

Natural

Editoren

Version 8.2.8

April 2023

Dieses Dokument gilt für Natural ab Version 8.2.8.

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

Copyright © 1979-2023 Software AG, Darmstadt, Deutschland und/oder Software AG USA, Inc., Reston, VA, USA, und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.

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

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

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

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

Dokument-ID: NATMF-NNATEDITORS-828-20230425DE

Inhaltsverzeichnis

Vorwort	ix
1 Über diese Dokumentation	1
Dokumentationskonventionen	2
Online-Informationen und Support	2
Datenschutz	3
I Sperrmechanismen für Quellcode-Objekte	5
2 Sperrmechanismen für Quellcode-Objekte	7
Quellcode-Objekte bearbeiten	9
Objekte speichern	9
Objekte entsperren	10
Objekte verschieben, löschen, umbenennen und ersetzen	10
II Editoren – Allgemeine Informationen	11
3 Editoren – Allgemeine Informationen	13
Split-Screen-Modus	14
Editor-Profil	17
III Programm-Editor	29
4 Programm-Editor	31
Programm-Editor aufrufen	32
Editor-Kommandozeile	33
Obere Informationszeile	34
Editierbereich	34
Untere Informationszeile	36
Editor-Kommandos	36
Editor-Kommandos zum Positionieren	44
Zeilenkommandos	45
Spezielle PF-Tastenfunktionen	48
Cursorsensitive Kommandos	52
Quellcode speichern und katalogisieren	53
Exit-Funktion	54
IV Datenbereich-Editor (Data Area Editor)	57
5 Datenbereich-Editor (Data Area Editor)	59
Data Area Editor aufrufen	60
Obere Informationszeile	62
Editor-Kommandozeile	63
Untere Informationszeile	64
Editierbereich benutzen	64
Spalten im Editierbereich	65
Erweiterte Felddefinition bearbeiten	70
Zeilenkommandos	75
Editor-Kommandos	81
Editor-Kommandos zum Positionieren	86
Data Area speichern und katalogisieren	87
User Exit für den Data Area Editor	88

Exit-Funktion	89
V Masken-Editor (Map Editor)	91
6 Komponenten des Masken-Editors	93
7 In vier Schritten zur fertigen Maske	95
Schritt 1 - Masken-Profil definieren	96
Schritt 2 - Maske definieren	96
Schritt 3 - Maskenfelder definieren	96
Schritt 4 - Masken-Definition speichern	97
8 Masken-Editor aufrufen und beenden	99
Masken-Editor aufrufen und neue Maske erstellen	100
Masken-Editor aufrufen und vorhandene Maske bearbeiten	101
Masken-Editor beenden	102
9 Funktionen im Menü 'Edit Map'	103
Feld- und Variablen-Definitionen bearbeiten – Funktion "Field and Variable Definitions"	104
Maske editieren – Funktion "Edit Map"	105
Feldumrahmungen editieren – Funktion "Outline Editor"	106
Neue Maske erstellen – Funktion "Initialize New Map"	106
Neue Hilfe-Maske erstellen – Funktion "Initialize a New Help Map"	106
Profile und Geräte pflegen – Funktion "Maintenance of Profiles & Devices"	107
Maske speichern – Funktion "Save Map"	107
Maske testen – Funktion "Test Map"	107
Maske speichern und katalogisieren – Funktion "Stow Map"	108
Hilfe zum Masken-Editor aufrufen – Funktion "Help"	108
10 Maske erstellen – Funktion 'Initialize New Map'	109
Trennzeichen – "Delimiters"	111
Masken-Format-Einstellungen – "Format"	113
Kontext-Einstellungen – "Context"	117
Füllzeichen – "Filler Characters"	119
11 Maske bearbeiten – Funktion 'Edit Map'	121
Bildschirm-Modi	122
PF-Tasten und Kommandos zum Positionieren	123
Zeilenkommandos	124
Feld-Kommandos	126
12 Maskenfelder definieren	129
Maskenfelder direkt im Bildschirm definieren	130
Definitionen aus anderen Objekten übernehmen	131
Systemvariablen in einer Masken-Definition benutzen	136
13 Erweiterte Feld-Bearbeitung	139
Funktion zur erweiterten Feld-Bearbeitung aufrufen und beenden	140
Felder im erweiterten Feld-Bearbeitungsbereich	142
Felder im erweiterten Text-Feld-Bearbeitungsbereich	146
14 Feldumrahmungen erstellen – mit dem Outline-Editor	149
Voraussetzungen	150

Editor für Feldumrahmungen (Outline-Editor) aufrufen	152
Box-Codes für Feldumrahmungen benutzen	153
Ende-Markierungen für Feldumrahmungen benutzen	155
Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen	156
Beispiel für Feldumrahmungen – Box-Coding	157
Zeilenkommandos und PF-Tasten	160
15 Nachträgliches Zuweisen von Feldern	163
16 Array- und Tabellen-Definitionen	165
Array-Definition	166
Array-Tabellen-Definition	170
17 Verarbeitungsregeln	175
Feldbezogene Verarbeitungsregeln	176
Funktionstastenbezogene Verarbeitungsregeln	177
Verarbeitungsregel bearbeiten	178
VI Map Editor Tutorial	185
18 Masken-Editor öffnen	187
19 Maskenfelder anlegen, positionieren und löschen	193
Felder anlegen und zentrieren	194
Felder verschieben	199
Felder löschen und Zeilen einfügen	214
20 Maske testen und speichern	225
21 Verarbeitungsregeln definieren	229
22 Felder benennen und Maske speichern/katalogisieren	239
23 Feld-Eigenschaften definieren	243
24 Hilfe-Maske erstellen und testen	249
25 Maske mit INPUT USING MAP aufrufen	255
26 Maske für WRITE erstellen und Feld-Definitionen kopieren	261
27 Layout einer Maske wiederverwenden	269
28 Maske mit WRITE USING MAP aufrufen	273
VII DDM-Editor (SYSDDM Utility)	277
29 Funktionsbeschreibung	279
DDMs speichern	280
Einschränkungen bei der Benutzung	281
30 SYSDDM aufrufen und beenden	283
31 SYSDDM Pflege- und Service-Funktionen benutzen	285
Hilfe zu Funktionen	286
Funktionen ausführen	287
Beschreibung der Funktionen	287
DDM-Parameter angeben	291
32 DDMs erstellen	295
33 DDM-Editor aufrufen und beenden	299
Editor aufrufen	300
Editor-Sitzung beenden	301
34 DDM-Editor benutzen	303
DDM-Kopfzeilen-Informationen	304

Spalten für Feld-Attribute	305
Kommandos zum Editieren und Ausführen von Funktionen	311
Erweiterte Feld-Attribute angeben	317
35 DDM katalogisieren	321
36 DDMs auflisten	323
37 DDMs in unterschiedlichen Umgebungen pflegen	325
VIII Software AG Editor	327
38 Allgemeine Informationen zum Software AG Editor	329
39 Software AG Editor aufrufen	331
40 Benutzung des Editor-Bildschirms	333
Benutzung der Eingabebereiche	335
In den Daten im Editierbereich blättern	335
Zu einer bestimmten Zeile springen	338
Zeilen zeigen oder verbergen	339
Ränder, Tabulator- und Spaltenpositionen anzeigen	340
41 Kommandos benutzen	341
Ausführung eines Kommandos	342
42 Daten anlegen und ändern	345
Zeilen einfügen und löschen	346
Zeilen kopieren, verschieben, überlagern und wiederholen	348
Ein Fenster mit Daten kopieren oder verschieben	352
Horizontale und vertikale Bereichsgrenzen einstellen	358
Maskenzeile definieren	360
Daten zwischen Bereichsgrenzen anordnen	361
Daten zentrieren	362
Daten ausrichten	363
Physischen oder logischen Tabulator benutzen	366
Zeilen in alphabetischer Reihenfolge sortieren	371
Zeichenkette suchen	371
Zeichenkette ersetzen	375
43 Editor-Profil einstellen	379
Editor-Profil-Einstellungen	381
44 Daten speichern und den Software AG Editor verlassen	383
45 Zeilenkommandos (Übersicht)	385
46 Hauptkommandos (Übersicht)	391
Syntax-Elemente	393
Gemeinsame Kommando-Optionen	394
ADVANCE	396
AORDER	397
AUTOREN	397
AUTOSAVE	398
BNDS	398
BOTTOM	399
CANCEL	399
CAPS	399

CEnter	400
CHANGE	400
COLS	403
CURSOR	404
CWINDOW	404
DELETE	404
DOWN	406
DWINDOW	406
EMPTY	407
END	407
ESCAPE	407
EXCLUDE	408
FIND	410
FIX	413
HEX	413
INCLUDE	413
JLEFT	414
JRIGHT	414
JUSTIFY	415
LABEL	416
LC	417
LEFT	418
LIMIT	419
LOCATE	419
LOG	420
MASK	421
MWINDOW	421
NULLS	421
ORDER	422
POWER	423
PROFILE	423
PROTECT	423
RCHANGE	424
RECOVERY	424
RENUMBER	425
RESET	425
RFIND	426
RIGHT	426
SORT	426
TABS	427
TOP	428
UC	428
UNDO	429
UNREN	429
UP	430

WINDOW	430
XSWAP	431
Stichwortverzeichnis	433

Vorwort

Dieses Dokument beschreibt alle Editoren, die in der Natural-Entwicklungsumgebung zur Verfügung stehen.

Eine Anleitung zur Benutzung der 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.

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

Sperrmechanismen für Quellcode-Objekte	Beschreibt die Sperrmechanismen, die bei Natural-Quellcode-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.
Programm-Editor	Beschreibt den Editor, der benutzt wird, um Natural-Programme, Subprogramme, Subroutinen, Klassen, Copycodes, Helprountinen, Functions und Text-Objekte zu erstellen und zu ändern.
Datenbereich-Editor (<i>Data Area Editor</i>)	Beschreibt den Editor, der zum Erstellen und Ändern von lokalen, globalen und Parameter-Datenbereichen (Data Areas) benutzt wird.
Masken-Editor (<i>Map Editor</i>)	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 (<i>SYSDDM Utility</i>)	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 <i>Ordner.Unterordner.Dienst</i> , Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
<i>Kursivschrift</i>	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 Alarmer 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 Sperrmechanismen für Quellcode-Objekte

2 Sperrmechanismen für Quellcode-Objekte

- Quellcode-Objekte bearbeiten 9
- Objekte speichern 9
- Objekte entsperren 10
- Objekte verschieben, löschen, umbenennen und ersetzen 10

Natural bietet Sperrmechanismen, durch die ein konkurrierendes Ändern von Natural-Quellcode-Objekten verhindert wird. Durch diese Mechanismen können Quellcode-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*.

■ Quellcode-Objekte in lokalen Umgebungen und in SPoD-Umgebungen sperren

`SLOCK=PRE` aktiviert die Sperrung von Quellcode-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.

■ Quellcode-Objekte in SPoD-Umgebungen sperren

Die Standardeinstellung `SLOCK=SPoD` aktiviert die Sperrung von Quellcode-Objekten nur in einer SPoD-Umgebung. Ein Quellcode-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 Quellcode-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 Quellcode-Objekt aktualisieren. Dies geschieht durch den Vergleich der Zeitmarke des in der Datenbank gespeicherten Quellcode-Objekts mit der Zeitmarke des Quellcode-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 Quellcode-Objekt-Sperrung, wenn der Profilparameter `SLOCK=PRE` in einer lokalen Umgebung gesetzt ist.

Quellcode-Objekte bearbeiten

Bei folgenden Natural-Objekttypen werden die Quellcode-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 Quellcode-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 Quellcode-Objekt unter einem neuen Namen speichern.



Anmerkung: Wenn Sie eine Source mit dem Systemkommando `READ` in den Arbeitsbereich einlesen, erfolgt *keine* Sperrung des Quellcode-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 Quellcode-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 Quellcode-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 Quellcode-Objekt verschieben, löschen, umbenennen und ersetzen, wird der Sperrzustand des Objekts geprüft:

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

Einschränkungen

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

II Editoren — Allgemeine Informationen

3 Editoren — Allgemeine Informationen

- Split-Screen-Modus 14
- Editor-Profil 17

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, Helprountinen, 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 **cursor sensitives 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
<u>S</u> PLIT ++	Mit diesem Kommando können Sie den geteilten Bildschirm an das Ende des Objekts positionieren.
<u>S</u> PLIT B	
<u>S</u> PLIT --	Mit diesem Kommando können Sie den geteilten Bildschirm an den Anfang des Objekts positionieren.
<u>S</u> PLIT T	
<u>S</u> PLIT +	Mit diesem Kommando können Sie den geteilten Bildschirm um eine Seite vorwärts positionieren.
<u>S</u> PLIT +P	

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

Kommando	Funktion
<code>SPLIT SCAN [scan-value]</code>	<p>Quellcode durchsuchen.</p> <p>Dabei ist <i>scan-value</i> die Zeichenkette, nach der im aktuellen Quellcode gesucht werden soll.</p> <p>Jede Zeile, die den <i>scan-value</i> enthält, wird mit einem Größer-als-Zeichen (>) markiert. Um weiter nach demselben <i>scan-value</i> zu suchen, brauchen Sie nur <code>S SC</code> einzugeben.</p>
<code>SPLIT VIEW name [SHORT]</code>	<p>Objekttyp View (DDM, entsprechend Definition in Predict oder SYSDDM Utility) anzeigen.</p> <p>Wenn Sie <code>SHORT</code> 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).</p> <p>Im Data Area Editor können Sie bei <i>name</i> Stern-Notation (*) benutzen, um eine Liste des so angegebenen Bereichs von Objekten anzuzeigen.</p>
<code>SPLIT</code>	<p>Vorheriges Objekt anzeigen.</p> <p>Wenn Sie das Kommando <code>SPLIT</code> 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.</p> <p>Diese Funktion gibt es nur beim Programm-Editor.</p>

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 Profilenames 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	<p>Der Name des Editor-Profiles. 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 Profilenames SYSTEM mit Ihrer Benutzerkennung überschreiben und das umbenannte Profil in der Datenbank speichern.</p> <p>Wenn Sie den Namen Ihres Profils mit einem anderen gültigen Profilenames (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.</p> <p>Sie können aber den Profilenames des Profils eines anderen Benutzers mit Ihrer eigenen Benutzerkennung überschreiben und dann das umbenannte Profil in der Datenbank speichern.</p>
PF and PA Keys	<p>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.</p>

Feld	Erläuterung
Auto Renumber	<p>Ein Y gibt an, dass der Quellcode im Programm-Editor automatisch neu nummeriert werden soll, wenn einer der folgenden Fälle eintritt:</p> <ul style="list-style-type: none"> ■ 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. <p>Anmerkung: Siehe auch <i>Quellcode-Zeilennummern umnummerieren im Leitfaden zur Programmierung</i>.</p>
Auto Save Numbers	<p>Wenn ein numerischer Wert eingegeben 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 Quellcode-Objekt gespeichert, welches im Feld Source Save into angegeben ist.</p> <p>Auto Save Numbers gilt auch für den Masken-Editor.</p>
Source Save into	<p>Der Name des Quellcode-Objekts, in das automatisch eine Kopie der Source in der aktuellen Library gespeichert werden soll. Der Standard-Name EDITWORK kann geändert werden. Das angegebene Quellcode-Objekts wird jedes Mal, wenn die im Feld Auto Save Numbers angegebene Anzahl an Änderungen überschritten wird, überschrieben.</p>
Additional Options	<p>Siehe Abschnitt <i>Zusätzliche Optionen</i>.</p>

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 <i>Programm-Editor</i> und <i>Data Area Editor</i> . Mögliche Einstellungen: <table border="1" data-bbox="516 520 1482 814"> <tr> <td>Y</td> <td>Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt. Dies ist die Standard-Einstellung.</td> </tr> <tr> <td>N</td> <td>Leer gelassenen Zeilen werden beim Drücken von Enter <i>nicht</i> aus der Source entfernt.</td> </tr> </table>	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 <i>nicht</i> aus der Source entfernt.
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 <i>nicht</i> 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 <i>Programm-Editor</i> . Mögliche Einstellungen: <table border="1" data-bbox="516 940 1482 1234"> <tr> <td>Y</td> <td>Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt.</td> </tr> <tr> <td>N</td> <td>Leer gelassenen Zeilen werden beim Drücken von Enter <i>nicht</i> aus der Source entfernt. Dies ist die Standard-Einstellung.</td> </tr> </table>	Y	Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt.	N	Leer gelassenen Zeilen werden beim Drücken von Enter <i>nicht</i> aus der Source entfernt. Dies ist die Standard-Einstellung.
Y	Alle leer gelassenen Zeilen werden beim Drücken von Enter aus der Source entfernt.				
N	Leer gelassenen Zeilen werden beim Drücken von Enter <i>nicht</i> aus der Source entfernt. Dies ist die Standard-Einstellung.				
Source Size Information	<table border="1" data-bbox="516 1234 1482 1512"> <tr> <td>Y</td> <td>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.</td> </tr> <tr> <td>N</td> <td>Diese Informationen werden nicht angezeigt.</td> </tr> </table>	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.
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	<table border="1" data-bbox="516 1512 1482 1890"> <tr> <td>Y</td> <td> <p>Programm-Editor:</p> <p>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 <i>Änderungsanzeige</i> in der <i>Programm-Editor-Dokumentation</i>.</p> <p>Data Area Editor:</p> </td> </tr> </table>	Y	<p>Programm-Editor:</p> <p>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 <i>Änderungsanzeige</i> in der <i>Programm-Editor-Dokumentation</i>.</p> <p>Data Area Editor:</p>		
Y	<p>Programm-Editor:</p> <p>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 <i>Änderungsanzeige</i> in der <i>Programm-Editor-Dokumentation</i>.</p> <p>Data Area Editor:</p>				

Option	Erläuterung				
	<p>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 <i>Data Area Editor</i>-Dokumentation.</p> <hr/> <p>N Es wird keine Transaktionsmeldung und/oder kein Stern (*) angezeigt.</p>				
Absolute Mode for SCAN/CHANGE	<p>Diese Option gilt, wenn das Editor-Kommando SCAN oder CHANGE eingegeben wird.</p> <hr/> <table border="1" data-bbox="425 596 1378 806"> <tr> <td data-bbox="425 596 808 680">Y</td> <td data-bbox="808 596 1378 680">Entspricht dem Editor-Kommando SET ABS ON.</td> </tr> <tr> <td data-bbox="425 680 808 806">N</td> <td data-bbox="808 680 1378 806">Entspricht dem Editor-Kommando SET ABS OFF.</td> </tr> </table> <hr/> <p>Siehe <i>Editor-Kommandos</i> in den Abschnitten Programm-Editor und Data Area Editor.</p>	Y	Entspricht dem Editor-Kommando SET ABS ON.	N	Entspricht dem Editor-Kommando SET ABS OFF.
Y	Entspricht dem Editor-Kommando SET ABS ON.				
N	Entspricht dem Editor-Kommando SET ABS OFF.				
Range Mode for SCAN/CHANGE	<p>Diese Option gilt, wenn das Editor-Kommando SCAN oder CHANGE eingegeben wird.</p> <hr/> <table border="1" data-bbox="425 1064 1378 1274"> <tr> <td data-bbox="425 1064 808 1148">Y</td> <td data-bbox="808 1064 1378 1148">Entspricht dem Editor-Kommando SET RANGE ON.</td> </tr> <tr> <td data-bbox="425 1148 808 1274">N</td> <td data-bbox="808 1148 1378 1274">Entspricht dem Editor-Kommando SET RANGE OFF.</td> </tr> </table> <hr/> <p>Siehe Editor-Kommandos im Abschnitt <i>Programm-Editor</i>.</p>	Y	Entspricht dem Editor-Kommando SET RANGE ON.	N	Entspricht dem Editor-Kommando SET RANGE OFF.
Y	Entspricht dem Editor-Kommando SET RANGE ON.				
N	Entspricht dem Editor-Kommando SET RANGE OFF.				
Direction Indicator	<p>Die Richtungsanzeige zeigt die Richtung (+ oder -) an, in der bestimmte Editor-Kommandos funktionieren sollen. Siehe auch <i>Editor-Kommando Line</i> in den Abschnitten Programm-Editor und Data Area Editor.</p>				
Numeric Input as Line Number	<p>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.</p> <hr/> <table border="1" data-bbox="425 1646 1378 1793"> <tr> <td data-bbox="425 1646 808 1793">Y</td> <td data-bbox="808 1646 1378 1793"> <p>Eine numerische Eingabe wird als Quellcodezeilennummer übernommen.</p> <p>Entspricht dem Editor-Kommando SET SEQ OFF.</p> </td> </tr> </table>	Y	<p>Eine numerische Eingabe wird als Quellcodezeilennummer übernommen.</p> <p>Entspricht dem Editor-Kommando SET SEQ OFF.</p>		
Y	<p>Eine numerische Eingabe wird als Quellcodezeilennummer übernommen.</p> <p>Entspricht dem Editor-Kommando SET SEQ OFF.</p>				

Option	Erläuterung	
	N	Eine numerische Eingabe wird als Text übernommen. Entspricht dem Editor-Kommando SET SEQ ON.
Siehe Editor-Kommandos im Abschnitt <i>Programm-Editor</i> .		

Allgemeine Standardwerte - General Defaults

Option	Erläuterung	
Editing in Lower Case	Y	Kleinbuchstaben im Quellcode werden <i>nicht</i> automatisch in Großbuchstaben umgesetzt.
	N	Kleinbuchstaben im Quellcode werden automatisch in Großbuchstaben umgesetzt. Die automatische Umsetzung erfolgt standardmäßig.
Vorsicht: Benutzen Sie <i>nicht</i> 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: <ul style="list-style-type: none"> ■ Textzeichenketten, die durch Apostrophe eingeschlossen sind, ■ Textzeichenketten, die durch Ausrufungszeichen eingeschlossen sind, ■ Kommentare. 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.

Option	Erläuterung	
	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 <i>Editor-Kommandos</i> in den Abschnitten <i>Programm-Editor</i> und <i>Data Area Editor</i> .	
Prompt Window for Exit Function	Y	Wenn Sie das Kommando EXIT in der Editor-Kommandozeile absetzen, wird ein Bestätigungsfenster angezeigt, (siehe auch <i>Exit-Funktion</i> in den Abschnitten <i>Programm-Editor</i> und <i>Data Area Editor</i>).
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 <i>Sperrmechanismen für Quellcode-Objekte</i> .	

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-Profiles 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-Profiles. Es betrifft <i>nicht</i> 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 <i>nicht</i> die Profil-Wartungsfunktion.
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 .

Kommando	Entsprechende PF-Taste	Beschreibung
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	<p>Speichern und Verlassen:</p> <p>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.</p> <p>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.</p>
Exit without Saving	<p>Verlassen ohne Speichern:</p>

Funktion	Erläuterung
	<p>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 <i>nicht</i> in der Datenbank gespeichert.</p> <p>Das Verlassen ohne Speichern ist gleichbedeutend mit dem Drücken von Enter.</p>
Resume Function	<p>Funktion fortsetzen:</p> <p>Das Aufforderungsfenster schließt sich. Sie können mit der Bearbeitung der aktuellen Wartungsfunktion fortfahren.</p>

III

Programm-Editor

4 Programm-Editor

- Programm-Editor aufrufen 32
- Editor-Kommandozeile 33
- Obere Informationszeile 34
- Editierbereich 34
- Untere Informationszeile 36
- Editor-Kommandos 36
- Editor-Kommandos zum Positionieren 44
- Zeilenkommandos 45
- Spezielle PF-Tastenfunktionen 48
- Cursorsensitive Kommandos 52
- Quellcode speichern und katalogisieren 53
- Exit-Funktion 54

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.

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

eingetragen 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 Quellcode-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 <code>ADD</code> und <code>SCAN</code> und der Zeilenkommandos <code>.C</code> , <code>.I</code> und <code>.M</code> zu steuern. Das Pluszeichen (+) steht für <i>nach</i> und das Minuszeichen (-) für <i>vor</i> . Die genaue Interpretation können Sie der Beschreibung des betreffenden Kommandos entnehmen. Siehe auch die Editor-Profil-Option Direction Indicator im Abschnitt <i>Editor-Profil</i> .
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 <code>SET TYPE</code> 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 Quellcode-Objekt gespeichert worden ist. Der Stern (*) ist nur dann sichtbar, wenn die Editor-Profil-Option Source Status Message auf <code>Y</code> gesetzt ist (siehe <i>Editor-Profil</i>). Der Stern (*) verschwindet, wenn Sie den Quellcode erfolgreich speichern und katalogisieren, indem Sie ein <code>SAVE</code> - oder <code>STOW</code> -Kommando ausführen. Siehe auch <i>Exit-Funktion</i> .
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

	Quellcode-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-Objektyps 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 Quellcodes. 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 γ 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 γ 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
<u>ADD</u> [(n)]	<p>Hängt <i>n</i> 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.</p> <p>Der Wert für <i>n</i> darf im Bereich von 1 bis 9 liegen. Wenn Sie für <i>n</i> keinen oder einen nicht korrekten Wert angeben, werden standardmäßig 9 Zeilen (4 Zeilen im Split-Screen-Modus) hinzugefügt.</p> <p>Wenn Sie das nächste Mal Enter drücken, werden alle Zeilen, die dann noch leer sind, wieder entfernt.</p>
CANCEL oder . (Punkt)	<p>Die Editor-Sitzung wird abgebrochen. Alle seit dem letzten Speichern (SAVE) vorgenommenen Änderungen werden <i>nicht</i> gespeichert.</p>
<u>CATALOG</u> [object-name]	<p>Führt das Systemkommando CATALOG aus. Der aktuelle Quellcode wird geprüft und katalogisiert.</p> <p>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.</p>
<u>CHANGE</u> ['scan-value' replace-value]	<p>Durchsucht den Quellcode nach der als Suchwert (<i>scan-value</i>) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (<i>replace-value</i>) eingegebenen Zeichenkette.</p> <p>Jede Zeile, in der eine Zeichenkette ersetzt wurde, wird am linken Zeilenrand mit einem R markiert.</p> <p>Als Eingabebegrenzungszeichen kann jedes Sonderzeichen benutzt werden, dass in einem Natural-Variablennamen nicht gültig ist.</p> <p>Wenn Sie CHANGE ohne Parameter eingeben, wird das SCAN/REPLACE-Fenster angezeigt. Zusätzlich zum Suchwert (<i>scan-value</i>) und zum Ersetzungswert (<i>replace-value</i>) können Sie in diesem Fenster Folgendes angeben:</p>

Editor-Kommando	Funktion
	<ul style="list-style-type: none"> ■ die Nullwert-Option (siehe SET NUL), ■ den absoluten Suchmodus (siehe SET ABS), ■ den Bereichsmodus (siehe SET RANGE).
CHECK	<p>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.</p>
CLEAR	<p>Führt das Systemkommando CLEAR aus, das den Quellcode-Arbeitsbereich (einschließlich des Objektnamens und der Markierungen X und Y) löscht.</p>
DX oder DY	<p>Löscht die mit X oder Y markierte Zeile.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
DX - Y	<p>Löscht den mit den Markierungen X und Y eingegrenzten Zeilenblock.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EX oder EY	<p>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.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EX - Y	<p>Löscht alle Zeilen im Editierbereich mit Ausnahme des Blocks, der mit X und Y eingegrenzt ist.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EXIT	<p>Die Editor-Sitzung wird beendet. Alle am Quellcode vorgenommenen Änderungen werden in Abhängigkeit von der Einstellung im Editor-Profil gespeichert. Siehe Exit-Funktion.</p>
LET	<p>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.</p>
N [(nnnn)]	<p>Dieses Kommando entspricht dem Systemkommando RENUMBER. Es bewirkt, dass die Quellcode-Zeilen des Objekts, das sich gerade im Quellcode-Arbeitsbereich befindet, neu durchnummeriert werden.</p> <p>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.</p>

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

Editor-Kommando	Funktion	
	<p>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.</p> <p>Wenn Sie das SCAN-Kommando ohne Parameter eingeben, wird das SCAN/REPLACE-Fenster angezeigt. Sie können dann zusätzlich zum Suchwert (<i>scan-value</i>) Folgendes eingeben:</p> <ul style="list-style-type: none"> ■ einen Ersetzungswert (<i>replace-value</i>, siehe auch CHANGE), ■ die Nullwert-Option (siehe SET NUL), ■ den absoluten Suchmodus (siehe SET ABS), ■ den Bereichsmodus (siehe SET RANGE). <p>SCAN ist ein cursorsensitives Kommando, das zusätzliche Optionen bietet, siehe Abschnitt Cursorsensitive Kommandos.</p>	
SCAN=[+ -]	<p>Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts (<i>scan-value</i>).</p> <p>Die Suchrichtung für den gegebenen Suchvorgang können Sie explizit angeben, indem Sie SCAN=+ oder SCAN=- eingeben. Die Einstellung in der Richtungsanzeige wird dann ignoriert.</p> <p>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 <i>Parameter-Referenz-Dokumentation</i>), muss dieses Zeichen stattdessen benutzt werden.</p>	
SET ABS [ON OFF]	ON	<p>Die Kommandos SCAN und CHANGE arbeiten im absoluten Modus. Das bedeutet, dass es nicht nötig ist, den Suchwert (<i>scan-value</i>) und den Ersetzungswert (<i>replace-value</i>) durch Leerzeichen oder Sonderzeichen abzugrenzen.</p>
	OFF	<p>Die Kommandos SCAN und CHANGE arbeiten nicht im absoluten Modus. Das bedeutet, dass Sie den Suchwert (<i>scan-value</i>) und den Ersetzungswert (<i>replace-value</i>) durch Leerzeichen oder Sonderzeichen abgrenzen müssen.</p>
	Der Standardwert ist OFF.	

Editor-Kommando	Funktion	
	Das Kommando SET ABS entspricht der Editor-Profil-Option Absolute Mode for SCAN/CHANGE . Siehe Abschnitt <i>Editor-Profil</i> .	
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 <i>Editor-Profil</i> .	
	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.
	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 <i>character</i>	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

Editor-Kommando	Funktion	
		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.
	<p>Der Standardwert ist OFF.</p> <p>Das Kommando SET RANGE entspricht der Editor-Profil-Option Range Mode for SCAN/CHANGE. Siehe <i>Editor-Profil</i>.</p>	
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.
	<p>Die Standard-Einstellung ist ON.</p> <p>Dieses Kommando entspricht der Editor-Profil-Option Empty Line Suppression. Siehe Abschnitt <i>Editor-Profil</i>.</p>	
SET SEQ [ON OFF]	OFF	<p>Bei Eingabe eines numerischen Wertes werden die ersten vier Stellen im Editierbereich als die Zeilennummern betrachtet und werden in die Zeilennummer-Position verschoben, sobald Sie Enter drücken.</p> <p>Diese Funktion ist nützlich, wenn zum Beispiel eine Statement-Zeile von einer Quellcode-Zeilenummer in einer anderen Statement-Zeile referenziert werden soll. Wenn Sie den Quellcode neu nummerieren, wird die referenzierende Zeilennummer ebenfalls neu nummeriert.</p>
	ON	Die ersten vier Stellen einer numerischen Werteingabe bleiben so, wie sie eingegeben wurden.
	<p>Außer beim Objekttyp Text ist der Standardwert OFF.</p> <p>Das Kommando SET SEQ entspricht der Editor-Profil-Option Numeric Input as Line Number. Siehe <i>Editor-Profil</i>.</p>	
SET SIZE [ON OFF]	ON	Die Größe des Quellcodes wird in der unteren Informationszeile des Editor-Bildschirms und der

Editor-Kommando	Funktion	
		Programmiermodus wird in der Lineal-Zeile angezeigt.
	OFF	Diese Informationen werden nicht angezeigt.
	<p>Der Standardwert ist OFF.</p> <p>Das Kommando SET SIZE entspricht der Editor-Profil-Option Source Size Information. Siehe Editor-Profil.</p>	
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.
	<p>Der Standardwert ist OFF.</p> <p>Das Kommando SET STAY entspricht der Editor-Profil-Option Stay on Current Screen. Siehe Editor-Profil.</p>	
SET TYPE <i>object-type</i>	<p>Mit diesem Kommando können Sie den (in der Editor-Kommandozeile angezeigten) Objekttyp für die zurzeit im Quellcode-Arbeitsbereich befindliche Source ändern.</p> <p><i>object-type</i> steht für einen Objekttyp, der mit dem Kommando angegeben werden muss. Folgende Werte sind möglich:</p>	
	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]	<p>Dieses Kommando verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, nach links oder nach rechts. Der Parameter <i>nn</i> gibt die Anzahl der Zeichen an, um die die Zeile verschoben werden soll. Kommentarzeilen werden nicht verschoben.</p>	
SHIFT --	<p>Dieses Kommando verschiebt jede Zeile, die mit den Markierungen X und Y begrenzt ist, ganz nach links. Kommentarzeilen werden nicht verschoben.</p>	

Editor-Kommando	Funktion
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 [<i>parameter</i>]	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 dem zusätzliche Optionen zur Verfügung stehen. Siehe Cursorsensitive Kommandos . Wenn SPLIT ohne Parameter eingegeben wird, wird der Quellcode des zuvor im Split-Screen-Bereich angezeigten Objekts erneut angezeigt. Dies gilt für die Dauer der aktuellen Editor-Sitzung.
STOW [<i>object-name</i>]	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 .
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.
+ <i>nnnn</i>	Um <i>nnnn</i> (maximal 4 Ziffern) Zeilen vorwärts positionieren.
- <i>nnnn</i>	Um <i>nnnn</i> (maximal 4 Ziffern) Zeilen rückwärts positionieren.
<i>nnnn</i>	Auf Zeile Nummer <i>nnnn</i> 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 <i>.N</i> eingegeben wurde. Siehe auch Zeilenkommando <i>.P</i> .

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 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.

Zeilenkommandos	Funktion
	<p>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.</p> <p>Handelt es sich bei dem Objekt um einen Adapter, wird in die aktuelle Zeile automatisch ein PROCESS PAGE USING-Statement (siehe <i>Syntax 2 - PROCESS PAGE USING</i> in der <i>Statements</i>-Dokumentation) mit allen definierten Variablen eingefügt.</p>
.J	<p>Führt die aktuelle Zeile mit der nächsten Zeile zusammen.</p> <p>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.</p>
.L	<p>Macht alle seit dem letzten Drücken der Enter-Taste an der Zeile vorgenommenen Änderungen rückgängig.</p>
.MX oder .MY	<p>Verschiebt die mit X oder mit Y markierte Zeile. Siehe auch Zeilenkommandos .X und .Y sowie Anmerkungen zu den Zeilenkommandos.</p>
.MX-Y	<p>Verschiebt den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist (siehe auch Anmerkungen zu den Zeilenkommandos).</p>
.N	<p>Markiert (unsichtbar) eine Zeile, die mit den Editor-Kommando POINT an den Anfang des Quellcode-Arbeitsbereichs positioniert werden soll.</p> <p>Die Markierung wird automatisch gelöscht, wenn ein Fehler bei einem Zeilen- oder Editor-Kommando auftritt oder wenn das RESET-Kommando ausgeführt wird.</p>
.P	<p>Positioniert die mit diesem Kommando markierte Zeile an den Anfang des Bildschirms.</p>
.S	<p>Spaltet die Zeile an der mit dem Cursor markierten Position.</p>
.X	<p>Markiert eine Zeile mit einem X (siehe auch Anmerkungen zu den Zeilenkommandos).</p>
.Y	<p>Markiert eine Zeile mit einem Y (siehe auch Anmerkungen zu den Zeilenkommandos).</p>

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 <i>Editor-Profil</i>). 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 <i>Beispiel für *CURSOR auf PF-Taste</i> .
*X oder *Y	Wird einer PF-Taste der Wert *X oder *Y im Editor-Profil zugewiesen (siehe PF- und PA-Tasten in <i>Editor-Profil</i>), 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 <i>Beispiel für *X und *Y auf PF-Tasten</i> .

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 Quellcode-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 Quellcode-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 <i>nicht</i> gespeichert.
Resume Funktion	Die Editor-Sitzung wird <i>nicht</i> 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.

IV

Datenbereich-Editor (Data Area Editor)

5

Datenbereich-Editor (Data Area Editor)

▪ Data Area Editor aufrufen	60
▪ Obere Informationszeile	62
▪ Editor-Kommandozeile	63
▪ Untere Informationszeile	64
▪ Editierbereich benutzen	64
▪ Spalten im Editierbereich	65
▪ Erweiterte Felddefinition bearbeiten	70
▪ Zeilenkommandos	75
▪ Editor-Kommandos	81
▪ Editor-Kommandos zum Positionieren	86
▪ Data Area speichern und katalogisieren	87
▪ User Exit für den Data Area Editor	88
▪ Exit-Funktion	89

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.

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 Quellcode-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)
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:

<i>Data-Area-Typ</i>	<p>Zeigt den Typ der Data Area an, die sich zurzeit im Editor-Arbeitsbereich befindet: Local, Global oder Parameter.</p> <p>Sie können den Typ mit dem Editor-Kommando SET TYPE ändern.</p>
<p>Änderungsanzeige:</p> <p>*</p>	<p>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 Quellcode-Objekt gespeichert worden ist.</p> <p>Der Stern (*) ist nur dann vorhanden, wenn die Editor-Profil-Option Source Status Message auf Y gesetzt ist (siehe <i>Editor-Profil</i>).</p> <p>Der Stern (*) verschwindet, nachdem Sie auf dem Quellcode das Kommando SAVE oder STOW erfolgreich ausgeführt haben.</p> <p>Siehe auch <i>Exit-Funktion</i>.</p>

<i>Data Area Name</i>	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 <code>SAVE</code> , <code>CATALOG</code> oder <code>STOW</code> als Quellcode-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:		
	<table border="1"> <tr> <td data-bbox="344 380 649 510">+</td> <td data-bbox="649 380 1385 510">Zeigt an, dass zu der Variablen oder dem Feld mehr als einer der weiter unten aufgeführten Einträge vorhanden sind.</td> </tr> </table>	+	Zeigt an, dass zu der Variablen oder dem Feld mehr als einer der weiter unten aufgeführten Einträge vorhanden sind.
	+	Zeigt an, dass zu der Variablen oder dem Feld mehr als einer der weiter unten aufgeführten Einträge vorhanden sind.	
	<table border="1"> <tr> <td data-bbox="344 510 649 640">E</td> <td data-bbox="649 510 1385 640">Zeigt an, dass ein Definitionsfehler festgestellt wurde.</td> </tr> </table>	E	Zeigt an, dass ein Definitionsfehler festgestellt wurde.
	E	Zeigt an, dass ein Definitionsfehler festgestellt wurde.	
	<table border="1"> <tr> <td data-bbox="344 640 649 770">A</td> <td data-bbox="649 640 1385 770">Zeigt an, dass Array-Grenzen mit dem Zeilenkommando <code>.E</code> definiert worden sind.</td> </tr> </table>	A	Zeigt an, dass Array-Grenzen mit dem Zeilenkommando <code>.E</code> definiert worden sind.
	A	Zeigt an, dass Array-Grenzen mit dem Zeilenkommando <code>.E</code> definiert worden sind.	
	<table border="1"> <tr> <td data-bbox="344 770 649 900">I</td> <td data-bbox="649 770 1385 900">Zeigt an, dass ein Anfangswert mit dem Zeilenkommando <code>.E</code> definiert worden ist.</td> </tr> </table>	I	Zeigt an, dass ein Anfangswert mit dem Zeilenkommando <code>.E</code> definiert worden ist.
	I	Zeigt an, dass ein Anfangswert mit dem Zeilenkommando <code>.E</code> definiert worden ist.	
	<table border="1"> <tr> <td data-bbox="344 900 649 1031">M</td> <td data-bbox="649 900 1385 1031">Zeigt an, dass eine Editiermaske und/oder eine Überschrift mit dem Zeilenkommando <code>.E</code> definiert worden ist.</td> </tr> </table>	M	Zeigt an, dass eine Editiermaske und/oder eine Überschrift mit dem Zeilenkommando <code>.E</code> definiert worden ist.
	M	Zeigt an, dass eine Editiermaske und/oder eine Überschrift mit dem Zeilenkommando <code>.E</code> definiert worden ist.	
	<table border="1"> <tr> <td data-bbox="344 1031 649 1161">S</td> <td data-bbox="649 1031 1385 1161">Zeigt an, dass sowohl ein Anfangswert als auch eine Editiermaske mit dem Zeilenkommando <code>.E</code> definiert worden sind.</td> </tr> </table>	S	Zeigt an, dass sowohl ein Anfangswert als auch eine Editiermaske mit dem Zeilenkommando <code>.E</code> definiert worden sind.
	S	Zeigt an, dass sowohl ein Anfangswert als auch eine Editiermaske mit dem Zeilenkommando <code>.E</code> definiert worden sind.	
	Folgende Anzeigen treffen nur bei PDAs zu:		
	<table border="1"> <tr> <td data-bbox="344 1228 649 1358"><i>blank</i></td> <td data-bbox="649 1228 1385 1358">Zeigt die Parameterangabe Call-by-Reference an (Standardwert).</td> </tr> </table>	<i>blank</i>	Zeigt die Parameterangabe Call-by-Reference an (Standardwert).
	<i>blank</i>	Zeigt die Parameterangabe Call-by-Reference an (Standardwert).	
	<table border="1"> <tr> <td data-bbox="344 1358 649 1446">V</td> <td data-bbox="649 1358 1385 1446">Zeigt die Parameterangabe Call-by-Value an.</td> </tr> </table>	V	Zeigt die Parameterangabe Call-by-Value an.
V	Zeigt die Parameterangabe Call-by-Value an.		
<table border="1"> <tr> <td data-bbox="344 1446 649 1535">R</td> <td data-bbox="649 1446 1385 1535">Zeigt die Parameterangabe Call-by-Value-Result an.</td> </tr> </table>	R	Zeigt die Parameterangabe Call-by-Value-Result an.	
R	Zeigt die Parameterangabe Call-by-Value-Result an.		
<table border="1"> <tr> <td data-bbox="344 1535 649 1665">0</td> <td data-bbox="649 1535 1385 1665">Zeigt einen optionalen Parameter an, der übergeben werden <i>kann</i>.</td> </tr> </table>	0	Zeigt einen optionalen Parameter an, der übergeben werden <i>kann</i> .	
0	Zeigt einen optionalen Parameter an, der übergeben werden <i>kann</i> .		
Weitere Informationen siehe Funktionscode P in <i>Erweiterte Felddefinition bearbeiten</i> .			
T	Typ der Variablen oder des Feldes. Mögliche Typen:		

Spaltenüberschrift	Erklärung
B	Ein Datenblock innerhalb einer GDA.
C	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 <i>CONSTANT-Option</i> in der <i>Statements-Dokumentation</i> und <i>Internen Zähler für ein Datenbank-Array referenzieren</i> – <i>C*-Notation im Leitfaden zur Programmierung</i> .
G	Eine Gruppe in einer View (DDM).
M	Ein multiples Feld in einer View (DDM).
O	Handle eines Objekts.
P	Eine Periodengruppe in einer View (DDM).
R	Die Redefinition einer Variablen oder eines Feldes.
U	Der Globally Unique Identifier (GUID).
V	Gilt nicht bei PDAs. Eine View-Definition, die aus einer DDM erstellt wurde.
<i>blank</i>	Eine Benutzervariable oder ein benutzerdefiniertes Feld oder eine Gruppenstruktur (nicht in einer View).
*	Ein Kommentarfeld.
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 <i>Namenskonventionen für Benutzervariablen</i> in der Dokumentation <i>Natural benutzen</i> . Informationen zu Benutzerkonstante siehe <i>CONSTANT</i> in der <i>Statements-Dokumentation</i> .

Spaltenüberschrift	Erklärung
	Anstatt der Vergabe eines Variablennamen kann auch die Füllbyte-Notation $n\chi$ 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 <i>Format und Länge von Benutzervariablen</i> und <i>Spezielle Formate im Leitfaden zur Programmierung</i> . 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 <i>Format und Länge von Benutzervariablen im Leitfaden zur Programmierung</i> . 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 <i>Spalte "Miscellaneous" (Verschiedenes) benutzen</i> 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 angeben, 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 übergeordneter „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	<p>Definiert einen Anfangswert für die angegebene Variable oder das angegebene Feld im Einzelwertmodus (Single 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.</p> <p>Beispiel: Aus einem Anfangswert F1F2 für eine binäre Variable (B2) erzeugt der Editor INIT <H'F1F2'>.</p> <p>Handelt es sich bei der Variablen oder dem Feld um ein Array, kann ein Ausgangswert für jede einzelne Ausprägung definiert werden (muss aber nicht).</p> <p>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.</p> <p>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.</p> <p>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 <i>Parameter-Referenz-Dokumentation</i>.</p> <p>Um anstelle eines Anfangswertes einen Konstanten-Wert zu definieren, müssen Sie im Feld Define as CONSTANT (Y/N) ein Y eingeben.</p>
F	<p>Definiert einen Anfangswert für das angegebene Feld im Free-Modus (Free 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.</p> <p>Ausführliche Informationen zu Ausgangswerten siehe <i>Definition eines Ausgangswerts Ausgangswerte/Konstanten-Werte für ein Array</i> in der <i>Statements -Dokumentation</i>.</p> <p>Siehe auch Beispiele unter Ausgangswert.</p>
E	<p>Definiert eine Editiermaske und/oder eine Überschrift für das angegebene Feld gemäß den Natural-Regeln für die Angabe von Editiermasken und Überschriften.</p> <p>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.</p> <p>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>
P	<p>Diese Funktion gilt nur bei PDAs.</p> <p>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</p>

Funktionscode	Funktion										
	<p>Objekt (z.B. Subprogramm) übergeben werden soll. Im oberen Eingabefeld können Sie einen der folgenden Code-Werte eingeben:</p> <table border="1" data-bbox="310 365 1385 506"> <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> <p>Weitere Informationen siehe BY VALUE und BY VALUE RESULT in der Beschreibung des DEFINE DATA-Statement in <i>Definition von Parameterdaten</i> und operand2 in der Beschreibung des CALLNAT-Statement in der <i>Statements-Dokumentation</i>.</p> <p>Zusätzlich können Sie im unteren Eingabefeld angeben, ob ein Parameter übergeben werden muss oder kann:</p> <table border="1" data-bbox="310 793 1385 884"> <tr> <td>N</td> <td>Ein Parameter muss übergeben werden (Standardwert).</td> </tr> <tr> <td>Y</td> <td>Ein optionaler Parameter <i>kann</i> übergeben werden.</td> </tr> </table> <p>Weitere Informationen siehe entsprechende Option OPTIONAL in der Beschreibung des DEFINE DATA-Statement in <i>Definition von Parameterdaten</i> und operand2 in der Beschreibung des CALLNAT-Statement in der <i>Statements-Dokumentation</i>.</p>	D	Call-by-Reference (Standardwert).	V	Call-by-Value.	R	Call-by-Value-Result.	N	Ein Parameter muss übergeben werden (Standardwert).	Y	Ein optionaler Parameter <i>kann</i> übergeben werden.
D	Call-by-Reference (Standardwert).										
V	Call-by-Value.										
R	Call-by-Value-Result.										
N	Ein Parameter muss übergeben werden (Standardwert).										
Y	Ein optionaler Parameter <i>kann</i> übergeben werden.										
A	<p>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).</p>										
D	<p>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.</p> <p>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.</p>										

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 <i>Anmerkungen zu den Zeilenkommandos</i> .
.CX[(nnnn)] or .CY[(nnnn)]	Kopiert die Zeile, die mit X oder mit Y markiert ist. Siehe auch Zeilenkommandos .X und .Y sowie <i>Anmerkungen zu den Zeilenkommandos</i> .
.CX-Y[(nnnn)]	Kopiert den Zeilenblock, der mit den Markierungen X und Y eingegrenzt ist. Siehe auch Zeilenkommandos .X und .Y sowie <i>Anmerkungen zu den Zeilenkommandos</i> .

Kommando	Funktion
.D	<p>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).</p> <p>Wird das Zeilenkommando bei einem einzelnen Feld eingegeben wird, dann wird nur dieses Feld gelöscht.</p> <p>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.</p>
.D(<i>nnnn</i>)	<p>Löscht <i>nnnn</i> 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(<i>nnnn</i>) nur die angegebene Anzahl an Zeilen, und zwar unabhängig von einer hierarchischen Struktur.</p>
.E	<p>Ruft den Bildschirm Extended Field Definition Editing auf, der dazu dient, Array-Grenzen, Anfangswerte, Editiermasken, Überschriften und Parameter-Attribute zu definieren.</p> <p>Weitere Informationen siehe Erweiterte Felddefinition bearbeiten.</p>
.F(<i>file-name</i>)	<p>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.</p>
.I[(<i>n</i>)]	<p>Fügt <i>n</i> Leerzeilen hinzu. Dabei kann <i>n</i> im Bereich von 1 bis 9 liegen. Wird <i>n</i> nicht (oder nicht korrekt) angegeben, werden standardmäßig 10 Zeilen (5 Zeilen im Split-Screen-Modus) hinzugefügt.</p> <p>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.</p> <p>Anmerkung: Es kein nur jeweils ein .I-Kommando ausgeführt. Siehe auch Anmerkungen zu den Zeilenkommandos.</p>
.I(<i>obj</i>)	<p>Kopiert Variablen- oder Parameter-Definitionen aus einem anderen Natural-Objekt folgenden Typs:</p> <ul style="list-style-type: none"> Data Area Programm Subprogramm Subroutine

Kommando	Funktion
	<p>Helproutine Map Function</p> <p>Wenn das bei <i>obj</i> 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:</p> <ul style="list-style-type: none"> ■ 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 enthalten sind (einschließlich der, die aus PDAs übernommen wurden). ■ Nur diejenigen Parameter, die in dem angegebenen Objekt enthalten sind. <p>Außerdem können Folgendes wählen:</p> <ul style="list-style-type: none"> ■ 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. <p>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.</p> <p>Wenn Sie Variablen-Definitionen aus Objekten übernehmen, die keine <code>DEFINE DATA</code>-Statement-Definition enthalten (d.h., aus Objekten, die im Reporting Mode kodiert wurden), kann es vorkommen, dass Variablen-Redefinitionen (siehe <code>REDEFINE</code>-Statement in der <i>Statements</i>-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.</p> <p>Falls bei einer Variablen-Redefinition mehr als eine Variable resultieren, wird jede Variable als einzelne, eigene Redefinition eingebaut, wobei gegebenenfalls Füllbytes verwendet werden.</p> <p>Falls das angegebene Objekt mit dem Natural Optimizer Compiler katalogisiert worden ist, können Anfangswerte und Konstanten nicht übernommen werden.</p>

Kommando	Funktion
	<p>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.</p> <p>Siehe auch Anmerkungen zu den Zeilenkommandos.</p>
<p><code>.I(obj, ssss, nnnn)</code></p>	<p>Fügt eine GDA, eine LDA oder eine PDA ein. Dieses Merkmal wird nur bei Data Areas unterstützt, die keine Anfangswerte oder Editiermasken enthalten.</p> <p>Der Eintrag <code>ssss</code> kann benutzt werden, um anzugeben, bei welcher Zeile die Einfügung beginnen soll. Wenn Sie beispielsweise <code>ssss</code> auf 20 setzen, beginnt die Einfügung bei der 20. Zeile der Data Area.</p> <p>Der Eintrag <code>ssss</code> kann benutzt werden, um die Anzahl der einzufügenden Zeilen anzugeben.</p> <p>Wird <code>ssss</code> und/oder <code>nnnn</code> bei einem Objekt angegeben, bei dem es sich nicht um eine Data Area handelt (siehe Kommando <code>.I(obj)</code>), wird/werden der/die angegebene(n) Wert(e) nicht beachtet.</p> <p>Siehe auch Anmerkungen zu den Zeilenkommandos.</p>
<p><code>.L</code></p>	<p>Macht alle seit dem letzten Drücken der Enter-Taste an der Zeile vorgenommenen Änderungen rückgängig.</p>
<p><code>.MX</code> or <code>.MY</code></p>	<p>Verschiebt die mit <code>X</code> oder mit <code>Y</code> markierte Zeile.</p> <p>Siehe auch Zeilenkommandos <code>.X</code> und <code>.Y</code> und Anmerkungen zu den Zeilenkommandos.</p>
<p><code>.MX-Y</code></p>	<p>Verschiebt den Zeilenblock, der mit den Markierungen <code>X</code> und <code>Y</code> eingegrenzt ist.</p> <p>Siehe auch Zeilenkommandos <code>.X</code> und <code>.Y</code> und Anmerkungen zu den Zeilenkommandos.</p>
<p><code>.N</code></p>	<p>Markiert (unsichtbar) eine Zeile, die mit den Editor-Kommando <code>POINT</code> an den Anfang des Quellcode-Arbeitsbereichs positioniert werden soll. Siehe <i>Editor-Kommandos zum Positionieren</i>.</p> <p>Die Markierung wird automatisch gelöscht, wenn ein Fehler bei einem Zeilen- oder Editor-Kommando auftritt oder wenn das <code>RESET</code>-Kommando ausgeführt wird.</p>
<p><code>.P</code></p>	<p>Positioniert die mit diesem Kommando markierte Zeile an den Anfang des Bildschirms.</p>
<p><code>.R</code></p>	<p>Redefiniert eine Variable oder ein Feld als eine einzelne Variable oder eine Variablen-Gruppe.</p> <p>Mit der optionalen Füllbyte-Notation (<code>nX</code>), <code>n</code> können Sie in der Variablen oder dem Feld, die/das neu definiert wird, Füllbytes bezeichnen.</p> <p>Siehe auch Anmerkungen zu den Zeilenkommandos.</p>

Kommando	Funktion
.V[(ddm-name[,NOFL])]	<p>Gilt nicht bei PDAs.</p> <p>Definiert eine View aus einem DDM.</p> <p>Geben Sie bei <i>ddm-name</i> 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.</p> <p>Im Split-Screen-Modus wird das DDM, das sich zurzeit im geteilten Bildschirm befindet, im Editierbereich angezeigt, wenn Sie .V ohne <i>ddm-name</i> eingeben.</p> <p>Wird .V(<i>ddm-name</i>) in einer View angegeben, die denselben Namen hat wie bei <i>ddm-name</i> angegeben, werden die gewählten Felder in diese View eingefügt, und es wird keine neue View definiert.</p> <p>Wird NOFL angegeben, werden die gewählten Felder ohne Format/Längen-Angaben eingefügt.</p> <p>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.</p> <p>Wenn Predict aktiv ist, werden auch Redefinitionen und Kommentare eingefügt.</p> <p>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 <i>Natural for VSAM-Dokumentation</i>.</p>
.VG	<p>Gilt nur bei Views.</p> <p>Konvertiert alle Felder einer View-Definition in Felder einer Gruppenstruktur.</p> <p>Geben Sie .VG neben der Zeile ein, die in der Spalte T ein V enthält.</p> <p>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.</p> <p>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.</p>

Kommando	Funktion
	Anmerkung: <code>.VG</code> 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.
<code>.X</code>	Gilt nicht bei Periodengruppen, multiplen Feldern oder View-Definitionen. Markiert eine Zeile mit einem X. Siehe auch Anmerkungen zu den Zeilenkommandos .
<code>.Y</code>	Gilt nicht bei Views, Periodengruppen, multiplen Feldern oder Redefinitionen. Markiert eine Zeile mit einem Y. Siehe auch Anmerkungen zu den Zeilenkommandos .
<code>.*</code>	Generiert ein Zähler-Feld (C*-Variable) für multiple Felder oder Felder in einer Periodengruppe. Siehe auch Anmerkungen zu den Zeilenkommandos .
<code>number [(nnn[,m])]</code>	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 <code>number</code> in einer View eingegeben wird) oder als Benutzervariablen eingefügt werden. Wenn das gewählte Feld denselben Namen hat wie das Feld, bei dem das Kommando eingegeben wurde, dann wird es nicht eingefügt, sondern ersetzt. Wenn Sie die Notation <code>nnn</code> benutzen, können Sie mehrere Zeilen aus dem geteilten Bildschirm übernehmen. Dabei steht <code>nnn</code> für die Anzahl der einzufügenden Zeilen. Die Notation <code>m</code> 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[(<i>n</i>)]	<p>Fügt <i>n</i> 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.</p> <p>Der Wert für <i>n</i> darf im Bereich von 1 bis 9 liegen. Wenn Sie für <i>n</i> keinen oder einen nicht korrekten Wert angeben, werden standardmäßig 9 Zeilen (4 Zeilen im Split-Screen-Modus) hinzugefügt.</p> <p>Wenn Sie das nächste Mal Enter drücken, werden alle Zeilen, die dann noch leer sind, wieder entfernt.</p>
CANCEL oder . (Punkt)	Die Editor-Sitzung wird abgebrochen. Alle seit dem letzten Speichern (SAVE) vorgenommenen Änderungen werden <i>nicht</i> gespeichert.
CATALOG [<i>object-name</i>]	<p>Führt das Systemkommando CATALOG aus. Die aktuelle Data Area-Definition wird geprüft und katalogisiert.</p> <p>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.</p>
CHANGE [' <i>scan-value</i> ' <i>replace-value</i> ']	Durchsucht die Data Area nach der als Suchwert (<i>scan-value</i>) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (<i>replace-value</i>) eingegebenen Zeichenkette. Als Eingabebegrenzungszeichen kann jedes

Editor-Kommando	Funktion
	<p>Sonderzeichen benutzt werden, dass in einem Natural-Variablenamen nicht gültig ist.</p> <p>Jede Zeile, in der eine Zeichenkette ersetzt wurde, wird am linken Zeilenrand mit einem R markiert.</p> <p>Weitere Informationen siehe SCAN-Kommando.</p>
CHECK	<p>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.</p> <p>Bei Ausführung von CHECK werden die Einträge in der Spalte Miscellaneous (Verschiedenes) in folgender Reihenfolge sortiert:</p> <p>Array-Definition Anfangswert Editiermaske, Überschrift und/oder PM - Druck-/Anzeige-Modus Name eines DDM oder eines „Parent“-Blocks Kommentar</p>
CLEAR	<p>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.</p>
DX or DY	<p>Löscht die mit X oder Y markierte Zeile.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
DX - Y	<p>Löscht den mit den Markierungen X und Y eingegrenzten Zeilenblock.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EX or EY	<p>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.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EX - Y	<p>Löscht alle Zeilen im Editierbereich mit Ausnahme des Blocks, der mit X und Y eingegrenzt ist.</p> <p>Siehe auch Zeilenkommandos .X und .Y.</p>
EXIT	<p>Die Editor-Sitzung wird beendet. Alle am Quellcode vorgenommenen Änderungen werden in Abhängigkeit von der Einstellung im Editor-Profil gespeichert. Siehe Exit-Funktion.</p>

Editor-Kommando	Funktion				
GENERATE [<i>object-name</i>]	<p>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.</p> <p>Wird ein Objektname (<i>object-name</i>) angegeben, dann wird der generierte Copycode unter diesem Namen in der aktuellen Library in der aktuellen Systemdatei gespeichert.</p>				
M + -	<p>Blätterkommandos für die Spalte Miscellaneous (Verschiedenes).</p> <table border="1" data-bbox="672 646 1471 743"> <tr> <td data-bbox="672 646 1068 688">+</td> <td data-bbox="1068 646 1471 688">Blättert nach rechts.</td> </tr> <tr> <td data-bbox="672 688 1068 743">-</td> <td data-bbox="1068 688 1471 743">Blättert nach links.</td> </tr> </table>	+	Blättert nach rechts.	-	Blättert nach links.
+	Blättert nach rechts.				
-	Blättert nach links.				
PROFILE [<i>name</i>]	<p>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.</p>				
READ <i>object-name</i>	<p>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 <i>Systemkommandos</i>-Dokumentation.</p>				
RESET	<p>Löscht die aktuellen Zeilenmarkierungen X und Y sowie alle zuvor mit dem Zeilenkommando .N gesetzten Markierungen. Siehe auch Zeilenkommandos .X und .Y.</p>				
SAVE [<i>object-name</i>]	<p>Führt das Systemkommando SAVE aus, das die aktuelle Data Area-Definition speichert.</p> <p>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.</p>				
SCAN <i>scan-value</i>	<p>Durchsucht die Data Area nach einer Zeichenkette (<i>scan-value</i>) in der Spalte Name (standardmäßig) und/oder in der Spalte Miscellaneous des Editor-Bildschirms (abhängig davon, ob zuvor das Kommando SET SCAN ausgeführt wurde).</p> <p>Jede Zeile, in der der Suchwert (<i>scan-value</i>) gefunden wird, wird am linken Zeilenrand mit einem S gekennzeichnet.</p> <p>Die erste Zeile, die den Suchwert (<i>scan-value</i>) enthält wird, abhängig von der aktuellen Einstellung der Richtungsanzeige, in die oberste oder in die unterste Zeile positioniert.</p> <p>Anmerkung: Mit dem Kommando SCAN erfolgt eine exakte Suche nach dem angegebenen Suchwert (<i>scan-value</i>). Dies sollte bei der</p>				

Editor-Kommando	Funktion				
	<p>Suche nach DBCS-Zeichen (DBCS = Double-Byte Character Set) berücksichtigt werden.</p> <p>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.</p>				
SCAN [=+ -]	<p>Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts (<i>scan-value</i>).</p> <p>Die Suchrichtung für den gegebenen Suchvorgang können Sie explizit angeben, indem Sie SCAN=+ oder SCAN=- eingeben. Die Einstellung in der Richtungsanzeige wird dann ignoriert.</p> <p>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 <i>Parameter-Referenz-Dokumentation</i>), muss dieses Zeichen stattdessen benutzt werden.</p>				
SET ABS [ON OFF]	<p>Legt fest, ob das SCAN-Kommando im absoluten oder nicht im absoluten Modus arbeitet.</p> <table border="1" data-bbox="581 1098 1385 1465"> <tr> <td data-bbox="581 1098 854 1283">ON</td> <td data-bbox="854 1098 1385 1283">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.</td> </tr> <tr> <td data-bbox="581 1283 854 1465">OFF</td> <td data-bbox="854 1283 1385 1465">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.</td> </tr> </table> <p>Der Standardwert ist OFF.</p> <p>Das Kommando SET ABS entspricht der Editor-Profil-Option Absolute Mode for SCAN/CHANGE. Siehe Abschnitt <i>Editor-Profil</i>.</p>	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.
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.				
SET PREFIX <i>prefix</i> OFF	<p>Dient zur Angabe eines Präfixes für Variablen- oder Feldnamen.</p> <p>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.</p>				

Editor-Kommando	Funktion						
	<p>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.</p>						
<p>SET SAVEFORMAT V31 V41 or SET SF V31 V41</p>	<p>Gibt das Standard-Quellcode-Format für Data Areas an.</p> <p>Einstellung V31: Data Area-Quellcodes werden im Format gespeichert, das kompatibel zu Natural Version 3.1 ist.</p> <p>Einstellung V41: Data Area-Quellcodes werden im erweiterten Quellcode-Format gespeichert (Standardeinstellung).</p> <p>Siehe auch Quellcode-Format zum Speichern der Data Area.</p>						
<p>SET SCAN COMMENT NAME ALL</p>	<p>Legt die Spalte(n) fest, in denen das SCAN-Kommando nach einem Suchwert (<i>scan-value</i>) suchen soll:</p> <table border="1" data-bbox="678 814 1482 1020"> <tr> <td data-bbox="678 814 997 894">COMMENT</td> <td data-bbox="997 814 1482 894">Die Spalte Miscellaneous wird durchsucht.</td> </tr> <tr> <td data-bbox="678 894 997 940">NAME</td> <td data-bbox="997 894 1482 940">Die Spalte Name wird durchsucht.</td> </tr> <tr> <td data-bbox="678 940 997 1020">ALL</td> <td data-bbox="997 940 1482 1020">Die Spalten Name und Miscellaneous werden durchsucht.</td> </tr> </table> <p>Die Standardeinstellung ist NAME.</p>	COMMENT	Die Spalte Miscellaneous wird durchsucht.	NAME	Die Spalte Name wird durchsucht.	ALL	Die Spalten Name und Miscellaneous werden durchsucht.
COMMENT	Die Spalte Miscellaneous wird durchsucht.						
NAME	Die Spalte Name wird durchsucht.						
ALL	Die Spalten Name und Miscellaneous werden durchsucht.						
<p>SET SIZE ON OFF</p>	<p>Wenn SET SIZE auf ON gesetzt ist, wird die Größe der Data Area in der unteren Informationszeile des Editor-Bildschirms angezeigt.</p> <p>Das Kommando SET SIZE entspricht der Editor-Profil-Option Source Size Information. Siehe Editor-Profil.</p>						
<p>SET STAY ON OFF</p>	<p>Wenn STAY auf ON gesetzt ist, wird der aktuelle Bildschirm weiterhin angezeigt. Eine Vorwärts-/Rückwärtspositionierung kann nur mit den Positionierungskommandos bewirkt werden.</p> <p>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.</p> <p>Das Kommando SET STAY entspricht der Editor-Profil-Option Stay on Current Screen. Siehe Editor-Profil.</p>						
<p>SET TYPE G L A</p>	<p>Mit diesem Kommando können Sie den Typ der aktuellen Data Area ändern:</p> <table border="1" data-bbox="678 1766 1482 1900"> <tr> <td data-bbox="678 1766 1016 1812">G</td> <td data-bbox="1016 1766 1482 1812">Global Data Area</td> </tr> <tr> <td data-bbox="678 1812 1016 1858">L</td> <td data-bbox="1016 1812 1482 1858">Local Data Area</td> </tr> <tr> <td data-bbox="678 1858 1016 1900">A</td> <td data-bbox="1016 1858 1482 1900">Parameter Data Area</td> </tr> </table>	G	Global Data Area	L	Local Data Area	A	Parameter Data Area
G	Global Data Area						
L	Local Data Area						
A	Parameter Data Area						

Editor-Kommando	Funktion
<code>SPLIT parameter</code>	<p>Dieses Kommando bewirkt, dass der Editor-Bildschirm zwecks gleichzeitiger Anzeige eines anderen Objekts in zwei Hälften unterteilt wird. Siehe Split-Screen-Modus.</p> <p><i>parameter</i> steht für einen Parameter, den Sie mit dem Kommando eingeben müssen. Siehe Split-Screen-Kommandos.</p>
<code>STOW [object-name]</code>	<p>Führt das Systemkommando STOW aus, mit dem die aktuelle Data Area-Definition gespeichert und katalogisiert wird.</p> <p>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.</p>

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 <code>.N</code> eingegeben wurde. Siehe auch Zeilenkommando <code>.P</code> .

Data Area speichern und katalogisieren

Bevor eine Data Area in einem Natural-Programm (oder einem anderen Objekt) benutzt werden kann, muss sie als Quellcode-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 Quellcode-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 erweiterte Quellcode-Format. Der Editor kann beide Formate lesen. Er konvertiert das Natural Version 3.1-Format in das erweiterte 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 Quellcode-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 Quellcode-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 <i>nicht</i> gespeichert.
Resume Function	Die Editor-Sitzung wird <i>nicht</i> 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.

V 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 Quellcode-Objekt und als katalogisiertes Objekt in einer Library in einer Natural-Systemdatei gespeichert werden.



Anmerkung: 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.

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

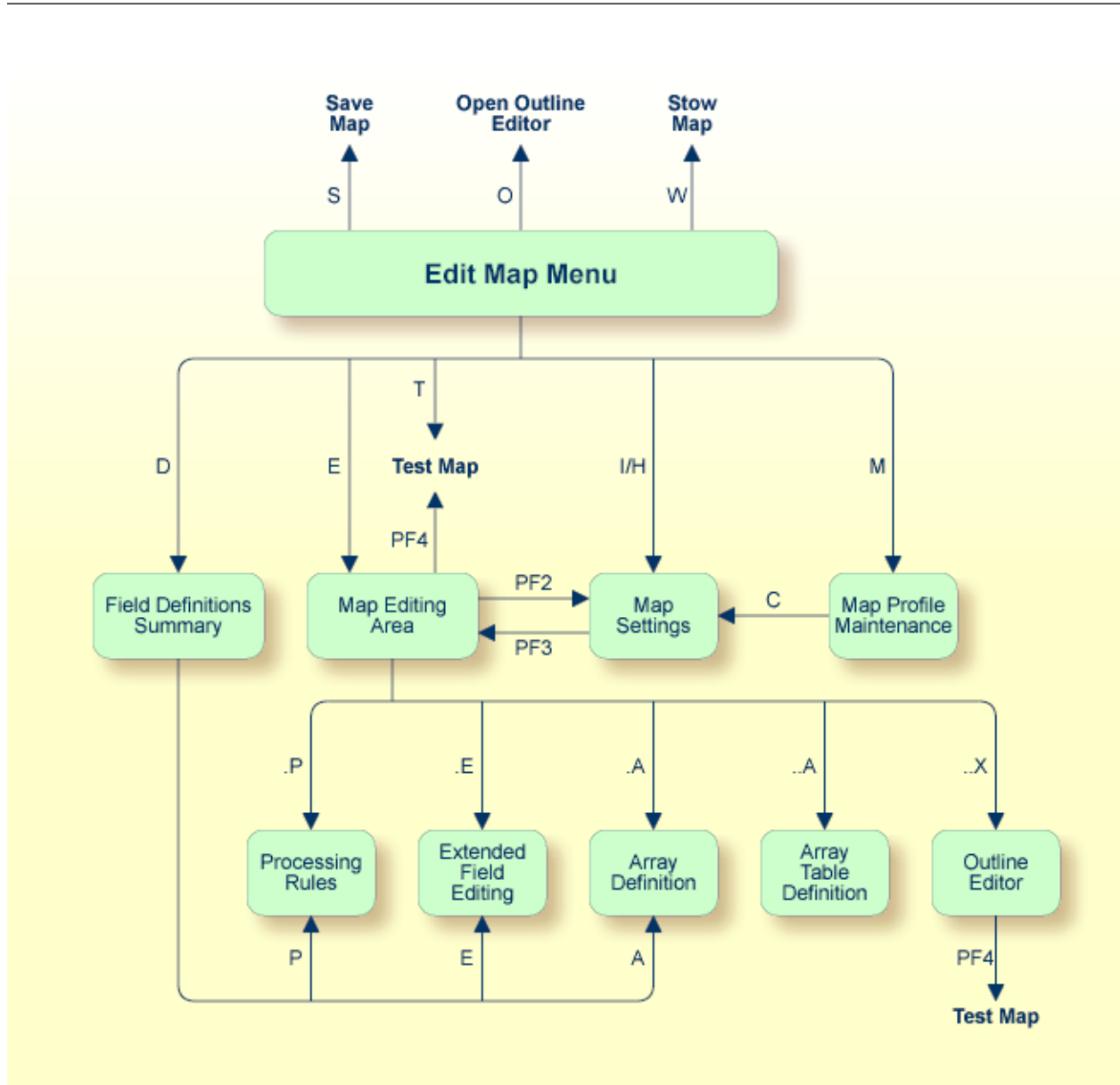
Nachträgliches Zuweisen von Feldern

Array- und Tabellen-Definitionen

Verarbeitungsregeln

6 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

7 In vier Schritten zur fertigen Maske

- Schritt 1 - Masken-Profil definieren 96
- Schritt 2 - Maske definieren 96
- Schritt 3 - Maskenfelder definieren 96
- Schritt 4 - Masken-Definition speichern 97

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-Objektyp Map) als Quellcode-Objekt und/oder katalogisiertes Objekt in der aktuellen Library in der aktuellen Natural-Systemdatei gespeichert werden. Nachdem die Definition als Quellcode-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*.

8

Masken-Editor aufrufen und beenden

- Masken-Editor aufrufen und neue Maske erstellen 100
- Masken-Editor aufrufen und vorhandene Maske bearbeiten 101
- Masken-Editor beenden 102

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-Benutzererkennung 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	<p>Der Name des Quellcode-Objekts, das die Map oder Help Map enthält.</p> <p>Bei mehrsprachig verfügbaren Maps sollte eine Stelle des Objektnamens für den Sprachschlüssel reserviert werden. Zum Beispiel:</p> <p>USERMAP1 (der Sprachcode für Englisch ist 1, für Deutsch 2)</p> <p>Aufgerufen wird die Map aus dem Programm mittels folgendem Statement:</p> <pre>INPUT USING MAP 'USERMAP&'</pre> <p>Dabei wird & zur Ausführungszeit durch den Inhalt der Systemvariablen *LANGUAGE ersetzt.</p> <p>Anmerkung: Eine Liste der in Natural verwendeten Sprachcodes finden Sie bei der Beschreibung der Systemvariablen *LANGUAGE in der <i>Systemvariablen</i>-Dokumentation.</p>
Profile	<p>Das zurzeit gültige Sitzungsprofil.</p> <p>Als Name des Profils wird der Name der aktuellen Library gesetzt. Falls diese Library nicht verfügbar ist, wird die aktuelle Benutzererkennung eingesetzt. Falls diese Profilkennung nicht vorhanden ist, wird der Name auf SYSPROF gesetzt.</p>

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 Quellcode-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 Quellcode-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*.

9

Funktionen im Menü 'Edit Map'

▪ Feld- und Variablen-Definitionen bearbeiten — Funktion "Field and Variable Definitions"	104
▪ Maske editieren — Funktion "Edit Map"	105
▪ Feldumrahmungen editieren — Funktion "Outline Editor"	106
▪ Neue Maske erstellen — Funktion "Initialize New Map"	106
▪ Neue Hilfe-Maske erstellen — Funktion "Initialize a New Help Map"	106
▪ Profile und Geräte pflegen— Funktion "Maintenance of Profiles & Devices"	107
▪ Maske speichern — Funktion "Save Map"	107
▪ Maske testen — Funktion "Test Map"	107
▪ Maske speichern und katalogisieren — Funktion "Stow Map"	108
▪ Hilfe zum Masken-Editor aufrufen — Funktion "Help"	108

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.
	<i>leer</i>	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.
Prr	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-Maske erstellt wird.

Die Funktion kann nur dann ausgeführt werden, wenn in der Natural-Systemdatei kein Quellcode-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 Quellcode-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 Quellcode-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.

10 Maske erstellen — Funktion 'Initialize New Map'

- Trennzeichen — "Delimiters" 111
- Masken-Format-Einstellungen — "Format" 113
- Kontext-Einstellungen — "Context" 117
- Füllzeichen — "Filler Characters" 119

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 ..... _____
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 ..... _        -----
                                Control Var .... _____
                                Optional, Partial ....
                                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 **SCIs** (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 **CIs**, **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 **CIs** (Class), **Att** (Attribute) und **CD** oder in den Feldern der Funktion zur [erweiterten Feldbearbeitung](#) eingegeben werden können.

Trennzeichen-Klasse (CIs)		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-Standard-einstellung verwendet. Siehe auch <i>AD - Attribut-Definition</i> in der <i>Parameter-Referenz-Dokumentation</i> .
	LC	Gibt an, dass keine Umsetzung von Klein- in Großbuchstaben erfolgen soll, d.h., der Session-Parameter AD=W wird als Feld-Standard-einstellung verwendet. Um den LC-Wert wirksam zu machen, müssen Sie den Natural-Profilparameter LC auf ON setzen. Siehe auch <i>AD - Attribut-Definition</i> und <i>LC - Umsetzung von Klein- in Großbuchstaben</i> in der <i>Parameter-Referenz-Dokumentation</i> .
	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 <i>nicht</i> 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 <i>DC - Dezimalstellenzeichen</i> in der <i>Parameter-Referenz</i> -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	<p>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.</p> <p>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.</p>	

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 <i>Funktionen im Menü "Edit Map"</i>).
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 <i>HE-Parameter-Syntax</i> in der <i>Parameter-Referenz-Dokumentation</i> .
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
	<p>der Name eines jeden Feldes individuell an die Helprououtine übergeben wird.</p> <hr/> <p>N</p> <p>Gibt an, dass der Name der Maske an die Helprououtine oder Hilfe-Maske übergeben werden soll.</p> <p>Dies ist die Standard-Einstellung.</p> <p>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.</p>
Position Line Col	<p>Die Position, an der die Hilfe-Maske zur Ausführungszeit auf dem Bildschirm erscheinen soll.</p> <p>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.</p>
Automatic Rule Rank	<p>Der Rang (Priorität), der automatischen Predict-Regel zugewiesen wird, wenn sie während der Maskenfeld-Definition mit der Maske verknüpft werden. Der Standardwert ist 1.</p>
Profile Name	<p>Der Name des Profils, das bei der Initialisierung der Maske aktiv war.</p> <p>Falls ENFORCED angezeigt wird, dann sind die folgenden Masken-Einstellungen geschützt:</p> <ul style="list-style-type: none"> ■ Alle Masken-Trennzeichen (Delimiters) ■ Statisches und dynamisches Layout ■ Geräte-Prüfung ■ WRITE und INPUT-Statements ■ Alle Füllzeichen ■ Automatische Regel-Rangordnung ■ Positionierung von Hilfe-Masken <p>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.</p>

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*.

11 Maske bearbeiten — Funktion 'Edit Map'

▪ Bildschirm-Modi	122
▪ PF-Tasten und Kommandos zum Positionieren	123
▪ Zeilenkommandos	124
▪ Feld-Kommandos	126

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 _
.
.
.
.
.
.
001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+--

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   (
.

```

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	.0	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 <i>nnn</i> Zeilen nach oben positionieren.
PF8	. +	Eine halbe Fensterseite nach unten positionieren.
	. + nnn	Um <i>nnn</i> 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 <i>nnn</i> Spalten nach links positionieren.
	. <<	An den linken Rand der Maske positionieren.
PF11	. >	Eine halbe Fensterseite nach rechts positionieren.
	. > nnn	Um <i>nnn</i> 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 <i>Funktionen im Menü "Edit Map"</i>) 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
..En	Dient zum Aufrufen des Bildschirms Field and Variable Definitions - Summary der Funktion Field and Variable Definitions (siehe <i>Funktionen im Menü "Edit Map"</i>) 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 <i>Funktionen im Menü "Edit Map"</i>) 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 .
..Fc	Füllt die Leerräume in einer Zeile mit dem Buchstaben <i>c</i> auf.
..I	Fügt eine einzelne Zeile ein. Um das Einfügen der Zeile zu ermöglichen, wird die letzte Leerzeile auf dem Bildschirm gelöscht.
..In	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.
..Jn	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 <i>Referenzen auf DDMs prüfen und korrigieren</i>).
..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.
..Mn	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
<code>..M*</code>	<p>Verschiebt die Zeile, in der das Kommando eingegeben wird, und alle unter ihr befindlichen Zeilen unter die mit dem Cursor markierte Zeile.</p> <p>Dieses Kommando ist nur zweckmäßig, wenn sich die mit dem Cursor markierte Zeile oberhalb der Zeile befindet, in der Das Kommando eingegeben wird.</p>
<code>..P</code>	<p>Ruft die Bearbeitung der PF-Tasten-Verarbeitungsregeln auf.</p> <p>Bei den PF-Tasten-Verarbeitungsregeln handelt es sich um spezielle Verarbeitungsregeln zum Definieren von Aktivitäten, die programmsensitiven PF-Tasten zugewiesen werden.</p>
<code>..Pn</code>	Ruft die Bearbeitung der PF-Tasten-Verarbeitungsregeln für Rangebene n auf.
<code>..P*</code>	Listet alle Verarbeitungsregeln auf, die für die PF-Tasten in dieser Maske definiert sind.
<code>..Q</code>	Beendet die Bearbeitung der Maske. Es erscheint wieder das Menü Edit Map .
<code>..R</code>	<p>Wiederholt alle Literal-Zeichenketten einmal in der Zeile, in der das Kommando eingegeben wurde. Die Cursor-Position dient zur Anzeige der Zielzeile.</p> <p>Zwei Kommandos <code>..R</code>, die auf demselben Bildschirm eingegeben werden, bewirken, dass die Literal-Zeichenketten innerhalb des mit diesen Kommandos eingegrenzten Zeilenblocks wiederholt werden.</p>
<code>..Rn</code>	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.
<code>..S</code>	<p>Spaltet die Zeile an der Cursor-Position auf.</p> <p>Zwei Kommandos <code>..S</code>, die auf demselben Bildschirm eingegeben werden, bewirken, dass der mit diesen Kommandos eingegrenzte Zeilenblock aufgespalten wird.</p>
<code>..Sn</code>	Spaltet die Zeile, in der das Kommando eingegeben wird, und die $n-1$ unter ihr an der Cursor-Position auf.
<code>..X</code>	<p>Nur verfügbar, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Anzeige von Feldumrahmungen gestattet.</p> <p>Umschalten zwischen dem Editierbereich des Masken-Editors und dem Outline Editor. Als Alternative zu diesem Zeilenkommando können Sie PF13 benutzen.</p>

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	<p>Definiert ein Array. Dieses Kommando können Sie nur bei einem einzelnen Feld (nicht bei einem Bereich von Feldern) anwenden.</p> <p>Die Array-Definition (siehe <i>Array- und Tabellen-Definitionen</i>) 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.</p> <p>Sie können ein Array redefinieren, indem Sie das Kommando .A bei einem seiner Elemente anwenden.</p>
.An	<p>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.</p>
.C	<p>Zentriert ein Feld oder einen Bereich von Feldern zwischen angrenzenden Feldern.</p> <p>Um ein einzelnes Feld zu zentrieren, müssen Sie das Kommando .C in das zu zentrierende Feld eingeben.</p> <p>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.</p> <p>Falls kein angrenzendes Feld oder keine angrenzenden Felder vorhanden sind, werden stattdessen die für die Masken-Definition wirksamen Spaltenbegrenzungen benutzt.</p>
.D	<p>Löscht ein Feld oder einen Bereich von Feldern.</p> <p>Um ein einzelnes Feld zu löschen, müssen Sie das Kommando .D in das zu löschende Feld eingeben.</p> <p>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.</p>

Kommando	Erläuterung
.E	<p>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.</p> <p>Zum Aufrufen der Funktion zur erweiterten Feld-Bearbeitung können Sie auch den Cursor in das ausgewählte Feld stellen und PF5 drücken.</p>
.J	<p>Fügt Felder zusammen, die sich in aufeinander folgenden Zeilen befinden.</p> <p>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.</p>
.M	<p>Verschiebt ein Feld oder einen Bereich von Feldern.</p> <p>Um ein einzelnes Feld zu verschieben, müssen Sie das Kommando .M in das zu verschiebende Feld eingeben und den Cursor in die Zielposition stellen.</p> <p>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>
.P[<i>n</i>]	<p>Ermöglicht die Bearbeitung von Verarbeitungsregeln (siehe entsprechenden Abschnitt) für ein Feld.</p> <p>Um die Priorität (den Rang) der zu bearbeitenden Verarbeitungsregel anzugeben, können Sie beim Kommando .P einen Parameter <i>n</i> mit angeben. Falls nötig, können Sie den Wert für <i>n</i> in runden Klammern () einschließen.</p>
.R	<p>Wiederholt (kopiert) ein Feld oder einen Bereich von Feldern.</p> <p>Um ein einzelnes Feld zu kopieren, müssen Sie das Kommando .R in das zu kopierende Feld eingeben und den Cursor in die Zielposition stellen.</p> <p>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.</p> <p>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).</p> <p>Anmerkung: Arrays können nicht kopiert werden.</p>
.S	<p>Spaltet (verschiebt) eine Zeile oder einen Bereich von Zeilen.</p> <p>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.</p>
.T	<p>Verkürzt (löscht) ein Feld oder einen Bereich von Feldern in einer Zeile.</p> <p>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.</p>

12 Maskenfelder definieren

- Maskenfelder direkt im Bildschirm definieren 130
- Definitionen aus anderen Objekten übernehmen 131
- Systemvariablen in einer Masken-Definition benutzen 136

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-Quellcode-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, Subprogramme, 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-Quellcode-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.

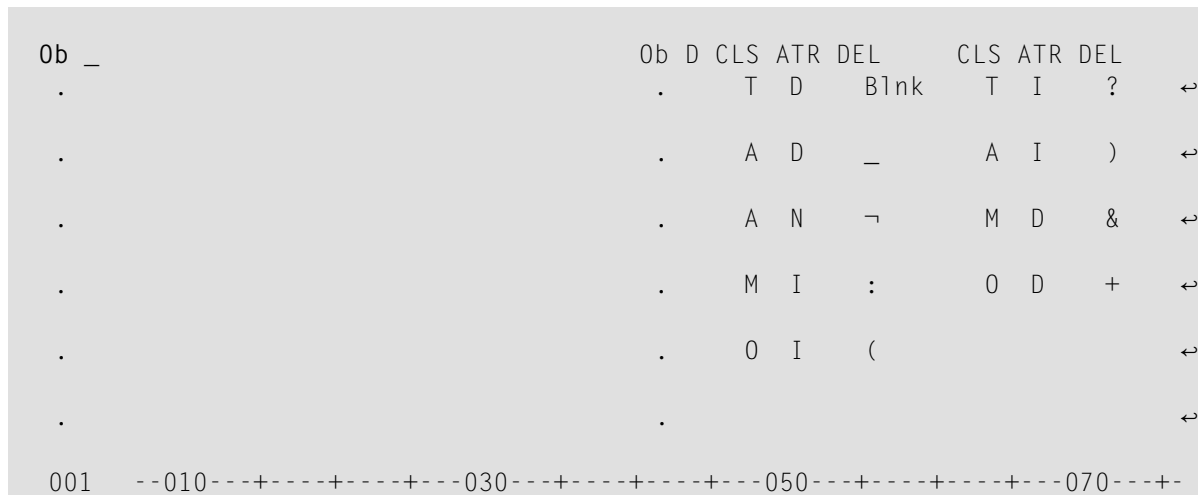
Dieser Abschnitt behandelt folgende Themen:

- [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.



- 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                             A #NAME-START           A20
2 #NAME-END                               B #NAME-END             A20
3 #MARK                                   C #PERS-ID              A8
. EMPLOYEES-VIEW                          D #MAKE                 A20
4 PERSONNEL-ID                            E #MODEL                A20
5 NAME                                     .
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	0	Der Objekttyp „General Object“ (0) 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 <code>0 objectname</code> eingeben. Dabei stellt <code>objectname</code> den Objektnamen dar. Der Masken-Editor ändert dann den Objekttyp von 0 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 Ausgabentrennzeichen (Delimiter) vorangestellt werden. Siehe folgendes Beispiel:

```
(*TIME  
(*DATE  
(*APPLIC-ID
```


13

Erweiterte Feld-Bearbeitung

- Funktion zur erweiterten Feld-Bearbeitung aufrufen und beenden 140
- Felder im erweiterten Feld-Bearbeitungsbereich 142
- Felder im erweiterten Text-Feld-Bearbeitungsbereich 146

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---+-----
.Eield1

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	<p>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.</p> <p>Der im Feld Fld bzw. Arr eingegebene Name ist abhängig davon, welche Methode Sie beim Anlegen des Feldes angewendet haben:</p> <ul style="list-style-type: none"> ■ 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. <p>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.</p> <p>Anmerkung: Feldnamen-Dubletten sind nur bei Feldern zulässig, die als reine Ausgabefelder definiert sind.</p> <p>Weitere Informationen siehe Maskenfelder definieren.</p>
Fmt	<p>Das Natural-Datenformat und die Länge des Feldes. Diese Angaben können Sie durch Überschreiben des aktuellen Eintrags ändern.</p> <p>Die Datenformate U (Unicode), C (Attribute Control) und Handle (Object Handle) sind <i>nicht</i> zulässig.</p>

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 <i>AL - Alphanumerische Länge der Ausgabe</i> , <i>FL - Gleitkomma-Mantissenlänge</i> , und <i>NL - Numerische Länge der Ausgabe</i> in der <i>Parameter-Referenz-Dokumentation</i> .
RIs	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 <i>ZP - Anzeige von Nullwerten</i> in der <i>Parameter-Referenz-Dokumentation</i> .
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 <i>SG - Vorzeichen-Stelle</i> in der <i>Parameter-Referenz-Dokumentation</i> .
PM	Druck-/Anzeige-Modus. Dieses Feld entspricht dem Session-Parameter PM. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>PM - Druck-/Anzeige-Modus</i> in der <i>Parameter-Referenz-Dokumentation</i> .
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 <i>DF - Datumsformat</i> in der <i>Parameter-Referenz-Dokumentation</i> .
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 <i>DY - Dynamische Attribute</i> in der <i>Parameter-Referenz-Dokumentation</i> . 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
	<p>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.</p> <p>Wichtig: Eine erweiterte DY-Definition können Sie nur dann eingeben, wenn noch keine Kontrollvariable für diese Maske definiert worden ist (siehe auch CV-Feld).</p>
HE	<p>Helproutine oder Hilfemaske.</p> <p>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 <i>HE - Helproutine</i> in der <i>Parameter-Referenz</i>-Dokumentation.</p> <p>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 <code>.E</code> oder geben Sie ein Pluszeichen (+) ein und drücken Sie Enter. Es erscheint ein Fenster mit zusätzlichen Eingabezeilen.</p> <p>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 <i>HE-Parameter-Syntax</i> in der <i>Parameter-Referenz</i>-Dokumentation. Zusätzlich zu den dort vorhandenen Syntax-Erläuterungen gilt bei der Benutzung des Masken-Editors Folgendes:</p> <p><i>operand1:</i></p> <ul style="list-style-type: none"> ■ 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. <p><i>operand2:</i></p> <ul style="list-style-type: none"> ■ Wird ein Variablenname angegeben, zu dem noch kein Maskenfeld existiert, wird automatisch ein Maskenparameter mit diesem Namen im Natural-Datenformat/Länge N7 definiert. <p>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 <i>Array-Definition</i>).</p>
AD	<p>Feldattribute.</p> <p>Dieses Feld entspricht dem Session-Parameter AD. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>AD - Attribut-Definition</i> in der <i>Parameter-Referenz</i>-Dokumentation.</p> <p>Um den Quellcode zu optimieren, werden die Standardwerte D, H, F und W akzeptiert aber nicht gespeichert (siehe auch Session-Parameter AD).</p>

Feld	Erläuterung
CD	<p>Farbattribute.</p> <p>Dieses Feld entspricht dem Session-Parameter CD. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>CD - Farbdefinition</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p>
BX	<p>Feldumrahmungen.</p> <p>Dieses Feld wird nur dann angezeigt, wenn Ihr Terminal bzw. Ihre Terminal-Emulation die Darstellung von Feldumrahmungen unterstützt.</p> <p>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 <i>BX - Feldumrahmung (Box-Definition)</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p> <p>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.</p>
CV	<p>Kontrollvariable für dynamische Feldattribute.</p> <p>Dieses Feld entspricht dem Session-Parameter CV. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>CV - Kontrollvariable</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p> <p>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.</p>
EM oder EMU	<p>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.</p> <p>Dieses Feld entspricht dem Session-Parameter EM bzw. EMU. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>EM - Editiermaske</i> und <i>EMU - Unicode-Editiermaske (Parameter-Referenz-Dokumentation)</i> und <i>EMU, ICU, LCU, TCU versus EM, IC, LC, TC (Unicode and Code Page Support-Dokumentation)</i>.</p> <p>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.</p> <p>Ein Feld, bei dem eine Editiermaske zur Anwendung kommt, wird im Masken-Editierbereich durch ein M gekennzeichnet.</p>
SB	<p>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.</p> <p>Es gilt das Format des Quellfeldes. Sie können es im Fenster Parameter Definitions ändern.</p>

Feld	Erläuterung	
	Siehe auch <i>SB - Auswahlfeld</i> in der <i>Parameter-Referenz-Dokumentation</i> .	
Mod	Das Feld Mode zeigt an, auf welche Weise das Feld erstellt wurde:	
	Data	Das Feld wurde durch Auswählen eines Feldes aus einer <code>DEFINE DATA-Statement-Definition</code> 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	<p>Feldattribute.</p> <p>Dieses Feld entspricht dem Session-Parameter <code>AD</code>. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>AD - Attribut-Definition</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p> <p>Standardmäßig ist der Wert <code>D</code> zugewiesen.</p>
CD	<p>Farbattribute.</p> <p>Dieses Feld entspricht dem Session-Parameter <code>CD</code>. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>CD - Farbdefinition</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p>

Feld	Erläuterung
PM	<p>Druck-/Anzeige-Modus.</p> <p>Dieses Feld entspricht dem Session-Parameter PM. Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>PM - Druck-/Anzeige-Modus</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p> <p>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).</p>
BX	<p>Feldumrahmung, die mit dem Parameter BX angegeben werden.</p> <p>Ausführliche Informationen zur Benutzung dieses Feldes und zu den gültigen Eingabewerten siehe <i>BX - Feldumrahmung (Box-Definition)</i> in der <i>Parameter-Referenz-Dokumentation</i>.</p> <p>Die Angaben zum BX-Parameter werden gewöhnlich automatisch mittels der Funktionalität des Outline-Editors vorgenommen.</p>
Hex Editing	<p>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.</p> <p>Ein Leerzeichen-Wert wird durch eine Reihe von hexadezimalen Werten 00 dargestellt.</p> <p>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.</p> <p>Die Standardeinstellung beim Hex Editing-Feld ist N (No).</p>
Value	<p>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.</p> <p>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.</p> <p>Das Feld Value darf <i>keine</i> Leerzeichen enthalten.</p>

14

Feldumrahmungen erstellen — mit dem Outline-Editor

▪ Voraussetzungen	150
▪ Editor für Feldumrahmungen (Outline-Editor) aufrufen	152
▪ Box-Codes für Feldumrahmungen benutzen	153
▪ Ende-Markierungen für Feldumrahmungen benutzen	155
▪ Liste der Box-Codes und BX-Parameterwerte für Feldumrahmungen	156
▪ Beispiel für Feldumrahmungen — Box-Coding	157
▪ Zeilenkommandos und PF-Tasten	160

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 Quellcode-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 Quellcode-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):

```

Outline Editor
Code BX  Code BX  Code BX  Code BX  Code BX  Code BX  Code BX  Code BX
T  ┌───┐  B  ┌───┐  I  ┌───┐  C  ┌───┐  1  ┌───┐  3  ┌───┐  5  ┌───┐  U  ┌───┐
001 L  ┌──┴──┐  R  ┌──┴──┐  H  ┌──┴──┐  O  ┌──┴──┐  2  ┌──┴──┐  4  ┌──┴──┐  6  ┌──┴──┐  E  END
    --010--++--++--++--030--++--++--++--050--++--++--++--070--++--
    -
        #Personal#Information

        First Name:  :XXXXXXXXXXXXXXXXXXXXX
        Last Name:   :XXXXXXXXXXXXXXXXXXXXX
        Street:      :XXXXXXXXXXXXXXXXXXXXX
        Zip Code / City: :99999999 :XXXXXXXXXXXXXXXXXXXXX

        Phone Office: :XXXXXXXXXXXXXXXXXXXXX
        Home:         :XXXXXXXXXXXXXXXXXXXXX
        Mobile:       :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help   Exit Test BoxDi --   -   +   Full <   >   Let
    
```

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 0 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 `Personal1`.

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: `0E`) 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.	0	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:

```

Outline Editor
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--+-----

      BPersonal#Information

      First Name:  OXXXXXXXXXXXXXXXXXXXXX

      Last Name:  2
                  L:XXXXXXXXXXXXXXXXXXXXX  R  E
                  U
                  Street:  OXXXXXXXXXXXXXXXXXXXXX

      Zip Code / City:  C99999999  :XXXXXXXXXXXXXXXXX1E

Phone L Office:  OXXXXXXXXXXXXXXXXXXXXX
      L Home:    :XXXXXXXXXXXXXXXXXXXXX
      L Mobile:  :XXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      Exit Test BoxDi -- - + Full < > Let
    
```

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:


```

Outline Editor
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--+-+--
                                     BPersonalBInformation
                                     First Name:  OXXXXXXXXXXXXXXXXXXXXX
                                     Last Name:    2.....
                                                L:XXXXXXXXXXXXXXXXXXXXX R
                                                U.....
                                     Street:  OXXXXXXXXXXXXXXXXXXXXX
                                     Zip Code / City: C99999999I.IXXXXXXXXXXXXXXXXX1
Phone L Office: OXXXXXXXXXXXXXXXXXXXXX
L Home:  :XXXXXXXXXXXXXXXXXXXXX
L Mobile: :XXXXXXXXXXXXXXXXXXXXX
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit Test BoxDi --  -  +      Full <  >      Let
    
```

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:

```

      Personal Information

First Name:  ZZZZZZZZZZZZZZZZZZZZZ
Last Name:  FFFFFFFFFFFFFFFFFFFFFF
Street:     PFFFFFFFFFFFFFFFFFFFFF
Zip Code / City:  12345678  YYYYYYYYYYYYYYYY

Phone | Office:  12345678901234567890
      | Home:    12345678901234567890
      | Mobile:  12345678901234567890
    
```

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.
..Dnn		Löscht alle Box-Codes und leeren Text-Felder in der Zeile, in der das Kommando eingegeben wurde, sowie <i>nn</i> Zeilen unterhalb dieser Zeile. Dabei steht <i>nn</i> 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 <i>alle</i> 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 <i>Bildschirm "Box-Code Display"</i> 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 <i>vor</i> 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.....
      ..... 0.....
      ..... 2.....
      ..... L.....R
      ..... U.....
      ..... 0.....
      ..... C.....I.I.....1
..... L ..... 0.....
      L .....
      L .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit

```

15

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.

16

Array- und Tabellen-Definitionen

- Array-Definition 166
- Array-Tabellen-Definition 170

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 .AXXXXXXXXXXXXXXXXXXXXX

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	<p>Zeigt die obere Dimension eines Array, d.h. die höchste Ausprägung (von links nach rechts) in der ersten, zweiten und dritten Dimension.</p> <p>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-Definitions-Bildschirm nicht überschrieben werden. Siehe auch Hinweis in <i>Top Dim ändern</i> weiter unten.</p> <p>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.</p> <p>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.</p>
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	<p>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.</p> <p>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.</p> <p>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.</p>
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   C1s/Ls
Third  Index:  Direction(H/V) V           _____           0   C1s/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:

```
(*DATE                                     (*TIME
      Map containing an array table of multi-dimensional arrays

ARRAY1 (1-dim.)                ARRAY2 (2-dim.)                ARRAY3 (3-dim.)

:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }  Third Index
                       Main Index                Second Index                (3 vertical
                       (2 vertical                (2 vertical                occurrences)
                       occurrences)            occurrences)
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }
:xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }          :xxxxxxxxxxxxxxxxx }

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12--
      Help Mset  Exit  Test  Edit  Top  -   +   Full <   >   Let
```

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.

17

Verarbeitungsregeln

- Feldbezogene Verarbeitungsregeln 176
- Funktionstastenbezogene Verarbeitungsregeln 177
- Verarbeitungsregel bearbeiten 178

Feldbezogene Verarbeitungsregeln

Es können drei Arten von feldbezogenen Verarbeitungsregeln definiert werden.

■ **Inline-Verarbeitungsregeln**

Diese Regeln werden innerhalb eines Map Quellcode-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 (`.Prr`) 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
<u>ADD</u> [(n)]	Fügt <i>n</i> Leerzeilen zum Quellcode hinzu. Weitere Informationen siehe ADD-Kommando in <i>Editor-Kommandos</i> in der <i>Programm-Editor</i> -Dokumentation.
<u>C</u> HANGE 'string1' string2'	Durchsucht den Quellcode nach der als Suchwert (<i>string1</i>) eingegebenen Zeichenkette und ersetzt jeden gefundenen Suchwert mit der als Ersetzungswert (<i>string2</i>) eingegebenen Zeichenkette.
<u>C</u> HECK	Prüft die Regel.
<u>C</u> LEAR	Löscht den Inhalt des Editierbereichs (auch die Markierungen X und Y).
<u>D</u> X oder D <u>Y</u> oder D <u>X</u> - <u>Y</u>	Löscht die mit X markierte Zeile oder die mit Y markierte Zeile oder den mit den Markierungen X und Y eingegrenzten Zeilenblock.
<u>E</u> X oder E <u>Y</u> oder E <u>X</u> - <u>Y</u>	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. .
<u>E</u> XIT .	Beendet die Bearbeitung der Verarbeitungsregel. Anschließend Rückkehr zur Maskenbearbeitung.
<u>P</u>	Positioniert vorwärts zur nächsten für das Feld definierten Regel.
<u>P</u> *	Auswahl einer Regel aus dem Auswahlm Menü.
<u>P</u> rr	Auswahl der Regel mit Rang <i>rr</i> .

Kommando	Erläuterung
P= <i>rr</i>	Ändert den Rang einer Verarbeitungsregel in den Rang <i>rr</i> .
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 <i>name</i>	Speichert die Verarbeitungsregel als Copycode unter dem Namen <i>name</i> .
SCAN [' <i>scan-value</i> ']	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 (<i>scan-value</i>) Folgendes eingeben: Wenn Sie SCAN ' <i>scan-value</i> ' eingeben, erfolgt die Suche nach dem Suchwert <i>scan-value</i> .
SCAN = [+ -]	Sucht nach der nächsten Ausprägung des mit dem SCAN-Kommando angegebenen Suchwerts in der angegebenen Richtung. Weitere Informationen siehe SCAN=[+ -] unter <i>Editor-Kommandos</i> beim Programm-Editor.
SHIFT [- + <i>nn</i>]	Verschiebt jede Quellcode-Zeile, die mit den Markierungen X und Y begrenzt ist, nach links oder nach rechts. Der Parameter <i>nn</i> 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 <i>Split-Screen-Modus</i> und Vollbild-Modus. Siehe auch SPLIT in <i>Editor-Kommandos zum Positionieren</i> .
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 vorhandenen 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.
<u>I</u> OP	An den Anfang der Regel positionieren.
-	
<u>B</u> OTTOM	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.
<u>S</u> PLIT[- - + ++]	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(<i>nnnn</i>)	Löscht eine oder mehrere Zeilen. Der Standardwert ist 1 Zeile.
.I(<i>n</i>)	Fügt <i>n</i> Leerzeilen ein. Zeilen, die leer gelassen werden, werden beim nächsten Drücken von Enter wieder entfernt.
.I(<i>obj</i> , <i>ssss</i> , <i>nnnn</i>)	Fügt in den Quellcode ein Objekt ein, das in der aktuellen Library oder in der Steplib enthalten ist. Der Eintrag <i>ssss</i> kann benutzt werden, um die Zeile anzugeben, ab der die Einfügung beginnen soll. Der Eintrag <i>nnnn</i> kann benutzt werden, um die Anzahl der einzufügenden Zeilen anzugeben. Weitere Informationen zu den Zeilenkommandos .I siehe Zeilenkommandos im Abschnitt <i>Maske bearbeiten</i> .
.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 <i>n</i> 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	<p>Markiert eine Zeile oder das Ende eines Zeilenblocks, die bzw. der bearbeitet werden soll.</p> <p>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.</p>

VI 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.

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.

18 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.



Tip: 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 ..... _  -----
Control Var .... _____  Optional, Partial ....
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 ..... _  -----
Control Var .... _____  Optional, Partial .... _
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:


```

0b _                               0b 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 Editierbereich, in dem Sie eine Maske entwerfen können.

Sie können nun fortfahren mit *Maskenfelder anlegen, positionieren und löschen*.

19 Maskenfelder anlegen, positionieren und löschen

■ Felder anlegen und zentrieren	194
■ Felder verschieben	199
■ Felder löschen und Zeilen einfügen	214

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:

```
0b _                0b 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:

```

0b _                               0b 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
*****

*****
001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----

```

2 Drücken Sie Enter.

Der Text wird zentriert.

➤ **Um Datenfelder anzulegen:**

1 Geben Sie Folgendes ein:

```
*****
(*DATX                PERSONNEL INFORMATION
(*TIMX

PLEASE Enter CITY::X(20)
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

*****
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:


```

*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

*****
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.

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

.MEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]

*****
001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----
```

2 Drücken Sie Enter.

Das Text-Feld, in das das Kommando eingegeben wurde, wird zur Position des Cursors verschoben:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

      Enter CITY::XXXXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

..M   Enter CITY::XXXXXXXXXXXXXXXXXXXXX
PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
[]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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE
    Enter CITY::XXXXXXXXXXXXXXXXXXXXX

*****
001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----
```

➤ **Um Zeilen zusammenzuführen:**

- 1 Geben Sie das Zeilenkommando `. . J` wie unten gezeigt ein:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
..JASE
    Enter CITY::XXXXXXXXXXXXXXXXXXXXX

*****
001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----
```

2 Drücken Sie Enter.

Die Zeile, in der das Kommando eingegeben wurde, und die Zeile darunter werden zusammengeführt:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

*****
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::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

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:


```

*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

.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.

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

        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.

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

        .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.

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                                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:

```
*****
(XXXXXXXXX          PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

          []

          .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:

```
*****
(XXXXXXX          PERSONNEL INFORMATION
(XXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                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:

```

*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

..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.)

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                                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:


```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                                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::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                                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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                                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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

                                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::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

                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:

```
*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

..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::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX

*****

001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----
```

> **Um Zeilen einzufügen:**

- 1 Geben Sie das Zeilenkommando `..I4` so wie unten gezeigt ein:


```

*****
(XXXXXXXXX                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXX
..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*.

20 Maske testen und speichern

In dieser Sitzung werden Sie das Layout einer Maske testen, den Masken-Editier-Bereich verlassen und eine Maske als Quellcode-Objekt speichern.

➤ **Um eine Maske zu testen und sie als Quellcode-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 wird 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 Quellcode-Objekt mit dem Namen MAP001 in der aktuellen Natural-Library in der aktuellen Systemdatei gespeichert.

Sie können nun fortfahren mit *Verarbeitungsregeln definieren*.

21

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:

```

0b _          0b 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---+-----
*****
(XXXXXXXXX          PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME: .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY: .XXXXXXXXXXXXXXXXXXXXXXXXX

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*.

22 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 Quellcode-Objekt *und* als katalogisiertes Objekt in der aktuellen Natural-Library in der aktuellen Systemdatei speichert.

Sie können nun fortfahren mit [Feld-Eigenschaften definieren](#).

23

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:

```

0b _                               0b 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---+---+
*****
(XXXXXXXXX                          PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXXXXXXXXX

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:

```

0b _                               0b 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---+----
*****
(XXXXXXXXX                          PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME: .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY: .XXXXXXXXXXXXXXXXXXXXXXXXX

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---+---
*****
(XXXXXXXXX                                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME:..XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY:..XXXXXXXXXXXXXXXXXXXXXXXXXX

TYPE IN?. TO STOP OR?? FOR HELP.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      HELP  Mset  Exit  <---  --->  --  -  +          <  >  Let

```

- 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---+----
*****
(XXXXXXXXX                                PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME:..XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY:..XXXXXXXXXXXXXXXXXXXXXXXXXXXX

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:

```

0b _                0b 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---+---
*****
(XXXXXXXXX          PERSONNEL INFORMATION
(XXXXXXXXX

PLEASE Enter NAME::XXXXXXXXXXXXXXXXXXXXX
PLEASE Enter CITY::XXXXXXXXXXXXXXXXXXXXX

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*.

24

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 ..... _
Control Var .... _____
Optional, Partial ....
Required, Partial ....
Optional, Complete ...
Required, Complete ...

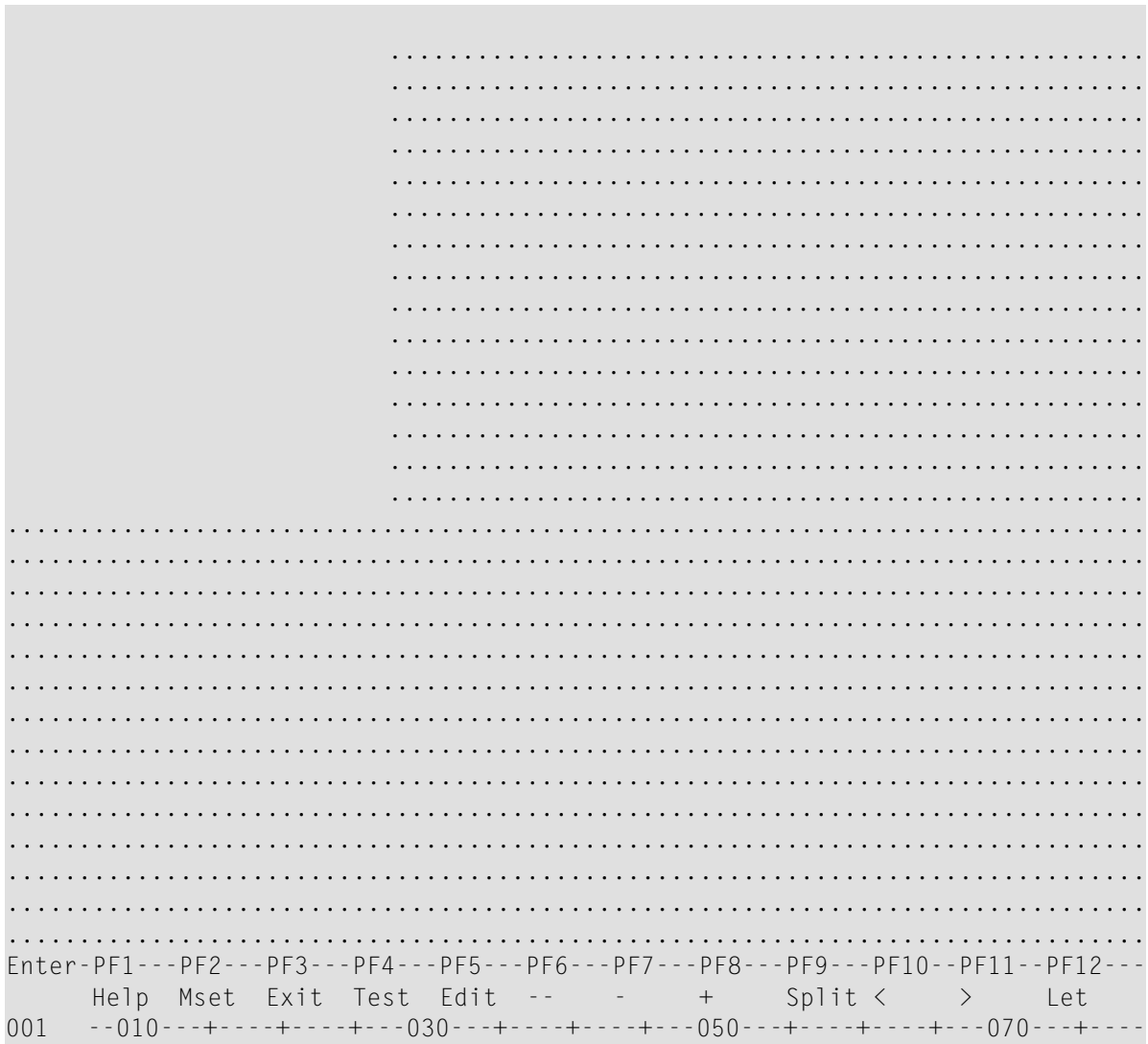
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Exit Let
    
```

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:


```

*****
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*.

25

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 `SYSEXP` 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 Quellcode-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*.

26

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 werden 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:

```
0b V EMPLOYEES                0b D CLS ATT DEL      CLS ATT DEL
1 PERSONNEL-ID                .   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:


```

0b V EMPLOYEES                0b 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  .           0  D   +
4 NAME                          A20 .   0  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:

```
0b V EMPLOYEES                0b 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  .
4 NAME                        A20 .   0 I   (
5 MIDDLE-NAME                 A20 .
001  --010---+---+---+---030---+---+---+---050---+---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXX
(XXXXXXXXXXXXXXXXXXXXXX

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:

```

0b V EMPLOYEES                0b D CLS ATT DEL      CLS ATT DEL
1 PERSONNEL-ID                .   T D   Blnk     T I   ?
. FULL-NAME                   *G1   .
2 FIRST-NAME                  A20   .
3 MIDDLE-I                    A1    .              0 D   +
4 NAME                        A20   .   0 I   (
5 MIDDLE-NAME                 A20   .
001  --010---+---+---+---030---+---+---+---050---+---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXXXXX

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:

```
0b V EMPLOYEES                0b D CLS ATT DEL      CLS ATT DEL
. FULL-ADDRESS                .   T D  Blnk    T I  ?
1 ADDRESS-LINE                .   A D  _      A I  )
2 CITY                        .   A N  -      M D  &
3 ZIP                          .   M I  :      O D  +
4 POST-CODE                   .   O I  (
5 COUNTRY                     .
001  --010---+---+---+---030---+---+---+---050---+---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXXXX(XXXXXXXXXXXXXXXXXXXXXX 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:

```

0b V EMPLOYEEES                0b D CLS ATT DEL      CLS ATT DEL
. FULL-ADDRESS                 .   T D  Blnk   T I  ?
1 ADDRESS-LINE                 .   A D  _      A I  )
2 CITY                         .   A N  -      M D  &
3 ZIP                          .   M I  :      O D  +
4 POST-CODE                    .   O I  (
5 COUNTRY                      .
001  --010---+---+---+---030---+---+---+---050---+---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                           CITY:(XXXXXXXXXXXXXXXXXXXXXXXXXXXX

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*.

27

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__
                                dynamic ..... N (Y/N)
                                Zero Print ..... N (Y/N)
                                Case Default ... UC (UC/LC)
                                Manual Skip .... N (Y/N)
0   D      +   Decimal Char ... .
0   I      (   Standard Keys .. N (Y/N)
                                Justification .. L (L/R)
                                Print Mode ..... _
                                Control Var .... _____

                                Automatic Rule Rank 1
                                Profile Name .... SYSPROF

                                Filler Characters
                                -----
                                Optional, Partial ....
                                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:(XXXXXXXXXXXXXXXXXXXX, 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:


```

0b _                               0b D CLS ATT DEL   CLS ATT DEL
.                                  .   T D   Blnk  T I   ?
.                                  .
.                                  .
.                                  .   0 D   +
.                                  .   0 I   (
.                                  .
001  --010---+---+---+---030---+---+---050---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXXXXXX  (XXXXXXXXXXXXXXXXXXXXXXXXX

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:

```
0b _                               0b D CLS ATT DEL   CLS ATT DEL
.      .      T D   Blnk   T I   ?
.
.
.      .      0 D   +
.      .      0 I   (
.
001  --010---+---+---+---030---+---+---050---+---+---070---+---
NAME:(XXXXXXXXXXXXXXXXXXXXX FIRST NAME:(XXXXXXXXXXXXXXXXXXXXX

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*.

28

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 Quellcode-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 Quellcode-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.

VII

DDM-Editor (SYSDDM Utility)

Dieses Dokument beschreibt, wie Sie Natural-Datendefinitionsmodule (*Data Definition Modules/DDMs*) erstellen, pflegen und löschen können.



Anmerkungen:

1. Bei Natural für Großrechner ist der Editor, mit dem die Datendefinitionsmodule (DDMs) erstellt, gepflegt und gelöscht werden können, als Utility SYSDDM implementiert.
2. Die Benutzung der Funktionen der SYSDDM Utility kann in Natural Security benutzerspezifisch eingeschränkt werden. Ausführliche Informationen siehe *SYSDDM Utility Profiles* in der *Natural Security*-Dokumentation.

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](#)

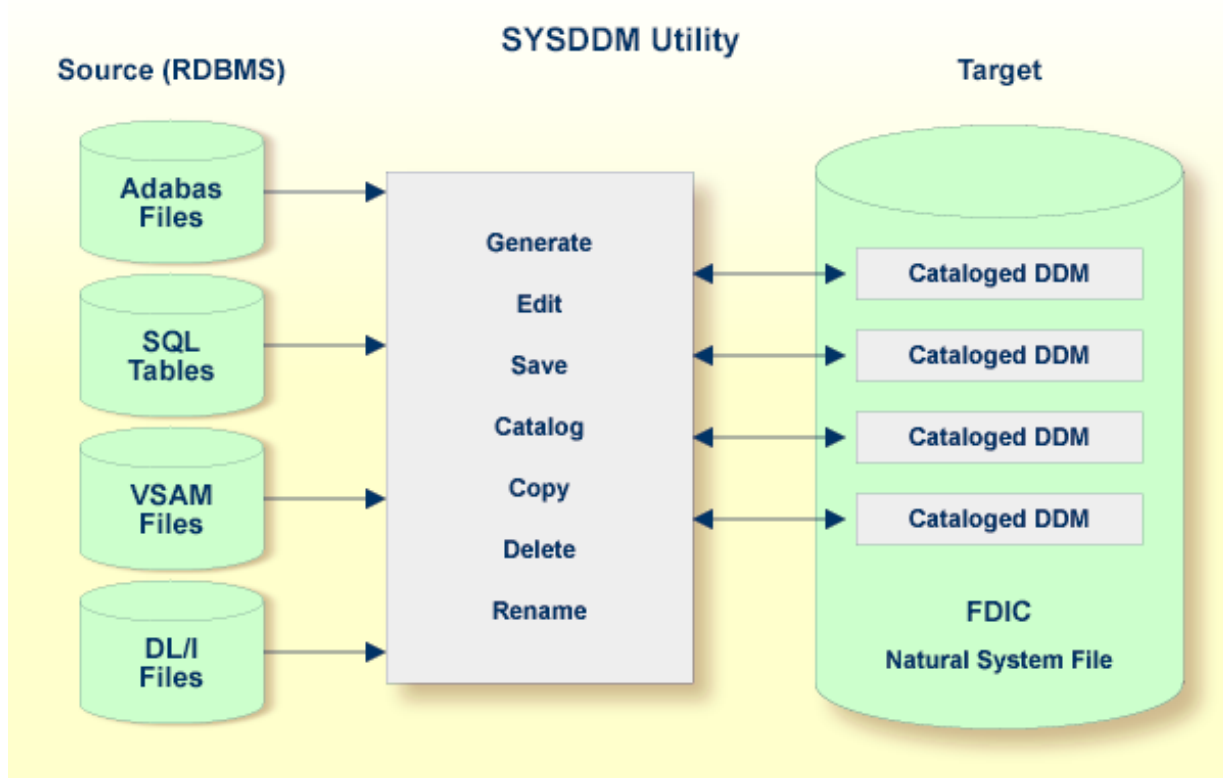
[DDMs in unterschiedlichen Umgebungen pflegen](#)

29 Funktionsbeschreibung

- DDMs speichern 280
- Einschränkungen bei der Benutzung 281

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 gleichwertige 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-Profiles geändert werden kann.

Informationen bezüglich des Schutzes von DDMs auf Großrechnern siehe *Protecting DDMs On Mainframes* in der *Natural Security*-Dokumentation.

30

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.

31

SYSDDM Pflege- und Service-Funktionen benutzen

■ Hilfe zu Funktionen	286
■ Funktionen ausführen	287
■ Beschreibung der Funktionen	287
■ DDM-Parameter angeben	291

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 Hilfetemas 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 <i>nicht</i> 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 <i>Namenskonventionen für Objekte</i> in der <i>Natural benutzen</i> -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.

Funktionscode	Feld	Funktion												
		<p>Bei einem DDM aus VSAM (DDM Type auf V gesetzt - siehe auch DDM Type in <i>DDM-Parameter angeben</i>) werden Sie aufgefordert, zusätzliche Angaben zu machen.</p> <p>Weitere Informationen siehe <i>Natural File Access</i> in der <i>Database Management System Interfaces</i>-Dokumentation.</p>												
U	Delete DDM	<p>Löscht ein (oder mehrere) katalogisierte(s) DDM(s) aus der Systemdatei FDIC.</p> <p>Der Inhalt des Editor-Source-Bereichs ist von der Löschung nicht betroffen.</p> <p>Beim Löschen eines DDM mit der Utility SYSDDM wird außerdem auch das entsprechende Natural Security-Dateiprofil gelöscht.</p>												
L	List DDMs	<p>Zeigt eine einzelne DDM-Source an (dazu wird der DDM-Editor <i>nicht</i> aufgerufen) oder eine Liste von DDMs, die in der angegebenen FDIC-Systemdatei gespeichert sind - ähnlich wie beim Natural-Systemkommando LIST DDM (siehe <i>Systemkommandos</i>-Dokumentation). Anders als beim Systemkommando LIST DDM 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 <i>Protecting DDMs On Mainframes</i> in der <i>Natural Security</i>-Dokumentation.</p> <p>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.</p>												
X	List DDMs with Additional Information	<p>Zeigt eine Liste von DDMs, die in der angegebenen FDIC-Systemdatei gespeichert sind. Die Liste enthält folgende Informationsfelder:</p> <table border="1" data-bbox="505 1287 1385 1803"> <tbody> <tr> <td data-bbox="505 1287 802 1335"></td> <td data-bbox="802 1287 1385 1335"></td> </tr> <tr> <td data-bbox="505 1335 802 1415">DBID</td> <td data-bbox="802 1335 1385 1415">Die Datenbankkennung: siehe DBID in <i>DDM-Parameter angeben</i>.</td> </tr> <tr> <td data-bbox="505 1415 802 1541">FNR</td> <td data-bbox="802 1415 1385 1541">Die Dateinummer: siehe FNR in <i>DDM-Parameter angeben</i>.</td> </tr> <tr> <td data-bbox="505 1541 802 1667">DDM Typ</td> <td data-bbox="802 1541 1385 1667">Der DDM-Typ: siehe DDM Type in <i>DDM-Parameter angeben</i>.</td> </tr> <tr> <td data-bbox="505 1667 802 1747">Bytes</td> <td data-bbox="802 1667 1385 1747">Die Größe des DDM in Bytes.</td> </tr> <tr> <td data-bbox="505 1747 802 1803"></td> <td data-bbox="802 1747 1385 1803"></td> </tr> </tbody> </table>			DBID	Die Datenbankkennung: siehe DBID in <i>DDM-Parameter angeben</i> .	FNR	Die Dateinummer: siehe FNR in <i>DDM-Parameter angeben</i> .	DDM Typ	Der DDM-Typ: siehe DDM Type in <i>DDM-Parameter angeben</i> .	Bytes	Die Größe des DDM in Bytes.		
DBID	Die Datenbankkennung: siehe DBID in <i>DDM-Parameter angeben</i> .													
FNR	Die Dateinummer: siehe FNR in <i>DDM-Parameter angeben</i> .													
DDM Typ	Der DDM-Typ: siehe DDM Type in <i>DDM-Parameter angeben</i> .													
Bytes	Die Größe des DDM in Bytes.													

Funktionscode	Feld	Funktion
		<p>Secur. Type</p> <p>Gilt nur, wenn Natural Security installiert ist.</p> <p>Der Security-Typ:</p> <ul style="list-style-type: none"> ■ Public ■ Private ■ Access ■ Undef <p>Siehe auch Status eines DDM im Abschnitt <i>Components of a File Profile</i> in der <i>Natural Security-Dokumentation</i>.</p> <hr/> <p>File Type</p> <p>Dateityp - Gilt nur bei DDMs, die aus VSAM erstellt wurden oder eine PersonalDB-Datei aus Super Natural.</p> <p>Der VSAM-Dateityp: <i>Log.View</i> (logische Datensicht), <i>Phy.File</i> (physische Datensicht) oder <i>Log.File</i> (logische Datei).</p> <p>Super Natural: <i>Userfile</i> zeigt an, dass das DDM in Super Natural wurde.</p> <hr/> <p>File Name</p> <p>Dateiname - Gilt nur bei DDMs aus VSAM-Dateien.</p> <p>Der Name einer logischen oder physischen Datei.</p> <hr/> <p>Remark</p> <p>Eine Bemerkung, z.B. <i>SupNat</i> (bei einer Benutzerdatei <i>Userfile</i> aus Super Natural) oder die VSAM-Dateiorganisation (<i>KSDS</i>, <i>RRDS</i>, <i>ESDS</i> oder <i>VRDS</i>).</p> <hr/> <p>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.</p>
S	Show Defined DBIDs and Used FNRs	See <i>Definierte Datenbankkennungen und verwendete Dateinummern anzeigen</i> .

Funktionscode	Feld	Funktion
M	Copy DDM to Another FDIC File	<p>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.</p> <p>Diese Funktion ruft das Fenster Copy DDMs der Natural Utility SYSMAIN auf, in dem Sie die Quell- und die Zielumgebung des oder der zu kopierenden DDMs angeben können. Weitere Informationen siehe <i>DDMs bearbeiten</i> in der <i>SYSMAIN Utility</i>-Dokumentation.</p>
G	Generate DDM from Adabas FDT	<p>Generiert ein DDM aus einer Adabas Feld Definition Table (FDT) und stellt es zur Weiterverarbeitung in den Source-Bereich.</p> <p>Wird als DBID eine 0 (Null) angegeben, dann wird die Standard-DBID benutzt, die im Profilparameter UDB im Natural-Parametermodul angegeben ist.</p>
B	SQL Services (NDB)	<p>Nur verfügbar, wenn Natural for DB2 (NDB) installiert ist.</p> <p>Die SQL Services (NDB) werden benutzt, um DDMs aus DB2-Tabellen zu generieren. Weitere Informationen siehe <i>SQL Services (NDB) (Natural for DB2)</i> in der <i>Database Management System Interfaces</i>-Dokumentation.</p>
D	DL/I Services	<p>Nur verfügbar, wenn Natural for DL/I installiert ist.</p> <p>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.</p> <p>Weitere Informationen zu <i>DL/I Services</i> siehe <i>Database Management System Interfaces</i>-Dokumentation.</p>
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 angeben, 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	<p>Der Name des Datendefinitionsmoduls, das verarbeitet werden soll.</p> <p>Sie können auch einen Namensbereich angeben, indem Sie Stern-Notation (*) benutzen, um alle DDMs oder nur bestimmte DDMs zu verarbeiten.</p> <p>Beispiel: Mit der Eingabe EMP* wählen Sie alle DDMs aus, deren Name mit EMP beginnt.</p> <p>Bei Bereichsangaben in den Funktionen Delete DDMs oder Copy DDM to Another FDIC File wird die SYSMAIN Utility aufgerufen, in der Sie die DDMs weiterverarbeiten können. Weitere Informationen siehe <i>DDMs bearbeiten</i> in der <i>SYSMAIN Utility</i> -Dokumentation.</p>

Feld	Erläuterung						
FNR	<p>Die Dateinummer (FNR) der Datenbankdatei, für die das Datendefinitionsmodul definiert wird bzw. werden soll.</p> <p>Die Dateinummer entspricht dem Typ der Datenbank. Wenn z.B. eine Adabas-Datei benutzt wird, müssen Sie die Adabas-Dateinummer eingeben.</p> <p>Wenn Sie einen DL/I-Segmenttyp benutzen, erfolgt die Angabe der Dateinummer intern durch Natural für DL/I.</p> <p>Bezüglich VSAM-Dateien siehe <i>Natural for VSAM</i> in der <i>Database Management System Interfaces</i>-Dokumentation.</p> <p>Gültige Werte: 0 - 5000.</p>						
DBID	<p>Die Kennung (DBID) der Datenbank, in der die Datenbankdatei enthalten ist, die von dem Datendefinitionsmodul referenziert wird.</p> <p>Gültige Werte: 1 - 65535, außer 255.</p> <p>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 <i>Parameter-Referenz</i>-Dokumentation.</p> <p>Wenn Sie keine DBID eingeben, wird diese dynamisch zur Ausführungszeit auf der Basis der DBID der Systemdatei FUSER generiert.</p>						
Replace	<p>Gibt an, ob ein DDM ersetzt werden soll:</p> <table border="1" data-bbox="336 1083 1383 1440"> <tr> <td data-bbox="336 1083 701 1125"></td> <td data-bbox="701 1083 1383 1125"></td> </tr> <tr> <td data-bbox="336 1125 701 1241">Y</td> <td data-bbox="701 1125 1383 1241">Ja. Ein DDM, das momentan kopiert oder katalogisiert wird, ersetzt ein DDM mit dem gleichen Namen.</td> </tr> <tr> <td data-bbox="336 1241 701 1440">N</td> <td data-bbox="701 1241 1383 1440">Nein. Existierende DDMs mit dem gleichen Namen werden nicht ersetzt. Dies ist die Standard-Einstellung.</td> </tr> </table> <p>Diese Funktion entspricht der REPLACE-Option des Kommandos CATALOG. Siehe <i>Editor- und Systemkommandos</i>.</p>			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.
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.						
FDIC Type	<p>Der Datenbanktyp der Systemdatei.</p> <p>Mögliche Typen wie bei DDM Type (siehe unten).</p> <p>Dieses Feld ist ein Ausgabefeld.</p>						
DDM Type	<p>Der Typ des Datendefinitionsmoduls, z.B.:</p> <table border="1" data-bbox="336 1787 1383 1833"> <tr> <td data-bbox="336 1787 745 1833"></td> <td data-bbox="745 1787 1383 1833"></td> </tr> </table>						

Feld	Erläuterung																		
	<table border="1"> <tr> <td data-bbox="431 243 841 321">A</td> <td data-bbox="841 243 1464 321">Adabas Dies ist der Standardtyp..</td> </tr> <tr> <td data-bbox="431 321 841 367">V</td> <td data-bbox="841 321 1464 367">VSAM</td> </tr> <tr> <td data-bbox="431 367 841 413">2</td> <td data-bbox="841 367 1464 413">DB2</td> </tr> <tr> <td data-bbox="431 413 841 459">D</td> <td data-bbox="841 413 1464 459">DL/I</td> </tr> <tr> <td data-bbox="431 459 841 506">P</td> <td data-bbox="841 459 1464 506">PROCESS (Entire System Server)</td> </tr> <tr> <td data-bbox="431 506 841 552">C</td> <td data-bbox="841 506 1464 552">Kommandoprozessor</td> </tr> <tr> <td data-bbox="431 552 841 598">S</td> <td data-bbox="841 552 1464 598">Super Natural</td> </tr> <tr> <td data-bbox="431 598 841 644">E</td> <td data-bbox="841 598 1464 644">Entire DB Engine</td> </tr> <tr> <td colspan="2" data-bbox="431 644 1464 762">Der Typ des Datendefinitionsmoduls entspricht dem Typ der Datenbank, die von dem DDM referenziert wird.</td> </tr> </table>	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.	
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	<p>Der Datenbanktyp der Datenbank, die im Feld DBID angegeben ist.</p> <p>Mögliche Typen wie bei DDM Type (siehe oben).</p> <p>Ausnahme: Bei einer Adabas-Datenbank wird die Adabas-Version (zum Beispiel 8) angezeigt.</p> <p>Dieses Feld ist ein Ausgabefeld.</p>																		

32 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 *SYSMAIN Utility*-Dokumentation.

Informationen, wie Sie Datendefinitionsmodule zwischen verschiedenen Systemdateien und/oder Hardware-Plattformen kopieren möchten, finden Sie im Abschnitt *DDMs in unterschiedlichen Umgebungen pflegen*.

33

DDM-Editor aufrufen und beenden

- Editor aufrufen 300
- Editor-Sitzung beenden 301

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
*      C=NNNNNNN
*      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*.

34 DDM-Editor benutzen

- DDM-Kopfzeilen-Informationen 304
- Spalten für Feld-Attribute 305
- Kommandos zum Editieren und Ausführen von Funktionen 311
- Erweiterte Feld-Attribute angeben 317

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 (<i>DDM-type</i>)	Der in Klammern neben dem Bildschirm-Titel Edit DDM angezeigte Wert bezeichnet den DDM-Typ, zum Beispiel: <table border="1" data-bbox="378 1108 1385 1522"> <tbody> <tr> <td>ADA</td> <td>Adabas</td> </tr> <tr> <td>VSAM</td> <td>VSAM</td> </tr> <tr> <td>DB2</td> <td>DB2</td> </tr> <tr> <td>DL/I</td> <td>DL/I</td> </tr> <tr> <td>PROCESS</td> <td>Entire System Server</td> </tr> <tr> <td>CMD-PROC</td> <td>Kommandoprozessor</td> </tr> <tr> <td>SNAT</td> <td>Super Natural</td> </tr> <tr> <td>ENTIREDB</td> <td>Entire DB Engine</td> </tr> </tbody> </table>	ADA	Adabas	VSAM	VSAM	DB2	DB2	DL/I	DL/I	PROCESS	Entire System Server	CMD-PROC	Kommandoprozessor	SNAT	Super Natural	ENTIREDB	Entire DB Engine
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 <i>DDM-Parameter angeben</i> .																
FNR	Die Nummer der Datei, die in der Datenbank referenziert wird. Weitere Informationen siehe FNR im Abschnitt <i>DDM-Parameter angeben</i> .																
DDM Name	Der Name des Datendefinitionsmoduls, das sich zurzeit im Arbeitsbereich des DDM-Editors befindet.																

Felder im DDM-Kopfbereich	Beschreibung
Def.Seq.	<p>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 <i>Statements</i>-Dokumentation.</p> <p>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.</p> <p>Der Inhalt dieses Feldes kann geändert werden.</p>

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 <i>Editor- und Systemkommandos</i> .
	S	Die mit diesem Buchstaben markierte Zeile enthält einen gesuchten Wert. Siehe auch SCAN in <i>Editor- und Systemkommandos</i> .
	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
	<p><i>blank</i> Elementarfeld. Dieser Feld-Typ kann Daten aufnehmen und enthält keine anderen Felder. Er kann innerhalb eines Datensatzes nur einen Wert haben.</p>
	<p>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 <i>Adabas-Dokumentation</i>.</p>
	<p>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.</p> <p>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.</p>
	<p>M Multiples Feld. Dieser Feld-Typ kann innerhalb eines Datensatzes mehr als einen Wert haben. Siehe auch <i>Multiple Felder im Leitfaden zur Programmierung</i>.</p>
	<p>P Periodengruppe. Eine Gruppe von Feldern, die innerhalb eines Datensatzes mehr als einen Wert haben kann. Siehe auch <i>Periodengruppen im Leitfaden zur Programmierung</i>.</p>
	<p>* Kommentarzeile.</p>
L	<p>Die einem Feld zugewiesene Level-Nummer.</p> <p>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 <i>Leitfaden zur Programmierung</i>).</p> <p>Gültige Level-Nummern sind 1 - 7.</p> <p>Level-Nummern müssen aufeinanderfolgend in aufsteigender Reihenfolge angegeben werden.</p>

Spaltenüberschrift	Feld-Attribute
DB	<p>Bei Adabas-Dateien wird in der Spalte DB der aus zwei Zeichen bestehende Kurzname des entsprechenden Feldes in der Datenbankdatei angezeigt.</p> <p>Bei DL/I-Segmenttypen wird in der Spalte DB der aus zwei Zeichen bestehende Code angezeigt, der in DL/I benutzt wird.</p> <p>Bei VSAM-Dateien siehe <i>Natural for VSAM</i> in der <i>Database Management System Interfaces</i>-Dokumentation.</p> <p>Bei Feldern des Typs C (siehe Attribut T) enthält diese Spalte den Kurznamen des Adabas-Deskriptors, der zur Dateikopplung benutzt wird.</p>
Name	<p>Der Name des Feldes.</p> <p>Dieser kann bei Adabas-Feldern und SQL-Spalten 3 - 32 Zeichen und bei DL/I-Namen 1 - 19 Zeichen lang sein.</p> <p>Bei der Generierung eines Feldes aus einer Adabas FDT weist der DDM-Editor einen Standard-Feldnamen <i>prefix</i>-FIELD zu. Beispiele: AA-FIELD für DB-Kurzname AA oder AUAL-FIELD für Kurzname Aa.</p> <p>Bei der Vergabe eines Namens gelten die Namenskonventionen für Benutzervariablen (siehe <i>Natural benutzen</i>-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).</p> <p>Der Feldname ist der Name, der in anderen Natural-Objekten (z.B. in einem Programm) benutzt wird, um das Feld zu referenzieren.</p> <p>Der Feldname ist innerhalb des gesamten Datendefinitionsmoduls eindeutig (unverwechselbar).</p> <p>Bei Feldern des Typs C (siehe Attribut T) enthält diese Spalte den Kurznamen des zur Dateikopplung verwendeten Adabas-Deskriptors.</p>
F	<p>Das Natural-Datenformat eines Elementarfeldes, z.B. A (alphanumerisch), P (gepackt numerisch) oder L (logisch).</p> <p>Informationen zu gültigen Natural-Datenformaten siehe <i>Format und Länge von Benutzervariablen</i> im <i>Leitfaden zur Programmierung</i>.</p>
Leng	<p>Die Standardlänge eines Elementarfeldes.</p> <p>Dieser Längenwert kann in einem Natural-Programm durch den Benutzer überschrieben werden.</p> <p>Bei numerischen Feldern (Natural-Datenformat N) wird die Länge in der Form <i>nn.m</i> angegeben, wobei <i>nn</i> die Anzahl der Stellen vor dem Dezimaltrennzeichen und <i>m</i> die Anzahl der Stellen nach dem Dezimaltrennzeichen ist.</p>

Spaltenüberschrift	Feld-Attribute										
	<p>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.</p> <p>Bei Feldern des Typs C (siehe Attribut T) enthält diese Spalte den Namen des zur Dateikopplung verwendeten Datendefinitionsmoduls.</p>										
S	<p>Gilt nicht bei VSAM..</p> <p>Nullwert-Unterdrückungsoption:</p> <table border="1" data-bbox="349 569 1377 1482"> <tr> <td data-bbox="349 569 553 615"></td> <td data-bbox="553 569 1377 615"></td> </tr> <tr> <td data-bbox="349 615 553 772"><i>blank</i></td> <td data-bbox="553 615 1377 772">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 data-bbox="349 772 553 890">F</td> <td data-bbox="553 772 1377 890">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 data-bbox="349 890 553 1283">N</td> <td data-bbox="553 890 1377 1283"> <p>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.</p> <p>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.</p> </td> </tr> <tr> <td data-bbox="349 1283 553 1482">M</td> <td data-bbox="553 1283 1377 1482">Zeigt an, dass das Feld mit der SQL-Nullwertoption <code>not null</code> 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>			<i>blank</i>	Zeigt an, dass die Adabas-Standardunterdrückung verwendet wird, d.h., dass nachfolgende Leerzeichen in alphanumerischen Feldern und vorangehende Nullen in numerischen Feldern unterdrückt werden.	F	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.	N	<p>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.</p> <p>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.</p>	M	Zeigt an, dass das Feld mit der SQL-Nullwertoption <code>not null</code> 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.
<i>blank</i>	Zeigt an, dass die Adabas-Standardunterdrückung verwendet wird, d.h., dass nachfolgende Leerzeichen in alphanumerischen Feldern und vorangehende Nullen in numerischen Feldern unterdrückt werden.										
F	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.										
N	<p>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.</p> <p>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.</p>										
M	Zeigt an, dass das Feld mit der SQL-Nullwertoption <code>not null</code> 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.										
D	<p>Der Adabas Deskriptor-Typ eines Elementarfeldes, das kein Array ist.</p> <p>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 S in 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.</p> <p>Mögliche Deskriptor-Typen sind:</p> <table border="1" data-bbox="349 1801 1377 1848"> <tr> <td data-bbox="349 1801 553 1848"></td> <td data-bbox="553 1801 1377 1848"></td> </tr> </table>										

Spaltenüberschrift	Feld-Attribute	
	<i>blank</i>	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. Für 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 (<i>collation descriptor</i>). 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.: <i>Collation 5 on AA</i> .
X	Alternativer Subdeskriptor oder Superdeskriptor, d.h. ein alternativer Index für eine VSAM-Datei.
	Informationen zu VSAM-Dateien siehe <i>Natural for VSAM</i> in der <i>Database Management System Interfaces</i> -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@xxxxxx-Feld ist zurzeit keine spezifische Abruffunktion zugewiesen. Es ist für zukünftige Erweiterungen reserviert.

Ein 0@xxxxxx-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 <i>nn</i> Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CX(nn)	Kopiert die Zeile, die mit <i>.X</i> einmal oder <i>nn</i> Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CY(nn)	Kopiert die Zeile, die mit <i>.Y</i> markiert ist, einmal oder <i>nn</i> Mal unter die Zeile, in der das Kommando eingegeben wurde.
.CX-Y(nn)	Kopiert den Zeilenblock einmal oder <i>nn</i> Mal, siehe Um einen Zeilenblock zu kopieren oder zu verschieben:
.D(nnnn)	Löscht die Zeile, in der das Kommando eingegeben wurde, oder löscht <i>nnnn</i> Zeilen ab der Zeile, in der das Kommando eingegeben wurde. Wenn <i>nnnn</i> nicht angegeben, dann wird standardmäßig eine Zeile gelöscht.
.Enn	Ruft die Funktion zum Bearbeiten der erweiterten Feld-Attribute auf. Siehe Erweiterte Feld-Attribute angeben.
.I(nn)	Fügt <i>nn</i> Leerzeilen unter der Zeile, in der das Kommando eingegeben wurde, ein. Für <i>nn</i> können Sie eine Zahl im Bereich von 1 bis 10 angeben. (Beim nächsten Drücken von Enter werden alle leer gelassenen Zeilen wieder entfernt.) Falls <i>nn</i> 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 <i>.X</i> markierte Zeile unter die Zeile, in der das Kommando eingegeben wurde.
.MY	Verschiebt die mit <i>.Y</i> 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 <i>X</i> in der Spalte I angezeigt.

Zeilenkommando	Funktion
	Siehe auch Um einen Zeilenblock zu kopieren oder zu verschieben: .
.Y	<p>Markiert die letzte Zeile eines Zeilenblocks zum Kopieren oder Verschieben.</p> <p>Eine markierte Zeile wird durch ein Y in der Spalte I angezeigt.</p> <p>Siehe auch Um einen Zeilenblock zu kopieren oder zu verschieben:.</p>

» **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 <code>.I</code> .
<u>C</u> ATALOG	Führt eine Syntaxprüfung durch und speichert den zurzeit Im Source-Bereich des Editors befindlichen DDM-Quellcode als katalogisiertes Objekt. <code>CATALOG [DDM-name] [REPLACE]</code> Falls der DDM-Quellcode bereits katalogisiert worden ist, müssen Sie die <code>REPLACE</code> -Option benutzen. Äquivalente PF-Taste: PF11
<u>C</u> HECK	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
<u>C</u> LEAR	Löscht den Source-Bereich. Weitere Informationen siehe entsprechendes Natural-Systemkommando CLEAR in der <i>Systemkommandos</i> -Dokumentation.
<u>C</u> ODEPAGE	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 <code>.X</code> markierte Zeile.
DY	Löscht die mit dem Zeilenkommando <code>.Y</code> markierte Zeile.
DX-Y	Löscht den mit den Zeilenkommandos <code>.X</code> und <code>.Y</code> eingegrenzten Zeilenblock.
EX	Löscht alle Zeilen oberhalb der mit dem Zeilenkommando <code>.X</code> markierten Zeile.
EY	Löscht alle Zeilen unterhalb der mit dem Zeilenkommando <code>.Y</code> markierten Zeile.
EX-Y	Löscht alle Zeilen mit Ausnahme des Blocks, der mit den Zeilenkommandos <code>.X</code> und <code>.Y</code> eingegrenzt ist.
HELP oder ?	Ruft den Editor Help Info -Bildschirm auf, der Hilfe-Informationen zu den Editor-Kommandos anzeigt.
LENGTH oder SIZE	$\left\{ \begin{array}{l} \text{LENGTH} \\ \text{SIZE} \end{array} \right\} [\textit{from-field to-field}]$
<u>L</u> IST DDM oder <u>L</u> IST <u>V</u> IEW	$\text{LIST} \left\{ \begin{array}{l} \text{DDM} \\ \text{VIEW} \end{array} \right\} [\textit{DDM-name}]$ Zeigt eine einzelne DDM-Source oder gemäß Angabe mit <i>DDM-name</i> eine Liste mit Datendefinitionsmodulen an. Weitere Informationen siehe Natural-Systemkommando LIST in der <i>Systemkommandos</i> -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 <code>Lower ON</code> angezeigt. Wenn Datendefinitionsmodule, die Kleinbuchstaben enthalten, zum Bearbeiten geöffnet werden, ist immer <code>Lower ON</code> gesetzt. Siehe auch UPPERCASE .

Kommando	Funktion
QUIT oder ·	<p>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.</p> <p>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 <i>Editor-Profil</i> im Abschnitt <i>Editoren – Allgemeine Informationen</i>.</p> <p>Äquivalente PF-Taste: PF3</p>
READ	<p><code>READ [DDM-name]</code></p> <p>Liest eine DDM-Source in den Source-Bereich. Falls sich zurzeit ein anderes DDM im Source-Bereich befindet, wird dieses überschrieben.</p>
RESET	<p>Entfernt alle Markierungen von Zeilen, die mit einem X (siehe Zeilenkommando <code>.X</code>), mit einem Y (siehe Zeilenkommando <code>.Y</code>) oder mit einem in der Spalte I angezeigten E (Fehler bei <code>CHECK</code>)markiert sind.</p>
SCAN	<p><code>SCAN [scan-value]</code></p> <p>Durchsucht den Quellcode nach einer als <code>scan-value</code> angegebenen Zeichenkette, zum Beispiel:</p> <p>SCAN ABC oder SCAN ABC D.</p> <p>Wird der angegebene Suchwert gefunden, dann wird die Zeile (bzw. die Zeilen), die den <code>scan-value</code> enthält, mit einem in der Spalte I angezeigten S markiert. (Drücken Sie Enter, um die Markierungen zu entfernen.)</p>
UNCATALOG	<p><code>UNCATALOG [DDM-name]</code></p> <p>Löscht ein oder mehrere Datendefinitionsmodule in der aktuellen Systemdatei FDIC, wenn <code>DDM-name</code> angegeben wird (siehe auch DDM Name im Abschnitt <i>DDM-Parameter angeben</i>).</p> <p>Wird <code>DDM-name</code> nicht angegeben, dann wird das zurzeit im Source-Bereich befindliche DDM gelöscht.</p> <p>Dieses Kommando entspricht dem Natural-Systemkommando UNCATALOG in der <i>Systemkommandos-Dokumentation</i>.</p>
UPPERCASE oder UC	<p>Gilt nur bei Datendefinitionsmodulen der Typen A (Adabas) und 2 (DB2).</p> <p>Aktiviert die automatische Umwandlung von Klein- und Großbuchstaben. Dies ist der Standard-Umwandlungsmodus.</p> <p>Siehe auch LOWERCASE.</p>

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 <code>.X</code> oder <code>.Y</code> markierte Zeile positionieren.
B oder ++	PF9	Nach unten zur letzten Seite blättern.
T oder --	PF6	Nach oben zur ersten Seite blättern.
+nn		Um <i>nn</i> Zeilen nach unten blättern.
-nn		Um <i>nn</i> 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.

35

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*.

36 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.

37

DDMs in unterschiedlichen Umgebungen pflegen

Sie können die Natural Utility SYSMAIN (siehe *Utilities*-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, OpenVMS und Windows) können Sie den Object Handler (siehe *Utilities*-Dokumentation) benutzen.

VIII

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 on z/OS, Installing Software AG Editor on BS2000 bzw. Installing Software AG Editor on z/VSE* in der *Natural Installation-Dokumentation* für das verwendete Betriebssystem
- *Operating the Software AG Editor* in der *Natural Operations-Dokumentation*
- *Editoren in der SPoD-Umgebung* in der *Natural Unicode- und Codepage-Unterstützung-Dokumentation*
- *SYSEDIT 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*

38

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 auch *General Defaults in Editor Profile in Editors - General Information*).

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
- z/VSE Members
- 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

39

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.

40

Benutzung des Editor-Bildschirms

▪ Benutzung der Eingabebereiche	335
▪ In den Daten im Editierbereich blättern	335
▪ Zu einer bestimmten Zeile springen	338
▪ Zeilen zeigen oder verbergen	339
▪ Ränder, Tabulator- und Spaltenpositionen anzeigen	340

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](#).

■ Präfixbereich

Die sechs Spalten am linken Rand des Editor-Bildschirms werden als Präfixbereich bezeichnet. Dieser Bereich kann eine Reihe von Apostrophen (' ' ' ' ' ') oder Sternchen (*****), eine Quellcode-Zeilenummer oder Text enthalten. Der Präfixbereich dient zur Eingabe von Zeilenkommandos. Weitere Informationen siehe [Zeilenkommandos \(Übersicht\)](#).

■ Editierbereich

Rechts neben dem Präfixbereich 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
<i>number</i>	Anzahl (<i>number</i>) der Zeilen, um die nach oben oder nach unten geblättert wird. Anzahl (<i>number</i>) 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 <i>n</i>	Blättert um <i>n</i> Zeilen vorwärts.
+ <i>n</i>	Blättert um <i>n</i> Zeilen vorwärts.
UP	Blättert um den im SCROLL-Feld angegebenen Betrag rückwärts.
UP <i>n</i>	Blättert um <i>n</i> Zeilen rückwärts.
- <i>n</i>	Blättert um <i>n</i> Zeilen rückwärts.
LEFT	Blättert um den im SCROLL-Feld angegebenen Betrag nach Links.
LEFT <i>n</i>	Blättert um <i>n</i> Spalten nach Links.
RIGHT	Blättert um den im SCROLL-Feld angegebenen Betrag nach Rechts.
RIGHT <i>n</i>	Blättert um <i>n</i> Spalten nach Rechts.
FIX <i>n</i>	Gibt die Anzahl <i>n</i> 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 Präfixbereich 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
<code>X</code>	Die markierte Zeile wird von der Anzeige ausgeschlossen.
<code>Xn</code>	Die nächsten <i>n</i> Zeilen werden von der Anzeige ausgeschlossen.
<code>XX</code>	Markiert die erste Zeile eines Datenblocks, der von der Anzeige ausgeschlossen werden soll. Ein zweites <code>XX</code> -Zeilenkommando ist nötig, um den Block abzugrenzen. Das Ausschließen erfolgt nach Eingabe des zweiten <code>XX</code> -Zeilenkommandos.
<code>F</code>	Bringt die markierte Zeile wieder zur Anzeige bzw. bringt die erste Zeile des ausgeschlossenen Blocks zur Anzeige.
<code>Fn</code>	Bringt die ersten <i>n</i> Zeilen des ausgeschlossenen Blocks wieder zur Anzeige bzw. bringt die erste Zeile des ausgeschlossenen Blocks zur Anzeige.
<code>Ln</code>	Bringt die letzten <i>n</i> 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
<code>BNDS</code>	Zeigt die Randpositionen auf dieser Zeile an.
<code>COLS</code>	Zeigt die Spaltenpositionen auf dieser Zeile an.
<code>TABS</code>	Zeigt die Tabulatorpositionen auf dieser Zeile an.

Weitere Informationen siehe [Daten anlegen und ändern](#).

41 Kommandos benutzen

- Ausführung eines Kommandos 342

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`.

42

Daten anlegen und ändern

▪ Zeilen einfügen und löschen	346
▪ Zeilen kopieren, verschieben, überlagern und wiederholen	348
▪ Ein Fenster mit Daten kopieren oder verschieben	352
▪ Horizontale und vertikale Bereichsgrenzen einstellen	358
▪ Maskenzeile definieren	360
▪ Daten zwischen Bereichsgrenzen anordnen	361
▪ Daten zentrieren	362
▪ Daten ausrichten	363
▪ Physischen oder logischen Tabulator benutzen	366
▪ Zeilen in alphabetischer Reihenfolge sortieren	371
▪ Zeichenkette suchen	371
▪ Zeichenkette ersetzen	375

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.
Dn	Löscht die nächsten <i>n</i> 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 <i>.X</i> markierte Zeile.
DY	Löscht die mit <i>.Y</i> markierte Zeile.
DX-Y	Löscht den Zeilenblock ab der mit <i>.X</i> markierten Zeile bis zur mit <i>.Y</i> 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 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. Wenn Sie eine Maskenzeile definiert haben und das Kommando MASK ist auf ON gesetzt, wird die Maskenzeile eingefügt, wenn Sie dieses Kommando absetzen.
In	Fügt <i>n</i> 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 <i>POWER-Kommando</i> .
W	Öffnet ein einzeliges Fenster. Es wird keine neue Zeile eingefügt, wenn Sie in dieses Fenster Daten eingeben und Enter drücken.
Wn	Öffnet ein Fenster mit <i>n</i> 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, Cn, CC) oder Verschieben (M, Mn, MM). Die Daten werden <i>nach</i> dieser Zeile eingefügt.
B	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (C, Cn, CC) oder Verschieben (M, Mn, MM). Die Daten werden <i>vor</i> dieser Zeile eingefügt.
C	Kopiert diese Zeile an die Stelle, die mit einem Zeilenkommando A, B or O markiert ist.
Cn	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.
Mn	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, Cn, CC) oder zum Verschieben (M, Mn, MM). Die Daten werden mit dieser Zeile zusammengeführt. (Es werden nur Leerzeichen <i>innerhalb</i> der Bereichsgrenzen geändert). Siehe auch Beispiel für das Überlagern von Daten .
On	Markiert diese Zeile und die nächsten n Zeilen als Zielzeilen für ein Kopier-Zeilenkommando (C, Cn, CC) oder Verschieben-Zeilenkommando (M, Mn, 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
00	<p>Markiert die erste Zeile eines Zielzeilenblocks für ein Kopier-Zeilenkommando (C, Cn, CC) oder Verschiebe-Zeilenkommando (M, Mn, MM). Ein zweites Zeilenkommando 00 ist nötig, um die letzte Zeile des Zielzeilenblocks zu markieren.</p> <p>Die kopierten oder verschobenen Zeilen werden mit diesen Zeilen zusammengeführt, d.h. Leerzeichen in den Zeilen werden überlagert.</p> <p>Siehe auch Beispiel für das Überlagern von Daten.</p>
R	Wiederholt diese Zeile einmal.
Rn	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.
RRn	<p>Wiederholt den Block, der durch zwei Zeilenkommandos RRn eingegrenzt ist, n Mal.</p> <p>Klammerkommandos) (siehe weiter unten) verschieben Text um die angegebene Anzahl an Spalten, jedoch nur <i>innerhalb</i> der angegebenen Begrenzungen, deshalb ist es möglich, dass der Text teilweise verschwindet.</p>
)	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 eingegeben 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	<p>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 eingegeben worden ist. Zwei nicht qualifizierte Zeilenkommandos ((verschieben den Block um eine Spalte nach Links.</p> <p>Wenn Sie die Symbole > (größer als) oder < (kleiner als) benutzen (siehe unten), um Daten zu verschieben, dann ist das Verschieben <i>innerhalb</i> der gesetzten Begrenzungen maximal bis zum nächsten Zeichen, das kein Leerzeichen ist, möglich.</p>
>	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 >> eingegeben 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
<< <i>n</i>	Markiert die erste Zeile eines um <i>n</i> Spalten nach Links zu verschiebenden Blocks. Ein zweites Zeilenkommando << ist nötig, um den Block einzugrenzen. Das Verschieben erfolgt, nachdem das zweite Kommando << eingegeben worden ist. Zwei nicht qualifizierte Zeilenkommandos << verschieben den Block um eine Spalte nach Links.

Beispiel für das Überlagern von Daten

Mit einem Überlagerungszeilenkommando (0, 0*n* oder 00) 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')
000140 //STEPLIB DD
000150 //          DD
000160 //          DD
000170 //          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 00 (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	<p>Definiert ein Fenster. Als Parameter werden die Startzeile und Startspalte und die Endzeile und Endspalte angegeben. Es muss mindestens ein Parameter angegeben werden. Beispiele:</p> <p>WINDOW 5 10 24 13</p> <p>Definiert ein Fenster, das in Zeile 5/Spalte 24 beginnt und in Zeile 10/Spalte 13 endet.</p> <p>WINDOW 5 10 24</p> <p>Definiert ein Fenster, das in Zeile 5/Spalte 24 beginnt und in Zeile 10/letzte Spalte endet.</p> <p>WINDOW 5 10</p> <p>Definiert ein Fenster, das in Zeile 5/erste Spalte beginnt und in Zeile 10/letzte Spalte endet.</p> <p>WINDOW 5 5</p> <p>Definiert ein Fenster, das in Zeile 5/erste Spalte beginnt und in Zeile 5/letzte Spalte endet.</p>
CWINDOW	<p>Kopiert ein Fenster, das mit dem WINDOW-Kommando definiert wurde. Optionale Parameter geben die Zeile an, bei der das Fenster eingefügt werden soll. Beispiele:</p> <p>CWINDOW 5</p> <p>Kopiert das Fenster nach Zeile 5.</p> <p>CWINDOW 5 24</p> <p>Spaltet Zeile 5 bei Spalte 24 und kopiert das Fenster zwischen die beiden Teile.</p>
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 Anwendung 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`, `T0`, `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, setzen 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:

```
I 2
```

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.
T0	Markiert eine Zeile, die ausgerichtet werden soll.
T00	Markiert die erste Zeile eines Datenblocks, der ausgerichtet werden soll. Ein zweites Zeilenkommando T00 ist nötig, um den Block einzugrenzen. Die Ausrichtung erfolgt, nachdem das zweite T00-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 T00 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==> BND5 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 `T00` (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](#)
- [Ausrichtungparameter 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 bestimmten 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.

Ausrichtungparameter 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 Ausrichtungparameter einstellen

Um Daten in einer bestimmten Spalte mit einer bestimmten Einrückung tabellarisch darzustellen, sind mehrere logische Tabulatorzeichen und gemischte Ausrichtungparameter 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 Ausrichtungparameter 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.

43 Editor-Profil einstellen

- Editor-Profil-Einstellungen 381

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-Profiles 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-Profiles 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
ms0	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 <i>n</i>
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 <i>n m</i>
user	Aktuell angemeldeter Benutzer. Nicht änderbarer Parameter.	Keines

44

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.

45 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
<< <i>n</i>	Markiert die erste Zeile eines um <i>n</i> 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 <i>n</i> , CC) oder Verschieben (M, M <i>n</i> , MM). Die kopierte(n) oder verschobene(n) Zeile(n) wird bzw. werden <i>nach</i> dieser Zeile eingefügt.
B	Markiert die Zielzeile für ein Zeilenkommando zum Kopieren (C, C <i>n</i> , CC) oder Verschieben (M, M <i>n</i> , MM). Die kopierte(n) oder verschobene(n) Zeile(n) wird bzw. werden <i>vor</i> dieser Zeile eingefügt.
BNDS	Zeigt die Begrenzungspositionen in dieser Zeile.
C	Kopiert diese Zeile an die Stelle, die mit einem Zeilenkommando A, B oder 0 markiert ist.
C <i>n</i>	Kopiert die nächsten <i>n</i> Zeilen an die Position, die mit einem Zeilenkommando A, B oder 0 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 0 angegeben worden ist.
CX	Kopiert die mit . <i>X</i> markierte Zeile. Fügt die Daten nach dieser Zeile ein.
CY	Kopiert die mit . <i>Y</i> markierte Zeile. Fügt die Daten nach dieser Zeile ein.
CX- <i>Y</i>	Kopiert den Zeilenblock ab der Zeile, die mit . <i>X</i> markiert ist, bis zur Zeile, die mit . <i>Y</i> markiert ist. Fügt die Daten nach dieser Zeile ein.
COLS	Zeigt die Spaltenpositionen in dieser Zeile an.
D	Löscht diese Zeile.
D <i>n</i>	Löscht die nächsten <i>n</i> 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 . <i>X</i> markierte Zeile.
DY	Löscht die mit . <i>Y</i> markierte Zeile.
DX- <i>Y</i>	Löscht den Zeilenblock ab der mit . <i>X</i> markierten Zeile bis zur mit . <i>Y</i> markierten Zeile.
F	Schließt die erste ausgeschlossene Zeile ein.
F <i>n</i>	Schließt die ersten <i>n</i> ausgeschlossenen Zeilen ein.
I	<p>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.</p> <p>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).</p> <p>Sie können außerdem eine eingefügte Zeile mit einem vordefinierten Inhalt füllen. (Siehe auch Hauptkommando MASK).</p>

Zeilenkommando	Erläuterung
I <i>n</i>	Fügt <i>n</i> 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 <i>n</i>	Schließt die letzten <i>n</i> ausgeschlossenen Zeilen ein.
LC	Ändert diese Zeile auf Kleinschreibung.
LC <i>n</i>	Ändert die folgenden <i>n</i> 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 0 markiert worden ist.
M <i>n</i>	Verschiebt die nächsten <i>n</i> Zeilen an die Position, die mit einem Zeilenkommando A, B oder 0 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 0 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 <i>n</i>) 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.
0	Markiert diese Zeile als Zielzeile für ein Zeilenkommando zum Verschieben (M, M <i>n</i> , MM) oder zum Kopieren (C, C <i>n</i> , CC). Die verschobene(n) oder kopierte(n) Zeile(n) wird (werden) mit dieser Zeile zusammengeführt; d.h., es werden Leerzeichen in der Zeile überlagert.
0 <i>n</i>	Markiert die folgenden <i>n</i> Zeilen als Zielzeilen für ein Zeilenkommando zum Verschieben (M, M <i>n</i> , MM) oder zum Kopieren (C, C <i>n</i> , CC). Die verschobenen oder kopierten Zeilen werden mit diesen Zeilen zusammengeführt; d.h., es werden Leerzeichen in den Zeilen überlagert.

Zeilenkommando	Erläuterung
00	<p>Markiert die erste Zeile eines Zielzeilenblocks für ein Verschiebe-Zeilenkommando (M, Mn, MM) oder Kopier-Zeilenkommando (C, Cn, CC). Ein zweites Zeilenkommando 00 ist nötig, um die letzte Zeile des Zielzeilenblocks zu markieren.</p> <p>Die kopierten oder verschobenen Zeilen werden mit diesen Zeilen zusammengeführt, d.h. Leerzeichen in den Zeilen werden überlagert.</p>
R	Wiederholt diese Zeile einmal.
Rn	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.
RRn	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.
TFn	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).
T0	Führt diese Zeile mit der nächsten Zeile zusammen.

Zeilenkommando	Erläuterung
T00	Markiert die erste Zeile eines Datenblocks, der innerhalb der eingestellten Grenzen zusammengeführt werden soll. Ein zweites Zeilenkommando T00 ist nötig, um die letzte Zeile des zusammenzuführenden Blocks zu markieren. Die Funktion wird ausgeführt, nachdem das zweite Kommando T00 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.
UCn	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.
Wn	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.
WCn	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.
WMn	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.
WSn	Das Datenfenster beginnt in Spalte n dieser Zeile.
.X	Diese Zeile wird mit .X markiert.
X	Diese Zeile wird ausgeschlossen.
Xn	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.

46 Hauptkommandos (Übersicht)

▪ Syntax-Elemente	393
▪ Gemeinsame Kommando-Optionen	394
▪ ADVANCE	396
▪ AORDER	397
▪ AUTOREN	397
▪ AUTOSAVE	398
▪ BNDS	398
▪ BOTTOM	399
▪ CANCEL	399
▪ CAPS	399
▪ CEnter	400
▪ CHANGE	400
▪ COLS	403
▪ CURSOR	404
▪ CWINDOW	404
▪ DELETE	404
▪ DOWN	406
▪ DWINDOW	406
▪ EMPTY	407
▪ END	407
▪ ESCAPE	407
▪ EXCLUDE	408
▪ FIND	410
▪ FIX	413
▪ HEX	413
▪ INCLUDE	413
▪ JLEFT	414
▪ JRIGHT	414
▪ JUSTIFY	415
▪ LABEL	416
▪ LC	417
▪ LEFT	418

▪ LIMIT	419
▪ LOCATE	419
▪ LOG	420
▪ MASK	421
▪ MWINDOW	421
▪ NULLS	421
▪ ORDER	422
▪ POWER	423
▪ PROFILE	423
▪ PROTECT	423
▪ RCHANGE	424
▪ RECOVERY	424
▪ RENUMBER	425
▪ RESET	425
▪ RFIND	426
▪ RIGHT	426
▪ SORT	426
▪ TABS	427
▪ TOP	428
▪ UC	428
▪ UNDO	429
▪ UNREN	429
▪ UP	430
▪ WINDOW	430
▪ XSWAP	431

Dieser Abschnitt enthält eine Kurzbeschreibung der einzelnen Hauptkommandos, die beim Software AG Editor zur Verfügung stehen, und eine vollständige Übersicht über die Kommando-syntax.

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

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 <i>nicht</i> 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

ADVANCE	<input type="checkbox"/> ON <input type="checkbox"/> OFF <input type="checkbox"/> PAGE
---------	--

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

$\left\{ \begin{array}{l} \text{AUTOSAVE} \\ \text{ASAVE} \end{array} \right\} \left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right]$

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

$\text{BNDS} \left[\begin{array}{l} n \ m \\ n \end{array} \right]$
--

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](#), [T0](#), [LJ](#) oder [RJ](#)).

<i>n</i>	Die Nummer der Spalte, bei der die linke Begrenzung gesetzt werden soll.
<i>m</i>	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

CEnter $\left\{ \begin{array}{l} \text{ALL} \\ n \\ n\ m \end{array} \right\}$

Mit diesem Kommando können Sie Daten zentrieren.

ALL	Zentriert die Daten in allen Zeilen.
<i>n</i>	Zentriert die Daten ab Zeile <i>n</i> bis zur letzten Zeile.
<i>n m</i>	Zentriert die Daten ab Zeile <i>n</i> bis zur Zeile <i>m</i> .

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

$\left\{ \begin{array}{l} \text{CHANGE} \\ \text{CHG} \end{array} \right\} \left[\begin{array}{l} * \\ [T]'string1' \\ C' string1' \\ X' string1' \\ P' string1' \end{array} \right] \left\{ \begin{array}{l} * \\ [X]'string2' \end{array} \right\}$

$\left[\begin{array}{l} .X \\ .X .Y \end{array} \right] \left[\begin{array}{l} n \\ n\ m \end{array} \right] \left[\begin{array}{l} \text{ALL} \\ \text{NEXT} \\ \text{PREV} \\ \text{FIRST} \\ \text{LAST} \end{array} \right] \left[\begin{array}{l} \text{CHARS} \\ \text{WORD} \\ \text{PREFIX} \\ \text{SUFFIX} \end{array} \right] \left[\begin{array}{l} \text{NX} \\ X \end{array} \right]$

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' <i>string1</i> '	Ersetzt die Zeichenkette <i>string1</i> unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
' <i>string1</i> '	Wie bei T' <i>string1</i> '.
C' <i>string1</i> '	Ersetzt die Zeichenkette <i>string1</i> nur, wenn sie genauso wie angegeben auftritt.
X' <i>string1</i> '	Ersetzt die Zeichenkette, die der angegebenen Zeichenkette <i>string1</i> im Hexadezimalformat entspricht. Ersetzt sie durch die Zeichenkette <i>string2</i> im Hexadezimalformat.
P' <i>string1</i> '	Ersetzt die Zeichenkette <i>string1</i> , 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 $\left[\begin{matrix} n \\ n \ m \end{matrix} \right]$

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 $\left[\begin{matrix} * \\ [T]'string' \\ 'string' \\ C' string' \\ X' string' \\ P' string' \end{matrix} \right] \left[\begin{matrix} .X \\ .X \ .Y \end{matrix} \right] \left[\begin{matrix} n \\ n \ m \end{matrix} \right] \left[\begin{matrix} ALL \\ NEXT \\ PREV \\ FIRST \\ LAST \end{matrix} \right] \left[\begin{matrix} CHARS \\ WORD \\ PREFIX \\ SUFFIX \end{matrix} \right] \left[\begin{matrix} NX \\ X \end{matrix} \right]$

Mit diesem Kommando können Sie Zeilen löschen.

Sie können angeben, dass nur Zeilen, die eine angegebene Zeichenkette (*string*) enthalten, gelöscht werden sollen:

T' <i>string</i> '	Löscht Zeilen, die die Zeichenkette <i>string</i> enthalten, unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
' <i>string</i> '	Wie bei T' <i>string</i> '.
C' <i>string</i> '	Löscht Zeilen, die die Zeichenkette <i>string</i> exakt wie angegeben enthalten.
X' <i>string</i> '	Löscht Zeilen, die die Zeichenkette enthalten, die der angegebenen Zeichenkette (<i>string</i>) im Hexadezimalformat entspricht.
P' <i>string</i> '	Löscht Zeilen, die die Zeichenkette <i>string</i> 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`, `Dn` 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

EMPTY	[ON OFF]
-------	---------------

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:

END

Wenn `AUTOSAVE` auf `OFF` gesetzt ist und wenn Sie Daten geändert haben, werden Sie aufgefordert, ein `SAVE`- oder `CANCEL`-Kommando abzusetzen.

ESCAPE

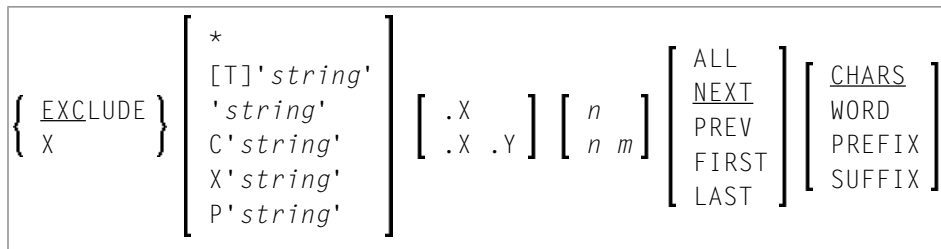
ESCAPE	[ON OFF]	[<i>character</i>]
--------	---------------	----------------------

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:

<code>T 'string'</code>	Schließt alle Zeilen aus, die die Zeichenkette (<i>string</i>) enthalten, unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
<code>'string'</code>	Wie bei <code>T 'string'</code> .
<code>C 'string'</code>	Schließt alle Zeilen aus, die die Zeichenkette (<i>string</i>) exakt wie angegeben enthalten
<code>X 'string'</code>	Schließt Zeilen aus, die die Zeichenkette enthalten, die der angegebenen Zeichenkette (<i>string</i>) im Hexadezimalformat entspricht.
<code>P 'string'</code>	Schließt Zeilen aus, die die Zeichenkette <i>string</i> 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
<code>*</code>	Verwendet die Suchzeichenkette, die in einem vorigen Kommando angegeben wurde (zum Beispiel FIND , CHANGE , EXCLUDE).
<code>.X</code>	Erklärung siehe Zeilenangabe .
<code>.X .Y</code>	

<i>n</i> <i>n m</i>	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
```

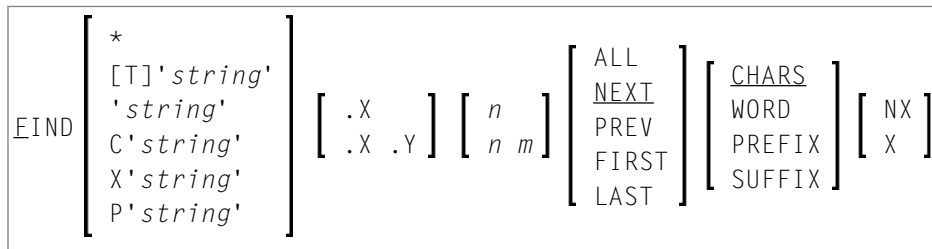
Dieses Kommando schließt Zeile 10 von der Anzeige aus.

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' <i>string</i> '	Sucht nach der Zeichenkette unabhängig davon, ob diese in Klein- oder Großschreibung vorliegt. Dies ist die Standardeinstellung.
' <i>string</i> '	Wie bei T' <i>string</i> '.
C' <i>string</i> '	Sucht nach der Zeichenkette exakt wie in <i>string</i> angegeben.
X' <i>string</i> '	Sucht nach der Zeichenkette, die der angegebenen Zeichenkette im Hexadezimalformat entspricht.
P' <i>string</i> '	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	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 .

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

```
FIX [ ON ]
    [ OFF ] n
```

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

```
HEX [ ON ]
     [ OFF ]
```

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

```
INCLUDE [ *
         [T]'string'
         'string'
         C'string'
         X'string'
         P'string' ] [ .X ] [ n ] [ n m ] [ ALL
         NEXT
         PREV
         FIRST
         LAST ] [ CHARS
                 WORD
                 PREFIX
                 SUFFIX ]
```

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

JLEFT	{ ALL }
	{ n }
	{ n m }

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

JRIGHT	{ ALL }
	{ n }
	{ n m }

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

JUSTIFY	{	LEFT	}
		RIGHT	
		BOTH	

Mit diesem Kommando können Sie den Ausrichtungsmodus für die Zeilenkommandos `T0` und `T00` einstellen.

Die Zeilenkommandos `T0` und `T00` 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 T0 (oder einen Zeilenblock mit zwei Zeilenkommandos T00) 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:

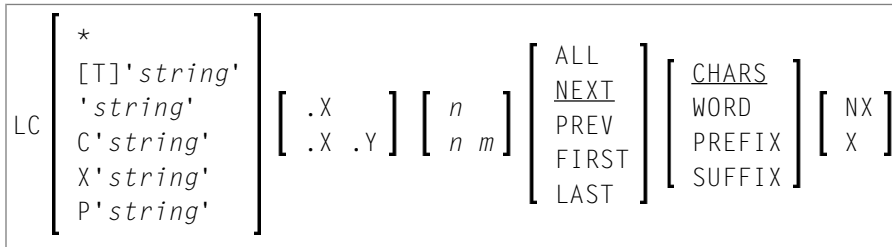
Um die aktuelle Zeile mit `.X` zu markieren, geben Sie folgendes Kommando ein:

```
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' <i>string</i> '	Ändert die Zeichenkette <i>string</i> unabhängig davon, ob sie in Klein- oder in Großschreibung auftritt. Dies ist die Standardeinstellung.
' <i>string</i> '	Wie bei T' <i>string</i> '.
C' <i>string</i> '	Ändert Zeilen, die die Zeichenkette <i>string</i> exakt wie angegeben enthalten.
X' <i>string</i> '	Ändert Zeilen, die die Zeichenkette enthalten, die der angegebenen Zeichenkette <i>string</i> im Hexadezimalformat entspricht.
P' <i>string</i> '	Ändert die Zeilen, die eine Zeichenkette (<i>string</i>) 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 <i>string</i> enthalten, die in dem vorigen Kommando verwendet wurde, in dem eine Zeichenkette angegeben wurde.
.X	Erklärung siehe Zeilenangabe .
.X .Y	
n	Erklärung siehe Spaltenangabe .

<i>n m</i>	
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

```
LEFT [ n ]  
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] $\left\{ \begin{array}{l} 0 \\ n \\ .label \end{array} \right\}$

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 identisch 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.

<i>n</i>	Die Nummer der Zeile, in der das Datenfenster eingefügt werden soll.
<i>m</i>	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

ORDER	$\left\{ \begin{array}{l} \text{ALL} \\ n \\ n \ m \end{array} \right\}$
-------	--

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`, `T0` und `T00` 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-Bildschirms 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 <code>TABS</code> angegeben) an.
7	Wie bei 6, zusätzlich wird die Maskenzeile (so wie mit dem Kommando <code>MASK</code> angegeben) angezeigt.
8	Wie bei 7, zusätzlich werden die Bereichsgrenzen (so wie mit dem Kommando <code>BNDS</code> angegeben) angezeigt.
9	Wie bei 8, zusätzlich werden die Spaltennummern (so wie mit dem Kommando <code>COLS</code> angegeben) angezeigt.

PROTECT

```
PROTECT [ INS ]
          [ ON  ]
          [ OFF ]
```

Mit diesem Kommando können Sie den **Präfixbereich** schützen. Um bei einem geschützten 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` $\left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right] [n]$

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.
<i>n1</i>	Inkrement der Nummerierung (der Standardwert ist in Ihrem Editor-Profil festgelegt).
<i>n2</i>	Start-Spalte für die neue Zeilennummerierung (Standardwert: 73).
<i>n3</i>	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 $\left[\begin{matrix} n \\ \text{RIGHT} \end{matrix} \right]$

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 $[n\ m]$ $\left[\begin{matrix} \cdot X \\ \cdot X \cdot Y \end{matrix} \right]$ $\left[\begin{matrix} A \\ D \end{matrix} \right]$

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 .
$\cdot X$	Sortiert ab der mit $\cdot X$ markierten Zeile bis zum Ende des Objekts.
$\cdot X \cdot Y$	Sortiert ab der mit $\cdot X$ markierten Zeile bis zu der mit $\cdot Y$ markierten Zeile (dabei sind $\cdot X$ und $\cdot Y$ 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

TABS	[ON [tab-character]]
		OFF	
		[
		LEFT]
		RIGHT	
		DECIMAL	
			[tab-character] [column...]

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	*				
	[T]'string'			ALL	CHARS
	'string'	.X	n	NEXT	WORD
	C' string'	.X .Y	n m	PREV	PREFIX
	X' string'			FIRST	SUFFIX
P' string'			LAST		NX
					X

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.
<i>n</i>	Die letzten <i>n</i> Änderungen werden rückgängig gemacht.

UNREN

```
UNREN [ n m ]
```

Deaktiviert die Neunummerierung von Zeilen.

<i>n</i>	Start-Spalte für die neue Zeilennummerierung (Standardwert: 73).
<i>m</i>	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 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, 346
- align
 - text in Software AG Editor, 363

C

- catalog
 - source in Programm-Editor, 53
- center
 - line in Software AG Editor, 362
- column
 - show position in Software AG Editor, 340
- command
 - cursor-sensitive commands
 - in Programm-Editor, 52
 - Editor-Kommandos in Programm-Editor, 36
 - line commands in Software AG Editor, 385
 - main commands in Software AG Editor, 391
- copy
 - line in Software AG Editor, 348
 - specific text section
 - in Software AG Editor, 352
- cursor position
 - assign to PF key
 - in Programm-Editor, 48
- customize
 - features of Programm/Data Area Editor, 17
 - features of Software AG Editor, 379

D

- Data Area Editor
 - customize editor features, 17
 - Zeilenkommandos, 75
- DDM
 - indicator field, 310
- delete
 - line in Software AG Editor, 346

E

- editor screen
 - split to show two sources, 14
- editors

overview of, ix

- exit
 - from Programm-Editor, 54
 - from Software AG Editor, 383

F

- find
 - text in Software AG Editor, 371
- Format
 - Einstellungen im Masken-Editor, 113
- format
 - center line in Software AG Editor, 362

H

- help
 - for SYSDDM utility functions, 286
- hide
 - text in Software AG Editor, 339

I

- indicator
 - for field in DDM, 310

K

- Kommando
 - Data Area Editor, 75
 - Zeilenkommandos im Programm-Editor, 45

L

- line
 - add/delete in Software AG Editor, 346
 - center in Software AG Editor, 362
 - copy in Software AG Editor, 348
 - move in Software AG Editor, 348
 - order in target zone in Software AG Editor, 361
 - repeat with mask in Software AG Editor, 360
 - show at top of screen in Software AG Editor, 338
 - show/hide in Software AG Editor, 339
 - sort in Software AG Editor, 371
- lock
 - activate/deactivate for object, 7
(Siehe auch unlock)
- lower case
 - change in Programm-Editor, 35

M

- Map-Editor
 - Formateinstellungen, 113
- Masken-Editor
 - Formateinstellungen, 113
 - Masken-Standard Einstellungen, 107
- merge
 - lines in Software AG Editor, 348
- move
 - line in Software AG Editor, 348
 - specific text section
 - in Software AG Editor, 352

O

- open
 - Programm-Editor, 32
 - Software AG Editor, 331

P

- Profil
 - Masken-Standard Einstellungen
 - Masken-Editor, 107
- profile
 - set for Programm/Data Area Editor, 17
 - set for Software AG Editor, 379
- Programm-Editor
 - change lower/upper case, 35
 - cursor sensitive commands, 52
 - customize editor features, 17
 - Editor-Kommandos, 36
 - exit from, 54
 - open, 32
 - PF key for cursor position, 48
 - save/catalog source, 53
 - use, 31
 - Zeilenkommandos, 45

Q

- Quellcode
 - save/catalog in Programm-Editor, 53

R

- replace
 - text in Software AG Editor, 375

S

- save
 - source in Programm-Editor, 53
 - source in Software AG Editor, 383
- scroll
 - data in Software AG Editor, 335
- show
 - column position in Software AG Editor, 340
 - tab position in Software AG Editor, 340
 - target zone in Software AG Editor, 340

- text in Software AG Editor, 339
- Software AG Editor
 - add/remove line, 346
 - align text, 363
 - center line, 362
 - copy/move line, 348
 - copy/move specific text section, 352
 - customize editor features, 379
 - define target zone for command, 358
 - exit from, 383
 - find text, 371
 - line commands, 385
 - main commands, 391
 - merge lines, 348
 - open, 331
 - order lines in target zone, 361
 - repeat line with template, 360
 - replace text, 375
 - save source, 383
 - scroll data, 335
 - set tab, 366
 - show line at top of screen, 338
 - show target zone/tab/column, 340
 - show/hide text, 339
 - sort lines, 371
- sort
 - lines in Software AG Editor, 371
- source code
 - save in Software AG Editor, 383
- split
 - editor screen, 14
- SYSDDM utility
 - help information for, 286

T

- tab
 - set in Software AG Editor, 366
 - show position in Software AG Editor, 340
- target zone
 - define for command in Software AG Editor, 358
 - show in Software AG Editor, 340
- template
 - for new line in Software AG Editor, 360
- terminate
 - Programm-Editor, 54
 - Software AG Editor, 383
- text
 - align in Software AG Editor, 363
 - copy/move specific section
 - in Software AG Editor, 352
 - show/hide in Software AG Editor, 339

U

- unlock
 - object, 10
 - (Siehe auch lock)
- upper case
 - change in Programm-Editor, 35
- use
 - Programm-Editor, 31