein existierendes DDM zu ändern:

- Geben Sie im Feld **Code** im Menü der SYSDDM Utility den Funktionscode `E` und im Feld **DDM Name** den Namen eines DDM ein.

  Wenn das angegebene DDM existiert, wird der DDM-Quellcode in den Source-Bereich gelesen und der Bildschirm **Edit DDM** erscheint (Beispiel):

```
16:03:29                 ***** Edit DDM (ADA) *****                2004-11-22
DDM Name EMPLOYEES                        Def.Seq.        DBID     0 FNR   316
Command
I T L DB Name                            F       Leng S D Remark
- - - -- -------------- top ------------ - ---------- - - --------------------
    1 AA PERSONNEL-ID                    A          8   D
  *      CNNNNNNN
  *      C=COUNTRY
  G 1 AB FULL-NAME
    2 AC FIRST-NAME                      A         20 N
    2 AD MIDDLE-I                        A          1 N
    2 AE NAME                            A         20   D
    1 AD MIDDLE-NAME                     A         20 N
    1 AF MAR-STAT                        A          1 F
  *      M=MARRIED
  *      S=SINGLE
  *      D=DIVORCED
  *      W=WIDOWED
    1 AG SEX                             A          1 F
    1 AH BIRTH                           D          6   D
    1 AH NJBIRTH                         I          2   D
  G 1 A1 FULL-ADDRESS

DDM EMPLOYEES read into source area.
```

Um den DDM-Editor aufzurufen und ein neues DDM anzulegen, benutzen Sie eine der folgenden Methoden:

- Vorgehensweise bei Adabas: Benutzen Sie die Funktion **Generate DDM from Adabas FDT**. Weitere Informationen siehe *SYSDDM Pflege- und Service-Funktionen benutzen*.

  Vorgehensweise bei Db2: Benutzen Sie die Funktion **Generate DDM from an SQL Table**. Weitere Informationen siehe *Database Management System Interfaces*-Dokumentation.

  Vorgehensweise bei DL/I: Benutzen Sie die Funktion **Generation of DDMs from DL/I Segment Types**. Weitere Informationen siehe *Database Management System Interfaces*-Dokumentation.

  Oder:

  Geben Sie im Feld **Code** im Menü der SYSDDM Utility den Funktionscode `E` ein und lassen Sie das Feld **DDM Name** leer.

  Es erscheint der Bildschirm **Edit DDM** (ähnlich dem oben aufgeführten Beispiel).

  Sie können den Source-Bereich löschen, indem Sie das Kommando `CLEAR` in der Kommandozeile absetzen.

Editor-Sitzung beenden

Dieser Abschnitt beschreibt, wie Sie die Editor-Sitzung beenden und zum Menü der SYSDDM Utility zurückkehren.

Um den DDM-Editor zu verlassen:

- Prüfen und katalogisieren Sie den DDM-Quellcode (siehe *Using the DDM Editor* und *Cataloging a DDM*), der sich im Source-Bereich befindet, und geben Sie in der Kommandozeile des Bildschirms **Edit DDM** einen Punkt (.) ein und drücken Sie Enter oder drücken Sie PF3.

  Das Menü der SYSDDM Utility erscheint.

Anmerkung: Beim DDM-Editor kommt die Editor-Profiloption **Leave Editor with Unlock** zur Anwendung. Sie dient dazu, den Quellcode beim Verlassen des Editors zu entsperren. Weitere Informationen zu dieser Option siehe *Allgemeine Standardwerte* in *Editor-Profil* im Abschnitt *Editoren — Allgemeine Informationen*.

40 DDM-Editor benutzen

- DDM-Kopfzeilen-Informationen 406
- Spalten für Feld-Attribute 407
- Kommandos zum Editieren und Ausführen von Funktionen 413
- Erweiterte Feld-Attribute angeben 419

Der Bildschirm des DDM-Editors (**Edit DDM**) ist in Form einer Tabelle aufgebaut. Darin sind die Daten der Felddefinitionen in Zeilen und Spalten enthalten. Alle Attribute, die zu einem Feld gehören, das zu einem DDM definiert ist, sind in einer Zeile (d.h. in einer Quellcodezeile) angeordnet, und sind durch Tabulatoren voneinander getrennt.

Dieser Abschnitt beschreibt die Spalten, die im Bildschirm **Edit DDM** enthalten sind, und die Kommandos, die zur Verfügung stehen, um zum Beispiel ein DDM-Feld anzulegen oder zu ändern, um in dem Bildschirm zu navigieren oder um eine DDM-Source zu katalogisieren.

DDM-Kopfzeilen-Informationen

Dieser Abschnitt beschreibt die Felder, die im Kopfbereich des Bildschirms **Edit DDM** enthalten sind.

- Beschreibung der Felder im DDM-Kopfbereich

Beschreibung der Felder im DDM-Kopfbereich

Felder im DDM-Kopfbereich	Beschreibung
Edit DDM (*DDM-type*)	Der in Klammern neben dem Bildschirm-Titel **Edit DDM** angezeigte Wert bezeichnet den DDM-Typ, zum Beispiel:
ADA	Adabas
VSAM	VSAM
DB2	Db2
DL/I	DL/I
PROCESS	Entire System Server
CMD-PROC	Kommandoprozessor
SNAT	Super Natural
ENTIREDB	Entire DB Engine
DBID	Die Datenbankkennung (DBID). Weitere Informationen siehe **DBID** im Abschnitt *DDM-Parameter angeben*.
FNR	Die Nummer der Datei, die in der Datenbank referenziert wird. Weitere Informationen siehe **FNR** im Abschnitt *DDM-Parameter angeben*.
DDM Name	Der Name des Datendefinitionsmoduls, das sich zurzeit im Arbeitsbereich des DDM-Editors befindet.

Felder im DDM-Kopfbereich	Beschreibung
Def.Seq.	Die Standard-Reihenfolge, in der die Datei gelesen wird, wenn auf sie mit einem `READ LOGICAL`-Statement in einem Natural-Programm zugegriffen wird. Siehe auch Beschreibung des `READ`-Statement in der *Statements*-Dokumentation. Die Standard-Reihenfolge wird mit dem aus zwei Zeichen bestehenden Kurznamen des Feldes angegeben. Das System wertet den Kurznamen anhand der gewählten Dateinummer aus. Wenn auf die Datenbank zugegriffen werden kann, wird der Kurzname gegen das entsprechende Feld in der Datenbankdatei geprüft. Falls in der Datenbank kein solches Feld existiert, wird eine Auswahlliste mit gültigen Kurznamen angezeigt. Wenn auf die Datenbank nicht zugegriffen werden kann, wird keine Auswahlliste erzeugt. Der Inhalt dieses Feldes kann geändert werden.

Spalten für Feld-Attribute

Dieser Abschnitt beschreibt die Feld-Attribute, die in den Zeilen und Spalten des Bildschirms **Edit DDM** definiert werden können.

Spaltenüberschrift	Feld-Attribute
I	Informationsfeld, das vom Editor ausgefüllt wird. Diese Spalte ist nicht änderbar. Die Buchstaben, die in diesem Feld angezeigt werden können, haben folgende Bedeutung:

`E`	Die mit diesem Buchstaben markierte Zeile enthält einen Fehler, der während der Ausführung eines `CHECK`-Kommandos festgestellt wurde. Siehe auch `CHECK` in ***Editor- und Systemkommandos***.
`S`	Die mit diesem Buchstaben markierte Zeile enthält einen gesuchten Wert. Siehe auch `SCAN` in ***Editor- und Systemkommandos***.
`X`	Die Zeile ist für einen Kopier- oder Verschiebevorgang markiert. Weitere Informationen siehe ***Zeilenkommandos***.
`Y`	Die Zeile ist für einen Kopier- oder Verschiebevorgang markiert. Weitere Informationen siehe ***Zeilenkommandos***.

T	Der Feldtyp:

Spaltenüberschrift	Feld-Attribute	
	blank	Elementarfeld. Dieser Feld-Typ kann Daten aufnehmen und enthält keine anderen Felder. Er kann innerhalb eines Datensatzes nur einen Wert haben.
	C	Gilt nur bei einem DDM, das sich auf eine Adabas-Datei bezieht. Gibt an, dass eine Datei mit diesem DDM physisch gekoppelt ist. Die Kopplung von Dateien erfolgt mittels Adabas-Deskriptoren. Weitere Informationen zur Kopplung von Dateien siehe *Adabas*-Dokumentation.
	G	Gruppe. Eine Gruppe besteht aus einer Anzahl von Feldern, die unter einem gemeinsamen Gruppennamen definiert werden. Dadurch haben Sie die Möglichkeit, mehrere Felder gemeinsam unter Angabe des Gruppennamens zu referenzieren, anstatt die Namen aller Einzelfelder angeben zu müssen. Solche Felder können keine Daten aufnehmen, sondern dienen nur als „Behälter“ für andere Felder. **Anmerkung:** Gruppen, die in einem DDM definiert werden, brauchen nicht notwendigerweise auch als Gruppen in dem (den) Natural-Objekt(en), das (die) dieses DDM referenziert (referenzieren), definiert zu werden.
	M	Multiples Feld. Dieser Feld-Typ kann innerhalb eines Datensatzes mehr als einen Wert haben. Siehe auch *Multiple Felder* im *Leitfaden zur Programmierung*.
	P	Periodengruppe. Eine Gruppe von Feldern, die innerhalb eines Datensatzes mehr als einen Wert haben kann. Siehe auch *Periodengruppen* im *Leitfaden zur Programmierung*.
	*	Kommentarzeile.
L	Die einem Feld zugewiesene Level-Nummer. Levels (Ebenen) werden benutzt, um die Struktur und die Gruppierung von Felddefinitionen anzuzeigen. Dies ist relevant bei View-Definitionen, Redefinitionen und Feldgruppen (siehe entsprechende Abschnitte im *Leitfaden zur Programmierung*). Gültige Level-Nummern sind 1 - 7. Level-Nummern müssen aufeinanderfolgend in aufsteigender Reihenfolge angegeben werden.	

Spaltenüberschrift	Feld-Attribute
DB	Bei Adabas-Dateien wird in der Spalte **DB** der aus zwei Zeichen bestehende Kurzname des entsprechenden Feldes in der Datenbankdatei angezeigt. Bei DL/I-Segmenttypen wird in der Spalte **DB** der aus zwei Zeichen bestehende Code angezeigt, der in DL/I benutzt wird. Bei VSAM-Dateien siehe *Natural for VSAM* in der *Database Management System Interfaces*-Dokumentation. Bei Feldern des Typs `C` (siehe **Attribut T**) enthält diese Spalte den Kurznamen des Adabas-Deskriptors, der zur Dateikopplung benutzt wird.
Name	Der Name des Feldes. Dieser kann bei Adabas-Feldern und SQL-Spalten 3 - 32 Zeichen und bei DL/I-Namen 1 - 19 Zeichen lang sein. Bei der Generierung eines Feldes aus einer Adabas FDT weist der DDM-Editor einen Standard-Feldnamen *prefix*`-FIELD` zu. Beispiele: `AA-FIELD` für **DB**-Kurzname `AA` oder `AUAL-FIELD` für Kurzname `Aa`. Bei der Vergabe eines Namens gelten die Namenskonventionen für Benutzervariablen (siehe *Natural benutzen*-Dokumentation), außer dass das erste Zeichen des Namens immer ein lateinischer Großbuchstabe (A - Z) sein muss. Außerdem darf der Name nicht mit `L@`, `N@` oder `O@` anfangen, wobei `@` das Zeichen mit dem Hexadezimalwert H'7C' ist. Diese Präfixe dienen zum Identifizieren von **Indikator-Feldern** (siehe weiter unten). Der Feldname ist der Name, der in anderen Natural-Objekten (z.B. in einem Programm) benutzt wird, um das Feld zu referenzieren. Der Feldname ist innerhalb des gesamten Datendefinitionsmoduls eindeutig (unverwechselbar). Bei Feldern des Typs `C` (siehe **Attribut T**) enthält diese Spalte den Kurznamen des zur Dateikopplung verwendeten Adabas-Deskriptors.
F	Das Natural-Datenformat eines Elementarfeldes, z.B. `A` (alphanumerisch), `P` (gepackt numerisch) oder `L` (logisch). Informationen zu gültigen Natural-Datenformaten siehe *Format und Länge von Benutzervariablen* im *Leitfaden zur Programmierung*.
Leng	Die Standardlänge eines Elementarfeldes. Dieser Längenwert kann in einem Natural-Programm durch den Benutzer überschrieben werden. Bei numerischen Feldern (Natural-Datenformat `N`) wird die Länge in der Form *nn*`.`*m* angegeben, wobei *nn* die Anzahl der Stellen vor dem Dezimaltrennzeichen und *m* die Anzahl der Stellen nach dem Dezimaltrennzeichen ist.

<table>
<tr><th>Spaltenüberschrift</th><th>Feld-Attribute</th></tr>
<tr><td></td><td>In dem Eingabefeld Leng können Sie entweder die Feldlänge als numerischen Wert eingeben oder das Schlüsselwort DYNAMIC eingeben, um anzugeben, dass die Feldlänge variabel ist.

Bei Feldern des Typs C (siehe Attribut T) enthält diese Spalte den Namen des zur Dateikopplung verwendeten Datendefinitionsmoduls.</td></tr>
<tr><td>S</td><td>Gilt nicht bei VSAM..

Nullwert-Unterdrückungsoption:
<table>
<tr><td></td><td></td></tr>
<tr><td>blank</td><td>Zeigt an, dass die Adabas-Standardunterdrückung verwendet wird, d.h., dass nachfolgende Leerzeichen in alphanumerischen Feldern und vorangehende Nullen in numerischen Feldern unterdrückt werden.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>F</td><td>Zeigt an, dass das Feld mit der Adabas-Festspeicheroption definiert ist, d.h., es wird keine Unterdrückung angewendet und das Feld wird beim Speichern nicht komprimiert.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>N</td><td>Zeigt an, dass das Feld mit der Adabas-Nullwert-Unterdrückungsoption definiert ist, d.h., dass in der invertierten Liste keine Nullwerte für das Feld gespeichert werden und keine Nullwerte zurückgegeben werden, wenn das Feld in der WITH-Klausel eines FIND-Statements oder in einem HISTOGRAM- oder READ LOGICAL-Statements benutzt wird.

Falls die Spalte Remark (Bemerkung) den Wert NC (not counted/nicht gezählt) enthält, dann zeigt ein N in dieser Spalte an, dass das Feld mit der SQL-Nullwertoption definiert ist. Unterhalb dieses Feldes wird das entsprechende Null-Indikatorfeld gelistet.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>M</td><td>Zeigt an, dass das Feld mit der SQL-Nullwertoption not null definiert ist. Das Feld Remark (siehe Erweiterte Feld-Attribute angeben) enthält zu diesem Feld NN NC (not null, not counted). Unterhalb dieses Feldes wird das entsprechende Null-Indikatorfeld gelistet.</td></tr>
</table></td></tr>
<tr><td>D</td><td>Der Adabas Deskriptor-Typ eines Elementarfeldes, das kein Array ist.

Ein Deskriptor kann als Basis für eine mit einem READ- oder FIND-Statement durchgeführte Datenbanksuche benutzt werden. Beispiel: Ein Feld aus einer Adabas-Datenbank, das ein D oder ein Sin der Spalte D hat, kann in der BY-Klausel des READ-Statement benutzt werden. Nachdem ein Datensatz mit dem READ-Statement aus der Datenbank gelesen worden ist, kann ein DISPLAY-Statement jedes Feld referenzieren, das entweder ein D oder ein S in dieser Spalte hat.
Mögliche Deskriptor-Typ en sind:
<table>
<tr><td></td><td></td></tr>
</table></td></tr>
</table>

Spaltenüberschrift	Feld-Attribute	
	blank	Kein Deskriptor. Dieses Feld ist kein Deskriptor.
	A	Zeigt an, dass das Feld ein alternativer Index für eine VSAM-Datei ist.
	D	Elementar-Deskriptor. Adabas erstellt und verwaltet Wertelisten für dieses Feld, so dass dieses Feld als Suchkriterium in einem FIND-Statement, als Sortierschlüssel in einem FIND-Statement oder zum Steuern der logischen Lesefolge in einem READ- Statement verwendet werden kann.
	H	Hyperdeskriptor. Ein Hyperdeskriptor ist ein User Exit in Adabas. Fürr Natural stellt er die gleiche Funktionalität wie ein phonetischer Deskriptor zur Verfügung (siehe unten).
	N	Ein Nicht-Deskriptor. Ein Nicht-Deskriptor ist kein Deskriptor, kann aber als Suchfeld für eine Nicht-Deskriptor-Suche verwendet werden.
	P	Phonetischer Deskriptor. Ein phonetischer Deskriptor ermöglicht dem Benutzer eine phonetische Suche bei einem Feld (z.B. der Name einer Person). Als Ergebnis einer phonetischen Suche werden alle Werte zurückgegeben, die ähnlich klingen wie der Suchwert.
	S	Superdeskriptor. Wenn ein Superdeskriptor ein multiples Feld oder ein Feld aus einer Periodengruppe (oder einen Teil eines solchen Feldes) enthält, wird der Superdeskriptor in der Feldtypspalte **T** mit einem M oder einem P gekennzeichnet. Dadurch ist Natural in der Lage, für diesen Superdeskriptor. die richtigen Suchalgorithmen zu erstellen. Bei einem DL/I-Segmenttyp zeigt das S einen Superdeskriptor an, d.h., ein Suchfeld eines Parent-Segments.
	U	Subdeskriptor oder Sortier-Deskriptor (*collation descriptor*. Wenn ein Superdeskriptor ein multiples Feld oder ein Feld aus einer Periodengruppe (oder einen Teil eines solchen Feldes) enthält, müssen Sie den Subdeskriptor in der Feldtypspalte **T** mit einem M kennzeichnen.

Spaltenüberschrift	Feld-Attribute	
		Ein Sortier-Deskriptor wird verwendet, um Deskriptor-Feldwerte in einer nicht standardmäßigen Sortierfolge zu sortieren. Wenn ein Feld ein Sortier-Deskriptor ist, steht in der Spalte **Remark** (siehe unten): „Collation“ (Sortierung), die Nummer des Adabas-User Exit, der die Sortierfolge (1-8) enthält, und der Kurzname des Parent-Feldes, für das die Sortierfolge gilt, z.B.: `Collation 5 on AA`.
	`X`	Alternativer Subdeskriptor oder Superdeskriptor, d.h. ein alternativer Index für eine VSAM-Datei.
	Informationen zu VSAM-Dateien siehe *Natural for VSAM* in der *Database Management System Interfaces*-Dokumentation. Bei Feldern des Typs `C` (siehe **Attribut T**) enthält diese Spalte den Namen des zur Dateikopplung benutzten Datendefinitionsmoduls.	
Remark	Bemerkung. Ein Kommentar, der ein Feld und/oder das DDM betrifft.	

Indikator-Felder

Ein Indikator-Feld wird verwendet, um die Länge eines Feldes mit variabler Länge oder Informationen über die Datensignifikanz (`NULL`-Wertindikator) eines Datenbankfeldes abzurufen. Ein Indikator-Feld liefert *nicht* den Inhalt eines Datenbankfeldes.

Wenn der Name eines Datenbankfeldes mit `L@`, `N@` oder `O@` beginnt (wobei `@` das Zeichen mit dem Hexadezimalwert H'7C' ist), wird das Feld als Indikator-Feld interpretiert. Deshalb darf der Name eines Datenbankfeldes nicht mit einem dieser Zeichen beginnen - es sei denn, es handelt sich um ein Indikator-Feld.

Beim erstmaligen Generieren eines Datendefinitionsmoduls geschieht Folgendes:

- Für jedes Feld mit variabler Länge wird automatisch ein `L@`*`xxxxx`*-Feld angelegt, wobei *`xxxxx`* der Name des zugehörigen Feldes ist.

  In einer Adabas-Datei gilt dies für LA-Felder (Long Alpha) und LB-Felder (Large Object). In einer Db2-Tabelle gilt dies für VARCHAR- und LOB-Felder.

  Wenn sich der Längenanzeiger auf ein LA-, LB- oder LOB-Feld bezieht, muss das/die Natural-Datenformat/-länge I4 sein. Bei einem VARCHAR-Feld muss sie I2 sein.
- Für ein Feld, das einen `NULL`-Wert enthalten kann, wird automatisch ein `N@`*`xxxxx`*-Feld angelegt, wobei *`xxxxx`* der Name des zugehörigen Feldes ist.

Dies gilt für Adabas-Felder, die mit der SQL-Nullwertoption (*SQL Null Value Option*) definiert sind, und für Db2-Felder, die per Definition einen `NULL`-Wert haben. Das/die Natural-Datenformat/-länge eines `NULL`-Indikator-Feldes muss I2 sein.

- Einem `0@xxxxx`-Feld ist zurzeit keine spezifische Abruffunktion zugewiesen. Es ist für zukünftige Erweiterungen reserviert.

  Ein `0@xxxxx`-Feld wird automatisch für ein Zeiger-Feld (*Locator*) eines Db2-LOB-Feldes angelegt. Das/die Natural-Datenformat/-länge eines Zeiger-Feldes muss I4 sein.

Kommandos zum Editieren und Ausführen von Funktionen

Dieser Abschnitt beschreibt die Kommandos, die im Bildschirm **Edit DDM** zur Verfügung stehen.

Zeilenkommandos werden verwendet, um einzelne oder mehrere Quellcode-Zeilen zu kopieren, zu löschen, einzufügen oder zu verschieben. Außerdem kann man mit einem Zeilenkommando die Funktion zum Bearbeiten der erweiterten Feld-Attribute aufrufen. Siehe *Erweiterte Feld-Attribute angeben*

Editor- und Systemkommandos werden z.B. verwendet, um spezielle Zeilenkommandos auszuführen, im Quellcode des Datendefinitionsmoduls zu navigieren oder eine SYSDDM-Funktion im **Edit DDM**-Bildschirm direkt aufzurufen.

- Hilfe zu Kommandos
- Zeilenkommandos
- Editor- und Systemkommandos

Hilfe zu Kommandos

Dieser Abschnitt beschreibt, wie Sie Hilfe-Informationen zu den Kommandos erhalten, die Ihnen im **Edit DDM**-Bildschirm zur Verfügung stehen.

Um Hilfe-Informationen zu Kommandos anzuzeigen:

1 Geben Sie in der Kommandozeile **Edit DDM**-Bildschirms `HELP` ein.

  Oder:

  Geben Sie in der Kommandozeile **Edit DDM**-Bildschirms ein Fragezeichen (?) ein.

  Der Bildschirm **Editor Help Info** erscheint.

2 Drücken Sie Enter, um im Hilfetext nach unten zu blättern und um den **Editor Help Info**-Bildschirm zu verlassen.

  Der Bildschirm **Edit DDM** erscheint wieder.

Zeilenkommandos

Dieser Abschnitt beschreibt die Zeilenkommandos, die im Bildschirm **Edit DDM** zur Verfügung stehen, und erklärt, wie Sie ein Zeilenkommando ausführen.

Um ein Zeilenkommando auszuführen:

1 Platzieren Sie im Bildschirm **Edit DDM** neben der Quellcode-Zeile (bzw. den Quellcode-Zeilen), für die das Kommando gelten soll, den Cursor in die Spalte **T** und geben Sie ein Zeilenkommando ein, indem Sie die Inhalte der Spalten **T**, **L**, **DB** oder **Name** überschreiben.

2 Drücken Sie Enter.

Zeilenkommando	Funktion
.C(*nn*)	Kopiert die Zeile einmal oder *nn* Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CX(*nn*)	Kopiert die Zeile, die mit .X einmal oder *nn* Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CY(*nn*)	Kopiert die Zeile, die mit .Y markiert ist, einmal oder *nn* Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CX-Y(*nn*)	Kopiert den Zeilenblock einmal oder *nn* Mal, siehe ***Um einen Zeilenblock zu kopieren oder zu verschieben:***.
.D(*nnnn*)	Löscht die Zeile, in der das Kommando eingegeben wurde, oder löscht *nnnn* Zeilen ab der Zeile, in der das Kommando eingegeben wurde. Wenn *nnnn* nicht angegeben, dann wird standardmäßig eine Zeile gelöscht.
.E*nn*	Ruft die Funktion zum Bearbeiten der erweiterten Feld-Attribute auf. Siehe ***Erweiterte Feld-Attribute angeben***.
.I(*nn*)	Fügt *nn* Leerzeilen unter der Zeile, in der das Kommando eingegeben wurde, ein. Für *nn* können Sie eine Zahl im Bereich von 1 bis 10 angeben. (Beim nächsten Drücken von Enter werden alle leer gelassenen Zeilen wieder entfernt.) Falls *nn* nicht (oder nicht korrekt) angegeben wird, werden standardmäßig 10 Zeilen eingefügt. Um an den Quellcode Zeilen anzuhängen, benutzen Sie das Editor-Kommando ADD.
.MX	Verschiebt die mit .X markierte Zeile unter die Zeile, in der das Kommando eingegeben wurde.
.MY	Verschiebt die mit .Y markierte Zeile unter die Zeile, in der das Kommando eingegeben wurde.
.MX-Y	Verschiebt einen Zeilenblock. Siehe ***Um einen Zeilenblock zu kopieren oder zu verschieben:***.
.X	Markiert eine einzelne Zeile oder die erste Zeile eines Zeilenblocks zum Kopieren oder Verschieben. Eine markierte Zeile wird durch ein X in der Spalte **I** angezeigt.

Zeilenkommando	Funktion
	Siehe auch *Um einen Zeilenblock zu kopieren oder zu verschieben:*.
.Y	Markiert die letzte Zeile eines Zeilenblocks zum Kopieren oder Verschieben. Eine markierte Zeile wird durch ein Y in der Spalte **I** angezeigt. Siehe auch *Um einen Zeilenblock zu kopieren oder zu verschieben:*.

Um einen Zeilenblock zu kopieren oder zu verschieben:

1 Geben Sie in der ersten Zeile eines Zeilenblocks, der kopiert oder verschoben werden soll, das folgende Zeilenkommando ein:

```
.X
```

Geben Sie in der letzten Zeile des Zeilenblocks, der kopiert oder verschoben werden soll, das folgende Zeilenkommando ein:

```
.Y
```

2 Drücken Sie Enter.

Der Zeilenblock wird, wie in der Spalte **I** angezeigt, durch ein X und ein Y begrenzt.

3 Geben Sie in der Zeile, unter die Sie den markierten Block kopieren oder verschieben möchten, eines der folgenden Zeilenkommandos ein:

```
.CX-Y (nn)
```

oder

```
.MX-Y
```

dabei steht C für Kopieren und M für Verschieben. *nn* gibt an, wie viele Male der markierte Block kopiert werden soll. Wird *nn* nicht angegeben, wird der Block standardmäßig ein Mal kopiert.

4 Drücken Sie Enter.

Der markierte Block wird kopiert (ein Mal oder *nn* Mal) bzw. wird unter die Zeile verschoben, in der das Kommando eingegeben wurde.

Editor- und Systemkommandos

Dieser Abschnitt beschreibt die Editor- und Systemkommandos, die im Bildschirm **Edit DDM** zur Verfügung stehen und gibt (falls relevant) die gleichbedeutenden PF-Tasten an.

Um ein Editor- und Systemkommando auszuführen:

- Geben Sie in der Kommandozeile im oberen Bereich des Bildschirms ein Editor- oder ein Systemkommando ein und drücken Sie Enter.

  Oder:

  Falls eine der PF-Tasten im Bildschirm **Edit DDM** mit dem gewünschten Editor- oder Systemkommando belegt ist, drücken Sie diese Taste, um das Kommando auszuführen.

  Beispiel: Um ein DDM zu katalogisieren, können Sie entweder das Kommando CATALOG absetzen oder die Taste PF11 drücken.

Die in den Kommandos verwendeten Syntax-Symbole werden im Abschnitt *Systemkommando-Syntax* in der *Systemkommandos*-Dokumentation erklärt. Der unterstrichene Teil eines Kommandos kann als gültige Abkürzung verwendet werden.

Die Editor-Kommandos zu Navigieren im DDM-Quellcode werden in einer separaten Tabelle beschrieben. Siehe *Editor-Kommandos zum Positionieren*.

Kommando	Funktion
ADD	Hängt 10 Leerzeilen an den Quellcode an. (Beim nächsten Drücken von Enter werden leer gelassene Zeilen wieder entfernt.) Um Zeilen einzufügen, siehe Zeilenkommando .I.
CATALOG	Führt eine Syntaxprüfung durch und speichert den zurzeit Im Source-Bereich des Editors befindlichen DDM-Quellcode als katalogisiertes Objekt. `CATALOG [ DDM-name ] [ REPLACE ]` Falls der DDM-Quellcode bereits katalogisiert worden ist, müssen Sie die REPLACE-Option benutzen. **Äquivalente PF-Taste: PF11**
CHECK	Validiert den zurzeit im Source-Bereich des Editors befindlichen DDM-Quellcode gegen die von dem DDM referenzierte Adabas FDT. Falls eine Inkonsistenz festgestellt wird, wird die Quellcode-Zeile, die den Fehler verursacht, in der Spalte **I** mit einem E (Error) gekennzeichnet, so dass sie korrigiert werden kann. **Äquivalente PF-Taste: PF10**

Kommando	Funktion
CLEAR	Löscht den Source-Bereich. Weitere Informationen siehe entsprechendes Natural-Systemkommando CLEAR in der *Systemkommandos*-Dokumentation.
CODEPAGE	Zeigt den Quellcode des Datendefinitionsmoduls, das sich zurzeit im Source-Bereich des Editors befindet, und die Codepage der Session an. Die Codepage zurzeit im Source-Bereich des Editors befindlichen Datendefinitionsmoduls kann von der Codepage dieses Datendefinitionsmoduls in der Systemdatei abweichen, weil ein DDM beim Einlesen in den DDM-Editor konvertiert worden sein kann.
DX	Löscht die mit dem Zeilenkommando .X markierte Zeile.
DY	Löscht die mit dem Zeilenkommando .Y markierte Zeile.
DX-Y	Löscht den mit den Zeilenkommandos .X und .Y eingegrenzten Zeilenblock.
EX	Löscht alle Zeilen oberhalb der mit dem Zeilenkommando .X markierten Zeile.
EY	Löscht alle Zeilen unterhalb der mit dem Zeilenkommando .Y markierten Zeile.
EX-Y	Löscht alle Zeilen mit Ausnahme des Blocks, der mit den Zeilenkommandos .X und .Y eingegrenzt ist.
HELP oder ?	Ruft den **Editor Help Info**-Bildschirm auf, der Hilfe-Informationen zu den Editor-Kommandos anzeigt.
LENGTH oder SIZE	{ LENGTH / SIZE } [*from-field to-field*]
LIST DDM oder LIST VIEW	LIST { DDM / VIEW } [*DDM-name*] Zeigt eine einzelne DDM-Source oder gemäß Angabe mit *DDM-name* eine Liste mit Datendefinitionsmodulen an. Weitere Informationen siehe Natural-Systemkommando LIST in der *Systemkommandos*-Dokumentation.
LOWERCASE oder LC	Gilt nur bei Datendefinitionsmodulen der Typen A (Adabas) und 2 (Db2). Deaktiviert die automatische Umsetzung von Groß- in Kleinbuchstaben. Bei ausgeschalteter Umsetzung wird rechts oben im Editor-Bildschirm Lower ON angezeigt. Wenn Datendefinitionsmodule, die Kleinbuchstaben enthalten, zum Bearbeiten geöffnet werden, ist immer Lower ON gesetzt. Siehe auch UPPERCASE.

Kommando	Funktion
QUIT oder .	Beendet die DDM-Editor-Sitzung. Es erscheint das Menü der SYSDDM Utility. Die DDM-Source verbleibt so lange im Source-Bereich, bis eine andere Source in den Source-Bereich eingelesen wird (das kann auch durch einen beliebigen anderen Natural-Editor erfolgen) oder bis die Natural-Session beendet wird. **Anmerkung:** Beim DDM-Editor kommt die Editor-Profiloption **Leave Editor with Unlock** zur Anwendung. Sie dient dazu, den Quellcode beim Verlassen des DDM-Editors zu entsperren. Informationen zu dieser Option siehe ***Allgemeine Standardwerte*** in *Editor-Profil* im Abschnitt *Editoren — Allgemeine Informationen*. **Äquivalente PF-Taste: PF3**
READ	`READ [DDM-name]` Liest eine DDM-Source in den Source-Bereich. Falls sich zurzeit ein anderes DDM im Source-Bereich befindet, wird dieses überschrieben.
RESET	Entfernt alle Markierungen von Zeilen, die mit einem `X` (siehe Zeilenkommando `.X`), mit einem `Y` (siehe Zeilenkommando `.Y`) oder mit einem in der Spalte **I** angezeigten `E` (Fehler bei `CHECK`)markiert sind.
SCAN	`SCAN [scan-value]` Durchsucht den Quellcode nach einer als *scan-value* angegebenen Zeichenkette, zum Beispiel: `SCAN ABC` oder `SCAN ABC D`. Wird der angegebene Suchwert gefunden, dann wir die Zeile (bzw. die Zeilen), die den *scan-value* enthält, mit einem in der Spalte **I** angezeigten `S` markiert. (Drücken Sie Enter, um die Markierungen zu entfernen.)
UNCATALOG	`UNCATALOG [DDM-name]` Löscht ein oder mehrere Datendefinitionsmodule in der aktuellen Systemdatei FDIC, wenn *DDM-name* angegeben wird (siehe auch **DDM Name** im Abschnitt *DDM-Parameter angeben*). Wird *DDM-name nicht* angegeben, dann wird das zurzeit im Source-Bereich befindliche DDM gelöscht. Dieses Kommando entspricht dem Natural-Systemkommando `UNCATALOG` in der *Systemkommandos*-Dokumentation.
UPPERCASE oder UC	Gilt nur bei Datendefinitionsmodulen der Typen `A` (Adabas) und `2` (Db2). Aktiviert die automatische Umwandlung von Klein- und Großbuchstaben. Dies ist der Standard-Umwandlungsmodus. Siehe auch `LOWERCASE`.

Editor-Kommandos zum Positionieren

Editor-Kommando	Äquivalente PF-Taste:	Funktion

`+`
 	
oder
`+P`
	PF8
	
	oder
	Enter
		Eine Seite nach unten blättern (20 Zeilen).

`-`
 	
oder
`-P`
	PF7	Eine Seite nach oben blättern (20 Zeilen).

`+H`	PF5	Eine halbe Seite nach unten blättern (10 Zeilen).
`-H`	PF4	Eine halbe Seite nach oben blättern (10 Zeilen).

`X`
 	
oder
`Y`
		Auf die mit dem Zeilenkommando `.X` oder `.Y` markierte Zeile positionieren.

`B`
 	
oder
`++`
	PF9	Nach unten zur letzten Seite blättern.

`T`
 	
oder
`--`
	PF6	Nach oben zur ersten Seite blättern.

`+`*`nn`*		Um *nn* Zeilen nach unten blättern.
`-`*`nn`*		Um *nn* Zeilen nach oben blättern.

Erweiterte Feld-Attribute angeben

Die Funktion zur erweiterten Feldbearbeitung kann benutzt werden, um Standard-Attribute für Spaltenüberschrift (Field Header) und Editiermasken (Field Edit Mask) anzugeben, einen Feldkommentar (Bemerkung) einzugeben und eine Formatoption anzugeben, die gelten soll, wenn das Feld in einem anderen Natural-Objekt (z.B. in einem Programm) anzugeben. Bei einem DDM, das aus einer VSAM-Datei generiert wird, können Sie zusätzlich VSAM-spezifische Feld-Attribute anzeigen und bearbeiten.

Das Header-Attribut gibt die Standard-Spaltenüberschrift an, die über dem Feld angezeigt werden soll, wenn es z.B. mit einem `DISPLAY`-Statement ausgegeben wird. Der Header entspricht dem Text, der mit dem Parameter `HD` in Hochkommas (`HD='`*`text`*`'`) angegeben wird. Siehe *Parameter-Referenz*-Dokumentation. Wird kein Header angegeben, wird der Feldname als Spaltenüberschrift verwendet.

Das Editiermasken-Attribut gibt die Standard-Editiermaske an, die verwendet werden soll, wenn das Feld z.B. mit einem `DISPLAY`-Statement ausgegeben wird. Die Editiermaske muss den Natural-Syntaxregeln entsprechen und muss für das Natural-Datenformat und die Datenlänge des Feldes gültig sein.

Das Remark-Attribut dient zur Angabe eines Kommentars zu dem Feld.

Die Format-Option kann benutzt werden, um Felder mit variabler Länge zu definieren: Wird die Option auf `LA` gesetzt, dann wird das Feld als Long-Alpha-Feld definiert. Wird die Option auf `LB` gesetzt, dann wird das Feld als Large-Object-Feld definiert. Ein Long-Alpha-Feld kann das Format A oder U, ein Large-Object-Feld das Format A, U oder B haben.

Verwandte Themen:

- `DISPLAY` und `INPUT` in der *Statements*-Dokumentation
- *EM - Editiermaske* in der *Parameter-Referenz*-Dokumentation

Der nachfolgende Abschnitt behandelt folgende Themen:

- Erweiterte Feld-Attribute bearbeiten

Erweiterte Feld-Attribute bearbeiten

Dieser Abschnitt beschreibt, wie Sie die Funktion zur erweiterten Feldbearbeitung für ein Einzelfeld oder für einen Bereich von aufeinanderfolgenden Feldern aufrufen und beenden.

Um die Funktion zur erweiterten Feldbearbeitung aufzurufen:

1 Für ein Einzelfeld:

Platzieren Sie den Cursor in der Spalte **T** neben dem betreffenden Feld und tippen Sie das Zeilenkommando `.E` über die Werte in den Spalten **T** und **L** ein.

Der Bildschirm **Extended Field Editing** wird für das mit dem Kommando markierte Feld angezeigt. Das folgende Beispiel stammt von einem DDM aus Adabas:

```
12:07:49                ***** Edit DDM (ADA) *****                2006-02-08
                         - Extended Field Editing -
DDM Name DDM-TEST                         Def.Seq.        DBID     0 FNR   316

I T L DB Name                             F       Leng S D
- - - -- -------------- top ------------- - ---------- - -
    1 AF LA-FIELD                         A      16381 F
-------------------------------------------------------------

Remark ........... LA_________________________
Field Header ..... _____________________________________________________
Field Edit Mask .. _____________________________________________________

Format Option .... LA Long Alpha      (LA = LA field, LB = LOB field)
```

Im Bildschirm **Extended Field Editing** können Sie eine Bemerkung (Kommentar), eine Feldüberschrift, eine Editiermaske und eine Format-Option angeben.

Informationen zur erweiterten Feldbearbeitung in Datendefinitionsmodulen aus VSAM siehe *Extended Editing at Field Level* in der *Natural for VSAM*-Dokumentation).

2 Für einen Bereich von aufeinanderfolgenden Feldern:

1. Platzieren Sie den Cursor in der Spalte **T** neben dem ersten auszuwählenden Feld und tippen Sie das folgende Zeilenkommando über die Werte in den Spalten **T** und **L** ein.

   ```
   .Enn
   ```

   dabei ist *nn* die Anzahl der auszuwählenden Felder, einschließlich des aktuellen Feldes.

   Der Bildschirm **Extended Field Editing** erscheint für das erste ausgewählte Feld.
2. Geben Sie die erforderlichen Feld-Attribute ein oder ändern Sie sie. Drücken Sie Enter oder PF3.

   Der Bildschirm **Extended Field Editing** erscheint für das nächste Feld in der Abfolge.

Um die Funktion zur erweiterten Feldbearbeitung zu beenden:

- Drücken Sie Enter oder PF3.

  Alle Änderungen werden gespeichert und der Bildschirm **Edit DDM** erscheint.

41 DDM katalogisieren

Sie können ein Datendefinitionsmodul als katalogisiertes Objekt in der angegebenen Natural-Systemdatei speichern (siehe auch *DDMs speichern*).

Informationen zu Namenskonventionen, die für ein Objekt gelten, siehe *Namenskonventionen für Objekte* in der *Natural benutzen*-Dokumentation.

Um ein DDM zu katalogisieren:

- Geben Sie im Menü der SYSDDM Utility das DDM an, das Sie katalogisieren möchten (siehe auch *DDM-Parameter angeben*) und führen Sie die Funktion **Catalog DDM** aus (siehe auch *Beschreibung der Funktionen*).

  Oder:

  Setzen Sie im Bildschirm **Edit DDM** das Kommando `CATALOG` ab. Siehe *Editor- und Systemkommandos*.

42 DDMs auflisten

Dieser Abschnitt beschreibt, wie Sie unter Benutzung der SYSDDM Utility oder des Natural-Systemkommandos `LIST` eine Liste aller Datendefinitionsmodule, die in einer FDIC-Systemdatei vorhanden sind, anzeigen können.

Um alle Datendefinitionsmodule mit SYSDDM aufzulisten:

- Geben Sie im Menü der SYSDDM Utility Folgendes an:

  Geben Sie im **Code**-Feld den Code `L` oder `X` ein (siehe auch *Beschreibung der Funktionen*).
  Geben Sie im Feld **Name** einen Stern (*) und in den Feldern **DBID** und **FNR** die Datenbankkennung und die Dateinummer der Datenbank an, auf die das Datendefinitionsmodule zugreifen sollen.
  Weitere Informationen zu gültigen Feldeingaben siehe *DDM-Parameter angeben*.

  Der Bildschirm **LIST DDMs** erscheint. Er enthält eine Auswahlliste mit allen Datendefinitionsmodulen, die in der angegebenen FDIC-Systemdatei vorhanden sind.

Um alle Datendefinitionsmodule mit `LIST` aufzulisten:

- Geben Sie in der Kommandozeile folgendes Systemkommando ein:

  ```
  LIST DDM *
  ```

  Der Bildschirm **LIST DDMs** erscheint. Er enthält eine Auswahlliste mit allen Datendefinitionsmodulen, die in der aktuellen FDIC-Systemdatei vorhanden sind.

Informationen zu allen Optionen, die beim Systemkommando `LIST` zur Verfügung stehen, siehe entsprechende Abschnitte in der *Systemkommandos*-Dokumentation.

43 DDMs in unterschiedlichen Umgebungen pflegen

Sie können das Dienstprogramm (Utility) SYSMAIN (siehe *Debugger und AE-Dienstprogramme*-Dokumentation) benutzen, um Datendefinitionsmodule zwischen verschiedenen FDIC-Systemdateien zu übertragen (z.B. Kopieren oder Verschieben) und um eine DDM-Operation in einer anderen Umgebung auszuführen.

Um Datendefinitionsmodule zwischen unterschiedlichen Hardware-Plattformen zu übertragen (Großrechner, UNIX und Windows) können Sie den Object Handler (siehe *Utilities*-Dokumentation) benutzen.

IX Software AG Editor

Der Software AG Editor wird verwendet, um Objekte in verschiedenen Natural-Utilities, Natural-Add-on-Produkten und sonstigen Software AG-Produkten zu erstellen und zu bearbeiten.

Allgemeine Informationen zum Software AG Editor

Software AG Editor aufrufen

Benutzung des Editor-Bildschirms

Kommandos benutzen

Daten anlegen und ändern

Editor-Profil einstellen

Daten speichern und den Software AG Editor verlassen

Zeilenkommandos (Übersicht)

Hauptkommandos (Übersicht)

Verwandte Themen:

- *Installing Software AG Editor* in der Natural *Installation*-Dokumentation
- *Operating the Software AG Editor* in der Natural *Operations*-Dokumentation
- *Editoren in der SPoD-Umgebung* in der Natural *Unicode- und Codepage-Unterstützung*-Dokumentation
- *SYSEDT Utility* in der *Debugger und Dienstprogramme*-Dokumentation
- *EDBP - Software AG Editor Buffer Pool-Definitionen* und *EDPSIZE - Größe des Software AG Editor-Hilfs-Buffer-Pool* in der Natural *Parameter-Referenz*-Dokumentation

44 Allgemeine Informationen zum Software AG Editor

Der Software AG Editor wird verwendet, um Objekte in Natural (einschließlich Utilities, z.B. SYSRPC und SYSBPM) und in folgenden anderen Software AG-Produkte zu erstellen und zu bearbeiten:

- Natural ISPF
- Entire Operations
- Entire Output Management
- Predict

In jedem dieser Produkte wird eine Teilmenge der Gesamtfunktionalität des Software AG Editor verwendet.

Falls Sie den Software AG Editor als Alternative zum Natural-Programm-Editor verwenden möchten, müssen Sie dies im Editor-Profil entsprechend angeben (siehe *Editor-Profil einstellen*).

Der Software AG Editor bietet eine ähnliche Funktionalität wie Natural ISPF. Sie können damit zum Beispiel folgende Objekte anzeigen und bearbeiten:

- Natural-Objekte des Typs Programm, Subprogramm, Subroutine, Helproutine, Copycode, Text, Class oder Function
- PDS Members und sequenzielle Dateien
- PANVALET Members
- CA-LIBRARIAN Members
- Job SYSOUT (nur Anzeige möglich)
- Ausgabe im Benutzer-Work-Pool
- Liste mit Systemobjekten (nur Anzeige möglich)
- Beliebige Text-Objekte oder Tabellen

45 Software AG Editor aufrufen

Von welchem Bildschirm aus Sie den Software AG Editor aufrufen, hängt davon ab, welche Anwendung Sie momentan benutzen und welchen Objekttyp Sie anzeigen oder bearbeiten möchten.

Beispiel: Wenn Sie in Natural ISPF ein Natural-Objekt bearbeiten möchten, werden Sie den Software AG Editor aus dem Objekt-Eingabe-Bildschirm aufrufen.

Wie Sie in den Software AG Editor gelangen, wird in den entsprechenden Dokumentationsabschnitten der Anwendung beschrieben, die Sie benutzen.

Die Kommandos, die Ihnen zur Verfügung stehen, sind ebenfalls von der benutzen Anwendung abhängig.

Beispiele für mit dem Software AG Editor definierte Objekte finden Sie in den entsprechenden Dokumentationsabschnitten der Anwendung beschrieben, die Sie benutzen.

46 Benutzung des Editor-Bildschirms

- Benutzung der Eingabebereiche 437
- In den Daten im Editierbereich blättern 437
- Zu einer bestimmten Zeile springen 440
- Zeilen zeigen oder verbergen 441
- Ränder, Tabulator- und Spaltenpositionen anzeigen 442

Wenn Sie den Namen eines neuen Objekts angeben, erscheint ein Editor-Bildschirm, der so ähnlich aussieht, wie der folgende Beispiel-Bildschirm:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-78K ------------ Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ****************************** top of data ******************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** ***************************** bottom of data *****************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Je nachdem, welches Software AG-Produkt Sie benutzen, kann der Bildschirm auch geringfügig anders aussehen.

In der ersten Zeile dieses Bildschirms wird das zu bearbeitende Objekt gemäß der Umgebung angezeigt, aus der der Software AG Editor aufgerufen wurde. Im obigen Beispiel ist `JOB1JCL` der Name des neuen Objekts, und `NATLIB1` kennzeichnet die Objekt-Bibliothek, bei der es sich um eine Natural Library handelt.

Die Eingabebereiche in dem Editor-Bildschirm werden im folgenden Abschnitt beschrieben.

Benutzung der Eingabebereiche

Der Bildschirm des Software AG Editor enthält vier Eingabebereiche:

- **Kommandozeile**

  Die Kommandozeile wird eingeleitet durch **COMMAND===>**.

  Sie dient zur Eingabe eines Editor-Hauptkommandos. Weitere Informationen siehe *Hauptkommandos (Übersicht)*.

- **SCROLL-Feld**

  Das SCROLL-Feld ist gekennzeichnet durch **SCROLL===>**.

  Es dient dazu, die standardmäßig zu verwendenden Blätterwerte anzugeben. Weitere Informationen siehe *Einstellungen im SCROLL-Feld*.

- **Prafixbereich**

  Die sechs Spalten am linken Rand des Editor-Bildschirms werden als Prafixbereich bezeichnet. Dieser Bereich kann eine Reihe von Apostrophen (' ' ' ' ' ') oder Sternchen (******), eine Quellcode-Zeilennummer oder Text enthalten. Der Prafixbereich dient zur Eingabe von Zeilenkommandos. Weitere Informationen siehe *Zeilenkommandos (Übersicht)*.

- **Editierbereich**

  Rechts neben dem Prafixbereich befindet sich der Editierbereich. Er dient zur Eingabe von Daten.

  Sie können eine Editor-Session beginnen, indem Sie Daten in den Bildschirm des Software AG Editor eingeben und Editor-Hauptkommandos und Zeilenkommandos benutzen. Siehe *Hauptkommandos (Übersicht)*) und (*Zeilenkommandos (Übersicht)*).

In den Daten im Editierbereich blättern

Dieser Abschnitt beschreibt die Kommandos und Einstellungen, die Sie benutzen können, um in den Daten zu blättern, die im Editierbereich des Editor-Bildschirms enthalten sind.

- PF Keys
- Einstellungen im SCROLL-Feld

- Hauptkommandos zum Blättern

PF Keys

Die Blätterkommandos können auch folgenden PF-Tasten zugewiesen sein:

- PF7 (Hauptkommando `UP`) zum Blättern an den Anfang der Daten.
- PF8 (Hauptkommando `DOWN`) zum Blättern an das Ende der Daten.
- PF10 (Hauptkommando `LEFT`) zum Blättern nach Links.
- PF11 (Hauptkommando `RIGHT`) zum Blättern nach Rechts.

Einstellungen im SCROLL-Feld

Im SCROLL-Feld (**SCROLL===>**) können Sie Einstellungen für das Blättern eingeben. Mit diesen Einstellungen wird der Betrag festgelegt, um den mit den PF-Tasten (s.o.) weitergeblättert werden kann. Einige Angaben werden außerdem bei den Hauptkommandos zum Blättern auf der Folgeseite verwendet.

Folgende Einstellungen sind im SCROLL-Feld möglich:

SCROLL-Feld-Einstellung	Erläuterung
number	Anzahl (*number*) der Zeilen, um die nach oben oder nach unten geblättert wird. Anzahl (*number*) der Spalten, um die nach Rechts oder nach Links geblättert wird.
`CSR` (Standardwert)	Blättert nach unten bis zur Cursor-Position, wenn sich der Cursor in einer Textzeile befindet. Die Cursor-Zeile wird zur ersten Textzeile. Beim Blättern nach oben wird die Cursorzeile zur letzten Textzeile. Weiterblättern um eine Seite, wenn sich der Cursor in der Kommandozeile befindet. Nach Rechts oder Links bis zur Cursor-Position blättern.
`DATA`	Weiterblättern um eine Seite minus eine Zeile. Beim Blättern nach unten wird die unterste Zeile zur obersten Zeile. Beim Blättern nach oben wird die oberste Zeile zur untersten Zeile. Beim Blättern nach Rechts wird die letzte Spalte zur ersten Spalte. Beim Blättern nach Links wird die erste Spalte zur letzten Spalte.
`HALF`	Eine halbe Seite in beliebiger Richtung blättern.
`LINE`	Nach oben blättern bis zum Zeilenanfang oder nach unten bis zum Zeilenende.
`MAX`	Zum Anfang oder zum Ende der Daten blättern. An den rechten oder linken äußersten Rand der Daten blättern.
`PAGE`	Eine ganze Seite in beliebiger Richtung blättern.

SCROLL-Feld-Einstellung	Erläuterung
PARA	Nach oben oder unten zum ersten Zeichen des nächsten Abschnitts blättern.
SENT	Nach oben zum ersten Zeichen des aktuellen Satzes oder nach unten zum ersten Zeichen des folgenden Satzes blättern. Wenn sich beim Blättern nach oben der Cursor auf dem ersten Zeichen des Satzes befindet, wird zum ersten Zeichen des vorhergehenden Satzes geblättert.
WORD	Blättern nach oben zum ersten Zeichen des nächsten Worts oder nach unten zum ersten Zeichen des folgenden Worts.

Hauptkommandos zum Blättern

Außer dem Hauptkommando LOCATE, mit dem Sie in den Daten zur angegebenen Zeile blättern können, gibt es mehrere Hauptkommandos zum Blättern in vertikaler oder horizontaler Richtung.

Die folgende Tabelle enthält die verfügbaren Blätterkommandos und erklärt ihre Funktion:

Hauptkommando	Erläuterung
BOTTOM or ++	Blättert zum Ende des Objekts, das gerade bearbeitet wird.
TOP or --	Blättert zum Anfang des Objekts, das gerade bearbeitet wird.
DOWN	Blättert um den im SCROLL-Feld angegebenen Betrag vorwärts.
DOWN *n*	Blättert um *n* Zeilen vorwärts.
+*n*	Blättert um *n* Zeilen vorwärts.
UP	Blättert um den im SCROLL-Feld angegebenen Betrag rückwärts.
UP *n*	Blättert um *n* Zeilen rückwärts.
-*n*	Blättert um *n* Zeilen rückwärts.
LEFT	Blättert um den im SCROLL-Feld angegebenen Betrag nach Links.
LEFT *n*	Blättert um *n* Spalten nach Links.
RIGHT	Blättert um den im SCROLL-Feld angegebenen Betrag nach Rechts.
RIGHT *n*	Blättert um *n* Spalten nach Rechts.
FIX *n*	Gibt die Anzahl *n* der Spalten ab Spalte 1 an, die beim Blättern nach Rechts in der Anzeige bleiben sollen.

Zu einer bestimmten Zeile springen

Sie können das Hauptkommando LOCATE benutzen, um eine bestimmte Zeile am Anfang des Editor-Bildschirms erscheinen zu lassen (d.h. um sie zur aktuellen Zeile zu machen). Zusätzlich zu dem Kommando müssen Sie einen Parameter angeben, der die Zeile definiert, die als erste Zeile angezeigt werden soll.

Beispiele:

```
L 32
```

Dieses Kommando bewirkt, dass Zeile 32 die erste Zeile wird.

```
32
```

Wie zuvor.

```
L .X
```

Bewirkt, dass die mit .X markierte Zeile zur aktuellen Zeile wird.

```
L 'ABC'
```

Bewirkt, dass die erste Zeile, die mit der Zeichenkette ABC beginnt, zur aktuellen Zeile wird (nützlich beim Durchblättern von sortierten Daten, z.B. Verzeichnislisten).

Unterschiede zwischen den Kommandos LOCATE und FIND

Beachten Sie bitte die folgenden Unterschiede, die zwischen den Kommandos LOCATE und FIND bestehen:

- Wenn Sie das Kommando LOCATE mit einer Zeichenkette absetzen (L 'ABC'), wird die Zeichenkette nur dann gefunden, wenn sie in Spalte 1 beginnt. Das Kommando FIND durchsucht den gesamten Quellcode-Arbeitsbereich.
- Beim Kommando LOCATE wird davon ausgegangen, dass die zu durchsuchenden Daten in aufsteigender alphabetischer Reihenfolge sortiert sind.
- Wenn eine Zeile mit dem Kommando LOCATE gefunden wird, wird der Cursor in den Prafixbereich gestellt. Beim Kommando FIND wird der Cursor auf die gefundene Zeichenkette platziert, und die Zeile wird nicht notwendigerweise zur aktuellen Zeilen.

Zeilen zeigen oder verbergen

Mit dem Hauptkommando EXCLUDE können Sie bestimmte Zeilen von der Anzeige ausschließen. Beispiel: Durch das Kommando

```
EXC 'ABC' .X .Y ALL
```

werden alle Zeilen mit der Zeichenkette ABC innerhalb des Blocks mit der Bezeichnung .X und .Y von der Anzeige ausgeschlossen. Durch ein nicht qualifiziertes EXCLUDE-Kommando wird die aktuelle Zeile ausgeschlossen. Jede einzelne, von der Anzeige ausgeschlossene Zeile bzw. Zeilenblock wird ersetzt durch eine Zeile mit EXCLUDEs und einer Meldung, die Sie informiert, wie viele Zeilen ausgeschlossen sind.

Um ausgeschlossene Zeilen wieder zur Anzeige zu bringen, können Sie das Hauptkommando INCLUDE benutzen. Beispiel: Durch das Kommando

```
IN C'Abc' ALL
```

werden alle ausgeschlossenen Zeilen wieder angezeigt, die die Zeichenkette Abc genauso, wie hier eingegeben, enthalten. Ein nicht qualifiziertes INCLUDE-Kommando bringt die erste Zeile in dem ausgeschlossenen wieder zur Anzeige.

Die Hauptkommandos EXCLUDE und INCLUDE können mit den gleichen Zeichenketten- und Suchoperanden abgesetzt werden, die beim FIND-Kommando beschrieben sind. Ausnahme: Der Operand für die Suchrichtung ALL bedeutet, dass alle Zeilen mit der angegebenen Zeichenkette von der Anzeige ausgeschlossen bzw. zur Anzeige gebracht werden.

Darüber hinaus ist es möglich, mit den nachfolgend aufgeführten Zeilenkommandos Zeilen von der Anzeige auszuschließen bzw. zur Anzeige zu bringen.

Zeilenkommando	Erläuterung
X	Die markierte Zeile wird von der Anzeige ausgeschlossen.
X*n*	Die nächsten *n* Zeilen werden von der Anzeige ausgeschlossen.
XX	Markiert die erste Zeile eines Datenblocks, der von der Anzeige ausgeschlossen. werden soll. Ein zweites XX-Zeilenkommando ist nötig, um den Block abzugrenzen. Das Ausschließen erfolgt nach Eingabe des zweiten XX-Zeilenkommandos.
F	Bringt die markierte Zeile wieder zur Anzeige bzw. bringt die erste Zeile des ausgeschlossenen Blocks zur Anzeige.
F*n*	Bringt die ersten *n* Zeilen des ausgeschlossenen Blocks wieder zur Anzeige bzw. bringt die erste Zeile des ausgeschlossenen Blocks zur Anzeige.
L*n*	Bringt die letzten *n* Zeilen des ausgeschlossenen Blocks wieder zur Anzeige.

Durch Absetzen des Hauptkommandos XSWAP können Sie ausgeschlossene Zeilen gegen angezeigte Zeilen austauschen.

Ränder, Tabulator- und Spaltenpositionen anzeigen

Mit den unten aufgeführten Zeilenkommandos können Sie sich die (mit dem Hauptkommando `BNDS` gesetzten) Positionen der Ränder, die (mit dem Hauptkommando `TABS` gesetzten) Tabulatoren sowie die Spaltenpositionen des Editierbereichs auf jeder beliebigen Zeile anzeigen lassen:

Zeilenkommando	Erläuterung
`BNDS`	Zeigt die Randpositionen auf dieser Zeile an.
`COLS`	Zeigt die Spaltenpositionen auf dieser Zeile an.
`TABS`	Zeigt die Tabulatorpositionen auf dieser Zeile an.

Weitere Informationen siehe *Daten anlegen und ändern*.

47 Kommandos benutzen

- Ausführung eines Kommandos 444

Der Software AG Editor bietet zwei Arten von Kommandos, mit denen Sie die Bearbeitung beeinflussen können:

- **Hauptkommandos**

  Hauptkommandos betreffen die gesamten, zurzeit im Quellcode-Arbeitsbereich des Editors enthaltenen Daten. Die Hauptkommandos, die Ihnen zur Verfügung stehen, sind abhängig von der Anwendung, die Sie momentan benutzen.

  Das in den Beispielen in der *Software AG Editor*-Dokumentation verwendete Kommandoformat entspricht der Kurzform der Hauptkommandosyntax. Diese Beispiele sind keineswegs ausschließlich. Eine vollständige Übersicht finden Sie im Abschnitt ***Hauptkommandos (Übersicht)***.

- **Zeilenkommandos**

  Zeilenkommandos betreffen die Quellcodezeile, vor der sie eingegeben werden oder auf einen Datenblock, der durch Zeilenkommandos eingegrenzt wird.

  Eine vollständige Zusammenfassung der Zeilenkommandos finden Sie im Abschnitt ***Zeilenkommandos (Übersicht)***.

Die Ausführung der Kommandos geschieht in folgender Reihenfolge:

- Datenänderungen
- Zeilenkommandos
- Hauptkommandos

Ausführung eines Kommandos

Je nachdem, was anlässlich der Installation in der Konfiguration angegeben wurde, können Hauptkommandos und Zeilenkommandos in Kleinschreibung eingegeben werden. In den Kommandobeschreibungen in der *Software AG Editor*-Dokumentation werden alle Kommandos in Großschreibung dargestellt, um sie als Kommandos zu unterscheiden.

Um ein Hauptkommando auszuführen:

- Geben Sie das Hauptkommando in der Kommandozeile (**COMMAND===>**) ein, und drücken Sie Enter.

  Sie können während eines Eingabevorgangs auch mehrere Kommandos eingeben, indem Sie diese durch ein Semikolon (;) voneinander abtrennen.

Um ein Zeilenkommando auszuführen:

- Geben Sie ab der äußersten linken Spalte im **Präfixbereich** neben der betreffenden Quellcode-Zeile das Zeilenkommando ein, indem Sie die in diesen Spalten enthaltenen Zeichen überschreiben. Drücken Sie Enter.

  Wenn Sie Enter drücken, werden Apostrophe (' ' ' ' ' ') oder Sternchen (******) in den benutzten Zeilen durch Zeilennummern ersetzt. Nicht benutzte Zeilen werden automatisch gelöscht.

  Oder:

  Geben Sie in der ersten Spalte des Editierbereichs ein Zeilenkommando mit vorangehendem **Fluchtzeichen** („Escape“-Zeichen) ein, und drücken Sie Enter. Das Standard-Fluchtzeichen ist der Punkt (.).

  Oder:

  In der Kommandozeile geben Sie ein Zeilenkommando mit vorangestelltem Doppelpunkt (:) ein.

  Der Cursor markiert dann die betreffende Zeile. Drücken Sie Enter.

48 Daten anlegen und ändern

- Zeilen einfügen und löschen 448
- Zeilen kopieren, verschieben, überlagern und wiederholen 450
- Ein Fenster mit Daten kopieren oder verschieben 454
- Horizontale und vertikale Bereichsgrenzen einstellen 460
- Maskenzeile definieren 462
- Daten zwischen Bereichsgrenzen anordnen 463
- Daten zentrieren 464
- Daten ausrichten 465
- Physischen oder logischen Tabulator benutzen 468
- Zeilen in alphabetischer Reihenfolge sortieren 473
- Zeichenkette suchen 473
- Zeichenkette ersetzen 477

Dieser Abschnitt beschreibt die Funktionen, die bei allen Objekttypen zur Verfügung stehen, bei denen Hauptkommandos oder Zeilenkommandos verwendet werden können.

Zeilen einfügen und löschen

Dieser Abschnitt beschreibt die Zeilenkommandos und Hauptkommandos, die zum Einfügen und Löschen von Zeilen zur Verfügung stehen.

- Verfügbare Zeilenkommandos
- POWER-Kommando
- DELETE-Kommando

Verfügbare Zeilenkommandos

Zum Einfügen und Löschen von Zeilen auf dem Editor-Bildschirm stehen Ihnen folgende Zeilenkommandos zur Verfügung:

Zeilenkommando	Erläuterung
`D`	Löscht die markierte Zeile.
`D`*n*	Löscht die nächsten *n* Zeilen.
`DD`	Markiert die erste Zeile eines zu löschenden Blocks. Zum Eingrenzen des Blocks ist ein zweites `DD`-Kommando nötig. Die Löschung erfolgt nach der Eingabe des zweiten `DD`-Kommandos..
`DX`	Löscht die mit `.X` markierte Zeile.
`DY`	Löscht die mit `.Y` markierte Zeile.
`DX-Y`	Löscht den Zeilenblock ab der mit `.X` markierten Zeile bis zur mit `.Y` markierten Zeile.
`I`	Fügt eine Zeile unterhalb der aktuellen Zeile ein. Der Editor wird auf den Einfügemodus umgeschaltet. Das heißt, wenn Sie in der neuen Zeile Daten oder ein Leerzeichen eintippen und Enter drücken, wird automatische eine neue Zeile eingefügt und der Cursor in diese Zeile platziert. Falls Sie in eine eingefügte Zeile keine neuen Daten eingeben und Enter drücken, verlässt der Editor den Einfügemodus, und die Leerzeile wird gelöscht. Wenn Sie eine Maskenzeile definiert haben und das Kommando `MASK` ist auf `ON` gesetzt, wird die Maskenzeile eingefügt, wenn Sie dieses Kommando absetzen.
`I`*n*	Fügt *n* Zeilen unterhalb der aktuellen Zeile ein. Sie können in die neuen Zeilen Daten eingeben. Wenn Sie Enter drücken, werden nicht benutzte Zeilen gelöscht, eine Leerzeile bleibt jedoch erhalten, und der Cursor wird in diese Zeile platziert (d.h. der Editor bleibt im Einfügemodus).

Zeilenkommando	Erläuterung
TE	Schaltet den Editor in den Texteingabemodus. Das heißt, dass ab dieser Zeile der Editor-Bildschirm leer ist (ohne Zeilennummern) und dass Sie Daten eingeben können. Wenn Sie Enter drücken, werden alle übrig gebliebenen Leerzeilen gelöscht, Zeilennummern werden wieder angezeigt und der Text wird, unter Berücksichtigung der Ausrichtungseinstellungen innerhalb der eingestellten Randbegrenzungen neu formatiert. Siehe auch *POWER-Kommando*.
W	Öffnet ein einzeiliges Fenster. Es wird keine neue Zeile eingefügt, wenn Sie in dieses Fenster Daten eingeben und Enter drücken.
W*n*	Öffnet ein Fenster mit *n* Zeilen. Wenn Sie Enter drücken, werden alle nicht benutzten Zeilen gelöscht.

POWER-Kommando

Sie können auch das Hauptkommando POWER benutzen, um in den Texteingabemodus umzuschalten.

Wenn Sie das Hauptkommando POWER absetzen, erscheint ein leerer Editor-Bildschirm (ohne Zeilennummern), und Sie können Daten eingeben. Wenn Sie Enter drücken, werden übrig gebliebene Leerzeilen gelöscht, Zeilennummern werden wieder angezeigt und der Text wird, unter Berücksichtigung der Ausrichtungseinstellungen innerhalb der eingestellten Randbegrenzungen neu formatiert.

DELETE-Kommando

Zum Löschen von Zeilen können Sie auch das Hauptkommando DELETE benutzen.

Beispiel:

```
DEL C'Abc' .X .Y 10 30 ALL
```

Dieses Kommando löscht alle Zeilen, die die Zeichenkette Abc genau so wie hier angegeben enthalten zwischen den Spalten 10 bis 30 innerhalb des Blocks, der mit den Markierungen .X und .Y eingegrenzt ist.

Sie können alle weiter oben für das FIND-Kommando beschriebenen Operanden angeben. Ausnahme: Der Richtungsoperand ALL gibt die Löschung aller Zeilen mit der angegebenen Zeichenkette an. Ein nicht qualifiziertes DELETE-Kommando löscht die aktuelle Zeile.

Zeilen kopieren, verschieben, überlagern und wiederholen

Mit den folgenden Zeilenkommandos können Sie Zeilen oder Datenblöcke kopieren, verschieben, überlagern und wiederholen.

Zeilenkommando	Erläuterung
A	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (C, C*n*, CC) oder Verschieben (M, M*n*, MM). Die Daten werden *nach* dieser Zeile eingefügt.
B	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (C, C*n*, CC) oder Verschieben (M, M*n*, MM). Die Daten werden *vor* dieser Zeile eingefügt.
C	Kopiert diese Zeile an die Stelle, die mit einem Zeilenkommando A, B or O markiert ist.
C*n*	Kopiert die nächsten *n* Zeilen an die Position, die mit einem Zeilenkommando A, B or O markiert ist.
CC	Markiert die erste Zeile des zu kopierenden Blocks. Ein zweites Kommando CC ist nötig, um den Block einzugrenzen. Das Kopieren erfolgt, nachdem das Ziel mit einem Zeilenkommando A, B oder O markiert worden ist.
CX	Kopiert die mit .X markierte Zeile. Fügt die Daten nach dieser Zeile ein.
CY	Kopiert die mit .Y markierte Zeile. Fügt die Daten nach dieser Zeile ein.
CX-Y	Kopiert den Zeilenblock ab der Zeile, die mit .X markiert ist, bis zur Zeile, die mit .Y markiert ist. Fügt die Daten nach dieser Zeile ein.
M	Verschiebt diese Zeile an die Position, die mit einem Zeilenkommando A, B oder O markiert ist.
M*n*	Verschiebt die nächsten *n* Zeilen an die Position, die mit einem Zeilenkommando A oder B markiert ist.
MM	Markiert die erste Zeile des zu verschiebenden Blocks. Ein zweites Zeilenkommando MM ist nötig, um den Block abzugrenzen. Das Verschieben erfolgt, nachdem das Ziel mit einem Zeilenkommando A, B oder O markiert worden ist.
MX	Verschiebt die mit .X markierte Zeile. Fügt die Daten nach dieser Zeile ein.
MY	Verschiebt die mit .Y markierte Zeile. Fügt die Daten nach dieser Zeile ein.
MX-Y	Verschiebt den Zeilenblock ab der Zeile, die mit .X markiert ist, zur Zeile, die mit .Y markiert ist. Fügt die Daten nach dieser Zeile ein.
O	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (C, C*n*, CC) oder zum Verschieben (M, M*n*, MM). Die Daten werden mit dieser Zeile zusammengeführt. (Es werden nur Leerzeichen *innerhalb* der Bereichsgrenzen geändert). Siehe auch ***Beispiel für das Überlagern von Daten***.
O*n*	Markiert diese Zeile und die nächsten *n* Zeilen als Zielzeilen für ein Kopier-Zeilenkommando (C, C*n*, CC) oder Verschieben-Zeilenkommando (M, M*n*, MM). Die kopierten oder verschobenen Zeilen werden mit diesen Zeilen zusammengeführt, d.h. Leerzeichen in den Zeilen werden überlagert. Siehe auch ***Beispiel für das Überlagern von Daten***.

Zeilenkommando	Erläuterung
OO	Markiert die erste Zeile eines Zielzeilenblocks für ein Kopier-Zeilenkommando (C, C*n*, CC) oder Verschiebe-Zeilenkommando (M, M*n*, MM). Ein zweites Zeilenkommando OO ist nötig, um die letzte Zeile des Zielzeilenblocks zu markieren. Die kopierten oder verschobenen Zeilen werden mit diesen Zeilen zusammengeführt, d.h. Leerzeichen in den Zeilen werden überlagert. Siehe auch ***Beispiel für das Überlagern von Daten***.
R	Wiederholt diese Zeile einmal.
R*n*	Wiederholt diese Zeile *n* Mal.
RR	Markiert die erste Zeile eines Blocks, der wiederholt werden soll. Ein zweites Kommando RR ist nötig, um den Block einzugrenzen. Die Wiederholung erfolgt, nachdem das zweite Kommando RR eingegeben worden ist.
RR*n*	Wiederholt den Block, der durch zwei Zeilenkommandos RR*n* eingegrenzt ist, *n* Mal. Klammerkommandos) (siehe weiter unten) verschieben Text um die angegebene Anzahl an Spalten, jedoch nur *innerhalb* der angegebenen Begrenzungen, deshalb ist es möglich, dass der Text teilweise verschwindet.
)	Verschiebt diese Zeile ab der linken Begrenzung um eine Spalte nach Rechts.
)*n*	Verschiebt diese Zeile um *n* Spalten nach Rechts.
))*n*	Markiert die erste Zeile eines um *n* Spalten nach Rechts zu verschiebenden Blocks. Ein zweites Kommando))*n* ist nötig, um den Block einzugrenzen. Das Verschieben erfolgt, nachdem das zweite Kommando))*n* eingeben worden ist. Zwei nicht qualifizierte Zeilenkommandos)) verschieben den Block um eine Spalte nach Rechts.
(	Verschiebt diese Zeile um eine Spalte nach Links.
(*n*	Verschiebt diese Zeile um *n* Spalten nach Links.
((*n*	Markiert die erste Zeile eines um *n* Spalten nach Links zu verschiebenden Blocks. Ein zweites Zeilenkommando ((*n* ist nötig, um den Block einzugrenzen. Das Verschieben erfolgt, nachdem das zweite Kommando ((*n* eingeben worden ist. Zwei nicht qualifizierte Zeilenkommandos ((verschieben den Block um eine Spalte nach Links. Wenn Sie die Symbole > (größer als) oder < (kleiner als) benutzen (siehe unten), um Daten zu verschieben, dann ist das Verschieben *innerhalb* der gesetzten Begrenzungen maximal bis zum nächsten Zeichen, das kein Leerzeichen ist, möglich.
>	Verschiebt diese Zeile um eine Spalte nach Rechts.
>*n*	Verschiebt diese Zeile um *n* Spalten nach Links.
>>*n*	Markiert die erste Zeile eines um *n* Spalten nach Rechts zu verschiebenden Blocks. Ein zweites Zeilenkommando >> ist nötig, um den Block einzugrenzen. Das Verschieben erfolgt, nachdem das zweite Kommando >> eingeben worden ist. Zwei nicht qualifizierte Zeilenkommandos >> verschieben den Block um eine Spalte nach Rechts.
<	Verschiebt diese Zeile um eine Spalte nach Links.
<*n*	Verschiebt diese Zeile um *n* Spalten nach Links.

Zeilenkommando	Erläuterung
<<*n*	Markiert die erste Zeile eines um *n* Spalten nach Links zu verschiebenden Blocks. Ein zweites Zeilenkommando << ist nötig, um den Block einzugrenzen. Das Verschieben erfolgt, nachdem das zweite Kommando << eingeben worden ist. Zwei nicht qualifizierte Zeilenkommandos << verschieben den Block um eine Spalte nach Links.

Beispiel für das Überlagern von Daten

Mit einem Überlagerungszeilenkommando (O, O*n* oder OO) können Sie einspaltige Listen in ein mehrspaltiges Format (d.h., in eine Tabellenform) zusammenführen. Sie können ein Überlagerungszeilenkommando in Verbindung mit einem Kopier-Zeilenkommando (C, C*n* oder CC) oder einem Verschiebe-Zeilenkommando (M, M*n* oder MM) benutzen.

Dies wird in den folgenden zwei Abbildungen veranschaulicht:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------------ Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
000090 //JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 // 'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
OO0140 //STEPLIB DD
000150 //        DD
000160 //        DD
OO0170 //        DD
MM0180                    DISP=SHR,DSN=SPF.SYSF.ADALOAD
000190                    DISP=SHR,DSN=SPF.SYSF.LOAD
000200                    DISP=SHR,DSN=SPF.SYSF.PROD.INST   * OPS INSTALL
MM0210                    DISP=SHR,DSN=SPF.SYSF.SOURCE      * OPS DOCUMENTS
000220 //DDCARD  DD *
000230 ADARUN DA=9,DE=3380,SVC=249
000240 //CMPRINT DD SYSOUT=X
000250 //CMPRT01 DD SYSOUT=X
000260 //CMWKF01 DD DUMMY
000270 //CMSYNIN DD *
000280 LOGON SYSMAIN2
000290 CMD C C * FM  #FL DBID  #FD FNR  #FF TO  #TL DBID  #TD FNR  #TF REP
000300 FIN
000310 /*
****** **************************** bottom of data ****************************



Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

In der obigen Abbildung sind die Zeilen 180 bis 210 mit dem Zeilenkommando `MM` (Verschieben) markiert. Sie werden über die Zeilen 140 bis 170 gelegt, die mit dem Zeilenkommando `OO` (Überlagern) markiert sind.

Die folgende Abbildung zeigt das Ergebnis der Ausführung der Zeilenkommandos, die in der obigen Abbildung zu sehen sind. Die Zeilen 180 bis 220 sind über die Zeilen 140 bis 170 gelegt worden:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-76K ------------ Checkpoint done
 COMMAND===>                                                    SCROLL===> CSR
000090 //JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 // 'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
000140 //STEPLIB DD        DISP=SHR,DSN=SPF.SYSF.ADALOAD
000150 //        DD        DISP=SHR,DSN=SPF.SYSF.LOAD
000160 //        DD        DISP=SHR,DSN=SPF.SYSF.PROD.INST   * OPS INSTALL
000170 //        DD        DISP=SHR,DSN=SPF.SYSF.SOURCE      * OPS DOCUMENTS
000180 //DDCARD  DD *
000190 ADARUN DA=9,DE=3380,SVC=249
000200 //CMPRINT DD SYSOUT=X
000210 //CMPRT01 DD SYSOUT=X
000220 //CMWKF01 DD DUMMY
000230 //CMSYNIN DD *
000240 LOGON SYSMAIN2
000250 CMD C C * FM  #FL DBID  #FD FNR  #FF TO  #TL DBID  #TD FNR  #TF REP
000260 FIN
000270 /*
****** **************************** bottom of data ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Ein Fenster mit Daten kopieren oder verschieben

Sie können ein Fenster mit Daten für Verschiebe- oder Kopieraktionen markieren. Auf diese Weise können Sie Daten, die nicht am Anfang einer Zeile beginnen und nicht am Ende einer Zeile enden, kopieren oder verschieben. Diese Aktionen können Sie mittels Editor-Zeilenkommandos und/oder Editor-Hauptkommandos durchführen.

Wenn Sie ein Fenster definieren, werden alle Daten auf dem Bildschirm, die sich zwischen dem Anfang und dem Ende des Fensters befinden, zu einem Teil des Fensters.

Dieser Abschnitt behandelt folgende Themen:

- Verfügbare Zeilenkommandos
- Verfügbare Hauptkommandos
- Beispiel für die Benutzung eines Datenfensters

Verfügbare Zeilenkommandos

Zeilenkommando	Erläuterung
`WS`	Markiert den Anfang eines Datenfensters. Die Cursor-Position markiert die Spalte, ab der die Daten gelesen werden. Falls sich der Cursor nicht in der Zeile befindet, bei der das Kommando eingegeben wurde, beginnt das Fenster in Spalte 1.
`WS`*n*	Das Datenfenster beginnt in Spalte *n* dieser Zeile.
`WE`	Markiert das Ende eines Datenfensters. Funktion wie bei `WS`. Falls das Fenster in derselben Zeile beginnen und enden soll, tippen Sie das `WS`-Kommando über das `WE`-Kommando. Der Editor bestätigt das eingestellte Fenster mit den Meldungen `WS` (`WS`*n*) und `WE` (`WE`*n*) im Präfixbereich bzw. mit `WW`, wenn sich Anfang und Ende des Fensters in derselben Zeile befinden. Bevor Sie ein neues Fenster definieren, müssen Sie das alte Fenster mit dem `RESET`-Kommando zurücksetzen, um einen Kommandokonflikt zu vermeiden.
`WE`*n*	Das Datenfenster endet in Spalte *n* dieser Zeile.
`WC`	Kopiert das Datenfenster. Die Cursor-Position markiert die Spalte, an der die Zeile aufgespalten werden soll, um die kopierten Daten einzufügen.
`WC`*n*	Spaltet diese Zeile in Spalte *n*, und kopiert die Daten zwischen die beiden Teilen dieser Zeile.
`WM`	Verschiebt das Datenfenster. Funktion wie bei `WC`, jedoch werden die ursprünglichen Daten nach dem Kopiervorgang gelöscht.
`WM`*n*	Spaltet diese Zeile in Spalte *n*, und verschiebt die Daten zwischen die beiden Teilen dieser Zeile.

Verfügbare Hauptkommandos

Main Command	Erläuterung

WINDOW	Definiert ein Fenster. Als Parameter werden die Startzeile und Startspalte und die Endzeile und Endspalte angeben. Es muss mindestens ein Parameter angegeben werden. Beispiele:
	`WINDOW 5 10 24 13`
	Definiert ein Fenster, das in Zeile 5/Spalte 24 beginnt und in Zeile 10/Spalte 13 endet.
	`WINDOW 5 10 24`
	Definiert ein Fenster, das in Zeile 5/Spalte 24 beginnt und in Zeile 10/letzte Spalte endet.
	`WINDOW 5 10`
	Definiert ein Fenster, das in Zeile 5/erste Spalte beginnt und in Zeile 10/letzte Spalte endet.
	`WINDOW 5 5`
	Definiert ein Fenster, das in Zeile 5/erste Spalte beginnt und in Zeile 5/letzte Spalte endet.

CWINDOW	Kopiert ein Fenster, das mit dem WINDOW-Kommando definiert wurde. Optionale Parameter geben die Zeile an, bei der das Fenster eingefügt werden soll. Beispiele:
	`CWINDOW 5`
	Kopiert das Fenster nach Zeile 5.
	`CWINDOW 5 24`
	Spaltet Zeile 5 bei Spalte 24 und kopiert das Fenster zwischen die beiden Teile.

DWINDOW	Löscht ein Datenfenster, das mit dem WINDOW-Befehl definiert wurde.
MWINDOW	Verschiebt ein Fenster, das mit dem WINDOW-Befehl definiert wurde. Funktion wie beim CWINDOW-Kommando, jedoch werden die ursprünglichen Daten nach dem Kopiervorgang gelöscht.

Beispiel für die Benutzung eines Datenfensters

Dieser Abschnitt enthält ein Beispiel für das Definieren und Verschieben von Text mittels eines Datenfensters, wobei entweder Zeilenkommandos oder die entsprechenden Hauptkommandos zur Anwwendung kommen.

Das Beispiel bezieht sich auf den Text, der in Schritt 1 weiter unten gezeigt wird. Es wird davon ausgegangen, dass Sie den ganzen Satz, der mit `Note that when...` (Zeile 8) beginnt, so verschieben wollen, dass er auf den angezeigten Text folgt, der mit `...copy operations` (Zeile 3) endet.

Um ein Fenster mit Zeilenkommandos zu definieren und zu verschieben:

1 Geben Sie den unten abgebildeten Text ein:

```
EDIT-NAT:NATLIB1(WINEX)-Text->Struct-Free-78K ---------------- Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ***************************** top of data ******************************
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 editor line commands and/or main commands.
000007
000008 Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
000010 the window become part of the window. Available line commands are:
****** **************************** bottom of data ****************************
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

2 Geben das Zeilenkommando `WS` in Zeile 8 (der ersten zu verschiebenden Datenzeile) ein. Positionieren Sie den Cursor in die betreffende Spalte (`N` des Worts `Note`) und drücken Sie Enter.

Die Meldung `WS55` erscheint im Präfixbereich der Zeile 8 und zeigt die Nummer der gewählten Spalte an:

```
EDIT-NAT:NATLIB1(WINEX)-Text->Struct-Free-78K ---------------- Block is pending
 COMMAND===>                                                    SCROLL===> CSR
****** ***************************** top of data ******************************
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 editor line commands and/or main commands.
000007
WS55   Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
000010 the window become part of the window. Available line commands are:
****** **************************** bottom of data ****************************
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

3 Geben Sie das Zeilenkommando `WE` in Zeile 10 (die letzte Zeile der zu verschiebenden Daten) ein, platzieren Sie den Cursor auf die letzte zu verschiebende Spalte (Satzende-Zeichen (`.`) nach `window`) und drücken Sie Enter.

Die Meldung `WE37` erscheint im Präfixbereich der Zeile 10:

```
EDIT-NAT:NATLIB1(WINEX)-Text->Struct-Free-78K ---------------- Block is pending
 COMMAND===>                                                    SCROLL===> CSR
****** ***************************** top of data ******************************
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 editor line commands and/or main commands.
000007
WS55   Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
WE37   the window become part of the window. Available line commands are:
****** **************************** bottom of data ****************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

4 Geben Sie das Zeilenkommando `WM` in Zeile 3 ein (die Daten werden in die Folgezeile, Zeile 4, verschoben), und bewegen Sie den Cursor in die Spalte, bei der Zeile 3 gesplittet werden soll (das Leerzeichen vor den Wort `This`). Drücken Sie Enter.

Der angegebene Textabschnitt wird verschoben:

```
EDIT-NAT:NATLIB1(WINEX)-Text->Struct-Free-78K ---------------- Checkpoint done
 COMMAND===>                                                  SCROLL===> CSR
****** ***************************** top of data ******************************
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations.
000004 Note that when
000005 you define a window, all data on your screen between start and end of
000006 the window become part of the window.
000007 This
000008 allows you to copy or move data that does not start or end at the
000009 beginning or end of a line. This function can be performed using
000010 editor line commands and/or main commands.
000011
000012 Below are some examples of copying windows with data.
000013                                        Available line commands are:
****** ************************** bottom of data ***************************
```

```
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Sie können das gleiche Ergebnis wie in der zuvor beschriebenen Anleitung erzielen, indem Sie die folgende Kommandoabfolge benutzen.

Um ein Fenster mit Hauptkommandos zu definieren und zu verschieben:

1 Geben Sie den Text ein, der unter **Schritt 1** weiter oben abgebildet ist.

2 Geben Sie folgende Hauptkommandos in der Kommandozeile ein:

```
WINDOW 8 10 55 37;MWINDOW 3 64
```

Der angegebene Textabschnitt wird wie in **Schritt 4** dargestellt verschoben.

Horizontale und vertikale Bereichsgrenzen einstellen

Es gibt Editor-Kommandos, mit denen Sie bei bestimmten Funktionen die horizontalen und vertikalen Bereichsgrenzen einstellen können, zum Beispiel die Hauptkommandos `FIND`, `CHANGE`, `CEnter`, `ORDER`, `JLEFT` und `JRIGHT` sowie die entsprechenden Zeilenkommandos (Zum Beispiel `TC`, `TO`, `LJ` oder `RJ`).

Dieser Abschnitt behandelt folgende Themen:

- Bereichsgrenzen einstellen
- Einzelne oder mehrere Zeilen markieren

Bereichsgrenzen einstellen

Mit dem Hauptkommando `BNDS` können Sie die horizontalen Bereichsgrenzen so wie in der nachfolgenden Beispiel-Anleitung beschrieben einstellen.

Um Bereichsgrenzen einzustellen und anzuzeigen:

1 Setzen Sie folgendes Hauptkommando ab:

```
BNDS 20 50
```

Die horizontalen Grenzen werden bei den Spalten 20 und 50 gesetzt.

2 Um die aktuelle Einstellung der Bereichsgrenzen anzuzeigen, setzten Sie das folgende Zeilenkommando ab:

```
BNDS
```

Die folgende Abbildung zeigt das Ergebnis des Hauptkommandos `BNDS 20 50`, auf das ein Zeilenkommando `BNDS` in Zeile 2 folgt:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------------ Columns 001 072
 COMMAND===>                                                    SCROLL===> CSR
****** ***************************** top of data ****************************
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
=bnds>                  <                           >
000010   RESET #JOBNAME(A8)
000020   RESET #FD(N3) #FL(A8) #FF(N3)
000030   RESET #TD(N3) #TL(A8) #TF(N3)
000040   COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050   SET CONTROL 'WL60C6B005/010F'
000060   INPUT  'Enter PARAMETERS FOR LIBRARY COPY:'
000070    /     'FROM:   DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080    /     'TO  :   DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090 // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 //       'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
000140 //STEPLIB  DD  DISP=SHR,DSN=OPS.SYSF.V5.ADALOAD
000150 //         DD  DISP=SHR,DSN=OPS.SYSF.PROD.LOAD
000160 //DDCARD   DD  *
000170 ADARUN DA=9,DE=3380,SVC=249
000180 //CMPRINT  DD  SYSOUT=X
000190 //CMPRT01 DD SYSOUT=X
000200 //CMWKF01 DD DUMMY
000210 //CMSYNIN DD *
000220 LOGON SYSMAIN2
000230 CMD C C * FM  #FL DBID  #FD FNR  #FF TO  #TL DBID  #TD FNR  #TF REP
000240 FIN
000250 /*
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Einzelne oder mehrere Zeilen markieren

Sie können die aktuelle Zeile (die sich zurzeit am Anfang des Editierbereichs befindet) oder einen Zeilenblock markieren, indem Sie entweder das Hauptkommando `LABEL` oder entsprechende Zeilenkommandos benutzen.

Um aktuelle Zeilen mit `LABEL` zu markieren:

1 Setzen Sie das folgende Hauptkommando ab:

```
LABEL .X
```

Die aktuelle Zeile wird mit `.X` markiert.

2 Um einen Zeilenblock einzugrenzen, können Sie jetzt zur nächsten gewünschten Begrenzungszeile blättern und dann das folgende Hauptkommando absetzen:

```
LABEL .Y
```

Die neue aktuelle Zeile wird mit `.Y` markiert, womit die letzte Zeile eines Zeilenblocks gekennzeichnet ist.

Um bestimmte Zeilen mittels Zeilenkommandos zu markieren:

1 Setzen Sie neben der ersten zu markierenden Zeile folgendes Zeilenkommando ab:

```
.X
```

Die angegebene Zeile wird mit `.X` markiert.

2 Setzen Sie neben der letzten zu markierenden Zeile folgendes Zeilenkommando ab:

```
.Y
```

Die angegebene Zeile wird mit `.Y` markiert, womit ein Zeilenblocks gekennzeichnet ist, der bei der mit `.X` markierten Zeile beginnt.

Tipp: Sie können zum Markieren von Zeilen eine beliebige Zeichenkette benutzen, z.B. `.START` und `.END`.

Beispiele für die Verwendung von Markierungszeilen siehe *Zeichenkette suchen* und *Zeichenkette ersetzen*.

Maskenzeile definieren

Sie können Daten definieren, die automatisch in eine Zeile platziert werden, und zwar mittels einer Zeileneinfüge-Operation (z.B. mit dem Zeilenkommando `I` oder `W`). Eine solche Zeile wird als Maskenzeile bezeichnet. Eine Maskenzeile ist nützlich, wenn Sie mehrere Codezeilen schreiben müssen, die identisch oder sehr ähnlich sind.

Anmerkung: Während einer Bearbeitungssitzung können Sie nur eine Maskenzeile haben. Wenn Sie eine neue Maskenzeile definieren, wird eine bereits vorhandene Maskenzeilendefinition automatisch mit dem neuen Wert aktualisiert.

Um eine Maskenzeile zu definieren und zu benutzen:

1 Setzen Sie folgendes Zeilenkommando ab:

```
MASK
```

Über der Zeile, in der Sie das Kommando eingegeben haben, erscheint eine Leerzeile mit `=mask>`.

2 Geben Sie die Daten, die Sie als Maskenzeile definieren wollen, in die Leerzeile ein, und drücken Sie Enter.

Die Maskenzeile steht nun solange für den aktuellen Quellcode zur Verfügung, bis Sie die Maske mit einer neuen Maskenzeile aktualisieren oder bis Sie die Maskenfunktion deaktivieren.

3 Setzen Sie folgendes Hauptkommando ab:

```
MASK ON
```

Die Maskenfunktion wird aktiviert. Die definierte Maske erscheint nun in allen Zeilen, die durch eine Zeileneinfüge-Operation hinzugefügt werden.

4 Setzen Sie ein Einfüge-Zeilenkommando ab, zum Beispiel:

```
I2
```

Es werden zwei neue Zeilen mit dem Text aus der Maskenzeile in den Quellcode eingefügt. Der Text einer Maskenzeile erscheint in allen Zeilen, die durch ein Einfüge-Zeilenkommando hinzugefügt werden.

5 Ändern Sie den Text in den neuen Zeilen. Wenn Sie den Text nicht ändern, wird jede eingefügte Zeile gelöscht, wenn Sie das nächste Mal Enter drücken.

6 Falls erforderlich, deaktivieren Sie die Maskenfunktion mit dem folgende Hauptkommando:

```
MASK OFF
```

Das Kommando `MASK OFF` deaktiviert die Maskenfunktion, löscht aber nicht den Inhalt der Maskenzeile.

Siehe auch `MASK` in *Hauptkommandos (Übersicht)*.

Daten zwischen Bereichsgrenzen anordnen

Sie können die Einrückung angegebener Zeilen ändern, indem Sie das Hauptkommando `ORDER` zusammen mit einer Bereichsgrenzen-Einstellung benutzen. Beispiel: Die Kommandoabfolge

```
BNDS 3;ORDER 5 20
```

verschiebt die Zeilen 5 bis 20 soweit nach Rechts, dass sie in Spalte 3 beginnen.

Anmerkung: Falls das Ende einer neu angeordneten Zeile jenseits der rechten Bereichsgrenze zu liegen käme, wird sie automatisch gesplittet.

Um die verschobenen Daten wieder nach links auszurichten, benutzen Sie das Kommando `JLEFT`.

Sie können außerdem die Einrückung von Zeilen oder eines Zeilenblocks ändern, indem Sie Zeilenkommandos absetzen. Dabei gilt ebenfalls, dass, falls das Ende einer Zeile jenseits der rechten Bereichsgrenze zu liegen käme, diese Zeile automatisch gesplittet wird.

Zeilenkommando	Erläuterung
TF	Ordnet die Daten ab der Zeile neu an, in der es eingegeben wurde, bis zum Ende des Paragraphs oder bis zur nächsten Leerzeile bei der rechten Bereichsgrenze. Dieses Zeilenkommando kann mit einem numerischen Wert, der die rechte Bereichsgrenze angibt, eingegeben werden. Beispiel: Das Zeilenkommando TF50 ordnet die Daten bei Spalte 50.
TO	Markiert eine Zeile, die ausgerichtet werden soll.
TOO	Markiert die erste Zeile eines Datenblocks, der ausgerichtet werden soll. Ein zweites Zeilenkommando TOO ist nötig, um den Block einzugrenzen. Die Ausrichtung erfolgt, nachdem das zweite TOO-Kommando abgesetzt worden ist.

Die Daten können innerhalb der eingestellten Bereichsgrenzen neu angeordnet und bündig mit der linken, der rechten oder mit beiden Bereichsgrenzen angeordnet werden. Dazu dient das Kommando JUSTIFY. Beispiel: Das Kommando:

```
BNDS 6 60;JUSTIFY BOTH
```

aktiviert die Ausrichtung auf die Spalten 5 und 60. Um die Neuausrichtung auszuführen, müssen Sie einen Datenblock mit dem Zeilenkommando TOO markieren.

Außerdem bietet der Editor ein Zeilenkommando, mit dem Sie eine einzelne Zeile in zwei Zeilen aufspalten können. Geben Sie das Zeilenkommando S im Präfixbereich der Zeile ein, die Sie aufspalten möchten. Platzieren Sie den Cursor an die Stelle, an der die Aufspaltung erfolgen soll, und drücken Sie Enter.

Daten zentrieren

Der Editor bietet Kommandos, mit denen Sie angegebene Daten innerhalb eingestellter Bereichsgrenzen zentrieren können. Beispiel: Die Kommandosequenz

```
BNDS 5 60;CEnter 5 15
```

zentriert die Daten in den Zeilen 5 bis 15 zwischen den Spalten 5 und 60.

Anmerkung: Nur der Text zwischen den Bereichsgrenzen wird zentriert. Der Text links und rechts der Bereichsgrenzen ist nicht betroffen.

Alternativ können Sie Zeilenkommandos zum Ausführen der Zentrierungsfunktion benutzen:

Zeilenkommando	Erläuterung
TC	Zentriert diese Zeile innerhalb der eingestellten Bereichsgrenzen.
TCC	Markiert die erste Zeile in einem Datenblock, der zentriert werden soll. Ein zweites Zeilenkommando TCC ist nötig, um den Block einzugrenzen. Die Zentrierung erfolgt, nachdem das zweite Kommando TCC abgesetzt worden ist.

Daten ausrichten

Der Editor bietet verschiedene Hauptkommandos und Zeilenkommandos, mit denen Sie Zeilen oder Datenblöcke auf dem Bildschirm neu ausrichten können, und zwar in Abhängigkeit von den horizontalen Bereichsgrenzen, die Sie eingestellt haben (Hauptkommando BNDS); siehe *Horizontale und Vertikale Bereichsgrenzen einstellen*.

Die Hauptkommandos JLEFT und JRIGHT richten die Daten bündig mit der linken bzw. der rechten Bereichsgrenze aus. Beispiel: Die Kommandosequenz

```
BNDS 16 80;JLEFT 140 170
```

richtet die Daten zwischen den Spalten 16 bis 80 in den Zeilen 140 bis 170 linksbündig mit Spalte 16 aus.

Die folgende Abbildung veranschaulicht dieses Beispiel:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------------ Columns 001 072
 COMMAND===> BNDS 16 80;JLEFT 140 170                           SCROLL===> CSR
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000090 // #JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 //       'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
000140 //STEPLIB  DD           DISP=SHR,DSN=SPF.SYSF.ADALOAD
000150 //         DD           DISP=SHR,DSN=SPF.SYSF.LOAD
000160 //         DD           DISP=SHR,DSN=SPF.SYSF.PROD.INST  * OPS INSTALL
000170 //         DD           DISP=SHR,DSN=SPF.SYSF.SOURCE     * OPS DOCUMENTS
000180 //DDCARD   DD  *
000190 ADARUN DA=9,DE=3380,SVC=249
000200 //CMPRINT  DD  SYSOUT=X
000210 //CMPRT01 DD SYSOUT=X
000220 //CMWKF01 DD DUMMY
000230 //CMSYNIN DD *
000240 LOGON SYSMAIN2
000250 CMD C C * FM  #FL DBID  #FD FNR  #FF TO  #TL DBID  #TD FNR  #TF REP
000260 FIN
000270 /*
****** **************************** bottom of data ****************************




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Der folgende Bildschirm zeigt das Ergebnis des Kommandos, das in der Kommandozeile des obigen Bildschirms angezeigt wird:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------ File has been ordered
 COMMAND===>                                                  SCROLL===> CSR
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000090 // #JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 //       'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
000140 //STEPLIB  DD  DISP=SHR,DSN=SPF.SYSF.ADALOAD
000150 //         DD  DISP=SHR,DSN=SPF.SYSF.LOAD
000160 //         DD  DISP=SHR,DSN=SPF.SYSF.PROD.INST  * OPS INSTALL
000170 //         DD  DISP=SHR,DSN=SPF.SYSF.SOURCE     * OPS DOCUMENTS
000180 //DDCARD   DD  *
000190 ADARUN DA=9,DE=3380,SVC=249
000200 //CMPRINT  DD  SYSOUT=X
000210 //CMPRT01 DD SYSOUT=X
000220 //CMWKF01 DD DUMMY
000230 //CMSYNIN DD *
000240 LOGON SYSMAIN2
000250 CMD C C * FM  #FL DBID  #FD FNR  #FF TO  #TL DBID  #TD FNR  #TF REP
000260 FIN
000270 /*
****** *************************** bottom of data ***************************









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Die Kommandosequenz:

```
BNDS 10;JRIGHT 15
```

richtet die Daten rechts von Spalte 10 in Zeile 15 bis zum Ende des Bildschirms bündig mit der letzten Spalte des Editor-Bildschirms (Spalte 88 des Terminal-Bildschirms) aus.

Alternativ können Sie Zeilen ausrichten, indem Sie eines der nachfolgend aufgeführten Zeilenkommandos benutzen:

Zeilenkommando	Erläuterung
LJ	Richtet die Daten innerhalb der eingestellten Begrenzungen in dieser Zeile bündig mit der linken Begrenzung aus.
LJJ	Markiert die die erste Zeile eines Datenblocks innerhalb der Begrenzungen (die mit dem Hauptkommando BNDS eingestellt wurden), der linksbündig ausgerichtet werden soll. Ein zweites Zeilenkommando LJJ ist nötig, um den Block einzugrenzen. Die Ausrichtung erfolgt, nachdem das zweite Kommando LJJ abgesetzt worden ist.
RJ	Richtet die Daten innerhalb der eingestellten Begrenzungen in dieser Zeile bündig zur rechten Begrenzung aus.
RJJ	Markiert die erste Zeile eines Datenblocks innerhalb der Begrenzungen (die mit dem Hauptkommando BNDS eingestellt wurden), der rechtsbündig ausgerichtet werden soll. Ein zweites Zeilenkommando RJJ ist nötig, um den Block einzugrenzen. Die Ausrichtung erfolgt, nachdem das zweite Kommando RJJ abgesetzt worden ist.

Sie können Daten auch bündig zur linken Begrenzung ausrichten oder Daten zwischen der linken und der rechten Begrenzung ausrichten, indem Sie das Kommando JUSTIFY benutzen.

Beispiel: Die Kommandosequenz

```
BNDS 10 60;JUSTIFY LEFT
```

ermöglicht die Ausrichtung mit der linken Begrenzung. Markieren Sie einen Datenblock mit zwei Zeilenkommandos TOO (Erläuterung weiter unten), um die Daten zwischen den Spalten 10 und 60 so neu zu formatieren, dass sie bündig mit Zeile 10 ausgerichtet sind.

Physischen oder logischen Tabulator benutzen

Dieser Abschnitt beschreibt, wie Sie das Hauptkommando TABS und/oder das Zeilenkommando TABS benutzen können, um die Tabulatorstopps zu steuern.

In den Tabellierungsbeispielen, die in diesem Abschnitt enthalten sind, wird davon ausgegangen, dass das kaufmännische Und-Zeichen (&) als Tabulatorzeichen verwendet wird und dass das Zeilenkommando COLS abgesetzt worden ist, um die Spaltenpositionen anzuzeigen.

Dieser Abschnitt behandelt folgende Themen:

- Hauptkommando TABS benutzen
- Zeilenkommando TABS benutzen
- Beispiel 1 - Tab Positionen
- Beispiel 2 - TABS RIGHT
- Beispiel 3 - TABS DECIMAL
- Beispiel 4 - Gemischte Ausrichtung

- Beispiel 5 - Leerzeichen als Tabulationssymbol

Hauptkommando TABS benutzen

Wenn Sie das Hauptkommando `TABS ON` absetzen, werden die Standard-Tabulatorpositionen, die in Ihrem Editor-Profil eingestellt sind, eingeschaltet, und `tabs on std` erscheint in Ihrem Profil. Um die Tabulatorstopps wieder auszuschalten, müssen Sie das Hauptkommando `TABS OFF` absetzen.

Ausführliche Informationen zur Kommandosyntax, die für das Hauptkommando `TABS` gilt, finden Sie im entsprechenden Abschnitt unter *Hauptkommandos (Übersicht)*.

Dieser Abschnitt behandelt folgende Themen:

- Standard-Tabulatorpositionen einstellen
- Logisches Tabulatorzeichen einstellen
- Ausrichtungsparameter einstellen

Standard-Tabulatorpositionen einstellen

Um die Tabulatorfunktion einzuschalten und die Standard-Tabulatorpositionen für Ihr Profil beispielsweise auf die Spalten 10, 20, 30, 40 und 50 einzustellen, müssen Sie das Hauptkommando `TABS 10 20 30 40 50` absetzen.

Logisches Tabulatorzeichen einstellen

Um die Tabulatorfunktion einzuschalten und das logische Tabulatorzeichen einzustellen, beispielsweise das &-Zeichen, müssen Sie das Hauptkommando `TABS &` absetzen.

Sie können Daten eingeben und diese automatisch zu einer bestimmtem Tabulatorposition verschieben, indem Sie vor den Daten ein logisches Tabulatorzeichen eingeben. Ein einzelnes Tabulatorzeichen verschiebt die Daten an die nächste Tabulatorposition, zwei Tabulatorzeichen verschieben die Daten an die zweite Tabulatorposition usw.

Ausrichtungsparameter einstellen

Außer den Tabulatorpositionen können Sie mit dem Hauptkommando `TABS` noch folgende Parameter angeben:

Parameter	Erläuterung
DECIMAL	Ordnet die Daten so an, dass sich das darin enthaltene Dezimalzeichen an der Tabulatorposition befindet. Siehe auch *Beispiel 3 - TABS DECIMAL*.
LEFT	Ordnet die Daten links von der Tabulatorposition an. Siehe auch *Beispiel 1 - Tab Positions*.
RIGHT	Ordnet die Daten rechts von der Tabulatorposition an. Siehe auch *Beispiel 2 - TABS RIGHT*.

Um das aktuelle logische Tabulatorzeichen und den Verschiebeparameter (unter Ausschluss der Tabulatorpositionen) anzuzeigen, setzen Sie das Hauptkommando PROFILE ab.

Zeilenkommando TABS benutzen

Wenn Sie das Zeilenkommando TABS in einer beliebigen Zeile absetzen, werden in dieser Zeile die in Ihrem Editor-Profil eingestellten Tabulatorpositionen angezeigt und mit einem Stern (*) gekennzeichnet, falls kein logisches Tabulatorzeichen gesetzt worden ist. Mit diesem Kommando wird die Tabulatorfunktion nicht eingeschaltet.

Beispiel: Setzen Sie das Zeilenkommando TABS ab, um die Positionen anzuzeigen, die mit dem Hauptkommando TABS 10 20 30 40 50 eingestellt worden sind.

Dadurch werden die aktuellen Tabulatorpositionen folgendermaßen angezeigt:

```
=tabs>              *         *         *         *         *         *
```

Mehrere logische Tabulatorzeichen und gemischte Ausrichtungsparameter einstellen

Um Daten in einer bestimmten Spalte mit einer bestimmten Einrückung tabellarisch darzustellen, sind mehrere logische Tabulatorzeichen und gemischte Ausrichtungsparameter möglich.

Um mehrere logische Tabulatorzeichen einzustellen, setzen Sie das Zeilenkommando TABS ab und überschreiben Sie jeden Stern (*) mit einem Sonderzeichen. Jegliche Daten, die Sie unter Voranstellung eines dieser Sonderzeichen eingeben, werden in der entsprechenden Spalte tabuliert.

Um gemischte Ausrichtungsparameter einzustellen, geben Sie rechts neben jedem logischem Tabulatorzeichen L (Links), R (Rechts) oder D (Dezimal) für linksbündige, rechtsbündige bzw. am Dezimalzeichen orientierte Ausrichtung ein.

Siehe auch *Beispiel 4 - Gemischte Ausrichtung*.

Beispiel 1 - Tab Positionen

Das Kommando

```
TABS 10 20 40 LEFT
```

aktiviert logische Tabulatoren bei den Tabulatorspalten 10, 20 und 40 mit linksbündiger Ausrichtung. Nachdem Sie Enter gedrückt haben, wird die eingegebene Textzeile

```
&abc &def &ghi
```

folgendermaßen angezeigt:

```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
               abc       def                 ghi
```

Beispiel 2 - TABS RIGHT

Das Kommando

```
TABS RIGHT
```

aktiviert logische Tabulatoren mit rechtsbündiger Ausrichtung. Nachdem Sie Enter gedrückt haben, wird die eingegebene Textzeile

```
&abc &def &ghi
```

folgendermaßen angezeigt:

```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
             abc       def                 ghi
```

Beispiel 3 - TABS DECIMAL

Das Kommando

```
TABS DECIMAL
```

aktiviert logische Tabulatoren mit Ausrichtung am Dezimalzeichen an der Tabulatorposition. Nachdem Sie Enter gedrückt haben, wird die eingegebene Textzeile

```
&15.27$ &16.3 EUR &13 IS
```

folgendermaßen angezeigt:

```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
             15.27$    16.3 EUR             13 IS
```

Beispiel 4 - Gemischte Ausrichtung

Setzen Sie folgendes Kommando ab:

```
TABS 10 20 30 40 50
```

Setzen Sie dann das Zeilenkommando `TABS` ab. Dadurch werden die aktuellen Tabulatorpositionen folgendermaßen angezeigt:

```
=tabs>          *         *         *         *         *
```

Geben Sie je nach Bedarf neben jeder Tabulatorposition ein `L`, `R` oder `D` ein (an nicht markierten Tabulatorpositionen wird der Wert des letzten `TAB`-Kommandos übernommen):

```
=tabs>          *R        *D        *D        *D        *L
```

Nachdem Sie Enter gedrückt haben, wird die eingegebene Textzeile

```
&start &0.01 &0.02 &0.03 &end
```

folgendermaßen angezeigt:

```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
            start        0.01      0.02      0.03       end
```

Beispiel 5 - Leerzeichen als Tabulationssymbol

Setzen Sie folgendes Kommando ab:

```
TABS ' '
```

Es aktiviert die Tabulatorfunktion mit einem Leerzeichen als Tabulatorzeichen. Nachdem Sie Enter gedrückt haben, wird die eingegebene Textzeile

```
this is a blank tabulation
```

folgendermaßen angezeigt:

```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
               this      is        a         blank     tabulation
```

Zeilen in alphabetischer Reihenfolge sortieren

Sie können Datenzeilen in aufsteigender oder absteigender alphabetischer Reihenfolge nach Sortierkriterien sortieren. Beispiel: Das Kommando

```
SORT 10 15
```

sortiert alle Zeilen im Quellcode in aufsteigender Reihenfolge entsprechend den Zeichen ab Spalte 10 bis zur Spalte 15.

Um nur einen Zeilenblock zu sortieren, können Sie beispielsweise die Zeilen, an denen der Block anfangen und enden soll, mit `.X` bzw. `.Y` markieren. Das Kommando:

```
SORT .X .Y D
```

sortiert alle Zeilen in dem mit `.X` und `.Y` markierten Block in absteigender Reihenfolge.

Um beispielsweise einen Zeilenblock entsprechend den Zeichen, die in Spalte 5 beginnen und in Spalte 20 enden, zu sortieren, markieren Sie die Zeilen, wo der Block anfangen und enden soll, mit `.X` und `.Y`. Das Kommando

```
SORT 5 20 .X .Y
```

sortiert alle Zeilen in dem mit `.X` und `.Y` markierten Block in aufsteigender Reihenfolge entsprechend den Zeichen ab Spalte 5 bis zur Spalte 20.

Zeichenkette suchen

Um eine bestimmte Zeichenkette zu finden, können Sie das Hauptkommando `FIND` mit Operanden benutzen, um die die Zeichenkette, den zu durchsuchenden Bereich und die Suchrichtung zu definieren. Der Cursor wird auf das erste Zeichen der gefundenen Zeichenkette platziert. Falls die Zeile, die die gesuchte Zeichenkette enthält, von der Anzeige ausgeschlossen ist, wird sie jetzt in die Anzeige aufgenommen.

Die folgenden Abschnitte beschreiben die möglichen Kommando-Operanden.

Operand für die Definition der Zeichenkette

Der Operand für die Definition der Zeichenkette definiert die zu suchende Zeichenkette.

Folgende Angaben sind möglich:

Operand	Erläuterung
*	Findet die im vorangehenden Kommando FIND angegebene Zeichenkette.
'abc'	Findet die Zeichenkette abc unabhängig davon, ob sie in Klein- oder Großschreibung vorliegt.
C'Abc'	Findet exakt die hier Zeichenkette eingegebene Zeichenkette.
P'a*(char)*c'	Findet die Zeichenkette, deren erstes Zeichen ein a und deren drittes Zeichen ein c ist. *(char)* steht für ein Sonderzeichen, das als Platzhalterzeichen (Wildcard) fungiert. Bedeutung: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ nicht-Leerzeichen - nicht-numerisches Zeichen < Zeichen in Kleinschreibung > Zeichen in Großschreibung
T'abc'	Findet die Zeichenkette abc unabhängig davon, ob sie in Klein- oder Großschreibung vorliegt.
X'D4A8'	Findet die Zeichenkette, die dem Hexadezimalwert D4A8 entspricht.

Operand für die Suche nach Übereinstimmungen

Der Operand für die Suche nach Übereinstimmungen gibt an, ob eine besondere Ausprägung der Zeichenkette gefunden werden soll. Folgende Optionen sind verfügbar:

Operand	Erläuterung
CHARS	Ohne Einschränkungen (jede beliebige Ausprägung der Zeichenkette).
PREFIX	Nur die Ausprägungen, die die Vorsilbe (Präfix) eines Wortes sind.
SUFFIX	Nur die Ausprägungen, die die Nachsilbe (Suffix) eines Wortes sind.
WORD	Nur die Ausprägungen, die ein Wort bilden.

Die Standardeinstellung ist CHARS.

Operand für die Angabe der Suchrichtung

Der Operand für die Angabe der Suchrichtung gibt an, in welcher Richtung die Suche erfolgen soll.

Folgende Optionen sind verfügbar:

Operand	Erläuterung
ALL	Jede beliebige Ausprägung der Zeichenkette (Suche in allen Richtungen).
FIRST	Erstes Vorkommen der Zeichenkette.
LAST	Letztes Vorkommen der Zeichenkette.
NEXT	Nächstes Vorkommen der Zeichenkette ab der Cursor-Position.
PREV	Vorheriges Vorkommen der Zeichenkette.

Die Standardeinstellung ist NEXT.

Operand für die Angabe des Zeilentyps

Der Operand für die Angabe des Zeilentyps gibt an, ob nur nach ausgeschlossenen oder nur nach eingeschlossenen Zeilen gesucht werden soll. Folgende Optionen sind verfügbar:

Operand	Erläuterung
X	Nur nach Zeilen suchen, die von der Anzeige ausgeschlossen sind.
NX	Nur nach Zeilen suchen, die nicht von der Anzeige ausgeschlossen sind.

Wenn dieser Operand weggelassen wird, durchsucht der Editor alle Daten nach der gegebenen Zeichenkette, in von der Anzeige ausgeschlossenen und nicht von der Anzeige ausgeschlossenen Zeilen. Wird die Zeichenkette in einer ausgeschlossenen Zeile gefunden, dann wird die Zeile wieder zur Anzeige gebracht.

Block-Operand

Wenn Sie Zeilen oder einen Zeilenblock markiert haben, können Sie den Block-Operanden benutzen, um den Suchbereich für das FIND-Kommando einzuschränken.

Beispiele für den Gebrauch des Block-Operanden:

Operand	Erläuterung
.X	Suche ab der mit .X markierten Zeile bis zum Ende der Daten.
.X .Y	Suche ab der mit .X markierten Zeile bis zur Zeile, die mit .Y markiert ist.

Dabei können X und Y eine beliebige alphabetische oder eine aus vier Zeichen bestehende Zeichenkette sein.

Spalten-Operand

Mit dem Spalten-Operanden können Sie die Suche nach der gegebenen Zeichenkette zwischen bestimmten Spalten eingrenzen. Beispiele für den Gebrauch des Spalten-Operanden:

Operand	Erläuterung
20	Suche nach der gegebenen Zeichenkette ab Spalte 20 (das erste Zeichen der Zeichenkette muss in Spalte 20 sein).
20 40	Suche nach der gegebenen Zeichenkette zwischen Spalte 20 bis 40.

Beispiele für das FIND-Kommando

```
F C'HILITE' X PREV
```

Suche nach dem vorangegangenen Vorkommen der Zeichenkette HILITE, und zwar exakt wie hier eingegeben. Suche nur nach ausgeschlossenen Zeilen.

```
F P'RCV#' .X .Z 20 30
```

Suche nach der Zeichenkette, die mit RCV beginnt, mit einem numerischen vierten Zeichen zwischen dem Block .X .Z, und zwischen den Spalten 20 bis 30.

```
F X'6C' SUFFIX NX
```

Suche nach dem Zeichen, das dem Hexadezimalwert 6C entspricht, und zwar nur in nicht von der Anzeige ausgeschlossenen Zeilen. Das Zeichen muss am Ende eines Wortes stehen.

Command	Erläuterung
F C'HILITE' X PREV	Suche nach dem ersten Vorkommen der Zeichenkette HILITE, und zwar exakt wie hier eingegeben. Suche nur in ausgeschlossenen Zeilen.
F P'RCV#' .X .Z 20 30	Suche nach der Zeichenkette, die mit RCV beginnt, mit einem numerischen vierten Zeichen zwischen dem Block .X .Z, und zwischen den Spalten 20 bis 30.
F X'6C' SUFFIX NX	Suche nach dem Zeichen, das dem Hexadezimalwert 6C entspricht, und zwar nur in nicht von der Anzeige ausgeschlossenen Zeilen. Das Zeichen muss am Ende eines Wortes stehen.

Falls die zu suchende Zeichenkette einzelne Hochkommazeichen enthält, müssen Sie im FIND-Kommando ein anderes Trennzeichen benutzen, zum Beispiel doppelte Hochkommazeichen:

```
FIND C"'string'"
```

Mit dem Hauptkommando RFIND können Sie ein vorangegangenes FIND-Kommando wiederholen.

Zeichenkette ersetzen

Sie können eine gegebene Zeichenkette suchen und durch eine andere Zeichenkette ersetzen, indem Sie das Hauptkommando CHANGE benutzen.

Falls die zu ersetzende Zeichenkette Apostrophe enthält, müssen Sie im CHANGE-Kommando ein anderes Trennzeichen benutzen, zum Beispiel Anführungszeichen:

```
CHANGE "'string1'" "'string2'"
```

Beim CHANGE-Kommando können Sie die gleichen Operanden wie beim FIND-Kommando benutzen. Beim CHANGE-Kommando bedeuten jedoch die Richtungsoperanden ALL, dass alle Vorkommen der angegebenen Zeichenkette geändert werden sollen.

Nachdem der Ersetzungsvorgang abgeschlossen ist, erscheint im Präfixbereich der geänderten Zeile die Meldung ==chg>

Dieser Abschnitt behandelt folgende Themen:

- Beispiele für das CHANGE-Kommando
- Kommandos RFIND und RCHANGE

Beispiele für das CHANGE-Kommando

Kommando	Erläuterung
`CHG 'LOW' 'HIGH'`	Ersetzt das erste Vorkommen von HIGH (Groß-/Kleinschreibung wird nicht beachtet).
`CHG C'OPS' 'SPF' .X .Y 28 32 all`	Ersetzt OPS (exakt wie hier eingegeben) durch SPF; ersetze alle Vorkommen in dem mit .X und .Y markierten Block und zwischen den Spalten 28 und 32. Siehe auch Beispiel-Bildschirme weiter unten.
Wiederholte CHANGE-Kommandos:	
`CHG * 'NEW'`	Ersetzt das nächste Vorkommen der Zeichenkette, die in dem letzten CHANGE-Kommando angegeben wurde, durch die neue Zeichenkette NEW.

Kommando	Erläuterung
`CHG 'OLD' *`	Ersetzt `OLD` durch dieselbe neue Zeichenkette, die im letzten `CHANGE`-Kommando angegeben wurde.

Der folgende Bildschirm veranschaulicht das zweite Beispiel *bevor* das Kommando mit Enter ausgeführt wurde:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------------ Columns 001 072
 COMMAND===> CHG C'OPS' 'SPF' .X .Y 28 32 ALL                   SCROLL===> CSR
****** ***************************** top of data ****************************
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000010   RESET #JOBNAME(A8)
000020   RESET #FD(N3) #FL(A8) #FF(N3)
000030   RESET #TD(N3) #TL(A8) #TF(N3)
000040   COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050   SET CONTROL 'WL60C6B005/010F'
000060   INPUT  'Enter PARAMETERS FOR LIBRARY COPY:'
000070    /     'FROM:   DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080    /     'TO  :   DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090 // #JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 //       'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
.X     //STEPLIB  DD  DISP=SHR,DSN=OPS.SYSF.ADALOAD
000150 //         DD  DISP=SHR,DSN=OPS.SYSF.LOAD
000160 //         DD  DISP=SHR,DSN=OPS.SYSF.PROD.INST  * OPS INSTALL
.Y     //         DD  DISP=SHR,DSN=OPS.SYSF.SOURCE     * OPS DOCUMENTS
000180 //DDCARD   DD  *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Der folgende Bildschirm veranschaulicht das zweite Beispiel *nachdem* das Kommando mit Enter ausgeführt wurde:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-77K ------ 4 char  'OPS' changed
 COMMAND===>                                                  SCROLL===> CSR
****** ***************************** top of data ****************************
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000010   RESET #JOBNAME(A8)
000020   RESET #FD(N3) #FL(A8) #FF(N3)
000030   RESET #TD(N3) #TL(A8) #TF(N3)
000040   COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050   SET CONTROL 'WL60C6B005/010F'
000060   INPUT  'Enter PARAMETERS FOR LIBRARY COPY:'
000070    /     'FROM:   DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080    /     'TO  :   DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090 // #JOBNAME JOB NAT,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 //       'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
.X     //STEPLIB  DD  DISP=SHR,DSN=SPF.SYSF.ADALOAD
==chg> //         DD  DISP=SHR,DSN=SPF.SYSF.LOAD
==chg> //         DD  DISP=SHR,DSN=SPF.SYSF.PROD.INST  * OPS INSTALL
.Y     //         DD  DISP=SHR,DSN=SPF.SYSF.SOURCE     * OPS DOCUMENTS
000180 //DDCARD   DD  *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Alle Vorkommen der Zeichenkette `OPS` sind zwischen den Zeile 140 und 170 und innerhalb der Spalten 28 durch die Zeichenkette `SPF` bis 32 ersetzt worden.

Kommandos RFIND und RCHANGE

Beim Ändern von Zeichenketten sind die folgenden Kommandos hilfreich: `RFIND` (wiederholtes `FIND`) und `RCHANGE` (wiederholtes `CHANGE`)

Beispiel: Mit der Kommandosequenz

```
FIND 'abc'
  CHANGE 'abc' 'def'
  RFIND
  RCHANGE
```

können Sie Vorkommen einer bestimmten Zeichenkette finden und diese wahlweise mit relativ geringem Aufwand ändern.

49 Editor-Profil einstellen

- Editor-Profil-Einstellungen 483

Jeder Benutzer hat ein Editor-Profil mit Parametern, die entsprechend seinen individuellen Erfordernissen gesetzt werden können. Wenn Sie den Editor zum ersten Mal aufrufen, kommen die vom Administrator festgelegten Standardwerte zum Einsatz.

Mittels dafür vorgesehener Kommandos können Sie nun einzelne Einstellungen in Ihrem Editor-Profil ändern. Die neuen Einstellungen gelten für die restliche Editor-Sitzung bzw. solange, bis Sie sie mit den dafür vorgesehenen Kommandos ändern.

Um die aktuellen Einstellungen Ihres Editor-Profils anzuzeigen:

- Setzen Sie folgendes Hauptkommando ab:

```
PROFILE
```

Weitere Informationen siehe PROFILE in *Hauptkommandos (Übersicht)*.

Im oberen Bereich des Editor-Bildschirms erscheinen die folgenden Zeilen:

```
EDIT-NAT:NATLIB1(JOB1JCL)-Program->Struct-Free-78K ------------ Columns 001 072
 COMMAND===>                                                   SCROLL===> CSR
****** ****************************** top of data ******************************
=prof> date: 16/04/08 15:37:17 user: SAG      init size: 0 size: 0
=prof> var   - 88,..recovery on  (7 0)...autosave off... empty line off
=prof> mask off.caps on .hex off  nulls on  std.autoren off .auto order off
=prof> log on  1.mso on .fix off .escape off   tabs off
=prof> advance on .protect off.limit off
****** ***************************** bottom of data ****************************
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Split End   Suspe Rfind Rchan Up    Down  Swap  Left  Right Curso
```

Die einzelnen Einstellungen des Editor-Profils werden im folgenden Abschnitt beschrieben.

Editor-Profil-Einstellungen

Der folgende Abschnitt beschreibt die Editor-Profil-Einstellungen und die entsprechenden Hauptkommandos, mit denen Sie sie ändern können.

Erläuterung der bei der Darstellung der Hauptkommandos verwendeten Syntaxsymbole siehe *Syntax-Elemente*.

Weitere Informationen zu Hauptkommandos siehe *Hauptkommandos (Übersicht)*.

Profil-Parameter	Beschreibung	Hauptkommando
`advance`	Gibt an, ob der Cursor nach einer Änderung in einer Zeile automatisch zur nächsten Zeile springt.	`ADVANCE ON|OFF|PAGE`
`auto order`	Automatische Ausrichtung von Text innerhalb definierter Grenzen.	`AORDER ON|OFF`
`autoren`	Gibt an, ob der Editor, die `RENUMBER`-Funktion aktiviert oder deaktiviert.	`AUTOREN ON|OFF`
`autosave`	Aktiviert oder deaktiviert das automatische Speichern, wenn das Kommando `END` abgesetzt wird.	`AUTOSAVE ON|OFF`
`caps`	Gibt an, ob die Daten in Großschreibung umgesetzt werden sollen.	`CAPS ON|OFF|PGM`
`date`	Aktuelles Datum und Uhrzeit. Nicht änderbarer Parameter.	Keines
`empty line`	Gibt an, ob Zeilen, die nur Leerzeichen enthalten, automatisch gelöscht werden sollen.	`EMPTY ON|OFF`
`escape`	Gibt an, ob das Fluchtzeichen verwendet werden soll, um es Zeilenkommandos voranzustellen.	`ESCAPE ON|OFF [char]`
`fix`	Gibt an, ob eine feste Anzahl von Spalten angezeigt wird, und wie viele Spalten festgelegt sein sollen.	`FIX ON|OFF n`
`hex`	Gibt an, ob die Daten im Hexadezimalformat angezeigt werden sollen.	`HEX ON|OFF`
`init size`	Anzahl der Zahlen im Objekt, wenn der Editor aufgerufen wurde. Nicht änderbarer Parameter.	Keines
`limit`	Gibt die maximale Anzahl der Zeilen an, die mittels eines `FIND`- oder `RFIND`-Kommandos durchsucht werden sollen.	`LIMIT n`
`log`	Schaltet die Protokolldatei ein oder aus. Bei eingeschalteter Protokolldatei kann das Kommando `UNDO` benutzt werden, um die letzten Änderungen rückgängig zu machen.	`LOG ON|OFF`
`mask`	Aktiviert oder deaktiviert die Maskenzeilenfunktion.	`MASK ON|OFF`

Profil-Parameter	Beschreibung	Hauptkommando
`mso`	Zeigt an, dass Daten zwischen mehreren Editor-Sessions ausgetauscht werden können, z.B. das Kopieren von Daten von einem Objekt in ein anderes. Nicht änderbarer Parameter.	Keines
`nulls`	Gibt an, ob das Ende jeder Quellcode-Zeile mit Null-Zeichen aufgefüllt werden soll.	`NULLS ON|OFF`
`protect`	Gibt die Schutzart des **Präfixbereichs** an.	`PROTECT ON|OFF|INS`
`recovery`	Aktiviert oder deaktiviert die Wiederherstellungsfunktion. Gibt an, wie oft eine Checkpoint-Speicherung erfolgt.	`RECOVERY ON|OFF` *n*
`size`	Aktuelle Zeilenzahl in dem Objekt, wobei Informationszeilen (z.B. Profilzeilen und Meldungszeilen) nicht inbegriffen sind. Nicht änderbarer Parameter.	Keines
`tabs`	Aktiviert oder deaktiviert die Tabulatorfunktion.	`TABS ON|OFF`
`var`	Gibt die aktuelle Zeilenlänge an.	`BNDS` *n m*
`user`	Aktuell angemeldeter Benutzer. Nicht änderbarer Parameter.	Keines

50 Daten speichern und den Software AG Editor verlassen

Sie können Daten speichern und den Software AG Editor verlassen, indem Sie eines der folgenden Hauptkommandos absetzen:

Hauptkommando	Erläuterung
CANCEL	Abbruch der Editor-Session. Bei der Bearbeitung vorgenommene Änderungen werden nicht wirksam.
END	Wenn AUTOSAVE auf ON gesetzt ist, werden die Daten einschließlich jeglicher Änderungen gespeichert, und der Editor wird verlassen. Wenn AUTOSAVE auf OFF gesetzt ist, wirkt das Hauptkommando END wie das CANCEL-Kommando, falls keine Daten geändert wurden. Falls jedoch Änderungen vorgenommen wurden, erscheint eine Meldung, in der Sie aufgefordert werden, ein SAVE- oder ein CANCEL-Kommando abzusetzen.
SAVE	Speichert die Daten (einschließlich jeglicher Änderungen), die sich zurzeit im Quellcode-Arbeitsbereich des Editors befinden. Die Editor-Session wird fortgesetzt.

51 Zeilenkommandos (Übersicht)

Dieser Abschnitt enthält eine Kurzbeschreibung der einzelnen Zeilenkommandos, die beim Software AG Editor zur Verfügung stehen.

Allgemeine Informationen zur Benutzung von Zeilenkommandos siehe *Befehle benutzen*.

Zeilenkommando	Erläuterung
)	Verschiebt diese Zeile um eine Spalte nach Rechts.
) *n*	Verschiebt diese Zeile um *n* Spalten nach Rechts, und zwar ohne Rücksicht auf irgendwelche anderen Daten: Sie können in den verschobenen Zeilen Daten verlieren.
)) *n*	Markiert die erste Zeile eines um *n* Spalten nach Rechts zu verschiebenden Blocks. Ein zweites Kommando)) *n* ist nötig, um die letzte Zeile des Blocks zu markieren. Das Verschieben erfolgt ohne Rücksicht auf irgendwelche anderen Daten in dem Block: Sie können in dem verschobenen Block Daten verlieren.
(	Verschiebt diese Zeile um eine Spalte nach Links.
(*n*	Verschiebt diese Zeile um *n* Spalten nach Links, und zwar ohne Rücksicht auf irgendwelche anderen Daten: Sie können in den verschobenen Zeilen Daten verlieren.
((*n*	Markiert die erste Zeile eines um *n* Spalten nach Links zu verschiebenden Blocks. Ein zweites Kommando ((*n* ist nötig, um die letzte Zeile des Blocks zu markieren.
<	Verschiebt die Daten in dieser Zeile um eine Spalte nach Links.
>	Verschiebt die Daten in dieser Zeile um eine Spalte nach Rechts.
> *n*	Verschiebt die Daten in dieser Zeile um *n* Spalten nach Rechts (bis zum letzten Zeichen, das kein Leerzeichen ist: Es gehen keine Daten verloren).
> > *n*	Markiert die erste Zeile eines um *n* Spalten nach Rechts (oder bis zum letzten Zeichen, das kein Leerzeichen ist) zu verschiebenden Blocks. Ein zweites Kommando > > ist nötig, um die letzte Zeile des Blocks zu markieren.
< *n*	Verschiebt diese Zeile um *n* Spalten (oder bis zum letzten Zeichen, das kein Leerzeichen ist) nach Links.

Zeilenkommando	Erläuterung
`<<`*n*	Markiert die erste Zeile eines um *n* Spalten nach Links (oder bis zum letzten Zeichen, das kein Leerzeichen ist) zu verschiebenden Blocks. Ein zweites Kommando `<<` ist nötig, um die letzte Zeile des Blocks zu markieren.
`A`	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (`C`, `C`*n*, `CC`) oder Verschieben (`M`, `M`*n*, `MM`). Die kopierte(n) oder verschobene(n) Zeile(n) wird bzw. werden *nach* dieser Zeile eingefügt.
`B`	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (`C`, `C`*n*, `CC`) oder Verschieben (`M`, `M`*n*, `MM`). Die kopierte(n) oder verschobene(n) Zeile(n) wird bzw. werden *vor* dieser Zeile eingefügt.
`BNDS`	Zeigt die Begrenzungspositionen in dieser Zeile.
`C`	Kopiert diese Zeile an die Stelle, die mit einem Zeilenkommando `A`, `B` oder `O` markiert ist.
`C`*n*	Kopiert die nächsten *n* Zeilen an die Position, die mit einem Zeilenkommando `A`, `B` oder `O` markiert ist.
`CC`	Markiert die erste Zeile des zu kopierenden Blocks. Ein zweites Kommando `CC` ist nötig, um die letzte Zeile des Blocks zu markieren. Das Kopieren erfolgt an die Stelle, die durch ein Zeilenkommando `A`, `B` oder `O` angegeben worden ist.
`CX`	Kopiert die mit `.X` markierte Zeile. Fügt die Daten nach dieser Zeile ein.
`CY`	Kopiert die mit `.Y` markierte Zeile. Fügt die Daten nach dieser Zeile ein.
`CX-Y`	Kopiert den Zeilenblock ab der Zeile, die mit `.X` markiert ist, bis zur Zeile, die mit `.Y` markiert ist. Fügt die Daten nach dieser Zeile ein.
`COLS`	Zeigt die Spaltenpositionen in dieser Zeile an.
`D`	Löscht diese Zeile.
`D`*n*	Löscht die nächsten *n* Zeilen.
`DD`	Markiert die erste Zeile eines zu löschenden Blocks. Ein zweites Kommando `DD` ist nötig, um die letzte Zeile des zu löschenden Blocks zu markieren. Die Löschung erfolgt, nachdem das zweite `DD`-Kommando eingegeben worden ist.
`DX`	Löscht die mit `.X` markierte Zeile.
`DY`	Löscht die mit `.Y` markierte Zeile.
`DX-Y`	Löscht den Zeilenblock ab der mit `.X` markierten Zeile bis zur mit `.Y` markierten Zeile.
`F`	Schließt die erste ausgeschlossene Zeile ein.
`F`*n*	Schließt die ersten *n* ausgeschlossenen Zeilen ein.
`I`	Fügt eine Zeile ein. Der Editor wird auf den Einfügemodus umgeschaltet. Das heißt, wenn Sie in der neuen Zeile Daten oder ein Leerzeichen eingeben und Enter drücken, wird automatisch eine neue Zeile eingefügt und der Cursor in diese Zeile platziert. Falls Sie in eine eingefügte Zeile keine neuen Daten eingeben und Enter drücken, verlässt der Editor den Einfügemodus, und die Leerzeile wird gelöscht. (Siehe auch Hauptkommando `EMPTY`). Sie können außerdem eine eingefügte Zeile mit einem vordefinierten Inhalt füllen. (Siehe auch Hauptkommando `MASK`).

Zeilenkommando	Erläuterung
I*n*	Fügt *n* Zeilen ein. Sie können in die neuen Zeilen Daten eingeben. Wenn Sie Enter drücken, werden nicht benutzte Zeilen gelöscht, eine Leerzeile bleibt jedoch erhalten und der Cursor wird in diese Zeile platziert (der Editor bleibt im Einfügemodus).
J	Führt die nächste Zeile mit dieser Zeile zusammen. Identisch mit dem Zeilenkommando TJ.
L*n*	Schließt die letzten *n* ausgeschlossenen Zeilen ein.
LC	Ändert diese Zeile auf Kleinschreibung.
LC*n*	Ändert die folgenden *n* Zeilen auf Kleinschreibung.
LCC	Markiert die erste Zeile eines auf Kleinschreibung umzustellenden Blocks. Ein zweites Kommando LCC ist nötig, um die letzte Zeile in dem Block zu markieren.
LJ	Richtet die Daten innerhalb der eingestellten Begrenzungen in dieser Zeile bündig mit der linken Begrenzung aus.
LJJ	Markiert die die erste Zeile eines Datenblocks, der innerhalb der eingestellten Begrenzungen linksbündig ausgerichtet werden soll. Ein zweites Zeilenkommando LJJ ist nötig, um den Block einzugrenzen. Die Ausrichtung erfolgt, nachdem das zweite Kommando LJJ abgesetzt worden ist.
M	Verschiebt diese Zeile an die Position, die mit einem Zeilenkommando A, B oder O markiert worden ist.
M*n*	Verschiebt die nächsten *n* Zeilen an die Position, die mit einem Zeilenkommando A, B oder O markiert ist.
MM	Markiert die erste Zeile des zu verschiebenden Blocks. Ein zweites Zeilenkommando MM ist nötig, um die letzte Zeile des zu verschiebenden Blocks zu markieren. Das Verschieben erfolgt an die Position, die mit einem Zeilenkommando A, B oder O markiert worden ist.
MASK	Fügt im Editor eine Leerzeile ein, in der Sie eine Maske erstellen können. Diese Zeile wird immer dann eingefügt, wenn Sie das Zeilenkommando zum Einfügen (I*n*) benutzen, um eine oder mehrere neue Zeilen zu erstellen. (Siehe auch Hauptkommando MASK).
MX	Verschiebt die mit .X markierte Zeile. Fügt die Daten nach dieser Zeile ein.
MY	Verschiebt die mit .Y markierte Zeile. Fügt die Daten nach dieser Zeile ein.
MX-Y	Verschiebt den Zeilenblock ab der Zeile, die mit .X markiert ist, zur Zeile, die mit .Y markiert ist. Fügt die Daten nach dieser Zeile ein.
N	Die in dieser Zeile vorgenommenen Änderungen werden nicht wirksam, wenn Enter gedrückt wird.
O	Markiert diese Zeile als Zielzeile für ein Zeilenkommando zum Verschieben (M, M*n*, MM) oder zum Kopieren (C, C*n*, CC). Die verschobene(n) oder kopierte(n) Zeile(n) wird (werden) mit dieser Zeile zusammengeführt; d.h., es werden Leerzeichen in der Zeile überlagert.
O*n*	Markiert die folgenden *n* Zeilen als Zielzeilen für ein Zeilenkommando zum Verschieben (M, M*n*, MM) oder zum Kopieren (C, C*n*, CC). Die verschobenen oder kopierten Zeilen werden mit diesen Zeilen zusammengeführt; d.h., es werden Leerzeichen in den Zeilen überlagert.

Zeilenkommando	Erläuterung
OO	Markiert die erste Zeile eines Zielzeilenblocks für ein Verschiebe-Zeilenkommando (M, M*n*, MM) oder Kopier-Zeilenkommando (C, C*n*, CC). Ein zweites Zeilenkommando OO ist nötig, um die letzte Zeile des Zielzeilenblocks zu markieren. Die kopierten oder verschobenen Zeilen werden mit diesen Zeilen zusammengeführt, d.h. Leerzeichen in den Zeilen werden überlagert.
R	Wiederholt diese Zeile einmal.
R*n*	Wiederholt diese Zeile *n* Mal.
RR	Markiert die erste Zeile eines Blocks, der wiederholt werden soll. Ein zweites Kommando RR ist nötig, um den zu wiederholenden Block einzugrenzen. Die Wiederholung erfolgt, nachdem das zweite Kommando RR eingegeben worden ist.
RR*n*	Wiederholt den Zeilenblock *n* Mal.
RJ	Richtet die Daten innerhalb der eingestellten Begrenzungen in dieser Zeile bündig zur rechten Begrenzung aus.
RJJ	Markiert die erste Zeile eines Datenblocks, der innerhalb der Begrenzungen rechtsbündig ausgerichtet werden soll. Ein zweites Zeilenkommando RJJ ist nötig, um die letzte Zeile des Blocks zu markieren. Die Ausrichtung erfolgt, nachdem das zweite Kommando RJJ abgesetzt worden ist.
S	Spaltet diese Zeile ab der Cursor-Position in zwei Zeilen auf. Geben Sie das Zeilenkommando ein. Platzieren Sie den Cursor an die Stelle, an der die Aufspaltung erfolgen soll, und drücken Sie Enter.
T	Blättert die Daten so, dass die markierte Zeile zur obersten Zeile wird.
TABS	Zeigt die Tabulatorpositionen in dieser Zeile an.
TC	Zentriert die Daten in dieser Zeile innerhalb der eingestellten Begrenzungen.
TCC	Markiert die erste Zeile in einem Datenblock, der innerhalb der eingestellten Begrenzungen zentriert werden soll. Ein zweites Zeilenkommando TCC ist nötig, um die letzte Zeile des Blocks zu markieren. Die Zentrierung erfolgt, nachdem das zweite Kommando TCC abgesetzt worden ist.
TE	Schaltet den Editor in den Texteingabemodus (der Editor-Bildschirm ist bis zum unteren Ende leer).
TF	Führt diese Zeile mit den nachfolgenden Zeilen bis zur nächsten Leerzeile zusammen.
TF*n*	Führt diese Zeile mit den nachfolgenden Zeilen bis zur nächsten Leerzeile zusammen, wobei die Daten, die sich rechts von Spalte *n* befinden, ignoriert werden.
TI	Kehrt die Reihenfolge aller Zeichen in der aktuellen Zeile und innerhalb der eingestellten Grenzen um.
TII	Markiert die erste Zeile eines Textblocks, dessen Zeichenreihenfolge innerhalb der eingestellten Grenzen umgekehrt werden soll. Ein zweites Zeilenkommando TII ist nötig, um die letzte Zeile des Blocks zu markieren.
TJ	Führt die nächste Zeile mit dieser Zeile zusammen (identisch mit dem Zeilenkommando J).
TO	Führt diese Zeile mit der nächsten Zeile zusammen.

Zeilenkommando	Erläuterung
TOO	Markiert die erste Zeile eines Datenblocks, der innerhalb der eingestellten Grenzen zusammengeführt werden soll. Ein zweites Zeilenkommando TOO ist nötig, um die letzte Zeile des zusammenzuführenden Blocks zu markieren. Die Funktion wird ausgeführt, nachdem das zweite Kommando TOO abgesetzt worden ist.
TS	Spaltet diese Zeile an der Cursor-Position in zwei Zeilen auf; außerdem wird automatisch eine Leerzeile eingefügt. Diese wird jedoch gelöscht, wenn Sie unbenutzt bleibt (identisch mit den Zeilenkommando S).
UC	Ändert diese Zeile in Großschreibung.
UC*n*	Ändert die die folgenden *n* Zeilen in Großschreibung.
UCC	Markiert die erste Zeile eines Blocks, der in Großschreibung geändert werden soll. Ein zweites Zeilenkommando UCC ist nötig, um die letzte Zeile des Blocks zu markieren.
W	Öffnet ein Fenster mit einer Zeile.
W*n*	Öffnet ein Fenster mit *n* Zeilen.
WC	Kopiert das Datenfenster. Die Cursor-Position markiert die Spalte, bei der diese Zeile aufgespalten werden soll, um die kopierten Daten einzufügen.
WC*n*	Spaltet die Zeile in Spalte *n* auf und kopiert die Daten zwischen die beiden Teile der Zeile.
WE	Markiert das Ende des Datenfensters. Funktioniert so wie das Zeilenkommando WS. Wenn das Fenster in derselben Zeile beginnen und enden, dann nehmen Sie statt des Kommandos WS das Kommando WE. Der Editor bestätigt das eingestellte Fenster mit der Meldung WW im Präfixbereich.
WM	Verschiebt das Datenfenster. Funktion wie bei WC, jedoch werden die ursprünglichen Daten nach dem Kopiervorgang gelöscht.
WM*n*	Spaltet diese Zeile in Spalte *n*, und verschiebt die Daten zwischen die beiden Teilen dieser Zeile.
WS	Markiert den Anfang eines Datenfensters. Die Cursor-Position markiert die Spalte, ab der die Daten gelesen werden. Falls sich der Cursor nicht in der Zeile befindet, bei der das Kommando eingegeben wurde, beginnt das Fenster in Spalte 1.
WS*n*	Das Datenfenster beginnt in Spalte *n* dieser Zeile.
.X	Diese Zeile wird mit .X markiert.
X	Diese Zeile wird ausgeschlossen.
X*n*	Die folgenden *n* Zeilen werden ausgeschlossen.
XX	Markiert die erste Zeile des Blocks, der ausgeschlossen werden soll. Ein zweites Zeilenkommando XX ist nötig, um die zweite Zeile des Blocks zu markieren.
.*label*	Markiert die Zeile mit einem .*label*. Das *label* kann eine beliebige Zeichenkette mit 1 bis 4 alphabetischen Zeichen sein. Siehe auch Hauptkommando LABEL.
.Y	Diese Zeile wird mit .Y markiert.

52 Hauptkommandos (Übersicht)

- Syntax-Elemente 495
- Gemeinsame Kommando-Optionen 496
- ADVANCE 498
- AORDER 499
- AUTOREN 499
- AUTOSAVE 500
- BNDS 500
- BOTTOM 501
- CANCEL 501
- CAPS 501
- CEnter 502
- CHANGE 502
- COLS 505
- CURSOR 506
- CWINDOW 506
- DELETE 506
- DOWN 508
- DWINDOW 508
- EMPTY 509
- END 509
- ESCAPE 509
- EXCLUDE 510
- FIND 512
- FIX 515
- HEX 515
- INCLUDE 515
- JLEFT 516
- JRIGHT 516
- JUSTIFY 517
- LABEL 518
- LC 519
- LEFT 520

- LIMIT 521
- LOCATE 521
- LOG 522
- MASK 523
- MWINDOW 523
- NULLS 523
- ORDER 524
- POWER 525
- PROFILE 525
- PROTECT 525
- RCHANGE 526
- RECOVERY 526
- RENUMBER 527
- RESET 527
- RFIND 528
- RIGHT 528
- SORT 528
- TABS 529
- TOP 530
- UC 530
- UNDO 531
- UNREN 531
- UP 532
- WINDOW 532
- XSWAP 533

Dieser Abschnitt enthält eines Kurzbeschreibung der einzelnen Hauptkommandos, die beim Software AG Editor zur Verfügung stehen, und eine vollständige Übersicht über die Kommandosyntax.

Allgemeine Informationen zur Benutzung von Kommandos siehe *Kommandos benutzen*.

Anmerkung: Falls im Präfixbereich Zeilennummern erscheinen, dann entspricht die Spalte 8 auf dem Bildschirm der Spalte 1 des Editierbereichs. Dies ist bei der Angabe von Spalten in Hauptkommandos zu beachten.

Syntax-Elemente

In den Diagrammen, in denen die Syntax der Editor-Hauptkommandos dargestellt ist, werden folgende Symbole verwendet:

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

Element	Beschreibung
	Ist das Element vor dem Komma und den drei Punkten ein in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
:...	Ein Doppelpunkt und drei Punkte hinter einem Element bedeuten, dass Sie das Element mehrmals angeben dürfen, wobei die einzelnen Elemente durch Doppelpunkte voneinander getrennt werden müssen. Gegebenenfalls gibt eine Zahl hinter dem Doppelpunkt und den drei Punkten an, wie oft das Element angegeben werden darf. Ist das Element vor dem Doppelpunkt und den drei Punkten ein in eckige oder geschweifte Klammern eingeschlossener Ausdruck, gilt die Wiederholmöglichkeit für den gesamten in Klammern stehenden Ausdruck.
Andere Symbole (außer [ ] { } ... ,... :...)	Alle anderen Symbole außer den in dieser Tabelle definierten müssen genau so eingegeben werden wie angegeben.

Gemeinsame Kommando-Optionen

Es gibt einige Optionen, die bei mehreren Hauptkommandos vorkommen. Diese Optionen werden im folgenden Abschnitt beschrieben.

- Funktion zur Wiederanzeige eines Kommandos
- Zeilenangaben
- Spaltenangaben
- Angezeigte oder nicht angezeigte Zeilen
- Richtungsangabe bei Kommandos
- Spezielle Ausprägungen einer Zeichenkette

Funktion zur Wiederanzeige eines Kommandos

Der Editor bietet eine Funktion zur Wiederanzeige eines Kommandos: Wenn Sie vor einem Kommando zwei Und-Zeichen (&&) eingeben, bleibt es in der Kommandozeile erhalten und wird jedes Mal ausgeführt, wenn Sie Enter drücken, und zwar so lange bis Sie das Kommando löschen oder durch ein anderes Kommando ersetzen.

Zeilenangaben

Mit diesen Optionen können Sie die Wirkung eines Kommandos auf einen bestimmten Zeilenbereich einschränken:

`.X`	Das Kommando betrifft nur die Zeilen ab der Zeile, die mit `.X` markiert ist, bis zur letzten Zeile.
`.X .Y`	Das Kommando betrifft nur die Zeilen ab der Zeile, die mit `.X` markiert ist, bis zur Zeile, die mit `.Y` markiert ist.

`X` und `Y` kann auch eine beliebige, aus vier alphabetischen Zeichen bestehende Markierung sein (siehe Kommando `LABEL`).

Spaltenangaben

Mit diesen Optionen können Sie die Wirkung eines Kommandos auf einen bestimmten Spaltenbereich einschränken. Die Spaltennummern beziehen sich auf die tatsächlichen Datenspalten, die Zeilennummern vor den Daten werden nicht mitgezählt. Wenn Sie also Spalte 1 bei einem Kommando angeben, kann das physisch die achte Spalte auf Ihrem Bildschirm, faktisch jedoch die erste Spalte der Daten sein, die Sie gerade editieren.

n	Das Kommando betrifft nur Zeilen, in denen die angegebene Zeichenkette in Spalte *n* beginnt (d.h., die Zeichenkette muss in Spalte *n* sein).
n m	Das Kommando betrifft nur Zeilen, in denen die angegebene Zeichenkette an beliebiger Stelle zwischen den Spalten *n* und *m* auftritt.

Angezeigte oder nicht angezeigte Zeilen

Mit einer der folgenden Optionen können Sie angeben, dass nur ausgeschlossene oder nur eingeschlossene Zeilen von einem Kommando betroffen sein sollen:

`NX`	Das Kommando betrifft nur nicht ausgeschlossene Zeilen; d.h., Zeilen, die zurzeit angezeigt werden.
`X`	Das Kommando betrifft nur ausgeschlossene Zeilen; d.h., Zeilen, die, so wie mit dem `EXCLUDE`-Kommando angegeben, zurzeit *nicht* angezeigt werden. Eine ausgeschlossene Zeile bleibt von der Anzeige ausgeschlossen, wenn bei ihr eine Hauptkommandofunktion ausgeführt wird.

Richtungsangabe bei Kommandos

Mit diesen Optionen können Sie die Richtung angeben, in der sich ein Kommando auswirken soll:

NEXT	Das Kommando betrifft nur die nächste Zeile (ab der Cursor-Position), in der die angegebene Zeichenkette auftritt.
PREV	Das Kommando betrifft die Zeile, in der die angegebene Zeichenkette zuvor aufgetreten ist.
FIRST	Das Kommando betrifft die erste Zeile, in der die angegebene Zeichenkette auftritt.
LAST	Das Kommando betrifft die letzte Zeile, in der die angegebene Zeichenkette auftritt.
ALL	Das Kommando betrifft alle Zeilen, in denen die angegebene Zeichenkette auftritt.

Spezielle Ausprägungen einer Zeichenkette

Mit diesen Optionen können Sie angeben, ob nur spezielle Ausprägungen der angegebenen Zeichenkette von einem Kommando betroffen sein sollen:

CHARS	Das Kommando betrifft jede beliebige Zeile, in der die angegebene Zeichenkette auftritt.
WORD	Das Kommando betrifft nur diejenigen Zeilen, in denen die angegebene Zeichenkette ein Wort bildet.
PREFIX	Das Kommando betrifft nur diejenigen Zeilen, in denen die angegebene Zeichenkette der Anfang eines Wortes ist.
SUFFIX	Das Kommando betrifft nur diejenigen Zeilen, in denen die angegebene Zeichenkette das Ende eines Wortes ist.

ADVANCE

Mit diesem Kommando können Sie angeben, ob der Cursor nach der Aktualisierung einer Zeile automatisch in die nächste Zeile springt.

ON	Der Cursor springt nach Aktualisierung einer Zeile automatisch in die nächste Zeile.
OFF	Der Cursor springt nach Aktualisierung einer Zeile nicht in die nächste Zeile.
PAGE	Die Zeile, in der der Cursor steht, wird nach erfolgter Aktualisierung an den Anfang des Editierbereichs gestellt.

Wird ein nichtqualifiziertes ADVANCE-Kommando abgesetzt, dann wird es als ADVANCE ON interpretiert. Die Standardeinstellung ist ADVANCE ON. Dies kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

AORDER

```
AORDER [ ON  ]
       [ OFF ]
```

Mit diesem Kommando können Sie angeben, ob Text innerhalb der angegebenen Begrenzungen automatisch ausgerichtet werden soll.

Wird ein nichtqualifiziertes AORDER-Kommando abgesetzt, dann wird es als AORDER ON interpretiert. Die Grundeinstellung kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

AUTOREN

```
AUTOREN [ ON  ]
        [ OFF ]
```

Nur bei PDS Members und sequenziellen Datasets.

Mit diesem Kommando können Sie angeben, ob der Editor die RENUMBER-Funktion automatisch aktiviert.

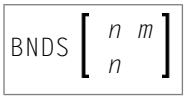
Wird ein nichtqualifiziertes AUTOREN-Kommando abgesetzt, dann wird es als AUTOREN ON interpretiert. Die Grundeinstellung kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

AUTOSAVE

Mit diesem Kommando können Sie angeben, ob der Editor automatisch ein `SAVE`-Kommando ausführt, wenn Sie das `END`-Kommando absetzen.

Wird ein nichtqualifiziertes `AUTOSAVE`-Kommando abgesetzt, dann wird es als `AUTOSAVE ON` interpretiert. Die Standardeinstellung ist `AUTOSAVE ON`. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

BNDS



Mit diesem Kommando können Sie die Auswirkung bestimmter Kommandos auf einen bestimmten Spaltenbereich einschränken.

Diese Begrenzungen gelten für die Kommandos `FIND`, `CHANGE`, `CEnter`, `ORDER`, `JLEFT` und `JRIGHT` sowie für die entsprechenden **Zeilenkommandos** (zum Beispiel `TC`, `TO`, `LJ` oder `RJ`).

n	Die Nummer der Spalte, bei der die linke Begrenzung gesetzt werden soll.
m	Die Nummer der Spalte, bei der die rechte Begrenzung gesetzt werden soll.

Wenn *n* und *m* weggelassen werden, dann werden die Begrenzungen bei der ersten und der letzten Spalte des Editierbereichs gesetzt.

Um die aktuellen Begrenzungseinstellungen anzuzeigen, können Sie das Zeilenkommando `BNDS` absetzen.

BOTTOM

```
BOTTOM
```

Mit diesem Kommando können Sie an das Ende des Objekts blättern, das Sie gerade bearbeiten.

CANCEL

```
CANCEL
```

Macht alle Datenänderungen, die Sie während der aktuellen Editor-Sitzung vorgenommen haben, rückgängig und schließt den Editor. Alle Änderungen, die Sie seit der letzten Speicherung der Daten gemacht haben, gehen verloren.

CAPS

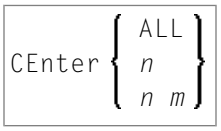
```
CAPS [ ON  ]
     [ OFF ]
     [ PGM ]
```

Mit diesem Kommando können Sie die Umsetzung in Großbuchstaben ein- und ausschalten.

ON	Die Daten werden in Großbuchstaben umgesetzt.
OFF	Die Daten werden nicht in Großbuchstaben umgesetzt; d.h., sie bleiben so, wie sie eingegeben werden.
PGM	Die Daten werden in Großbuchstaben umgesetzt. Ausnahme: Kommentare - diese bleiben so, wie sie eingegeben werden.

Wenn das CAPS-Kommando ohne Parameter abgesetzt wird, hat es die gleiche Wirkung wie CAPS ON. Die Standardeinstellung ist CAPS ON. Sie kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

CEnter



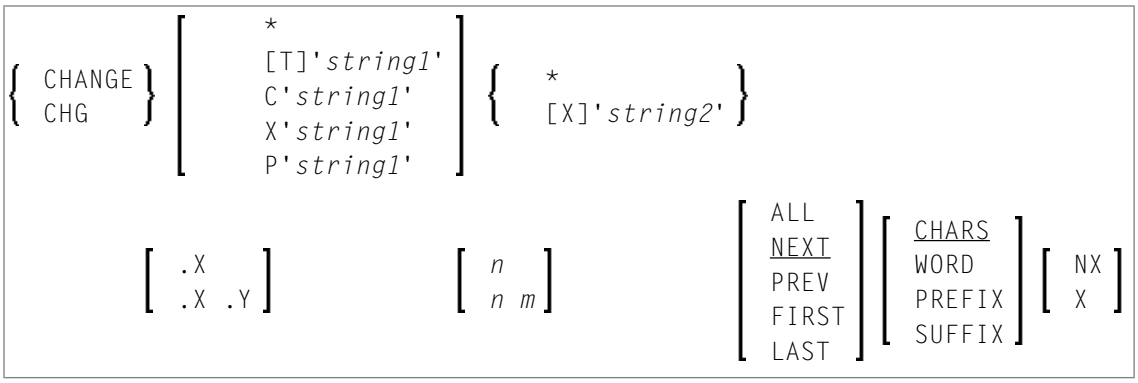
Mit diesem Kommando können Sie Daten zentrieren.

ALL	Zentriert die Daten in allen Zeilen.
n	Zentriert die Daten ab Zeile *n* bis zur letzten Zeile.
n m	Zentriert die Daten ab Zeile *n* bis zur Zeile *m*.

Das Kommando CEnter gilt nur innerhalb der Begrenzungen, die mit dem Hauptkommando BNDS gesetzt werden.

Zum Zentrieren können Sie auch die Zeilenkommandos TC und TCC benutzen.

CHANGE



Mit diesem Kommando können Sie eine Zeichenkette (*string1*) durch eine andere Zeichenkette (*string2*) ersetzen.

Wenn in *string1* oder *string2* ein Apostroph enthalten sein soll, müssen Sie es als zwei Apostrophe schreiben.

Die zu ersetzende Zeichenkette (*string1*) können Sie folgendermaßen angeben:

T'*string1*'	Ersetzt die Zeichenkette *string1* unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
'*string1*'	Wie bei T'*string1*'.
C'*string1*'	Ersetzt die Zeichenkette *string1* nur, wenn sie genauso wie angegeben auftritt.
X'*string1*'	Ersetzt die Zeichenkette, die der angegebenen Zeichenkette *string1* im Hexadezimalformat entspricht. Ersetzt sie durch die Zeichenkette *string2* im Hexadezimalformat.
P'*string1*'	Ersetzt die Zeichenkette *string1*, die folgende Platzhalterzeichen (Wildcard) enthält: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ Zeichen, das kein Leerzeichen ist - Zeichen, das kein numerisches Zeichen ist < Zeichen in Kleinschreibung > Zeichen in Großschreibung
*	Verwende die Zeichenkette, die in einem vorigen Kommando angegeben wurde (zum Beispiel FIND, CHANGE, EXCLUDE).
.X	Erklärung siehe ***Zeilenangabe***.
.X .Y	
n *n m*	Erklärung siehe ***Spaltenangabe***.
ALL NEXT PREV FIRST LAST	Erklärung siehe ***Richtungsangabe bei Kommandos***.
CHARS WORD PREFIX SUFFIX	Erklärung siehe ***Spezielle Ausprägungen einer Zeichenkette***.
NX	Erklärung siehe ***Angezeigte oder nicht angezeigte Zeilen***.

X	

Dieser Abschnitt behandelt folgende Themen:

- Kommando CHANGE mit anderen Kommandos benutzen
- Beispiele für das CHANGE-Kommando

Kommando CHANGE mit anderen Kommandos benutzen

Um die Ausführung eines CHANGE-Kommandos zu wiederholen, können Sie das Kommando RCHANGE benutzen.

Um die gesamten Daten nach einer Zeichenkette zu durchsuchen und dann bei jedem einzelnen Vorkommen zu entscheiden, ob sie durch eine andere Zeichenkette ersetzt werden soll, können Sie eine Kombination der Kommandos FIND, CHANGE, RFIND und RCHANGE benutzen:

Zuerst suchen Sie nach der Zeichenkette:

```
FIND 'string'
```

Wenn die Zeichenkette gefunden worden ist, können Sie entscheiden:

- ob Sie sie ersetzen:

  ```
  CHANGE 'string' 'new-string'
  ```

- oder ob Sie nach dem nächsten Vorkommen der Zeichenkette suchen, indem Sie das FIND-Kommando wiederholen:

  ```
  RFIND
  ```

Wenn das nächste Vorkommen der Zeichenkette gefunden worden ist, können Sie wieder entscheiden:

- ob Sie sie ersetzen, indem Sie das CHANGE-Kommando wiederholen:

  ```
  RCHANGE
  ```

- oder ob Sie nach dem nächsten Vorkommen der Zeichenkette suchen, indem Sie das FIND-Kommando wiederholen:

  ```
  RFIND
  ```

Beispiele für das CHANGE-Kommando

Beispiel 1:

```
CHG 'LOW' 'HIGH'
```

Dieses Kommando ersetzt das erste Vorkommen der Zeichenkette `LOW` durch `HIGH`, unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt.

Beispiel 2:

```
CHG C'OPS' 'SPF' .X .Y 28 32 ALL
```

Dieses Kommando ändert `OPS` (exakt wie hier eingegeben) in `SPF`. Es ändert alle Vorkommen in dem mit `.X` und `.Y` markierten Zeilenblock zwischen den Spalten 28 und 32.

Beispiel 3:

```
CHG C'NAME' 'APPL' .X .Y ALL PREFIX NX
```

Dieses Kommando ändert alle Vorkommen der Vorsilbe `NAME` (exakt wie hier eingegeben) in `APPL` in allen angezeigten Zeilen in dem mit `.X` und `.Y` markierten Block.

Beispiel 4:

```
CHG * 'NEW'
```

Dieses Kommando ersetzt das nächste Vorkommen der mit dem `CHANGE`-Kommando angegebenen Zeichenkette durch die Zeichenkette `NEW`.

Beispiel 5:

```
CHG 'OLD' *
```

Dieses Kommando ersetzt das nächste Vorkommen der Zeichenkette `OLD` durch dieselbe neue Zeichenkette, die beim letzten `CHANGE`-Kommando angegeben wurde.

COLS

```
COLS [ ON  ]
     [ OFF ]
```

Mit diesem Kommando können Sie am oberen Rand des Editierbereichs eine Zeile anzeigen, die die Spaltenpositionen zeigt.

Um die Spaltenpositionen anzuzeigen, können Sie auch das Zeilenkommando `COLS` benutzen.

CURSOR

```
CURSOR
```

Dieses Kommando bewirkt, dass der Cursor wieder in die Kommandozeile gestellt wird, wenn Sie das nächste Mal Enter drücken.

CWINDOW

```
CWINDOW [ n     ]
        [ n m   ]
```

Mit diesem Kommando können Sie ein Datenfenster gemäß den angegebenen Kommandoparametern kopieren.

n	Die Nummer der Zeile, in die das Datenfenster eingefügt werden soll.
m	Die Nummer der Spalte, in die das Datenfenster eingefügt werden soll.

DELETE

```
DELETE [ *           ] [ .X    ] [ n   ] [ ALL   ] [ CHARS  ] [ NX ]
       [ [T]'string' ] [ .X .Y ] [ n m ] [ NEXT  ] [ WORD   ] [ X  ]
       [ 'string'    ]                   [ PREV  ] [ PREFIX ]
       [ C'string'   ]                   [ FIRST ] [ SUFFIX ]
       [ X'string'   ]                   [ LAST  ]
       [ P'string'   ]
```

Mit diesem Kommando können Sie Zeilen löschen.

Sie können angeben, dass nur Zeilen, die eine angegebene Zeichenkette (*string*) enthalten, gelöscht werden sollen:

T'*string*'	Löscht Zeilen, die die Zeichenkette *string* enthalten, unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
'*string*'	Wie bei T'*string*'.
C'*string*'	Löscht Zeilen, die die Zeichenkette *string* exakt wie angegeben enthalten.
X'*string*'	Löscht Zeilen, die die Zeichenkette enthalten, die der angegebenen Zeichenkette (*string*) im Hexadezimalformat entspricht.
P'*string*'	Löscht Zeilen, die die Zeichenkette *string* enthalten, in der folgende Platzhalterzeichen (Wildcard) vorkommen: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ Zeichen, das kein Leerzeichen ist - Zeichen, das kein numerisches Zeichen ist < Zeichen in Kleinschreibung > Zeichen in Großschreibung
*	Verwendet die Suchzeichenkette, die in einem vorigen Kommando angegeben wurde (zum Beispiel FIND, CHANGE, EXCLUDE).
.X	Erklärung siehe ***Zeilenangabe***.
.X .Y	
n *n m*	Erklärung siehe ***Spaltenangabe***.
ALL NEXT PREV FIRST LAST	Erklärung siehe ***Richtungsangabe bei Kommandos***.
CHARS WORD PREFIX SUFFIX	Erklärung siehe ***Spezielle Ausprägungen einer Zeichenkette***.
NX	Erklärung siehe ***Angezeigte oder nicht angezeigte Zeilen***.

X	

Wenn Sie das DELETE-Kommando ohne Parameter eingeben, wird die aktuelle Zeile gelöscht.

Um Zeilen zu löschen, können Sie auch die Zeilenkommandos D, D*n* und DD benutzen.

Beispiel 1:

```
DEL C'NAME' 1 20 ALL PREFIX NX
```

Dieses Kommando löscht alle Zeilen, die die Zeichenkette NAME (in Großbuchstaben, exakt wie hier eingegeben) als Vorsilbe für ein Wort in allen nicht von der Anzeige ausgeschlossenen Zeilen enthalten, wenn NAME zwischen den Spalten 1 und 20 auftritt.

Beispiel 2:

```
DEL C'Abc' .X .Y 10 30 ALL
```

Dieses Kommando löscht alle Zeilen, die die Zeichenkette Abc (exakt wie hier eingegeben) zwischen den Spalten 10 und 30 innerhalb dem mit .X und .Y markierten Zeilenblock enthalten.

DOWN

```
DOWN [n]
```

Mit diesem Kommando können Sie in den Daten abwärts blättern.

Der Parameter *n* gibt an, um wie viele Zeilen nach unten geblättert werden. Wenn *n* weggelassen wird, dann wird der Blätterbetrag durch den Blättermodus bestimmt.

DWINDOW

```
DWINDOW
```

Dieses Kommando löscht das zuletzt definierte Datenfenster.

EMPTY

Dieses Kommando steuert die Löschung von Leerzeilen im Editor.

OFF	Leerzeilen werden nicht gelöscht.
ON	Leerzeilen werden gelöscht.

Wenn Sie EMPTY ohne Parameter eingeben, wird es als EMPTY ON interpretiert. Die Standardeinstellung ist EMPTY OFF (keine Leerzeilenunterdrückung). Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

END

Wenn Sie dieses Kommando absetzen, werden die Daten einschließlich aller Änderungen gespeichert und der Software AG Editor wird verlassen.

Kommandoformat:

Wenn AUTOSAVE auf OFF gesetzt ist und wenn Sie Daten geändert haben, werden Sie aufgefordert, ein SAVE- oder CANCEL-Kommando abzusetzen.

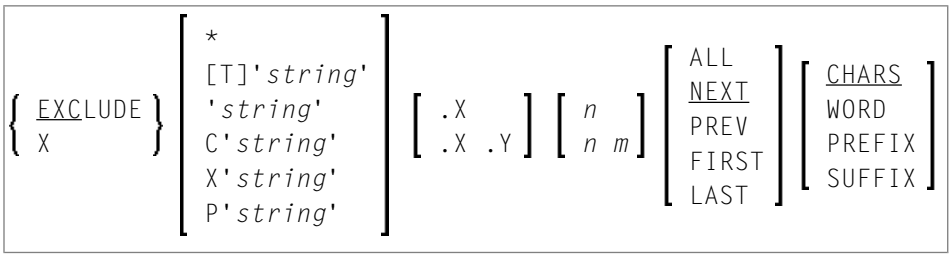
ESCAPE

Mit diesem Kommando können Sie das Fluchtzeichen aktivieren oder deaktivieren, das den Zeilenkommandos vorangestellt werden soll, die in der ersten Spalte der Daten eingegeben werden.

Der Parameter *character* ist das zu verwendende Sonderzeichen. Das Standard-Fluchtzeichen ist der Punkt (.).

Wenn Sie das ESCAPE-Kommando ohne Parameter eingeben, wird es als ESCAPE ON interpretiert. Die Standardeinstellung ist ESCAPE OFF. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

EXCLUDE



Mit diesem Kommando können Sie Zeilen von der Anzeige ausschließen.

Sie können angeben, dass nur Zeilen, die eine angegebene Zeichenkette (*string*) enthalten, von der Anzeige ausgeschlossen werden sollen:

T'*string*'	Schließt alle Zeilen aus, die die Zeichenkette (*string*) enthalten, unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
'*string*'	Wie bei T'*string*'.
C'*string*'	Schließt alle Zeilen aus, die die Zeichenkette (*string*) exakt wie angegeben enthalten
X'*string*'	Schließt Zeilen aus, die die Zeichenkette enthalten, die der angegebenen Zeichenkette (*string*) im Hexadezimalformat entspricht.
P'*string*'	Schließt Zeilen aus, die die Zeichenkette *string* enthalten, in der folgende Platzhalterzeichen (Wildcard) vorkommen: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ Zeichen, das kein Leerzeichen ist - Zeichen, das kein numerisches Zeichen ist < Zeichen in Kleinschreibung > Zeichen in Großschreibung
*	Verwendet die Suchzeichenkette, die in einem vorigen Kommando angegeben wurde (zum Beispiel FIND, CHANGE, EXCLUDE).
.X	Erklärung siehe ***Zeilenangabe***.
.X .Y	

n *n m*	Erklärung siehe ***Spaltenangabe***.
ALL NEXT PREV FIRST LAST	Erklärung siehe ***Richtungsangabe bei Kommandos***.
CHARS WORD PREFIX SUFFIX	Erklärung siehe ***Spezielle Ausprägungen einer Zeichenkette***.

Wenn Sie das EXCLUDE-Kommando ohne Parameter eingeben, wird die aktuelle Zeile von der Anzeige ausgeschlossen.

Um ausgeschlossene Zeilen wieder zur Anzeige zu bringen, benutzen Sie das Hauptkommando INCLUDE.

Beispiel 1:

```
EX 10
```

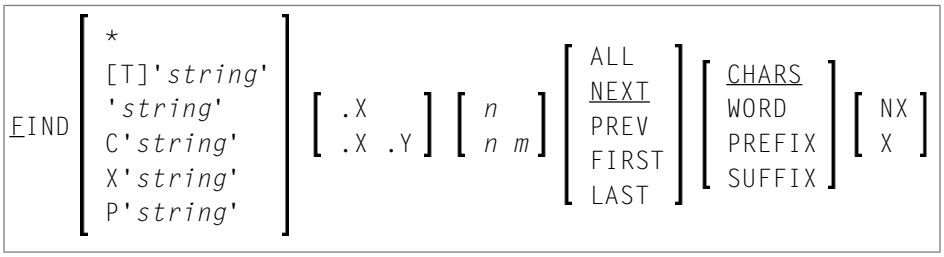
Dieses Kommando schließt Zeile 10 von der Anzeige aus.

Beispiel 2:

```
EX C'NAME' .X ALL PREFIX
```

Dieses Kommando schließt ab der mit .X markierten Zeile alle Zeilen von der Anzeige aus, die NAME (in Großschreibung, wie hier eingegeben) als Vorsilbe für ein Wort enthalten.

FIND



Mit diesem Kommando können Sie nach einer bestimmtem Zeichenkette (*string*) suchen. Der Cursor wird auf den Anfang der zuerst gefundenen Zeichenkette platziert. Falls die Zeile, die die Zeichenkette enthält, von der Anzeige ausgeschlossen worden war, wird sie nun angezeigt.

Wenn die Zeichenkette ein Apostroph enthalten soll, müssen Sie zwei Apostrophe schreiben.

Sie können die Zeichenkette folgendermaßen angeben:

T'*string*'	Sucht nach der Zeichenkette unabhängig davon, ob diese in Klein- oder Großschreibung vorliegt. Dies ist die Standardeinstellung.
'*string*'	Wie bei T'*string*'.
C'*string*'	Sucht nach der Zeichenkette exakt wie in *string* angegeben.
X'*string*'	Sucht nach der Zeichenkette, die der angegebenen Zeichenkette im Hexadezimalformat entspricht.
P'*string*'	Sucht nach einer Zeichenkette, in der folgende Platzhalterzeichen (Wildcards) vorkommen: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ Zeichen, das kein Leerzeichen ist - Zeichen, das kein numerisches Zeichen ist < Zeichen in Kleinschreibung > Zeichen in Großschreibung
*	Sucht nach der Zeichenkette, die im vorherigen FIND-Kommando angegeben wurde.
.X	Erklärung siehe ***Zeilenangabe***.
.X .Y	
n *n m*	Erklärung siehe ***Spaltenangabe***.

ALL NEXT PREV FIRST LAST	Erklärung siehe *Richtungsangabe bei Kommandos*.
CHARS WORD PREFIX SUFFIX	Erklärung siehe *Spezielle Ausprägungen einer Zeichenkette*.
NX X	Erklärung siehe *Angezeigte oder nicht angezeigte Zeilen*.

Das FIND-Kommando unterscheidet sich folgendermaßen vom LOCATE-Kommando:

- Das FIND-Kommando erweist sich bei Textsuchen als wirkungsvoller, während das LOCATE-Kommando in erster Linie dafür gedacht ist, Zeilennummern oder Zeilenmarkierung zu finden.
- Das LOCATE-Kommando findet Text nur in Großschreibung ab Zeile 1 des Editors. Außerdem müssen die Daten im Editor in alphabetischer Reihenfolge vorliegen, damit eine Zeichenkette gefunden wird.
- Wenn eine Zeile mit dem LOCATE-Kommando aufgefunden wird, dann wird der Cursor in den Präfixbereich dieser Zeile gestellt und die Zeile wird an den oberen Rand der des Editors platziert.

  Beim FIND-Kommando wird der Cursor auf die gesuchte Zeichenkette gesetzt und die betreffende Zeile wird nicht zwangsläufig an den oberen Rand der des Editierbereichs platziert.

Um die Ausführung eines FIND-Kommandos zu wiederholen, können Sie das Kommando RFIND benutzen.

Beispiel 1:

```
F C'NAME' .X .Y ALL PREFIX X
```

Dieses Kommando sucht nach allen Ausprägungen von NAME als Vorsilbe eines Wortes (exakt wie hier eingegeben) in einer beliebigen ausgeschlossenen Zeile innerhalb eines durch .X und .Y eingegrenzten Blocks.

Beispiel 2:

```
F C'HILITE' X PREV
```

Dieses Kommando sucht nach dem vorherigen Vorkommen von `HILITE` (exakt wie hier eingegeben) in einer beliebigen ausgeschlossenen Zeile.

Beispiel 3:

```
F P'RCV#' .X .Z 20 30
```

Dieses Kommando sucht nach einer aus vier Zeichen bestehenden Zeichenkette, die mit `RCV` beginnt und deren viertes Zeichen ein numerisches Zeichen ist. Die Suche erfolgt in einem durch `.X` und `.Z` eingegrenzten Block und zwischen den Spalten 20 bis 30.

Beispiel 4:

```
F X'6C' SUFFIX NX
```

Dieses Kommando sucht nach einem Zeichen in Hexadezimaldarstellung `6C`. Es werden nur solche Ausprägungen des Zeichens gefunden, die am Wortende stehen. Die Suche gilt nur bei nicht von der Anzeige ausgeschlossenen Zeilen.

Beispiel 5:

```
F '''w'
```

Dieses Kommando sucht nach folgender Zeichenkette: `'w`

Beispiel 6:

```
F 'r''w'
```

Dieses Kommando sucht nach folgender Zeichenkette: `r'w`.

Beispiel 7:

```
F ''''
```

Dieses Kommando sucht nach einem Apostroph.

FIX

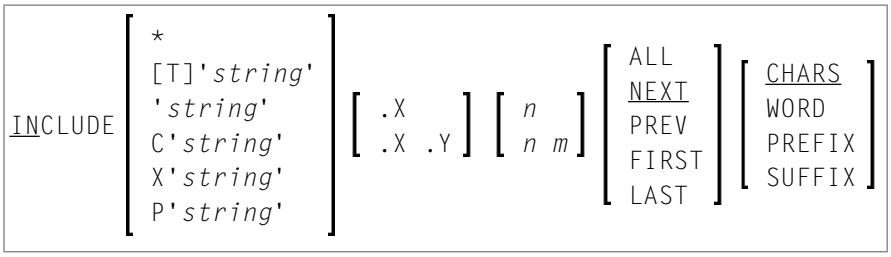
Mit diesem Kommando können Sie die Anzahl der Spalten *n* (ab Spalte) angeben, die in der Anzeige sichtbar bleiben sollen, wenn nach Rechts geblättert wird. Die Standardeinstellung ist `FIX OFF 000`. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

HEX

Mit diesem Kommando können Sie den Hexadezimal-Anzeigemodus ein- und ausschalten.

Die Standardeinstellung ist `HEX ON`. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

INCLUDE



Mit diesem Kommando können Sie Zeilen, die mit dem `EXCLUDE`-Kommando ausgeschlossen worden waren, wieder zur Anzeige bringen. Das Kommando kann mit den gleichen Parametern wie beim `EXCLUDE`-Kommando abgesetzt werden.

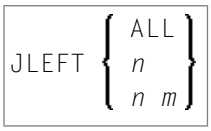
Wenn Sie das `INCLUDE`-Kommando ohne Parameter absetzen, wird die erste Zeile eines ausgeschlossenen Zeilenblocks wieder eingeschlossen.

Beispiel:

```
IN C'NAME' .X ALL PREFIX
```

Dieses Kommando bewirkt, dass alle ausgeschlossenen Zeilen, in denen NAME (exakt wie hier eingegeben) als Vorsilbe eines Wortes ab der mit .X markierten Zeile vorkommt.

JLEFT



Mit diesem Kommando können Sie Daten linksbündig ausrichten.

ALL	Die Daten werden in allen Zeilen ausgerichtet.
n	Die Daten werden ab Zeile *n* bis zur letzten Zeile ausgerichtet.
n m	Die Daten werden ab Zeile *n* bis zur Zeile *m* ausgerichtet.

Das Kommando JLEFT gilt nur innerhalb der horizontalen Begrenzungen, die mit dem Hauptkommando BNDS eingestellt werden.

Für die linksbündige Ausrichtung können Sie auch die Zeilenkommandos LJ und LJJ benutzen.

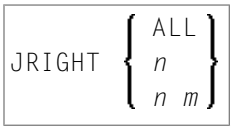
Siehe auch Hauptkommando JRIGHT.

Beispiel:

```
BNDS 10;JLEFT 15 20
```

Die Daten zwischen Spalten 10 und äußersten rechten Spalte des Bildschirms in den Zeilen 15 bis 20 werden linksbündig zur Spalte 10 ausgerichtet.

JRIGHT



Mit diesem Kommando können Sie Daten rechtsbündig ausrichten.

ALL	Die Daten werden in allen Zeilen ausgerichtet.
n	Die Daten werden ab Zeile *n* bis zur letzten Zeile ausgerichtet.
n m	Die Daten werden ab Zeile *n* bis zur Zeile *m* ausgerichtet.

Das Kommando JRIGHT gilt nur innerhalb der horizontalen Begrenzungen, die mit dem Hauptkommando BNDS eingestellt werden.

Für die rechtsbündige Ausrichtung können Sie auch die Zeilenkommandos RJ und RJJ benutzen.

Siehe auch Hauptkommando JLEFT.

Beispiel 1:

```
BNDS 4 40;JRIGHT 6 18
```

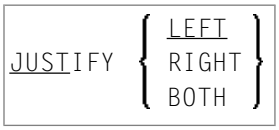
Die Daten zwischen den Spalten 4 bis 40 in den Zeilen 6 bis 18 werden rechtsbündig ausgerichtet.

Beispiel 2:

```
BNDS 10;JRIGHT 15
```

Die Daten rechts von Spalte 10 in Zeile 15 werden bündig zur äußersten rechten Spalte des Editierbildschirms neu ausgerichtet.

JUSTIFY



Mit diesem Kommando können Sie den Ausrichtungsmodus für die Zeilenkommandos TO und TOO einstellen.

Die Zeilenkommandos TO und TOO werden verwendet, um Datenzeilen mit nachfolgenden Zeilen zusammenzuführen. Beide Zeilenkommandos gelten nur innerhalb der horizontalen Begrenzungen, die mit dem Hauptkommando BNDS eingestellt werden.

LEFT	Die Daten werden bündig zur linken Begrenzung ausgerichtet.
RIGHT	Die Daten werden bündig zur rechten Begrenzung ausgerichtet.
BOTH	Die Daten werden bündig zu beiden Begrenzung ausgerichtet.

Beispiel:

Mit den folgenden Kommandos setzen Sie die horizontalen Begrenzungen auf die Spalten 10 und 60 und aktivieren die linksbündige Ausrichtung:

```
BNDS 10 60;JUSTIFY LEFT
```

Wenn Sie danach eine Zeile mit dem Zeilenkommando `TO` (oder einen Zeilenblock mit zwei Zeilenkommandos `TOO`) markieren, werden die Daten zwischen den Spalten 10 und 60 in der (den) markierten Zeile(n) linksbündig zur Zeile 10 ausgerichtet.

LABEL

```
LABEL .label
```

Mit diesem Kommando können Sie die aktuelle Zeile (d.h., die Zeile, die sich zurzeit am oberen Rand des Editierbereichs befindet) mit dem angegebenen *.label* markieren.

label kann eine Zeichenkette mit 1 bis 4 alphabetischen Zeichen sein.

Beispiel:

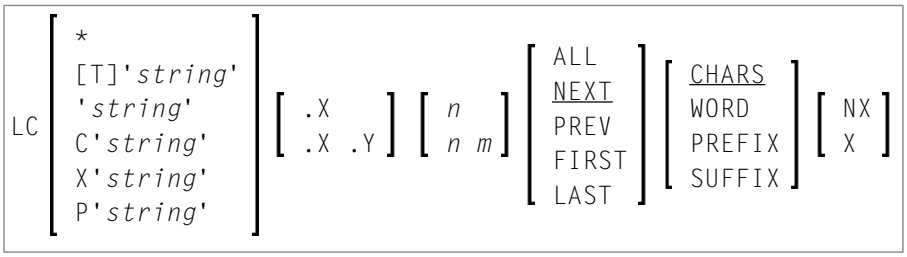
Um die aktuelle Zeile mit `.X` zu markieren, geben Sie folgendes Kommando ein:

```
LABEL .X
```

Außerdem können Sie einen Zeilenblock mit zwei Markierungen versehen. Um zum Beispiel einen Block mit den Markierungen `.X` und `.Y` zu versehen, markieren Sie zuerst die aktuelle Zeile (wobei angenommen wird, dass es sich um die erste Zeile des zu markierenden Blocks handelt) mit `.X` (wie im obigen Beispiel), dann blättern Sie so weit, bis die letzte Zeile des Blocks die aktuelle Zeile ist. Danach setzen Sie das Hauptkommando `LABEL .Y` ab, um diese Zeile mit `.Y` zu markieren.

Um eine Zeile mit einer Markierung zu versehen, können Sie auch das Zeilenkommando *.label* benutzen. Weitere Informationen siehe ***Zeilenkommandos (Übersicht)***.

LC



Mit diesem Kommando können Sie eine oder mehrere Zeilen in Kleinschreibung ändern.

Sie können angeben, dass nur Zeilen, die eine angegebene Zeichenkette (*string*) enthalten, auf Kleinschreibung umgeschaltet werden sollen. Wenn in der Zeichenkette ein Apostroph enthalten sein soll, müssen Sie es als zwei Apostrophe schreiben.

Die Zeichenkette (*string*) können Sie folgendermaßen angeben:

T'*string*'	Ändert die Zeichenkette *string* unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
'*string*'	Wie bei T'*string*'.
C'*string*'	Ändert Zeilen, die die Zeichenkette *string* exakt wie angegeben enthalten.
X'*string*'	Ändert Zeilen, die die Zeichenkette enthalten, die der angegebenen Zeichenkette *string* im Hexadezimalformat entspricht.
P'*string*'	Ändert die Zeilen, die eine Zeichenkette (*string*) enthalten, in der folgende Platzhalterzeichen (Wildcards) vorkommen: = beliebiges Zeichen § alphabetisches Zeichen # numerisches Zeichen $ Sonderzeichen ^ Zeichen, das kein Leerzeichen ist - Zeichen, das kein numerisches Zeichen ist < Zeichen in Kleinschreibung > Zeichen in Großschreibung
*	Ändert Zeilen, die die Zeichenkette *string* enthalten, die in dem vorigen Kommando verwendet wurde, in dem eine Zeichenkette angegeben wurde.
.X	Erklärung siehe ***Zeilenangabe***.
.X .Y	
n	Erklärung siehe ***Spaltenangabe***.

n m	
ALL NEXT PREV FIRST LAST	Erklärung siehe ***Richtungsangabe bei Kommandos***.
CHARS WORD PREFIX SUFFIX	Erklärung siehe ***Spezielle Ausprägungen einer Zeichenkette***.
NX X	Erklärung siehe ***Angezeigte oder nicht angezeigte Zeilen***.

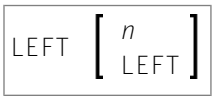
Wenn Sie das LC-Kommando ohne Parameter eingeben, wird die aktuelle Zeile in Kleinschreibung geändert.

Beispiel:

```
LC C'NAME' .X .Y ALL PREFIX NX
```

Dieses Kommando ändert alle Zeilen innerhalb des mit .X und .Y markierten Blocks in Kleinschreibung, wenn Sie die Zeichenkette NAME (genau wie hier in Großbuchstaben eingegeben) als Vorsilbe eines Wortes enthalten.

LEFT



Mit diesem Kommando können Sie in den Daten nach Links blättern.

n	Blättert um *n* Spalten nach Links.
LEFT	Blättert um die maximal mögliche Spaltenzahl nach Links.

Wenn *n* oder LEFT weggelassen wird, dann wird der Blätterbetrag durch den Blättermodus bestimmt.

LIMIT

```
LIMIT [n]
```

Mit diesem Kommando können Sie die maximale Anzahl Zeilen angeben, die mit einem FIND- oder RFIND-Kommando durchsucht werden sollen. Der Parameter *n* gibt die Anzahl der zu durchsuchenden Zeilen an.

Diese Einstellung kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

LOCATE

```
[LOCATE] { 0       }
         { n       }
         { .label  }
```

Mit diesem Kommando können Sie bewirken, dass eine bestimmte Zeile an den oberen Rand des Editierbereichs geblättert wird (sie wird dadurch zur aktuellen Zeile).

Das Kommando bietet folgende Optionen:

0	Die erste Zeile der Daten wird zur aktuellen Zeile.
n	Zeile *n* wird zur aktuellen Zeile.
.label	Die mit *.label* markierte Zeile wird zur aktuellen Zeile.

Das LOCATE-Kommando unterscheidet sich folgendermaßen vom FIND-Kommando:

- Das FIND-Kommando erweist sich bei Textsuchen als wirkungsvoller, während das LOCATE-Kommando in erster Linie dafür gedacht ist, Zeilennummern oder Zeilenmarkierung zu finden.

- Das LOCATE-Kommando findet Text nur in Großschreibung ab Zeile 1 des Editors. Außerdem müssen die Daten im Editor in alphabetischer Reihenfolge vorliegen, damit eine Zeichenkette gefunden wird.
- Wenn eine Zeile mit dem LOCATE-Kommando aufgefunden wird, dann wird der Cursor in den Präfixbereich dieser Zeile gestellt und die Zeile wird an den oberen Rand der des Editors platziert.

  Beim FIND-Kommando wird der Cursor auf die gesuchte Zeichenkette gesetzt und die betreffende Zeile wird nicht zwangsläufig an den oberen Rand der des Editierbereichs platziert.

Beispiele:

```
LOC 32
```

Platziert Zeile 32 an den oberen Rand des Editierbereichs.

```
32
```

Wie zuvor.

```
LOC .X
```

Platziert die mit .X markierte Zeile an den oberen Rand des Editierbereichs.

LOG

```
LOG [ ON  ]
    [ OFF ]
```

Mit diesem Kommando können Sie die Protokolldatei aktivieren bzw. deaktivieren.

Die Protokolldatei enthält die Historie aller Änderungen, die seit dem Beginn der Editor-Sitzung im Editor ausgeführt wurden. Wenn die Protokolldatei aktiv ist, werden jedes Mal, wenn Sie Enter drücken, die Änderungen, die seit dem letzten Enter gemacht wurden, in der Protokolldatei aufgezeichnet. Wenn Sie das UNDO-Kommando benutzen, können Sie nacheinander alle Änderungen soweit rückgängig machen, bis der ursprüngliche Zustand bei Beginn der Editor-Sitzung wiederhergestellt ist.

Diese Einstellung kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

⚠ **Wichtig:** Sie müssen sich vergewissern, dass das LOG-Kommando aktiviert ist, bevor Sie mit dem Editieren beginnen.

MASK

```
MASK [ ON  ]
     [ OFF ]
```

Mit diesem Kommando können Sie die Maskenfunktion aktivieren bzw. deaktivieren. Wenn die Maskenfunktion aktiviert ist, wird jedes Mal, wenn Sie eine Zeile im Editor einfügen, statt einer Leerzeile eine vordefinierte Textzeile eingegeben. Die Maskenzeile wird mit dem Zeilenkommando `MASK` definiert. Die Maskenfunktion ist nützlich, wenn Sie mehrere Zeilen Code schreiben müssen, die identsich oder sehr ähnlich sind.

Weitere Informationen siehe Abschnitt ***Um eine Maskenzeile zu definieren und zu benutzen***.

Die Standardeinstellung ist `MASK OFF`. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

MWINDOW

```
MWINDOW [ n   ]
        [ n m ]
```

Mit diesem Kommando können Sie ein Datenfenster unter Angabe von Parametern verschieben.

n	Die Nummer der Zeile, in der das Datenfenster eingefügt werden soll.
m	Die Nummer der Spalte, in der das Datenfenster eingefügt werden soll.

NULLS

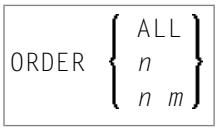
```
NULLS [ ON  ]
      [ OFF ]
```

Mit diesem Kommando können Sie bestimmen, ob die Datenzeilen mit Nullzeichen gefüllt werden.

ON	Das Ende jeder Zeile wird mit Nullzeichen gefüllt.
OFF	Es werden keine Zeilen mit Nullzeichen gefüllt.

Die Standardeinstellung ist NULLS ON. Sie kann in Ihrem Profil geändert werden. Siehe ***Editor-Profil einstellen***.

ORDER



Mit diesem Kommando können Sie Datenzeilen zusammenführen.

ALL	Führt alle Zeilen zusammen.
n	Führt die Zeilen ab Zeile *n* bis zur letzten Zeile zusammen.
n m	Führt die Zeilen ab Zeile*n* bis zur Zeile *m* zusammen.

Das ORDER-Kommando gilt nur innerhalb der eingestellten Grenzen, die mit dem Hauptkommando BNDS eingestellt werden.

Innerhalb der eingestellten Grenzen werden die Zeilen miteinander verkettet und so weit wie möglich gefüllt. Wörter die nicht in eine Zeile passen, werden automatisch in die nächste Zeile gestellt.

Um Datenzeilen zusammenzuführen, können Sie auch die Zeilenkommandos TF, TO und TOO benutzen.

POWER

```
POWER
```

Mit diesem Kommando können Sie den Editor in den Texteingabe-Modus schalten. Es erscheint ein leerer Bildschirm, in den Sie eine oder mehrere Textzeilen eingeben können. Nach der Eingabe drücken Sie Enter, und der Text wird in die erste Zeile des Editierbereichs eingefügt.

PROFILE

```
PROFILE [n]
```

Mit diesem Kommando können Sie sich Ihr Editor-Profil am oberen Rand des Editor-Bilschirms anzeigen lassen.

Mit *n* geben Sie zusätzlich anzuzeigende Zeilen an. Mögliche Werte für *n* sind:

6	Zeigt Ihr Editor-Profil und alle Tabulatorpositionen (so wie mit dem Kommando TABS angegeben) an.
7	Wie bei 6, zusätzlich wird die Maskenzeile (so wie mit dem Kommando MASK angegeben) angezeigt.
8	Wie bei 7, zusätzlich werden die Bereichsgrenzen (so wie mit dem Kommando BNDS angegeben) angezeigt.
9	Wie bei 8, zusätzlich werden die Spaltennummern (so wie mit dem Kommando COLS angegeben) angezeigt.

PROTECT

```
PROTECT [ INS ]
        [ ON  ]
        [ OFF ]
```

Mit diesem Kommando können Sie den **Präfixbereich** schützen. Um bei einem gegeschützten Präfixbereich Zeilenkommandos einzugeben, müssen Sie das Zeilenkommando mit vorangestelltem Fluchtzeichen in Spalte 1 des Editierbereichs eingeben.

INS	Schützt den Präfixbereich gegen das Hinzufügen von Zeilen, wenn das Einfüge-Kommando benutzt wird.
ON	Aktiviert den Schutz.
OFF	Deaktiviert den Schutz.

Die Standardeinstellung ist PROTECT ON. Sie kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

RCHANGE

RCHANGE

Mit diesem Kommando können Sie das zuletzt abgesetzte CHANGE-Kommando wiederholen.

RECOVERY

RECOVERY [ON / OFF] [n]

Mit diesem Kommando können Sie die Wiederherstellungsfunktion für die aktuelle Editor-Sitzung aktivieren bzw. deaktivieren. Außerdem können Sie angeben, ab wieviel geänderten Zeilen eine Checkpoint-Speicherung durchgeführt werden soll.

Wenn Sie den Parameter *n* benutzen, geben Sie die Anzahl der geänderten Zeilen an, nach denen eine Checkpoint-Speicherung durchgeführt wird.

Die Standardeinstellung ist RECOVERY ON. Sie kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

RENUMBER

```
RENUMBER [ ON       ]
         [ OFF      ]
         [ n1 n2 n3 ]
```

Nur bei PDS Members und sequenziellen Datasets.

Mit diesem Kommando können Sie Parameter für die Neunummerierung von Zeilen im Editierbereich angeben:

ON	Zeilen werden neu nummeriert.
OFF	Zeilen werden nicht neu nummeriert.
n1	Inkrement der Nummerierung (der Standardwert ist in Ihrem Editor-Profil festgelegt).
n2	Start-Spalte für die neue Zeilennummerierung (Standardwert: 73).
n3	Ende-Spalte für die neue Zeilennummerierung (Standardwert: 80).

Zum Deaktivieren der Neunummerierung von Zeilen siehe Kommando UNREN.

RESET

```
RESET
```

Mit diesem Kommando können Sie alle ausstehenden Zeilenkommandos zurücksetzen und alle Zeilenmarkierungen löschen.

RFIND

```
RFIND
```

Mit diesem Kommando können Sie das letzte FIND-Kommando wiederholen.

RIGHT

```
RIGHT [ n     ]
      [ RIGHT ]
```

Mit diesem Kommando können Sie in den Daten nach Rechts blättern.

n	Blättert um *n* Spalten nach Rechts.
RIGHT	Blättert um die maximal mögliche Spaltenzahl nach Rechts.

Wird *n* oder RIGHT weggelassen, dann wird der Spaltenbetrag, um den nach Rechts geblättert wird, durch den Blättermodus bestimmt.

SORT

```
SORT [n m] [ .X     ] [ A ]
           [ .X .Y  ] [ D ]
```

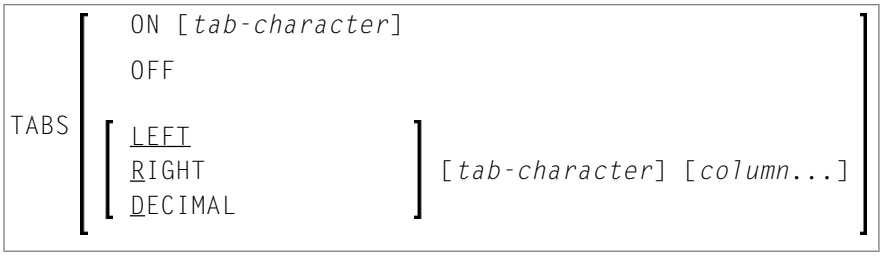
Das Kommando SORT sortiert die Zeilen im Editor in aufsteigender oder absteigender alphabetischer Reihenfolge. Wenn Sie ein SORT-Kommando ohne Parameter absetzen, werden alle Daten im Objekt in aufsteigender Reihenfolge sortiert.

n m	Sortiert ab Spalte *n* bis zur Spalte *m*.
.X	Sortiert ab der mit .X markierten Zeile bis zum Ende des Objekts.
.X .Y	Sortiert ab der mit .X markierten Zeile bis zu der mit .Y markierten Zeile (dabei sind .X und .Y eine beliebige Zeichenkette mit bis zu vier Zeichen).
A	Sortiert die Daten in aufsteigender Reihenfolge (A bis Z).

D	Sortiert die Daten in absteigender Reihenfolge (Z bis A).

Anmerkung: Die maximale Länge des Schlüssels beim Sortieren beträgt 80 Zeichen. Alle darüberhinausgehenden Zeichen werden als Sortierkriterien ignoriert.

TABS



Mit diesem Kommando können Sie Tabulatoreinstellungen steuern.

Mit dem Kommando `TABS ON` bzw. `TABS OFF` können Sie die logische oder physische Tabulierung ein- bzw. ausschalten. Die Tabulierung wird außerdem durch jedes Kommando eingeschaltet, das eine Tabulatoreinstellung ändert.

Mit dem folgenden Kommando wird beispielsweise die logische Tabulierung mit einem Und-Zeichen (&) als logischem Tabulatorzeichen eingeschaltet:

```
TABS &
```

Die Tabulatorpositionen setzen Sie mit dem Kommando `TABS`. Beispiel: Das folgende Kommando setzt Tabulatoren in den Spalten 10, 20 und 30.

```
TABS 10 20 30
```

Sie können Daten eingeben und diese automatisch an eine bestimmte Tabulatorposition verschieben, indem Sie den Daten ein logisches Tabulatorzeichen voranstellen. Ein einzelnes Tabulatorzeichen verschiebt die Daten an die nächste Tabulatorposition, zwei Tabulatorzeichen verschieben die Daten an die zweite Tabulatorposition usw.

Um das aktuellen Einstellungen des `TABS`-Kommandos anzuzeigen, setzen Sie das Hauptkommando `PROFILE` ab.

Die Standardeinstellung ist `TABS OFF`. Sie kann in Ihrem Profil geändert werden. Siehe *Editor-Profil einstellen*.

Außer den Tabulatorpositionen können Sie beim `TABS`-Kommando noch folgende Parameter angeben:

LEFT	Ordnet die Daten links von der Tabulatorposition an.
RIGHT	Ordnet die Daten rechts von der Tabulatorposition an.
DECIMAL	Ordnet die Daten so an, dass sich das darin enthaltene Dezimalzeichen an der Tabulatorposition befindet.

Um Daten in einer bestimmten Spalte mit einer bestimmten Einrückung tabellarisch darzustellen, sind mehrere Tabulatorzeichen möglich. Setzen Sie das Zeilenkommando TABS ab und überschreiben Sie jeden Stern (*) mit einem anderen Sonderzeichen. Jegliche Daten, die Sie unter Voranstellung eines dieser Sonderzeichen eingeben, werden in der entsprechenden Spalte tabuliert.

Sie können nach jedem Tabulatorzeichen ein L (für LEFT), ein R (für RIGHT) oder ein D (für DECIMAL) eingeben, um die Platzierung der Daten an der Tabulatorposition anzugeben.

Anmerkung: Weitere Informationen und Beispiel siehe *Physischen oder logischen Tabulator benutzen*.

TOP

```
TOP
```

Mit diesem Kommando können Sie an den Anfang des Objekts blättern, das Sie gerade bearbeiten.

UC

```
UC [ * ]                [ .X    ] [ n   ] [ ALL   ] [ CHARS  ] [ NX ]
   [ [T]'string' ]      [ .X .Y ] [ n m ] [ NEXT  ] [ WORD   ] [ X  ]
   [ 'string'    ]                        [ PREV  ] [ PREFIX ]
   [ C'string'   ]                        [ FIRST ] [ SUFFIX ]
   [ X'string'   ]                        [ LAST  ]
   [ P'string'   ]
```

Das Kommando UC setzt eine oder mehrere Zeile in Großschreibung um. Es können die gleichen Parameter wie beim Kommando LC angegeben werden. Wenn Sie das Kommando UC ohne Parameter eingeben, wird die aktuelle Zeile in Großschreibung umgesetzt.

UNDO

```
UNDO [ ALL ]
     [ n   ]
```

Falls die Protokolldatei aktiv ist (siehe Kommando LOG), werden alle Änderungen, die Sie vorgenommen haben, nachdem Sie das letzte Mal Enter gedrückt haben, rückgängig gemacht. Durch wiederholtes Absetzen des UNDO-Kommandos werden aufeinander folgende Änderungen in umgekehrter Reihenfolge rückgängig gemacht. Auf diese Weise können Sie nacheinander alle Änderungen soweit rückgängig machen, bis der ursprüngliche Zustand bei Beginn der Editor-Sitzung wiederhergestellt ist.

Beim UNDO-Kommando können Sie folgende Parameter angeben:

ALL	Alle während der aktuellen Editor-Sitzung vorgenommenen Änderungen werden rückgängig gemacht.
n	Die letzten *n* Änderungen werden rückgängig gemacht.

UNREN

```
UNREN [n m]
```

Deaktiviert die Neunummerierung von Zeilen.

n	Start-Spalte für die neue Zeilennummerierung (Standardwert: 73).
m	Ende-Spalte für die neue Zeilennummerierung (Standardwert: 80).

Zum Aktivieren der Neunummerierung von Zeilen siehe Kommando RENUMBER.

UP

```
UP [n]
```

Mit diesem Kommando können Sie in den Daten aufwärts blättern.

Der Parameter *n* gibt an, um wie viele Zeilen nach oben geblättert werden. Wenn *n* weggelassen wird, dann wird der Blätterbetrag durch den Blättermodus bestimmt.

WINDOW

```
WINDOW l1 l2 [ n   ]
             [ n m ]
```

Mit diesem Kommando können Sie ein Datenfenster definieren, das kopiert oder verschoben werden soll. Die Zeile und Spalte, in denen das Fenster beginnt, und die Zeile und Spalte, in denen das Fenster endet, werden mit den Kommando-Parametern angegeben. Mindestens *l1* und *l2* sind notwendig.

l1 l2	Definiert ein Fenster, das bei Spalte 1 von Zeile *l1* beginnt und bei der letzten Spalte von Zeile *l2* endet.
l1 l2 n	Definiert ein Fenster, das bei Spalte *n* von Zeile *l1* beginnt und bei der letzten Spalte von Zeile *l2* endet.
l1 l2 n m	Definiert ein Fenster, das bei Spalte *n* von Zeile *l1* beginnt und bei Spalte *m* von Zeile *l2* endet.

Beachten Sie, dass sämtliche Daten innerhalb der angegebenen Punkte im Quellcode-Arbeitsbereich Bestandteil des Fensters werden. Beispiel siehe ***Ein Fenster mit Daten kopieren oder verschieben***.

XSWAP

```
XSWAP
```

Mit diesem Kommando können Sie angezeigte Zeilen gegen ausgeschlossene Zeilen tauschen. Zum Ausschließen von Zeilen dient das Kommando `EXCLUDE`.

Stichwortverzeichnis

A

add
    line in Software AG Editor, 448
align
    text in Software AG Editor, 465

C

catalog
    source in Programm-Editor, 155
center
    line in Software AG Editor, 464
column
    show position in Software AG Editor, 442
command
    cursorsensitive commands
        in Programm-Editor, 154
    Editor-Kommandos in Programm-Editor, 138
    line commands in Software AG Editor, 487
    main commands in Software AG Editor
        , 493
copy
    line in Software AG Editor, 450
    specific text section
        in Software AG Editor, 454
cursor position
    assign to PF key
        in Programm-Editor, 150
customize
    features of Programm/Data Area Editor, 119
    features of Software AG Editor, 481

D

Data Area Editor
    customize editor features, 119
    Zeilenkommandos, 177
DDM
    indicator field, 412
delete
    line in Software AG Editor, 448

E

editor screen
    split to show two sources, 116
exit
    from Programm-Editor, 156
    from Software AG Editor, 485

F

find
    text in Software AG Editor, 473
Format
    Einstellungen im Masken-Editor, 215
format
    center line in Software AG Editor, 464

H

help
    for SYSDDM utility functions, 388
hide
    text in Software AG Editor, 441

I

indicator
    for field in DDM, 412

K

Kommando
    Data Area Editor, 177
    Zeilenkommandos im Programm-Editor, 148

L

line
    add/delete in Software AG Editor, 448
    center in Software AG Editor, 464
    copy in Software AG Editor, 450
    move in Software AG Editor, 450
    order in target zone in Software AG Editor, 463
    repeat with mask in Software AG Editor, 462
    show at top of screen in Software AG Editor, 440
    show/hide in Software AG Editor, 441
    sort in Software AG Editor, 473
lock
    activate/deactivate for object, 109
        (Siehe auch unlock)
lower case
    change in Programm-Editor, 137

M

Map-Editor
    Formateinstellungen, 215
Masken-Editor
    Formateinstellungen, 215
    Masken-Standardeinstellungen, 209
merge
    lines in Software AG Editor, 450
move
    line in Software AG Editor, 450
    specific text section
        in Software AG Editor, 454

O

open
    Programm-Editor, 134
    Software AG Editor, 433

P

Profil
    Masken-Standardeinstellungen
        Masken-Editor, 209
profile
    set for Programm/Data Area Editor, 119
    set for Software AG Editor
        , 481
Programm-Editor
    change lower/upper case, 137
    cursorsensitive commands, 154
    customize editor features, 119
    Editor-Kommandos, 138
    exit from, 156
    open, 134
    PF key for cursor position, 150
    save/catalog source, 155
    use, 133
    Zeilenkommandos, 148

Q

Quellcode
    save/catalog in Programm-Editor, 155

R

replace
    text in Software AG Editor
        , 477

S

save
    source in Programm-Editor, 155
    source in Software AG Editor, 485
scroll
    data in Software AG Editor, 437
show
    column position in Software AG Editor, 442
    tab position in Software AG Editor, 442
    target zone in Software AG Editor, 442
    text in Software AG Editor, 441
Software AG Editor
    add/remove line, 448
    align text, 465
    center line, 464
    copy/move line, 450
    copy/move specific text section, 454
    customize editor features, 481
    define target zone for command, 460
    exit from, 485
    find text, 473
    line commands, 487
    main commands, 493
    merge lines, 450
    open, 433
    order lines in target zone, 463
    repeat line with template, 462
    replace text, 477
    save source, 485
    scroll data, 437
    set tab, 468
    show line at top of screen, 440
    show target zone/tab/column, 442
    show/hide text, 441
    sort lines, 473
sort
    lines in Software AG Editor, 473
source code
    save in Software AG Editor, 485
split
    editor screen, 116
SYSDDM utility
    help information for, 388

T

tab
    set in Software AG Editor, 468
    show position in Software AG Editor, 442
target zone
    define for command in Software AG Editor, 460
    show in Software AG Editor, 442
template
    for new line in Software AG Editor, 462
terminate
    Programm-Editor, 156
    Software AG Editor, 485
text
    align in Software AG Editor, 465
    copy/move specific section
        in Software AG Editor, 454
    show/hide in Software AG Editor, 441

U

unlock
    object, 112
        (Siehe auch lock)
upper case
    change in Programm-Editor, 137
use
    Programm-Editor, 133