

—  
Komfortabler  
Bildschirmeditor  
MAXed (SINIX)

—  
Benutzerhandbuch

## **... und Schulung?**

Unsere SINIX-Kurse in  
München – Berlin – Essen – Frankfurt –  
Hannover – Wien – Zürich  
helfen Ihnen Betriebssystem, Kommu-  
nikation und Software optimal  
anzuwenden und Software effizient  
zu entwickeln.

**Zentrale Auskunft und Info-Material:  
Telefon (0 89) 92 75-33 32**

Siemens AG  
Schule für Daten- und  
Informationstechnik  
DI Schule S3  
Postfach 83 09 51, D-8000 München 83

Bestell-Nr. U2635-J-Z95-4  
Printed in the Federal Republic of Germany  
12210 AG 1902. (15270)

SINIX ist der Name der Siemens-Version des  
Softwareproduktes XENIX.  
SINIX enthält Teile, die dem Copyright © von Microsoft (1982)  
unterliegen; im übrigen unterliegt es dem Copyright von Siemens.  
Die Rechte an dem Namen SINIX stehen Siemens zu.  
XENIX ist ein Warenzeichen der Microsoft Corporation.  
XENIX ist aus UNIX-Systemen unter Lizenz der Firma AT&T entstanden.  
UNIX ist ein Warenzeichen der Bell Laboratories.

Copyright an der Übersetzung Siemens AG, 1984, alle Rechte  
vorbehalten.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und  
Mitteilung ihres Inhaltes nicht gestattet, soweit nicht ausdrücklich  
zugestanden.

Zuwerdhandlungen verpflichten zu Schadenersatz.  
Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung  
oder GM-Eintragung.

Liefermöglichkeiten und technische Änderungen vorbehalten.

**Copyright © Siemens AG 1990**

Alle Rechte vorbehalten.

Herausgegeben vom Bereich  
Daten- und Informationstechnik  
Postfach 830951, D-8000 München 83

Siemens Aktiengesellschaft

---

## Vorwort

Der MAXed V3.0 ist ein Allround-Bildschirmeditor für die SINIX-Systeme ab Version 5. Er eignet sich insbesondere für Software-Entwickler.

Dieses Handbuch beschreibt, wie Sie mit dem MAXed arbeiten und was er leistet. Es enthält:

- eine Einführung in den MAXed
- mehrere thematisch ausgerichtete Beispielsitzungen
- eine allgemeine Beschreibung, welche von "Grundlagen" bis zu "Erweiterten Möglichkeiten" die prinzipielle Handhabung des MAXed erläutert und funktionale Zusammenhänge deutlich macht
- eine Beschreibung der Aufrufmöglichkeiten des MAXed
- eine alphabetisch geordnete Beschreibung aller MAXed-Kommandos
- eine Beschreibung für den Umgang mit MAXed-Dienstprogrammen und MAXed-Steuerprogrammen
- viele eingestreute Anwendungsbeispiele in den Beschreibungen

### Wie lernen Sie den MAXed kennen?

Einen allgemeinen Überblick vermitteln Ihnen die "Einführenden Informationen" in Kapitel 1 und die beiliegende MAXed-Kurzbeschreibung.

Steigen Sie ein mit den Beispielsitzungen in Kapitel 2, wobei Sie sich, je nach Ihrem Kenntnisstand im Umgang mit dem MAXed, einen geeigneten Abschnitt herausgreifen. Lesen Sie sich parallel dazu in die thematisch passenden Abschnitte der Kapiteln 3 "Grundlagen" und 4 "Erweiterte Möglichkeiten" ein. Hier werden Sie auch später noch nützliche Hinweise und Detailinformationen zu bestimmten Kommando-Gruppen, also MAXed-Kommandos mit gleichem Einsatzgebiet finden.

### Wie werden Sie "MAXed-Experte"?

Diesem Ziel nähern Sie sich durch das Studium der Kapitel 3 und 4 (siehe oben), in denen Sie auch durch Verweise auf die weiteren Kapitel über die zugehörigen Beschreibungen zum Aufruf des MAXed (Kapitel 5), zu den einzelnen MAXed-Kommandos und Tasten-Makros (Kapitel 6) sowie zu den Dienstprogrammen und MAXed-Steuerprogrammen (Kapitel 7) unterrichtet werden.

---

In den Kapiteln 5 bis 7 sind alle Aufrufmöglichkeiten des MAXed und seiner Dienstprogramme, sämtliche MAXed-Kommandos mit ihren zahlreichen Attributierungsmöglichkeiten und die C-Schnittstellen-Funktionen für MAXed-Steuerprogramme zum Nachschlagen und weiteren Vertiefen aufgeführt.

### **Eine Bitte an Sie**

Sagen Sie uns, wie Sie mit dem Manual zurechtkommen. Wenn Sie zufrieden sind, dann sind wir auf dem richtigen Weg. Teilen Sie uns aber auch Ihre "Stolpersteine" mit, damit wir sie aus dem Weg räumen können.

**Manualredaktion DI ST QM 2**  
Otto-Hahn-Ring 6, 8000 München 83

---

## Änderungsprotokoll

### Änderung des Vorgänger-Handbuchs

Stand April 1988 (MAXed Version 2.0A)

durch die Neuauflage

vom Januar 1990 (MAXed Version 3.0A).

Der MAXed wurde von der Version 2 zur Version 3 auf den 8-Bit-Zeichensatz *ISO-8859/1* umgestellt. Daraus ergaben sich kleine Änderungen an der Benutzeroberfläche. Die MAXed-Dateien sind jedoch aufwärtskompatibel (siehe Anhang: Dateikonvertierung, Makros und Steuerprogramme aus der Version 2). Außerdem laufen Meldungen und *On-Line*-Dokumentation des MAXed nun über *NLS* (siehe Abschnitt 4.1.10 Die Sprach-Modi: Der Konversationsmodus).

### Neugestaltung des Handbuchs

Das Handbuch wurde zu einem Großteil neu erstellt. Dabei sind folgende Kapitel hervorzuheben:

- Kapitel 2 besteht aus mehreren unabhängig bearbeitbaren Beispielsitzungen, die jeweils einen eigenen Themenbereich haben und sich im Schwierigkeitsgrad steigern.
- Kapitel 3 bietet eine Einführung in die Arbeitsweise und den Kommandoumfang des MAXed. Kommandos sind zu Funktionsgruppen zusammengefaßt und die einzelnen Bestandteile des MAXed werden aufgeführt und beschrieben.
- Kapitel 4 setzt die Grundlagen von Kapitel 3 mit der Beschreibung weiterer Möglichkeiten fort.

### Referenzkarten

Der MAXed ist nun für verschiedene Terminaltypen konfigurierbar. Hierzu werden mehrere Referenzkarten mitgeliefert, die Ihnen die Tastenbelegungen verschiedener Datensichtstationen darstellen.

## Neue MAXed-Kommandos

Kommando	Funktion
CHLINELEN	Wird für bestimmte Terminaltypen bereitgestellt, die die zwischen verschiedenen Zeilenlängen umschaltbar sind.
ENDLINE, STARTLINE	Ersetzen als MAXed-Kommandos die früheren Tasten-Makros <LINE END> und <LINE START>.
QUIT	Beendet den MAXed ohne zusätzliches Sichern.
UNDO	Mit diesem Kommando können Sie schrittweise oder in einem definierten Umfang alle Textveränderungen seit Sitzungsbeginn rückgängig machen und wieder vorwärts durchspielen.
WAIT	Wartet auf die Beendigung eines EXECUTE-Kommandos und kann nur in Makros oder MAXed-Steuerdateien verwendet werden.

## Erweiterungen bei Kommandoattributierungen

Kommandoattribut	Betroffene Kommandos	Funktion
	DRAW	Schaltet den Zeichnen-Modus ein, in dem Sie mit den ARROW-Kommandos Linien zeichnen können.
a / b	DRAW	Stellt eine Linienstärke für den Zeichnen-Modus ein.
@f	EDIT	Mit <EDIT> kann zwischen zwei alternativen Schreibmarkenpositionen in derselben Datei hin- und hergesprungen werden.
@	ENTER	Blendet die Pseudo-Datei SETUP mit verschiedenen Parameter-Einstellungen des MAXed ein.
@v/@*/@>	ENTER	Zur Textersetzung im Eingabefeld gibt es weitere Parameter.
= string	ENTER	Wertet den string im Eingabefeld aus.

Kommando- attribut	Betroffene Kommandos	Funktion
` [proc] [ID]	GOTO	Erstellt die Pseudo-Datei TAGLIST und erzeugt dabei ggf. mit dem Kommando ctags Tag-Datei. Ersetzt ein ähnliches Kommando mit EDIT.
" [PAT]" [ID]	GOTO	Führt die Funktion FILE-SCAN aus und erstellt die Pseudo-Datei SCANLIST zur Auflistung gefundener Textstellen in den angeg. Dateien.
"	GOTO	Blendet die Pseudo-Datei SCANLIST wieder ein.
n/h/v/m	ARROW-Kommandos	Erlaubt verschiedene Einstellungen des Fenster-Modus, sodaß sich der Dateiausschnitt automatisch mit der Schreibmarke verschiebt oder ein Sprung in ein benachbartes Fenster möglich ist.
*\/</>/<>	FIND-Kommandos	Erlaubt verschiedene Einstellungen des Such-Modus mit intelligenter Suchfunktion oder bestimmten voreingestellten Sonderzeichen.
De / En	FILL, FORMAT	Stellt den Trennmodus für den Silbentrenner ein.
,	ERASE, PICK, ERASEWORD, PICKWORD	Beim Löschen bzw. Kopieren des aktuellen Wortes sind die folgenden Leerzeichen inbegriffen.

## Änderungen bei Dienst- und Steuerprogrammen

maxlpr	maxlpr ruft die Druckroutine lpr standardmäßig selbst auf. Mit der Umstellung auf 8859/1 erhielt maxlpr auch neue Schalter.
mdefine	FONT-Zeichen werden nun in Quelldateien für MACRO-Bibliotheken direkt eingegeben.
m_control	Die C-Funktion erhielt eine neue Definition der Argumentliste



---

# Inhalt

	<b>Seite</b>
<b>1</b>	<b>Einführende Informationen . . . . . 1-1</b>
1.1	Welche Vorzüge hat der MAXed? . . . . . 1-2
1.2	Zusätzliche Hilfsmittel . . . . . 1-4
1.2.1	Die Tastaturschablone . . . . . 1-4
1.2.2	Die Referenzkarten . . . . . 1-4
1.2.3	Die MAXed-Kurzbeschreibung . . . . . 1-5
1.3	Die verwendete Metasyntax . . . . . 1-6
1.4	MAXed-Kommandos in Quelldateien . . . . . 1-8
1.5	Besonderheiten bei der Texteingabe . . . . . 1-8
1.6	Hilfsinformationen . . . . . 1-9
<b>2</b>	<b>Beispielsitzungen . . . . . 2-1</b>
2.1	Beispielsitzung 1 - Grundlegende Kommandos . . . . . 2-4
2.2	Beispielsitzung 2 - Puffer und Textbereiche . . . . . 2-21
2.3	Beispielsitzung 3 - Pseudo-Dateien und Eingabefeld . . . . . 2-32
2.4	Beispielsitzung 4 - Suchen und Ersetzen . . . . . 2-43
2.5	Beispielsitzung 5 - Komplexes Suchen und Ersetzen . . . . . 2-54
2.6	Beispielsitzung 6 - Suchen, Sammeln und Gestalten . . . . . 2-67
<b>3</b>	<b>Wie arbeitet der MAXed? - Grundlagen . . . . . 3-1</b>
3.1	Die Dateihaltung des MAXed . . . . . 3-2
3.2	Starten und Beenden des MAXed . . . . . 3-6
	Den MAXed starten . . . . . 3-6
	Den MAXed beenden . . . . . 3-8
3.3	Arbeiten mit der aktuellen Datei . . . . . 3-9
3.3.1	Einblenden und Sichern einer Datei . . . . . 3-9
	Datei im aktuellen Fenster einblenden . . . . . 3-9
	Datei neu anlegen . . . . . 3-9
	Datei sichern . . . . . 3-10
	Datei auf einen früheren Versionsstand zurücksetzen . . . . . 3-11
3.3.2	Bewegen der Schreibmarke . . . . . 3-12
3.3.3	Verschieben des Dateiausschnitts . . . . . 3-17
3.4	Arbeiten mit mehreren Dateien . . . . . 3-21
3.4.1	Wechseln zwischen Dateien . . . . . 3-21
3.4.2	Dateiausschnitte in mehreren Fenstern bearbeiten . . . . . 3-25
3.5	Der Bildschirm des MAXed . . . . . 3-29
3.5.1	Das aktuelle Fenster . . . . . 3-32
3.5.2	Die Steuerzeile . . . . . 3-34
3.6	Prinzipielles zur Attributierung von Kommandos . . . . . 3-36
3.6.1	Das Format der Kommandoattributierung . . . . . 3-37
	Eine Alternative zum Abschicken von MAXed-Kommandos . . . . . 3-39

3.6.2	Das Eingabefeld und der Eingabepuffer . . . . .	3-41
	Verwenden des Eingabepuffers . . . . .	3-43
3.6.3	Sonderzeichen für die Kommandoattributierung . . . . .	3-45
3.6.4	Textersetzung im Eingabefeld . . . . .	3-46
	Parameter für Textersetzung . . . . .	3-46
	Ersetzen von MAXed-Variablen . . . . .	3-53
	Ersetzen von PICK-Puffernamen . . . . .	3-56
	Ersetzen von SINIX-Kommandoaufrufen . . . . .	3-57
	Entwertung und Auswertung im Eingabefeld . . . . .	3-58
3.7	Die Kommandoverstärkung . . . . .	3-63
3.8	Der IDENTIFIER als Kommandoattribut . . . . .	3-64
	Definition und Anwendung von IDENTIFIER . . . . .	3-64
3.8.1	Der Dateiname als IDENTIFIER . . . . .	3-67
	Sonderzeichen der IDENTIFIER für Dateien . . . . .	3-68
	Kopieren eines gesamten Dateiinhalts . . . . .	3-72
3.8.2	Der ERASE-Puffername als IDENTIFIER . . . . .	3-74
	Sonderzeichen der IDENTIFIER für ERASE-Puffer . . . . .	3-78
	Kopieren von Pufferinhalten zwischen ERASE-Puffern . . . . .	3-81
	Austauschen zweier Textbereiche . . . . .	3-82
3.8.3	Der PICK-Puffername als IDENTIFIER . . . . .	3-83
	Sonderzeichen der IDENTIFIER für PICK-Puffer . . . . .	3-86
	Kopieren von Pufferinhalten zwischen PICK-Puffern . . . . .	3-90
3.8.4	Der Markenname als IDENTIFIER . . . . .	3-91
	Sonderzeichen der IDENTIFIER für Marken . . . . .	3-92
3.8.5	Der Makroname als IDENTIFIER . . . . .	3-94
	Sonderzeichen der IDENTIFIER für Makros . . . . .	3-95
3.8.6	Der Name eines Steuerprogramms als IDENTIFIER . . . . .	3-98
3.9	Die Bereichsdefinition als Kommandoattribut . . . . .	3-99
	Die Bereichstypen . . . . .	3-99
	Die Eingabe der Bereichsgrenzen . . . . .	3-102
	Die Bedeutung des Bereichstyps . . . . .	3-103
3.9.1	Die Bereichsdefinition durch Bewegen der Schreibmarke . . . . .	3-105
	Die Definition eines RANGE-Bereichs . . . . .	3-105
	Die Definition eines RECTANGLE-Bereichs . . . . .	3-106
	Die Definition eines STREAM-Bereichs . . . . .	3-107
	Die Eingabe weiterer Kommandoattribute . . . . .	3-108
	Spezielle RECTANGLE- und STREAM-Bereiche . . . . .	3-109
3.9.2	Die Bereichsdefinition durch Zeichenketten-Attribut . . . . .	3-112
	Kombinieren von Zeichenketten-Attribut und Schreibmarkenposition . . . . .	3-112
	Sonderzeichen für die Bereichsdefinition . . . . .	3-112
	RANGE-Bereiche . . . . .	3-114
	RECTANGLE-Bereiche . . . . .	3-116
	STREAM-Bereiche . . . . .	3-118
	BLOCK als Spezialfall eines STREAM-Bereichs . . . . .	3-119

	RECORD als Spezialfall eines STREAM-Bereichs . . . . .	3-122
3.10	PATTERN als Kommandoattribut . . . . .	3-124
	Passende Texteinheiten bei der Suche . . . . .	3-125
	Der Aufbau eines PATTERNs . . . . .	3-127
3.10.1	Die Sonderzeichen des Standard-Suchmodus . . . . .	3-128
3.10.2	Erweiterte reguläre Ausdrücke . . . . .	3-135
3.10.3	Textersetzung innerhalb von PATTERN . . . . .	3-136
3.10.4	Verschiedene Einstellungen des Such-Modus . . . . .	3-137
3.11	Die Ersetzungszeichenkette als Kommandoattribut . . . . .	3-138
3.11.1	Die Sonderzeichen der Ersetzungszeichenkette . . . . .	3-140
3.11.2	Textersetzung innerhalb von Ersetzungszeichenketten . . . . .	3-145
3.12	Voreingestellte Kommandoattribute . . . . .	3-148
3.12.1	Kommandos mit konstanter Standard-Ausführung . . . . .	3-149
	Die aktuelle Zeile . . . . .	3-149
	Der aktuelle Zeilenrest . . . . .	3-150
	Das aktuelle Wort . . . . .	3-151
	Der aktuelle Absatz . . . . .	3-151
	Der aktuelle Absatzrest . . . . .	3-152
	Das aktuelle Fenster . . . . .	3-152
3.12.2	Kommandos mit Standard-Puffern . . . . .	3-153
	Die Alternativ-Datei . . . . .	3-153
	Der Standard-ERASE-Puffer . . . . .	3-154
	Der Standard-PICK-Puffer . . . . .	3-155
	Die Standard-Marke . . . . .	3-156
	Das aktuelle Makro . . . . .	3-157
	Das Standard-Makro . . . . .	3-158
3.12.3	Kommandos mit aktuell gespeicherten Kommandoattributen . . . . .	3-159
	Das aktuelle SINIX-Kommando . . . . .	3-159
	Das aktuelle Filter-Programm . . . . .	3-160
	Das aktuelle PATTERN . . . . .	3-161
	Die aktuelle Ersetzungszeichenkette . . . . .	3-162
	Gespeichertes PATTERN bei FILE-SCAN . . . . .	3-162
<b>4</b>	<b>Wie arbeitet der MAXed? - Erweiterte Möglichkeiten . . . . .</b>	<b>4-1</b>
4.1	Die Betriebs-Modi des MAXed . . . . .	4-2
4.1.1	Die Eingabe-Modi . . . . .	4-2
	Der Einfüge-Modus INSERT . . . . .	4-4
	Der Überschreibe-Modus OVERWRITE . . . . .	4-6
	Der Einfüge-Modus WORD . . . . .	4-8
	Der Einfüge-Modus LEXICAL . . . . .	4-10
	Der Einfüge-Modus STREAM . . . . .	4-11
4.1.2	Der Zeichnen-Modus . . . . .	4-12
4.1.3	Der Zeilenschaltungs-Modus . . . . .	4-14
	Der Zeilenschaltungs-Modus NORMAL . . . . .	4-14
	Der Zeilenschaltungs-Modus SPACE . . . . .	4-15

	Der Zeilenschaltungs-Modus EMPTY . . . . .	4-15
	Der Zeilenschaltungs-Modus TEXT . . . . .	4-16
4.1.4	Der Fenster-Modus . . . . .	4-17
	Der Fenster-Modus NORMAL . . . . .	4-18
	Der Fenster-Modus HORIZONTAL . . . . .	4-18
	Der Fenster-Modus VERTICAL . . . . .	4-19
	Der Fenster-Modus MOVE . . . . .	4-19
	Der Fenster-Modus CHANGEWINDOW . . . . .	4-20
4.1.5	Der Such-Modus . . . . .	4-21
	Der Standard-Such-Modus . . . . .	4-21
	Der \-Such-Modus . . . . .	4-22
	Der <-Such-Modus . . . . .	4-23
	Der >-Such-Modus . . . . .	4-23
	Der <>-Such-Modus . . . . .	4-24
4.1.6	Der Makro-Abarbeitungs-Modus . . . . .	4-25
4.1.7	Der Kommando-Abarbeitungs-Modus . . . . .	4-26
4.1.8	Der Ausgabe-Modus . . . . .	4-28
	Die Einstellungen OVERWRITE und APPEND . . . . .	4-28
	Die Einstellungen ECHO und NOECHO . . . . .	4-29
4.1.9	Der Zurücksetzen-Modus . . . . .	4-30
	Die Makrobehandlung . . . . .	4-30
	Die Dateiauswahl . . . . .	4-31
4.1.10	Die Sprach-Modi . . . . .	4-32
	Der Konversationsmodus . . . . .	4-32
	Der Trennmodus . . . . .	4-34
4.1.11	Der Lern-Modus . . . . .	4-35
4.2	Die Pseudo-Dateien . . . . .	4-37
4.2.1	Die Pseudo-Datei HELP . . . . .	4-40
4.2.2	Die Pseudo-Datei RESTORE . . . . .	4-41
4.2.3	Die Pseudo-Datei OPENFILES . . . . .	4-42
4.2.4	Die Pseudo-Datei EDITFILES . . . . .	4-43
4.2.5	Die Pseudo-Datei PUTFILES . . . . .	4-45
4.2.6	Die Pseudo-Datei ERASEBUFFERS . . . . .	4-46
4.2.7	Die Pseudo-Datei PICKBUFFERS . . . . .	4-47
4.2.8	Die Pseudo-Datei LABELS . . . . .	4-48
4.2.9	Die Pseudo-Datei FINDHISTORY . . . . .	4-49
4.2.10	Die Pseudo-Datei MACROS . . . . .	4-50
4.2.11	Die Pseudo-Datei SYSOUTPUT . . . . .	4-52
4.2.12	Die Pseudo-Datei SYSHISTORY . . . . .	4-53
4.2.13	Die Pseudo-Datei SYSCOMMAND . . . . .	4-54
4.2.14	Die Pseudo-Datei SCANLIST . . . . .	4-55
4.2.15	Die Pseudo-Datei TAGLIST . . . . .	4-58
4.2.16	Die Pseudo-Datei VARIABLES . . . . .	4-60
4.2.17	Die Pseudo-Datei SETUP . . . . .	4-61
4.3	Das Formatieren . . . . .	4-63

4.3.1	Tabulatoren . . . . .	4-64
4.3.2	Formatränder . . . . .	4-66
4.3.3	Flutter- und Blocksatz . . . . .	4-69
	Der Silbentrenner des MAXed . . . . .	4-71
4.3.4	Ausrichten und Vereinheitlichen . . . . .	4-73
4.3.5	Umbrechen und Zusammenfügen . . . . .	4-76
4.4	Die Zeichenattribute . . . . .	4-79
4.4.1	Einstellen und Aktivieren von Zeichenattributen . . . . .	4-81
	Setzen und Aktivieren in einem Kommando . . . . .	4-84
4.4.2	Hinzufügen und Entfernen von Zeichenattributen . . . . .	4-87
4.4.3	Zeichnen von Linien . . . . .	4-88
4.5	Die SINIX-Schnittstellen . . . . .	4-90
	Aufrufen von SINIX-Kommandos . . . . .	4-90
	Die Sub-Shells des MAXed . . . . .	4-91
4.5.1	Wechseln in die CLI-Sub-Shell . . . . .	4-92
4.5.2	Ausführen asynchroner SINIX-Kommandos . . . . .	4-94
4.5.3	Ausführen von Filter-Programmen . . . . .	4-99
4.5.4	Ausdrucken von MAXed-Dateien . . . . .	4-101
4.6	Arbeiten mit Textbausteinen . . . . .	4-103
4.7	Arbeiten mit Makros . . . . .	4-105
4.8	Arbeiten mit MAXed-Steuerprogrammen . . . . .	4-107
<b>5</b>	<b>Der Aufruf des MAXed . . . . .</b>	<b>5-1</b>
5.1	maxed - Den MAXed aufrufen . . . . .	5-2
5.2	Die letzte MAXed-Sitzung rekonstruieren . . . . .	5-7
<b>6</b>	<b>Die Kommandos . . . . .</b>	<b>6-1</b>
6.1	Die MAXed-Kommandos . . . . .	6-1
	ALIGN - Text an der Position der Schreibmarke ausrichten . . . . .	6-2
	ATTRIBUTE - Zeichenattribute setzen . . . . .	6-3
	BACKSPACE - Einzelne Zeichen löschen . . . . .	6-6
	BACKTAB - Tabulatorsprung nach links . . . . .	6-7
	CANCEL - Abbruch eines Kommandos . . . . .	6-9
	CENTER - Text zentrieren . . . . .	6-10
	CHLINELEN - Zwischen verschiedenen Zeilenlängen umschalten . . . . .	6-11
	CHWINDOW - Zwischen Fenstern wechseln . . . . .	6-12
	CLI - Den MAXed vorübergehend verlassen . . . . .	6-13
	CODE - Steuerzeichen eingeben . . . . .	6-15
	COMPARE - Textstellen vergleichen . . . . .	6-18
	COMPRESS - Überflüssige Leerzeichen entfernen . . . . .	6-20
	DEFDIR - Dateiverzeichnis anzeigen/wechseln . . . . .	6-21
	DELCHAR - Ein Zeichen löschen . . . . .	6-22
	DOWNARROW - Die Schreibmarke nach unten bewegen . . . . .	6-23
	DRAW - Linien und Umrahmungen zeichnen . . . . .	6-25

EDIT - Eine Datei anzeigen/erstellen . . . . .	6-28
ENDLINE - Sprung an das Zeilenende . . . . .	6-33
ENTER - Kommandoattribute eingeben . . . . .	6-34
ERASE - Textbereiche löschen . . . . .	6-38
EXCHANGE - Textbereiche vertauschen . . . . .	6-41
EXECUTE - Shell-Kommandos aufrufen . . . . .	6-44
EXIT - Den MAXed verlassen . . . . .	6-48
FILL - Im Flattersatz formatieren . . . . .	6-50
FILTER - Ein FILTER-Programm aufrufen . . . . .	6-53
FONT - Den Zeichensatz umschalten . . . . .	6-55
FORMAT - Im Blocksatz formatieren . . . . .	6-57
GOTO - Positionieren auf ausgewählten Textstellen . . . . .	6-60
HELP - Hilfsinformationen abrufen . . . . .	6-63
HOME - Schnelle Bewegungen der Schreibmarke . . . . .	6-64
HWINDOW - Den Bildschirm horizontal teilen . . . . .	6-66
INSERTMODE - Den Eingabemodus wählen . . . . .	6-69
JUSTIFY - Text links- und rechtsbündig ausrichten . . . . .	6-71
LABEL - Textstellen markieren . . . . .	6-72
LEARN - Ein Makro anlegen . . . . .	6-74
LEFTARROW - Die Schreibmarke nach links bewegen . . . . .	6-79
LINEFEED - Zeilenweises Verschieben des Dateiausschnitts . . . . .	6-81
LOWER - Kleinschreibung . . . . .	6-82
MACRO - Ein Makro ausführen . . . . .	6-83
MARGIN - Linken und rechten Formatrand setzen . . . . .	6-85
MINUSFIND - Textstelle suchen . . . . .	6-87
MINUSLINE - In der Datei zurückblättern . . . . .	6-91
MINUSPAGE - Seitenweise zurückblättern . . . . .	6-92
PICK - Textbereiche zwischenspeichern . . . . .	6-93
PLUSFIND - Textstelle suchen . . . . .	6-96
PLUSLINE - In der Datei vorblättern . . . . .	6-100
PLUSPAGE - Seitenweise vorblättern . . . . .	6-101
PRINT - Text ausdrucken . . . . .	6-102
PUT - Zwischengespeicherte Textbereiche einsetzen . . . . .	6-103
QUIT - Beenden des MAXed ohne Sichern . . . . .	6-108
REFRESH - Den Bildschirm neu schreiben . . . . .	6-109
REPLACE - Ersetzen von Textstellen . . . . .	6-110
RESTOR - Gelöschte Textbereiche wieder einsetzen . . . . .	6-114
RETURN - Zeilenschaltung . . . . .	6-116
RIGHTARROW - Die Schreibmarke nach rechts bewegen . . . . .	6-119
RVLINEFEED - Zeilenweises Verschieben des Dateiausschnitts . . . . .	6-121
SAVE - Eine Datei sichern . . . . .	6-122
SETATTRIB - Zeichenattribute setzen/löschen . . . . .	6-125
SPACE - Textbereiche mit Leerzeichen beschreiben . . . . .	6-129
SRTAB - Tabulatoren setzen/löschen . . . . .	6-131

	STARTLINE - Sprung an den Zeilenanfang . . . . .	6-133
	TAB - Tabulatorsprung nach rechts . . . . .	6-134
	UNDO - Textveränderungen rückgängig machen . . . . .	6-136
	UPARROW - Die Schreibmarke nach oben bewegen . . . . .	6-139
	UPEND - Die Schreibmarke ans vorige Zeilenende bewegen . . . . .	6-141
	UPPER - Großschreibung . . . . .	6-142
	VWINDOW - Den Bildschirm vertikal teilen . . . . .	6-143
	WAIT - Warten auf das Ende eines asynchronen Programms . . . . .	6-146
	WINL - Den Dateiausschnitt nach links verschieben . . . . .	6-147
	WINR - Den Dateiausschnitt nach rechts verschieben . . . . .	6-148
	WORDLEFT - Die Schreibmarke wortweise nach links bewegen . . . . .	6-149
	WORDRIGHT - Die Schreibmarke wortweise nach rechts bewegen . . . . .	6-150
	WRAP - Text in die nächste Zeile bringen . . . . .	6-151
6.2	MACRO-Definitionen zur Tastenbelegung . . . . .	6-152
<b>7</b>	<b>Dienstprogramme und Steuerprogramme des MAXed . . . . .</b>	<b>7-1</b>
7.1	Eine MACRO-Bibliothek erstellen . . . . .	7-2
7.1.1	Makros in einer MACRO-Bibliothek anlegen . . . . .	7-4
7.1.2	Textbausteine in einer MACRO-Bibliothek anlegen . . . . .	7-7
7.1.3	Beispiel für eine MACRO-Bibliothek . . . . .	7-9
7.2	Den Inhalt einer MACRO-Bibliothek anzeigen . . . . .	7-11
7.3	Eine neue Konfigurationsdatei erstellen . . . . .	7-13
7.4	Eine MAXed-Datei für den Ausdruck aufbereiten . . . . .	7-16
7.5	MAXed-Steuerprogramme . . . . .	7-21
7.5.1	Die C-Funktionen . . . . .	7-21
	Den MAXed steuern . . . . .	7-22
	Eingabe von der Tastatur lesen . . . . .	7-24
	Inhalt der aktuellen Zeile anfordern . . . . .	7-25
	Den MAXed-Status abfragen. . . . .	7-26
	MAXed-Zeichenketten-Argumente ersetzen . . . . .	7-27
7.5.2	MAXed-Steuerprogramme übersetzen und aufrufen . . . . .	7-29
7.5.3	Beispiele zu Steuerprogrammen . . . . .	7-31
<b>A</b>	<b>Anhang . . . . .</b>	<b>A-1</b>
	Maximalwerte bei der Dateigröße . . . . .	A-1
	Erstellen einer Tag-Datei . . . . .	A-2
	convert - Dateikonvertierung . . . . .	A-3
	Makros und Steuerprogramme aus der Version 2 . . . . .	A-4

**Fachwörter**  
**Literatur**  
**Stichwörter**



# 1 Einführende Informationen

Die Anforderungen bei der Programmierung und bei der Erstellung von Dokumenten sind gewachsen. Der MAXed wächst ebenfalls mit den gestiegenen Ansprüchen – er erreichte gerade Versionsalter 3.0 –, um Ihnen die Arbeit mit Dateien zu erleichtern.

Der MAXed ist ein Bildschirmeditor, mit dem Sie ASCII-Dateien bzw. Dateien mit dem Standard-8-Bit-Zeichensatz *ISO-8859/1* erstellen und bearbeiten können. In letzterem sind zusätzlich nationale und andere Sonderzeichen definiert (siehe Abschnitt 1.5 Besonderheiten bei der Texteingabe). Außerdem hält der MAXed einen speziellen Grafik-Zeichensatz und weitere Zeichenattribute bereit, mit denen Sie Ihre Texte gestalten können.

Wenn Sie bereits mit anderen Bildschirmeditoren vertraut sind, werden Sie beim MAXed dieselbe Grundkonzeption wiedererkennen: Der MAXed-Bildschirm stellt Ihnen ein Fenster auf Ihre Dateien zur Verfügung, damit Sie Texte erstellen und bearbeiten können. Der "MAXimale editor" bietet jedoch, über die Grundfunktionen eines Editors hinaus, komfortable Möglichkeiten zur Bearbeitung von Dateien und zur Programmerstellung in einer Shell-Umgebung.

Abschnitt 1.1 bietet Ihnen einen Überblick über den Funktionsumfang des MAXed.

Abschnitt 1.2 informiert Sie über mitgelieferte Hilfsmittel für die Arbeit mit dem MAXed: die Tastaturschablonen, die Referenzkarten und die MAXed-Kurzbeschreibung.

Abschnitt 1.3 erklärt Ihnen die in diesem Handbuch verwendete Metasyntax zur Darstellung von MAXed-Kommandos.

Der Abschnitt 1.4 stellt Ihnen die Angabe von MAXed-Kommandos innerhalb von Quelldateien für MACRO-Bibliotheken und MAXed-Steuerdateien vor.

Im Abschnitt 1.5 erfahren Sie die Besonderheiten für die Eingabe von Sonderzeichen.

Abschnitt 1.6 wird Sie schließlich kurz über verfügbare Hilfsinformationen wie die *On-Line*-Dokumentation des MAXed unterrichten.

## 1.1 Welche Vorzüge hat der MAXed?

- **Benutzerfreundlichkeit**

- übersichtliche Gestaltung des MAXed-Bildschirms
- *On-Line*-Hilfsinformation
- gleichzeitiges Bereithalten mehrerer Dateien
- Mehrfenster-Technik durch Unterteilung des MAXed-Bildschirms
- leicht erlernbare, einheitliche Kommandostruktur
- Voreinstellung von Parametern für zahlreiche Kommandos
- Merkfähigkeit durch Speichern verschiedener Kommandoangaben
- automatisches Erstellen von Arbeitsprotokollen und Auswahllisten
- benutzer- und terminalspezifische Konfigurierbarkeit

- **Allgemeine Funktionen**

- Ausschneiden und Kopieren mit verschiedenen Puffer-Strukturen
- Markieren mehrerer Dateipositionen
- schnelles Blättern, Suchen und Ersetzen
- SINIX-Schnittstellen zu Dateibearbeitung und Hintergrundverarbeitung
- Makroerstellung durch Aufzeichnung oder Definition in Quelldateien
- Erstellen von MACRO-Bibliotheken für Makros und Textbausteine
- C-Programmschnittstelle zur Erstellung von MAXed-Steuerprogrammen
- schrittweises Rück- und Vorspielen beliebiger Textveränderungen

- **Textverarbeitung**

- Bearbeiten ganzer Zeilen-, Rechteck- oder Zeichenkettenbereiche
- Randeinzug und Texteingassung durch Formatränder
- automatischer Zeilenumbruch und sprachabhängiger Silbentrenner
- Ausrichten, Zentrieren, Formatieren im Flatter- oder Blocksatz
- verschiedene Zeichenattribute zur Hervorhebung von Textstellen
- Zeichnen von Linien und Rahmen
- Druckprogramm zur Belieferung verschiedener Druckertypen

- **Programmentwicklung**
  - konfigurierbare BLOCK- und RECORD-Strukturen
  - Analysieren mehrerer Programmdateien über Tags
  - automatisches Erstellen von Tag-Dateien
  - Analysieren mehrerer Programmdateien durch FILE-SCAN
  - Programmübersetzung im Hintergrund
  - automatisches Umlenken von Compilermeldungen

## 1.2 Zusätzliche Hilfsmittel

Zum MAXed-Programm erhalten Sie – neben dem MAXed-Handbuch – noch zusätzliche Hilfsmittel, die Ihnen die Arbeit mit dem MAXed erleichtern: Tastaturschablonen, Referenzkarten und die MAXed-Kurzbeschreibung.

### 1.2.1 Die Tastaturschablone

Zur Bedienung des MAXed erhalten Sie beidseitig bedruckte Tastaturschablonen, die mit den Bezeichnungen fast aller MAXed-Kommandos beschriftet sind. Bringen Sie die Tastaturschablone mit der Seite, die zum Typ Ihrer Datensichtstation paßt, auf Ihrer Tastatur an.

### 1.2.2 Die Referenzkarten

Dem Handbuch liegen mehrere Referenzkarten bei, die alle Tastenbelegungen für die MAXed-Kommandos enthalten. Sie passen zu verschiedenen Typen von Datensichtstationen und sind in verschiedene Aufgabengebiete gegliedert. Sie sind für den täglichen Gebrauch gedacht und geben Ihnen Auskunft, auf welche Tasten bzw. Tastenkombinationen die MAXed-Kommandos unabhängig von der Tastaturschablone liegen.

Auf den Referenzkarten sind die MAXed-Kommandos in folgender Darstellungsweise den "real" existierenden Tasten bzw. Tastenkombinationen zugeordnet:

<KOMMANDO> Hinweis auf die Hauptfunktion des Kommandos

bzw.

<KOMMANDO> Hinweis auf die Hauptfunktion des Kommandos

Sie können sich über die Lage der Kommandotasten auf Ihrer Tastatur in der *On-Line*-Dokumentation informieren (siehe Abschnitt 1.6) mit  
<ENTER> keypad <HELP>

### **1.2.3 Die MAXed-Kurzbeschreibung**

Die dem MAXed-Handbuch angefügte MAXed-Kurzbeschreibung bietet Ihnen einen Überblick über die wichtigsten MAXed-Kommandos. Die Kurzbeschreibung ist thematisch gegliedert, sodaß Sie zu einem Problem schnell die geeigneten MAXed-Kommandos finden. Die Verweise werden es Ihnen erleichtern, ausführlichere Informationen in den korrespondierenden Abschnitten des MAXed-Manuals zu finden.

### 1.3 Die verwendete Metasyntax

Für die Darstellung der MAXed-Kommandos in diesem Handbuch, in der MAXed-Kurzbeschreibung und auf den Referenzkarten wurde folgende Schreibweise verwendet:

<KOMMANDO>

Dies steht für das Drücken der Kommandotaste bzw. der Tastenkombination, auf der ein MAXed-Kommando KOMMANDO definiert ist. Legen Sie sich die zu Ihrer Datensichtstation passende Referenzkarte zur Seite. Auf ihr finden Sie die Zuordnung zu den Tasten bzw. Tastenkombinationen Ihrer Tastatur (siehe Abschnitt 1.2.2 Die Referenzkarten).

*Beispiel*

<ENTER> steht für das Drücken der laut Referenzkarte für das MAXed-Kommando ENTER vorgesehenen Kommandotaste.

<ENTER> kommandoattribute <KOMMANDO>

Dies ist eine Kommandofolge, mit der Sie ein attributiertes MAXed-Kommando <KOMMANDO> eingeben. Eine Kommandoattributierung beginnen Sie grundsätzlich durch Drücken der Kommandotaste <ENTER> und schließen Sie durch Drücken einer weiteren Kommandotaste <KOMMANDO> ab (siehe Abschnitt 3.6 Prinzipielles zur Attributierung von Kommandos).

kommandoattribut

Normal gedruckte Zeichen oder Wörter innerhalb von Syntaxdarstellungen sind Platzhalter, für die Sie etwas eingeben müssen.

*Beispiel*

Ein Auszug aus der Syntaxdarstellung des Kommandos DOWNARROW:

<ENTER> n <DOWNARROW>

Für  $n$  geben Sie eine ganze positive Zahl ein. Hierbei dient  $n$  als Verstärkungsfaktor zur  $n$ -maligen Ausführung des Kommandos <DOWNARROW> (siehe Abschnitt 3.7 Die Kommandoverstärkung).

**halbfett**

Halbfett gedruckte Zeichen oder Wörter innerhalb von Syntaxdarstellungen sind Konstanten, die Sie so eingeben, wie sie angegeben sind. Bei der Beschreibung von Ausgaben oder Meldungen des MAXed sind jene Zeichen halbfett dargestellt, die im Klartext angezeigt werden.

[ option ]

Die beiden eckigen Klammern [ und ] werden innerhalb von Syntaxdarstellungen zur Kennzeichnung von optionalen Angaben verwendet. Beim Aufruf von MAXed-Kommandos können bestimmte Kommandoattribute optional sein, bei Aufruf von Kommandos oder Programmen auf der Shell-Ebene bestimmte Argumente oder Schalter. Die eckigen Klammern geben Sie in diesen Fällen selbstverständlich nicht mit ein!

/ Der Schrägstrich hat innerhalb der Syntaxdarstellung die Bedeutung eines "Oders" zur Trennung alternativer Kommandoattribute. Nur bei bestimmten Kommandos ist er außerdem als einzugebendes Sonderzeichen in halbfetter Darstellung aufgeführt.

*kursiv* Kursiv gedruckte Zeichen oder Wörter innerhalb des Textes stellen Zitate einer Texteingabe oder Namen dar.

**Beispiel**

Ein Auszug aus der Syntaxdarstellung des Kommandos PICK:  
 <ENTER> [bereich] [ > / > > ] [IDENTIFIER] <PICK>

Bei der Verwendung dieser Kommandoform des Kommandos PICK können Sie bis zu drei Kommandoattribute eingeben: eine Bereichsdefinition für *bereich* und/oder eine der beiden Angaben > bzw. >> und/oder einen PICK-Puffernamen für *IDENTIFIER*. Wenn Sie diese Kommandoattribute weglassen, arbeitet das Kommando PICK mit Standard-Einstellungen (siehe z.B. Abschnitt 3.12 Voreingestellte Kommandoattribute).

**Anmerkung**

Bei Beispielen sind Ein- und Ausgaben in normaler Schrift dargestellt.

## 1.4 MAXed-Kommandos in Quelldateien

Bei der Erstellung von Quelldateien für MACRO-Bibliotheken oder MAXed-Steuerprogrammen geben Sie MAXed-Kommandos durch ihre Bezeichnung an.

<KOMMANDO>

Innerhalb von MACRO-Bibliotheken verwenden Sie sogenannte Schlüsselwörter, die aus Bezeichnern in spitzen Klammern bestehen (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen). Die Bezeichnungen von MAXed-Kommandos stellen in spitzen Klammern auch Schlüsselwörter dar, mit denen Sie Makros definieren (siehe Abschnitt 4.7 Arbeiten mit Makros).

KOMMANDO

In C-Quelldateien für MAXed-Steuerprogramme verwenden Sie die Bezeichnungen von MAXed-Kommandos als Argumente der Schnittstellen-Funktion *m\_control* (siehe Abschnitt 7.5 MAXed-Steuerprogramme).

Zur Eingabe von Kommandobezeichnungen bietet sich das Kommando CODE an (siehe Kapitel 6: CODE):

<CODE> <KOMMANDO>

## 1.5 Besonderheiten bei der Texteingabe

Bei der Texteingabe dürfen Sie alle alphanumerischen Zeichen und Sonderzeichen verwenden. Das MAXed-Kommando FONT erlaubt Ihnen, Zeichen aus der "oberen Hälfte" des 8-Bit-Zeichensatzes 8859/1 einzugeben:

<FONT>

In der Standard-Konfiguration des MAXed liegen für diese Eingabefolge die nationalen Sonderzeichen  $\beta$ , die Umlaute und das Paragraphen-Zeichen auf den Zeichentasten für *s, a, A, o, O, u, U* und  $\$$ .

## 1.6 Hilfsinformationen

Die *On-Line*-Dokumentation:

Sie können sich während der MAXed-Sitzung Hilfsinformationen aus der *On-Line*-Dokumentation des MAXed einblenden lassen (siehe Abschnitt 4.2.1 Die Pseudo-Datei HELP). Ein eingeblendeter Hilfstext enthält entweder spezielle Informationen über ein MAXed-Kommando bzw. einen MAXed-spezifischen Begriff oder bietet eine Übersicht über MAXed-Kommandos. Zu jedem MAXed-Kommando ist ein eigener Hilfstext verfügbar.

*Beispiel*

<HELP>

Damit blenden Sie sich eine Übersicht über Begriffe ein, zu denen es spezielle Hilfstexte gibt.

<ENTER> range <HELP>

<ENTER> pattern <HELP>

<ENTER> <HELP> <PLUSFIND>

<ENTER> <HELP> <REPLACE>

Mit diesen HELP-Kommandos blenden Sie sich Hilfstexte zu den eingegebenen Begriffen bzw. zu den MAXed-Kommandos, deren Kommandotasten Sie betätigen, ein (siehe Kapitel 6: HELP).

In allen eingeblendeten Hilfstexten sind Begriffe, zu denen es ausführliche Informationen in bestimmten Hilfstexten gibt, in Großbuchstaben geschrieben. Durch Positionieren der Schreibmarke auf einem Begriff und nochmaliges <HELP> wählen Sie sich einen neuen Hilfstext aus.

Einblenden der Kommandobezeichnung:

Sie können sich in der Steuerzeile des MAXed die Bezeichnung eines MAXed-Kommandos KOMMANDO einblenden lassen, das auf eine Kommandotaste bzw. Tastenkombination gelegt ist (siehe Abschnitt 1.2.2 Die Referenzkarten). Dazu geben Sie folgende Kommandosequenz ein:

<ENTER> <CODE> <KOMMANDO>

Dies ist z.B. dann vorteilhaft, wenn Sie selbst Tasten-Makros definiert haben und die Tastenbelegung vor dem Aufruf eines Kommandos überprüfen wollen.

1

2

3

4

## 2 Beispielsitzungen

Mit den Beispielsitzungen dieses Kapitels können Sie sich in die Arbeit mit dem MAXed einführen lassen. Indem Sie die einzelnen Sitzungen durchspielen, sammeln Sie Erfahrung im Umgang mit den wichtigsten Kommandos des vielfältigen Angebots an MAXed-Kommandos. So erhalten Sie einen Einblick in die verschiedenen Einsatzgebiete des MAXed.

Jeder Abschnitt dieses Kapitels enthält eine in sich geschlossene Beispielsitzung. Der Schwierigkeitsgrad steigert sich von Sitzung zu Sitzung.

Jede Beispielsitzung hat ihren eigenen Schwerpunkt:

- Abschnitt 2.1 stellt Ihnen die grundlegenden Kommandos vor, mit deren Hilfe Sie bereits eigene Dateien erstellen, Texte erfassen und verändern können. Außerdem erlernen Sie nebenbei das Handwerkszeug, um Schreibmarke und Fenster innerhalb von Dateien zu bewegen.
- Abschnitt 2.2 erweitert Ihre Kenntnisse der Texteingabe. Sie lernen den geschickten Umgang mit Puffern und Textbereichen.
- Abschnitt 2.3 macht Sie mit komfortablen Möglichkeiten im Umgang mit verschiedenen Puffern, Pseudo-Dateien und Kommandoattributierungen bekannt und verdeutlicht Ihnen die Vorteile, die das Setzen von unsichtbaren Marken innerhalb von Dateien bietet.
- Abschnitt 2.4 präsentiert Ihnen einen Einblick in die Mächtigkeit der MAXed-Kommandos zum Suchen und Ersetzen.
- Abschnitt 2.5 führt die Thematik des Suchens und Ersetzens fort und macht Sie mit komplexeren Suchmustern (PATTERN) und Ersetzungszeichenketten bekannt.
- In Abschnitt 2.6 lernen Sie schließlich einige Textgestaltungsmöglichkeiten des MAXed, wie Formatieren, Zeilenumbruch und Einsatz von Zeichenattributen kennen. Dabei werden Sie sich auch der FILE-SCAN-Funktion des MAXed sowie spezieller Puffer bedienen, um schnell und unkompliziert Textstellen in mehreren Dateien aufzufinden und zu sammeln.

Bevor Sie nun anfangen, lesen Sie sich noch folgende Hinweise durch:

- Für die Beispielsitzung 1 benötigen Sie keinerlei Vorkenntnisse. Selbstverständlich können Sie sich auch ein für Sie interessantes Gebiet herausgreifen. Für das Verständnis einer der durchzuführenden Dateibearbeitungen ist es jedoch günstig, wenn Sie mit den wichtigsten Kommandos, die in den jeweils vorausgehenden Beispielsitzungen vorgestellt werden, einigermaßen vertraut sind.
- Zu jeder dieser Beispielsitzungen sind bereits eine oder mehrere Dateien vorbereitet, die das Rohmaterial für die Bearbeitung liefern: Ausgewählte Szenen aus einem Theaterstück eines bekannten Autors. Sie finden diese Dateien im Dateiverzeichnis `/usr/lib/nls/msg/De/maxedh/examples`. Erstellen Sie sich am besten ein neues Dateiverzeichnis unter Ihrem *HOME*-Verzeichnis, und kopieren Sie sich die benötigten Dateien dorthin. Dazu stellen Sie sicherheitshalber den sogenannten Konversationsmodus des MAXed auf Ihre Sprachkenntnisse (also vermutlich deutsch) ein. Das könnte in der Bourne-Shell so erledigt werden:

```
cd
mkdir max_training
cd max_training
LANG=De_DE.646
MAXLIB=/usr/lib/nls/msg/De/maxedh/examples
export MAXLIB LANG
cp $MAXLIB/abs* .
```

- Die MAXed-Tastaturschablone bringen Sie mit der zu Ihrer Datensichtstation passenden Seite an Ihrer Tastatur an. Sie wird Ihnen gute Dienste erweisen (siehe Abschnitt 1.2.1)!
- Suchen Sie sich auch die passende MAXed-Referenzkarte heraus und legen Sie sich diese für die Arbeit neben die Tastatur. Hier finden Sie zu jedem MAXed-Kommando die zugehörige Taste bzw. Tastenkombination (siehe Abschnitt 1.2.2).

### *Anmerkung*

Für die Abbildungen dieses Handbuchs, die den MAXed-Bildschirm darstellen, wurde das *Alpha-Terminal 97801* verwendet. In den Beispielsitzungen beziehen sich ebenfalls alle Angaben, die die Ausmaße des Bildschirms betreffen auf diese Datensichtstation!

- "Handlungsanweisungen":

- ▷ Mit diesem Zeichen werden Sie aufgefordert, das zu tun, was nachstehend angegeben ist! Dabei gibt es nur zwei Möglichkeiten:

- Sie tippen einen Text in die aktuelle Datei
- Sie geben ein MAXed Kommando ein (siehe unten)

<KOMMANDO>

Steht für das einmalige Drücken der laut Referenzkarte zugehörigen Kommandotaste bzw. Tastenkombination.

<ENTER> kommandoattributierung <KOMMANDO>

Tippen Sie die Kommandoattribute des MAXed-Kommandos KOMMANDO nach dem Drücken der Kommandotaste <ENTER> ein, und zwar genau so, wie sie angegeben sind! Der Text erscheint auf Ihrem MAXed-Bildschirm links unten im sogenannten Eingabefeld der Steuerzeile! Danach betätigen Sie die anschließend angegebene Kommandotaste <KOMMANDO>. Direkt vor und nach der Kommandoattributierung brauchen Sie keine Leerzeichen einzugeben. Leerzeichen sind aber notwendig, um mehrere Kommandoattribute zu trennen!

- Zu Beginn einer jeden Beispielsitzung wird vorausgesetzt, daß sich die sogenannten Betriebs-Modi des MAXed in ihrer Standard-Einstellung befinden (siehe Abschnitt 4.1 Die Betriebs-Modi)! Wenn Sie sicher gehen wollen, daß Sie diese nicht bereits umgestellt haben, geben Sie folgende MAXed-Kommandos zu Beginn einer Beispielsitzung ein:

```
<ENTER> i <INSERTMODE>
<ENTER> n <RETURN>
<ENTER> n <UPARROW>
<ENTER> * <PLUSFIND>
<ENTER> De <FILL>
```

- Die Theorie ist in den Beispielsitzungen auf das Notwendige reduziert. Beim Durcharbeiten einer Beispielsitzung ist es durchaus zu empfehlen, parallel dazu die thematisch passenden Abschnitte in den Kapiteln 3 und 4 zu studieren!

## 2.1 Beispielsitzung 1 - Grundlegende Kommandos

Sie haben sich die Beispieldateien, wie im Vorwort zu diesem Kapitel angegeben, in ein eigenes Dateiverzeichnis kopiert? Dann kann es losgehen!

- ▷ Wechseln Sie in dieses Verzeichnis und rufen Sie den MAXed von der Shell aus auf: `m`

Der MAXed wird gestartet. Nach der Ausgabe seiner Versionsnummer erscheint auf Ihrem Bildschirm der MAXed-Bildschirm. Wenn unter Ihrer Kennung an der aktuellen Datensichtstation der MAXed nun zum ersten Mal aufgerufen wird, blendet er Ihnen folgende Meldung ein:

```
Sie sehen diesen Text,  
  
* weil beim MAXed-Aufruf keine zu editierende Datei angegeben war  
  oder  
* weil MAXed die angegebene Datei nicht finden konnte  
  oder  
* weil die angegebene Datei keine Textdatei ist.  
  
Fuer weitere Informationen geben Sie bitte  
  <HELP>  
oder  
  <ENTER> help <ENTER>  
ein.  
  
File <UNDEFINED> Ln 1 Cl 1
```

Wenn der MAXed an Ihrer Datensichtstation unter Ihrer Benutzerkennung schon mal gestartet wurde, blendet er i.a. die Datei ein, die er zuletzt angezeigt hatte (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

Der MAXed präsentiert die abgebildete Meldung, um Sie darauf aufmerksam zu machen, daß er momentan keine bearbeitbare Datei geladen hat. Für die Durchführung der Beispielsitzungen ist es aber erforderlich, eine Datei zu laden. Sie entdecken auf Ihrer Tastaturschablone die Kommandotaste <HELP>.

▷ Drücken Sie auf <HELP>! Der MAXed blendet schon eine Datei ein:

```

Informationen zu einer bestimmten Funktion koennen mit
<ENTER> topic <HELP> angefordert werden oder durch Positionierung
des CURSORS auf das entsprechende Schluesselwort. Durch nochmaliges
Druuecken der <HELP>-Taste erscheint die Information dann auf dem
Bildschirm.

MAXED:      Ueberblick ueber MAXed Funktionalitaet
SCREEN:     Gestaltung des MAXed Bildschirms
KEYPAD:     Gestaltung der Tastatur
COMMAND:    Aufruf von MAXed aus dem Betriebssystem, Befehlsparameter
UTILITIES:  MAXed-Programme

Fuer folgende MAXed Befehle sind zusaetzliche Informationen abrufbar:

ALIGN      DEFDIR      FONT          LOMER        REPLACE      UPARROW
ATTRIBUTE  DELCHAR      FORMAT        MACRO        RESTOR       UPEND
BACKSPACE  DOWNARROW    GOTO          MARGIN       RETURN       UPPER
BACKTAB    DRAW         HELP          MINUSFIND    REVERSE      VWINDOW
CALCULATE  EDIT         HOME         MINUSLINE    RIGHTARROW   WAIT
CANCEL     ENDLINE      HMINDOW      MINUSPAGE    RVLINFEED   WINL
CENTER     ENTER        HYPHEN       PICK         SAVE         WINR

WAEHLE Begriff und gebe <HELP>           File <HELP>           Ln 1           Cl 1
    
```

In diesem Fall ist es eine MAXed-eigene Hilfsdatei, die Ihnen Auskunft über weitere *On-Line*-Dokumentation gibt. Lesen Sie sich diese einmal durch. Sie suchen zunächst nach dem Schlüsselwort, unter dem Sie Informationen dazu finden, wie Sie eine Datei zum "Editieren" laden. Deshalb entscheiden Sie sich für den Begriff EDIT.

▷ Ihre Schreibmarke (das blinkende, grafische Zeichen links oben) bewegen Sie auf diesen Begriff, indem Sie mehrmals die Kommandotasten <DOWNARROW> und <RIGHTARROW> betätigen! Drücken Sie nochmals <HELP>.



Die Schreibmarke springt aus dem aktuellen Fenster, wo sie an der aktuellen Position ein grafisches Zeichen hinterläßt, in die sogenannte Steuerzeile des MAXed, genauer: ins Eingabefeld links unten. Hier geben Sie grundsätzlich die Kommandoattribute an. Das sind die Angaben, die Sie zu einem MAXed-Kommando machen.

- ▷ Sie geben jetzt den gewünschten Dateinamen ein:

zettel

Das MAXed-Kommando, das Sie damit attributieren, rufen Sie danach auf:

<EDIT>

```

<ENTER>Dateiname<EDIT>
    Holt eine Datei in das aktuelle Fenster. Die Datei, die vorher im aktuellen Fenster war, wird zur alternativen Datei. Wenn die neue Datei das erste Mal mit MAXed bearbeitet wird, werden das Fenster und der CURSOR an den Anfang der Datei gesetzt, ansonsten an die Stelle, wo sie gesetzt waren zu dem Zeitpunkt, als die Datei verlassen worden ist. Der Name der neuen Datei ist in der Steuerzeile zu sehen.

    Enthaeft die Dateispezifikation sog. Wildcards, wird eine Liste von dazu passenden Dateinamen wiedergegeben. Daraus kann eine der Dateien durch Kennzeichnung mit dem CURSOR ausgewaehlt werden. Mittels <EDIT> kann die Datei zur Bearbeitung auf den Bildschirm gebracht werden, ebenso wie bei <ENTER> Dateiname <EDIT>.

    Handelt es sich bei der Spezifikation um ein Dateiverzeichnis, werden die dazugehoerigen Dateien aufgelistet. Daraus kann eine Datei wie oben beschrieben ausgewaehlt werden.
  
```

<EDIT> zum Anlegen von: zettel                      File <HELP>                      Ln 15                      Cl 1

Was ist passiert? Betrachten Sie die Steuerzeile: Der MAXed präsentiert hier seine Meldungen. Wenn wir von dem einigermaßen unwahrscheinlichen Fall absehen, daß Sie bereits im Besitz einer Datei namens *zettel* im aktuellen Dateiverzeichnis sein könnten, wird Sie der MAXed zu folgendem auffordern: <EDIT> zum Anlegen von: *zettel*

▷ Drücken Sie also die Kommandotaste <EDIT> nochmals.

Der MAXed wird Sie belohnen, indem er die Datei *zettel* im aktuellen Dateiverzeichnis anlegt und im aktuellen Fenster einblendet. Sie sehen zunächst nichts außer einer leeren Fläche, da eine neu angelegte Datei zu Beginn natürlich leer ist! Den Dateinamen zeigt der MAXed übrigens in der Mitte der Steuerzeile an: File *zettel*

Jetzt können Sie zu schreiben anfangen. Das Fenster ist auf den Anfang der leeren Datei positioniert. Die Schreibmarke befindet sich in Zeile 1 auf Spalte 1. Diese Information enthält wiederum die Steuerzeile; rechts unten ist die aktuelle Position der Schreibmarke eingeblendet: Ln 1 C1 1

Überdies werden Sie noch feststellen, daß der MAXed an der Umrandungslinie des aktuellen Fensters kenntlich macht, ob noch weiterer Text, außer dem im Fenster sichtbaren, in einer der vier "Himmelsrichtungen" vorhanden ist: Er wird gegebenenfalls kleine Striche, die nach außen weisen, an der Umrandungslinie anbringen.

▷ Tippen Sie also folgendes ein: Das wird

Was *das wird*, erfahren Sie gleich noch. Aber betrachten Sie zunächst den linken Fensterrand: Bei allen Zeilen außer der ersten weist ein kleiner Strich nach rechts. Auf diese Weise kennzeichnet der MAXed alle Zeilen nach der letzten beschriebenen Zeile einer Datei.

▷ Sie schreiben weiter: einige Traenen kosten

Die *Traenen* sind zwar schon ganz rührend, jedoch können Sie sie mit dem MAXed auch korrekt schreiben!

▷ Löschen Sie die Zeichen rückwärts bis zum *Tr*, indem Sie mehrmals <BACKSPACE> drücken.

Einige nationale Sonderzeichen wie Umlaute (*ä*, *ö*, usw.) und *ß* kann der MAXed in der Standard-Konfiguration aus seinem Zeichenvorrat hervorholen, wenn Sie ihm das "Zeichen" dafür geben:

- ▷ Drücken Sie die Kommandotaste <FONT>, und geben Sie direkt anschließend das Zeichen *a* ein! Sie schreiben also weiter:  
änen kosten bei einer wahrhaftigen Vorstellung.

Die erste Zeile ist schon ganz ordentlich geworden. Schreiben Sie in der nächsten weiter:

- ▷ Dazu springen Sie mit der Schreibmarke an den Anfang der nächsten Zeile mit <RETURN> und geben weiter ein:  
Wenn ichs mache, lasst die Zuhoerer nach ihren

Wie geht's besser? Richtig:

- ▷ Mit <LEFTARROW> wieder zurück auf das Zeichen direkt nach *oe*, mit zweimal <BACKSPACE> diese beiden Zeichen löschen und mit Hilfe der Kommandotaste <FONT> das Zeichen *ö* eingeben. Auf die gleiche Weise ersetzen Sie die Zeichen *ss* durch *ß*.

Sie springen mit der Schreibmarke ans Zeilenende: <ENDLINE>

Tippen Sie mal, ohne auf den Bildschirm zu sehen, weiter ein:

Augen sehn! Ich will Sturm erregen ich will einigermaßen  
lamentieren.

Sie schauen wieder auf den Bildschirm und erblicken:

```

Das wird einige Tränen kosten bei einer wahrhaftigen Vorstellung.
Wenn ichs mache, laßt die Zuhörer nach ihren Augen sehn! Ich will Sturm erreg

```

ACHTUNG: Blinde Eingabe!      **File zettel**      Ln 2      Cl 115

Daß Sie "blind" eingegeben haben, wußten Sie selber. Aber woher weiß dies der MAXed? Nun, er kann 77 und 1 zusammenzählen und kombiniert, daß das Ende des eingegebenen Textes den rechten Rand Ihres Fensters überschritten haben muß (es wird das *Alpha-Terminal 97801* vorausgesetzt). Das hindert ihn jedoch nicht daran, außerhalb des sichtbaren Dateiausschnitts, den das Fenster freigibt, weiterzuschreiben.

- ▷ Sie verschieben den Dateiausschnitt nach rechts mit: <WINR> <WINR> <WINR>. Da ist also Ihr eingegebener Zeilenrest. Drücken Sie jetzt wieder <RETURN>.

Die Schreibmarke springt an die erste Spalte; der Dateiausschnitt läßt sie nicht aus seinem sichtbaren Bereich und bewegt sich also auch hinterher. Sie stellen allerdings fest, daß die vorige Zeile immer noch zum Teil im "Dunkeln" bleibt. Sie bewegen den vorausgehenden Zeilenrest folgendermaßen in die nächste Zeile:

- ▷ Sie bewegen die Schreibmarke eine Zeile nach oben: <UPARROW>  
Sie bewegen die Schreibmarke eine Spalte nach links: <LEFTARROW>

Die Schreibmarke springt zum gegenüberliegenden Fensterrand um! Sie tun das in der Standard-Einstellung immer dann, wenn Sie sie auf eine der vier Randteile des aktuellen Fensters zu setzen versuchen.

- ▷ Sie bewegen die Schreibmarke rückwärts an den nächsten Wortanfang mit <WORDLEFT>. Jetzt trennen Sie den Zeilenrest ab der Position der Schreibmarke ab und schieben ihn an den Anfang einer neuen, nachfolgenden Zeile: <ENTER> <SPACE>  
Mit der Schreibmarke folgen Sie nach: <RETURN>

Ein bißchen Abstand vom Zeilenanfang würde sich nicht schlecht machen. Verschieben Sie doch den Textanfang auf Spalte drei. Nichts leichter als das:

- ▷ Sie drücken zweimal auf die Leerzeilentaste.

Bevor Sie dies in den beiden ersten Zeilen wiederholen, probieren Sie doch ein spezielles Kommando dafür aus:

- ▷ Bewegen Sie die Schreibmarke mit <UPARROW> nach oben auf die dritte Spalte drei der zweiten Zeile. Dann geben Sie das Kommando <ALIGN>.

Der Textanfang der Zeile verschiebt sich an die Position der Schreibmarke. Machen Sie das mit der ersten Zeile auch:

- ▷ <UPARROW> <ALIGN>

Für das weitere Vorgehen sollte sich der eingegebene Text "im Rahmen" halten und der Textanfang generell auf Spalte drei "festgeschrieben" sein. Dazu bietet es sich an, ein Textformat festzulegen, nach dem sich der eingebende Text automatisch ausrichtet. Dann müssen Sie sich bei der Eingabe nicht mehr darum kümmern, an welcher Spalte der eingegebene Text beginnt und endet.

Der MAXed ist im Besitz eines Kommandos mit Namen MARGIN. Damit setzen Sie sogenannte Formatränder, die bei der Texteingabe "Marginalitäten" links bzw. rechts der Spalten, auf denen sich die Formatränder befinden, verhindern.

- ▷ Drücken Sie die Kommandotaste <MARGIN>.

Dadurch haben Sie schon den linken Formatrand auf die aktuelle Spalte gesetzt! Der MAXed kennzeichnet dies am oberen Fensterrand an der Spalte drei durch einen kleinen Strich nach unten. Wenn Sie die Schreibmarke zukünftig über den linken Formatrand hinausbewegen sollten, wird Sie der MAXed darauf aufmerksam machen, indem er die Spaltennummer rechts unten in der Steuerzeile vorübergehend invers darstellt.

Am rechten Fensterrand bringen Sie einen zweiten Formatrand an. Sie nützen wiederum die Umspring-Eigenschaft der Schreibmarke am Fenster- rand aus, um zum rechten Fensterrand gelangen:

- ▷ Sie bewegen die Schreibmarke dreimal nach links mit:  
<LEFTARROW> <LEFTARROW> <LEFTARROW>

Zum Setzen der rechten Formatgrenze gibt es eine andere Kommandovariante:

- ▷ Drücken Sie nacheinander die Tasten <ENTER> <MARGIN>.

Auch hier erkennen Sie einen kleinen Strich am oberen Fensterrand. Eine der Auswirkungen des linken Formatrandes werden Sie sogleich bemerken:

- ▷ Drücken Sie <RETURN> <RETURN>.

Die Schreibmarke ist wieder in der dritten Zeile und hat auf dem Weg dahin nicht einmal den linken Fensterrand überschritten! Sie steht also auf der dritten Spalte. Die Texteingabe setzen Sie am Zeilenende fort:

- ▷ Sie springen mit <ENDLINE> hinter das letzte Zeichen der aktuellen Zeile und geben weiter ein:

Nun zu den übrigen... eigentlich habe ich doch das beste Genie zu einem Tyrannen;

Wie Sie bemerken, hat der MAXed den Text in der aktuellen Zeile automatisch umbrochen, nachdem Sie mit der Schreibmarke über den rechten Rand hinausgekommen waren und nach dem Eintippen des Wortes *eigentlich* die Leertaste drückten. Die Schreibmarke sprang danach ohne weiteres Zutun gleichzeitig mit, sodaß Sie ohne "Zeitverlust" sofort weitertippen konnten!

▷ Geben Sie weiter ein:

```
ich könnte einen Herkles kostbarlich spielen oder eine Rolle,
wo man alles kurz und klein schlagen muß:
```

Sie beginnen einen neuen Absatz, indem Sie eine Leerzeile voranstellen: <RETURN> <RETURN>

Der folgende Text erfordert eine "Zäsur", d.h. er soll rein optisch innerhalb des Textes stärker zur Geltung kommen. Es bietet sich z.B. eine Einrückung um einige Spalten nach rechts an.

▷ Sie springen mit der Schreibmarke an die nächste Tabulatorposition: <TAB>

Jene Spalten, an denen Tabulatoren gesetzt sind, kennzeichnet der MAXed am unteren Fensterrand durch kleine nach oben weisende Striche. Die Schreibmarke steht nun auf Spalte 5 Dieser Tabulator paßt Ihnen jedoch nicht:

▷ Sie löschen den Tabulator an der aktuellen Spaltenposition mit <SRTAB>. Die Schreibmarke bewegen Sie mit <RIGHTARROW> um 3 Spalten nach rechts und setzen hier einen neuen Tabulator mit <SRTAB>. Dann fahren Sie mit der Texteingabe fort:

```
Der Felsen Schoß <RETURN> <TAB>
Und toller Stoß <RETURN> <TAB>
Zerbricht das Schloß <RETURN> <TAB> <TAB> <TAB>
    Der Kerkertür, <RETURN> <TAB>
Und Phäbus' Karren <RETURN>
```

Die letzte Silbe stört den Sprachrhythmus. Am besten unterdrücken Sie den letzten Vokal:

- ▷ Sie springen mit der Schreibmarke ans Ende der letzten Zeile zurück mit <UPEND>, bewegen sie um zwei Positionen nach links mit <LEFTARROW>, löschen das *e* mit <DELCHAR> und schreiben weiter: <RETURN> <TAB>

```
Kommt angefahrn <RETURN> <TAB>
Und macht erstarrn <RETURN> <TAB> <TAB> <TAB>
Des stolzen Schicksals Zier: <RETURN> <RETURN>
```

Schreiben Sie noch den letzten Absatz:

```
Das ging prächtig!... Nun nennt die übrigen Akteurs... Dies ist
Herklessens Natur, eines Tyrannen Natur! Ein Liebhaber ist schon mehr
lamentabel. <RETURN>
```

Das Textstück, das Sie da eingetippt haben, existiert bis jetzt nur in einer temporären Datei (wenigstens für Ihren Rechner). Sie wollen es gleich auf Platte abspeichern, bevor Sie sich an weitere Aufgaben machen:

- ▷ Sie speichern den Inhalt der aktuellen Datei ab mit <SAVE>.

Nun werden Sie an einer bereits existierenden Datei Änderungen vornehmen. Wie Sie vielleicht schon ahnen, ist das soeben abgetippte Textstück aus einem größeren Werk entnommen. In der Beispieldatei *abs2.1* liegt eine Rohfassung der gesamten Szene vor. Sie sind nämlich der "Ghostwriter", dessen Aufgabe es ist, diese Szene zu vervollständigen. Wie Sie diese Datei ins aktuelle Fenster holen wissen Sie ja bereits:

- ▷ Sie laden die Datei *abs2.1* und blenden Sie im aktuellen Fenster ein: <ENTER> *abs2.1* <EDIT>

Wollen Sie sehen wie lang der Text dieser Datei ist?

- ▷ Sie springen mit der Schreibmarke direkt ans Dateiende, wenn Sie eingeben: <ENTER> . <GOTO>

In der Steuerzeile zeigt Ihnen der MAXed die aktuelle Zeilennummer an:  
Ln 105

Ihr "Arbeitsplatz" ist einige Zeilen weiter oben, wo der Bälkenflicker *Flaut* eine Sprechrolle hat und sich darüber beschwert, eine Weiberrolle zugeteilt zu bekommen. Die Stelle befindet sich in Zeile 49.

- ▷ Sie springen mit der Schreibmarke direkt in die Zeile 49  
<ENTER> 48 <GOTO>

Sie haben sicher schon sechs Zeilen weiter die Stelle erkannt, an der eine Ergänzung für die Rolle des Webers *Zettel* vorgesehen ist.

- ▷ Sie bewegen die Schreibmarke um sechs Zeilen nach unten, indem Sie das Kommando `DOWNARROW` verstärken, d.h. in einem Zug mehrmals ausführen lassen: <ENTER> 6 <DOWNARROW>

Die Schreibmarke steht nun auf der Zeile mit der Anmerkung für die Ergänzung der Rolle des *Zettel*. Die Anmerkung können Sie also entfernen:

- ▷ Sie löschen die aktuelle Zeile mit <ERASE>. Dann fügen Sie vor der aktuellen Zeile eine Leerzeile ein mit <SPACE>.

Die Schreibmarke kommt auf der Leerzeile zu stehen, da der nachfolgende Text um eine Zeile nach unten verschoben wurde, sodaß Sie sogleich mit der Texteingabe beginnen können. Sicher wollen Sie auch in dieser Datei wieder die Formatränder setzen. Die "Fußarbeit" mit der Schreibmarke ersparen Sie sich diesmal:

- ▷ Sie setzen beide Formatränder gleichzeitig, indem Sie einfach die gewünschten Spaltennummern eingeben: <ENTER> 3 77 <MARGIN>

Den Anschlußdialog für den "überdrehten" *Zettel* haben Sie als guter "Ghostwriter" schon im Kopf:

- ▷ Sie bewegen die Schreibmarke mit <RIGHTARROW> auf Spalte drei und tippen die erste Zeile:

Zettel: Wenn ich das Gesicht verstecken darf, so gebt mir Thisbe auch.  
<RETURN>

Vor Ihrem "geistigen Auge" erkennen Sie schon die weiteren Worte, die Sie "*Zettel* in den Mund legen" wollen. Auf dem Bildschirm steht die Schreibmarke zunächst mal auf der nächsten beschriebenen Zeile. Sie müssen sich vor der aktuellen Zeile Platz schaffen:

- ▷ Sie fügen 8 zusätzliche Leerzeilen ein: <ENTER> 8 <SPACE>

Oberflächlich betrachtet scheint es, als ob Sie das Kommando SPACE verstärkt haben wie zuvor ein Bewegungskommando. Tatsächlich stellt die 8 jedoch eine Bereichsangabe dar, nämlich "8 Zeilen". Auf Bereichsangaben werden wir in der folgenden Beispielsitzung noch näher eingehen.

- ▷ Sie wollen Ihren Dialog fortsetzen:

```
Ich will mit 'ner terribel feinen Stimme reden: "Thisne, Thisne! - Ach
Pyramus, mein Liebster schön! Deine Thisbe schön und Fräulein schön!"
<RETURN> <RETURN>
Squenz: Nein, nein! Ihr müßt den Pyramus spielen und, Flaut, Ihr, die
Thisbe.
<RETURN> <RETURN>
Zettel: Gut, nur weiter! <RETURN>
```

Für den unermüdlichen *Zettel* ist weiteres vorgesehen. Eine Löwenrolle will er sich auch noch "krallen". Die Stelle findet sich einige Zeilen weiter unten.

- ▷ Sie bewegen Dateiausschnitt und Schreibmarke gleichzeitig um 18 Bildschirmzeilen nach unten mit dem verstärkten Kommando <ENTER> 18 <LINEFEED>.

Die Erwiderung des *Squenz* an dieser Stelle könnte auch wesentlich gewitzter sein. Sie tauschen sie also aus. Dazu brauchen Sie nicht die ganze Zeile zu löschen. Der Anfang kann stehen bleiben:

- ▷ Sie bewegen die Schreibmarke mit <WORDRIGHT> auf den ersten Buchstaben des Wortes *schnell*. Den Zeilenrest ab der aktuellen Schreibmarkenposition löschen Sie mit <ENTER> <ERASE>. Sie geben den überarbeiteten Entwurf ein:

```
extempore machen: es ist nichts wie brüllen.
<RETURN> <RETURN>
```

Hier mischt sich nun wieder *Zettel* ein. "Sein" *Löwe* muß jedoch noch etwas stärker zur Geltung kommen. Sie ergänzen einen Satz, den Sie vor *Ich will brüllen...* einfügen. Dieser Satzanfang bietet sich geradezu zur Wiederholung an.

- ▷ Sie bewegen die Schreibmarke mit mehrmaligem <WORDRIGHT> zum *I* des *Ich*. Sie kopieren den ganzen Zeilenrest ab der aktuellen Position der Schreibmarke mit <ENTER> <PICK> in den sogenannten (unbenannten) Standard-PICK-Puffer. Von dort können Sie die Kopie später in den Text übertragen. Sie springen zunächst ans Zeilenende mit <ENDLINE> und ergänzen: es einem <RETURN>

Hier müssen Sie mal wieder mit <SPACE> Platz für eine Zeile schaffen um einzugeben:

Menschen im Leibe wohltun soll, mich zu hören.

Nach Eingabe des abschließenden Leerzeichens fügen Sie mit <PUT> den zuvor kopierten Zeilenrest an der Position der Schreibmarke nochmals ein.

Mit den Kommandos PICK und PUT haben Sie also den Standard-PICK-Puffer des MAXed verwendet- doch dazu mehr in der nächsten Beispielsitzung. Es stehen im Moment nämlich noch dringende Aufgaben an: Der in Klammern gesetzte *Widerspruch der Übrigen* sollte sich lautstark äußern!

- ▷ Sie setzen die Schreibmarke mit mehrmaligem <RETURN> wieder an den Anfang dieses Textes auf Zeile 85 und schalten den Eingabemodus auf Überschreiben um: <INSERTMODE>

Squenz: Matz Schlucker, Ihr müßt Thisbes Mutter spielen. Tom Schnauz, der Kesselflicker!

Schnauz: Hier Peter Squenz.

Squenz: Ihr, des Pyramus Vater, ich selbst, Thisbes Vater. Schnock, der Schreiner, Ihr, des Löwen Rolle! Und so wäre dann halt 'ne Komödie in den Schick gebracht.

Schnock: Habt Ihr des Löwen Rolle aufgeschrieben? Bitt Euch, wenn Ihr sie habt, so gebt sie mir; denn ich habe einen schwachen Kopf zum Lernen.

Squenz: Ihr könnt sie extempore machen: es ist nichts als brüllen.

Zettel: Laßt mich den Löwen auch spielen. Ich will brüllen, daß es einem Menschen im Leibe wohltun soll, mich zu hören. Ich will brüllen, daß der Herzog sagen soll: Nochmal brüllen!

(Widerspruch bei den Übrigen)

Mode: OVERWRITE

File abs2.1

Ln 85

Cl 3

Auf der Steuerzeile erkennen Sie, daß sich die Anzeige *Mode* links unten von INSERT (Einfügen) auf OVERWRITE (Überschreiben) geändert hat. Beim Tippen des folgenden Textes überschreiben Sie den darunterliegenden Text:

- ▷ Sie geben ein:

```
Sequenz: Wenn Ihr es gar zu fürchterlich machtet, so würdet Ihr die
Herzogin und die Damen erschrecken, daß sie schrien, und das brächte uns
<RETURN>
<SPACE>
```

Der unterliegende Text der aktuellen Zeile ist gelöscht! Das Kommando SPACE schreibt nämlich Leerzeichen, und wenn Sie den Eingabe-Modus auf Überschreiben stellen, werden die Leerzeichen nicht vor der aktuellen Zeile eingefügt, sondern sie überschreiben den darunterliegenden Text! Aber nehmen Sie das gelassen hin: Für solche Fälle hat der MAXed das Kommando UNDO!

- ▷ Sie machen die letzte Textveränderung (in diesem Falle das Überschreiben einer Zeile) wieder rückgängig mit <UNDO>. Sie schalten den Eingabe-Modus sicherheitshalber wieder auf INSERT um: <INSERTMODE>

Nun können Sie wieder beruhigt Platz schaffen: <ENTER> 10 <SPACE> – und die Sorgen und Zweifel der *Übrigen* zum Ausdruck bringen:

```
alle an den Galgen.
<RETURN> <RETURN>
```

```
Alle: Ja, das brächte uns an den Galgen, wie wir da sind.
<RETURN> <RETURN>
```

```
Zettel: Zugegeben, Freunde! Wenn ihr die Damen erst so erschreckt, daß
sie um ihre fünf Sinne kommen, so werden sie unvernünftig genug sein,
uns aufzuhängen. Aber ich will meine Stimme forcieren, ich will euch so
sanft brüllen wie ein saugendes Täubchen: ich will euch brüllen, als
wäre es 'ne Nachtigall. <RETURN>
```

Damit haben Sie dieses innige Bedürfnis *Zettels* auch abgewehrt. Wenigstens zu seiner eigenen Rolle soll er unwidersprochen etwas sagen dürfen. Und das haben Sie ja vorhin in der Datei *zettel* festgehalten!

- ▷ Sie bewegen die Schreibmarke mit <ENTER> 3 <MINUSPAGE> nach oben. Sie sehen es schon: *Zettel ergießt sich*. Sie bewegen die Schreibmarke mit mehrmaligem <RETURN> auf diese Zeile. Die Datei *zettel* hält der MAXed noch als sogenannte Alternativ-Datei. Sie blenden sie wieder ein: <EDIT>

Sie erkennen sofort, daß ganz oben noch der Rollename fehlt. Den kopieren Sie sich aus der Datei *abs2.1*, die vom MAXed momentan als Alternativ-Datei im Hintergrund gehalten wird:

- ▷ Sie geben wieder <EDIT> ein. Die Schreibmarke steht "zufällig" auf dem Wort *Zettel*. Zum Kopieren eines Wortes in den Standard-PICK-Puffer gibt es ein spezielles MAXed-Kommando: <PICKWORD>

Mit der "Beute" wechseln Sie durch <EDIT> wieder zur Alternativ-Datei. Sie kommen am schnellsten in die linke obere Fensterecke mit <HOME> und an den Textanfang mit <WORDRIGHT>.

Jetzt müssen Sie nur noch den kopierten Namen einfügen: <PUT> und einen Doppelpunkt sowie ein Leerzeichen am Wortende eingeben.

Die aktuelle Datei speichern Sie mit <SAVE> auf die Festplatte ab, denn da brauchen Sie sie nämlich gleich! Wieder wechseln Sie mit <EDIT> in die Szene, wo sich ein *Liebhaber... auf die honetteste Weise... umbringt*. Die aktuelle Zeile (*Zettel ergießt sich*) löschen Sie: <ERASE>. Jetzt fügen Sie vor der aktuellen Zeile den gesamten Inhalt der auf der Datei *zettel*, wie sie auf der Festplatte abgespeichert ist, ein: <ENTER> *zettel* <PUT>

Wenn Sie am Anfang der Datei den "Änderungsstand" kennzeichnen wollen:

- ▷ Die Schreibmarke springt mit <GOTO> in die erste Zeile der Datei. Hier fügen Sie am Zeilenende (<ENDLINE>) eine Notiz ein:

- fertig -

Sie finden den "Ausgang" aus dem MAXed mit der entsprechenden Taste: <EXIT>. Gleichzeitig speichert der MAXed unaufgefordert alle geänderten Dateien noch ab.

## 2.2 Beispielsitzung 2 - Puffer und Textbereiche

Sie befinden sich in Ihrem Dateiverzeichnis mit den kopierten Beispieldateien. Die Laienschauspielgruppe um den Zimmermann Peter Squenz hat kurzfristig eine Probe im Athener Stadtwald anberaumt (siehe Beispielsitzung 1). Jedoch waren die bisherigen Ideen für die Aufführung einer "Tragischen Komödie" aus Anlaß der herzoglichen Hochzeit noch nicht ganz ausgereift. In dieser schwierigen Situation wurden Sie als "Ghostwriter" der Truppe beauftragt, in aller Schnelle eine brauchbare Vorlage zu liefern! Sie greifen kurzentschlossen auf den Text der Beispieldatei *abs2.2* zurück:

- ▷ Sie rufen den MAXed mit dem Namen der Datei *abs2.2* auf:  
m abs2.2

Die Schreibmarke ist am Anfang der Datei positioniert. Sie wollen im Text noch an einigen Stellen was umändern. Die Position einer dieser Stellen ist Ihnen noch im Gedächtnis geblieben:

- ▷ Sie springen mit der Schreibmarke in Zeile 10 auf die Spalte 12:  
<ENTER> 10 12 <GOTO>

Den Anfang des Textes von *Thisbe* bis einschließlich *Blum* werden Sie also noch entsprechend verarbeiten. Bei diesem Textbereich handelt es sich um eine zusammenhängende Zeichenkette – in der "MAXed-Fachsprache" auch als STREAM bezeichnet -, die sich nur über eine Zeile erstreckt und deshalb auch "einzeiliger STREAM" genannt wird.

- ▷ Sie löschen den genannten Textbereich, indem Sie ihn zunächst durch Bewegen der Schreibmarke kennzeichnen (oder auch: definieren):  
<ENTER>

Die Schreibmarke springt ins Eingabefeld der Steuerzeile und hinterläßt an ihrer aktuellen Position in der Datei ein grafisches Zeichen. Dieses werden Sie als Markierung der einen von zwei benötigten Positionen verwenden, die einen Textbereich eingrenzen. Jetzt geben Sie keine Zeichen im Eingabefeld ein, sondern drücken eine Bewegungstaste: <WORDRIGHT>

```

[Puck tritt auf.

Puck: Welch hausgebacknes Volk macht hier sich breit,
So nah der Wiege unsrer Königin?
Wie? gibts ein Schauspiel? Ich will Hörer sein,
Mitspieler auch vielleicht, nachdem sichs fügt.

Squenz: Sprech, Pyramus; Thisbe, tretet vor.

Pyramus: Thisbe wie eine Blum von Düften duftet süß,
So tut dein Atem auch, o Thisbe, meine Zier!
Doch horch, ich hör ein' Stimm: es ist mein Vater gwiß;
Bleib eine Weile stehn, ich bin gleich wieder hier! [Ab

Puck (beiseit): Ein seltnes Stück von einem Pyramus! [Ab

Thisbe: Umstrahlter Pyramus, an Farbe Lilienweiß
Und rot wie eine Ros' auf triumphierendem Strauch;
Du muntrer Juvenil, der Männer Zier und Preis,
Treu wie das treueste Roß, das nie ermüdet auch:
Ich will dich treffen an, glaub mir, bei Ninus Grab.

```

STREAM Anfang bei Ln 10 Cl 12 File abs2.2 Ln 10 Cl 18

Die Schreibmarke ist momentan wieder im aktuellen Fenster aufgetaucht und hat sich, ausgehend vom grafischen Zeichen, nach rechts weiterbewegt. Der MAXed kommentiert in der Steuerzeile das, was Sie tun: Er hat erkannt, daß Sie einen STREAM-Bereich definieren wollen, dessen Anfangspunkt bei Ln 10 Cl 12 liegt. Steuern Sie das Ziel an:

- ▷ Bewegen mit weiteren <WORDRIGHT> oder <RIGHTARROW> die Schreibmarke auf das *v* des Wortes *von*. Löschen Sie diesen STREAM-Bereich mit <ERASE>.

Mit dieser kleinen Aktion haben Sie bereits ein sogenanntes attribuiertes Kommando ausgeführt. Das Kommandoattribut war diesmal eine Bereichsdefinition, das Kommando war ERASE! Nun zu den Textergänzungen.

- ▷ Geben Sie zunächst an der aktuellen Position noch drei Punkte ein, um darzustellen, daß der Sprecher des *Pyramus* unterbrochen wurde und an dieser Stelle fortfährt.

Vor einer weiteren Texteingabe setzen Sie die Formatgrenzen:

▷ <ENTER> 3 77 <MARGIN>

Bewegen Sie die Schreibmarke um eine Zeile nach oben: <UPARROW>

Wenn Sie die "Beispielsitzung 1" durchgespielt haben sollten, ist Ihnen sicher noch in Erinnerung geblieben, daß Sie mit dem Kommando SPACE Leerzeilen (bzw. Leerzeichen) einfügen können, wenn Sie den Eingabemodus auf Einfügen (INSERT) eingestellt haben. Für die Texteingabe innerhalb eines bestehenden Textes wäre es zweifellos angenehm, wenn Sie dies automatisieren könnten.

▷ Sie stellen den Zeilenschaltungs-Modus auf SPACE um mit:  
<ENTER> s <RETURN>

Beim Sprung der Schreibmarke in die nächste Zeile ist gleichzeitig eine Leerzeile eingefügt worden. Der Zeilenschaltungs-Modus ist umgestellt, d.h. Sie erreichen ab jetzt dasselbe immer wieder durch bloßes Drücken der Kommandotaste <RETURN>!

▷ Sie geben also den neuen Text ein: Pyramus:

Hier nun könnten Sie den Text, den Sie zuvor gelöscht haben, wieder gebrauchen! Kein Problem: Der MAXed hat ihn unaufgefordert im sogenannten (unbenannten) Standard-ERASE-Puffer zwischengespeichert!

▷ Sie fügen den Inhalt des Standard-ERASE-Puffers an der aktuellen Schreibmarkenposition ein mit: <RESTOR>

Mit <ENDLINE> setzen Sie die Schreibmarke ans Zeilenende und schreiben "ungerührt" weiter:

von Giften duftet süß...

("Harter Toback" – aber nicht vergessen: Vor einem neuen Abschnitt zweimal mit <RETURN> eine Leerzeile einfügen!)

Sequenz: Düften! Düften!<RETURN>

Dem Sprecher der *Thisbe* werden Sie auch noch eine Ungeschicklichkeit andichten. Sie vermeiden die Gefahr einer Blindeingabe durch Überschreiten des rechten Fensterrandes, indem Sie den Fenster-Modus MOVE einschalten:

▷ <ENTER> m <RIGHTARROW>

Die Schreibmarke hat sich zwar nicht bewegt, jedoch ist der Fenster-Modus umgestellt. Von nun an verschiebt sich der eingblendete Dateiauschnitt automatisch in dieselbe Richtung wie die Schreibmarke, wenn sie beim Schreiben oder beim Bewegen mit einem ARROW- oder TAB-Kommando an den Fensterrand stößt.

- ▷ Sie bewegen die Schreibmarke mit DOWNARROW 7 Zeilen tiefer auf die Leerzeile vor der *Thisbe* und geben <RETURN> ein. Sie werden bei der Texteingabe das Kommando <RETURN> nur zum Einfügen einer Leerzeile benötigen:

*Thisbe*: Muß ich jetzt reden?

Squenz: Ja, zum Henker, freilich müßt Ihr; Ihr müßt wissen, er geht nur weg, um ein Geräusch zu sehen, das er gehört hat, und wird gleich wiederkommen.

Nach zwei abschließenden <RETURN> und <PLUSLINE> bewegen Sie die Schreibmarke anschließend wiederum 6 Zeilen tiefer in die Leerzeile unter den *Thisbe*-Part und springen an dessen letztes Zeilenende zurück: <UPEND>

Mit dreimaligem <WORDLEFT> stehen Sie unter dem *Ninus* und bauen der *Thisbe* einen Versprecher ein. Dazu löschen Sie das aktuelle Wort mit <ERASEWORD> und fügen stattdessen folgendes ein: Nickels <RETURN> <RETURN>

Das muß natürlich eine Reaktion von *Squenz* zur Folge haben:

Squenz: Ninus' Grab, Kerl. Aber das müßt Ihr jetzt nicht sagen, das antwortet Ihr dem Pyramus. Ihr sagt Euren ganzen Part auf einmal her, Stichwörter und den ganzen Plunder. – Pyramus, tretet auf, Euer Stichwort ist schon dagewesen; es ist: e r m ü d e t a u c h.

Sie dagegen ermüden noch nicht! Ihre weiteren Stichworte warten in Form zweier Gesänge am Dateiende, die noch an die richtige Stelle verschoben werden müssen.

- ▷ Zunächst stellen Sie den Zeilenschaltungs-Modus wieder auf NORMAL zurück: <ENTER> n <RETURN>

Sie springen ans Dateiende mit: <ENTER> . <GOTO>

Nach kleinen Bewegungen nach oben mit <UPARROW> lassen Sie die Schreibmarke auf der Leerzeile über dem "Kuckuck" nieder. Sie löschen den "KuckucksGesang" und kopieren ihn gleichzeitig in einen ERASE-Puffer namens *vogell*. Das leiten Sie folgendermaßen ein: <ENTER>

Als erstes muß die Bereichsdefinition eingegeben werden: Es soll ein Zeilenbereich definiert werden, auch unter dem Begriff RANGE geläufig, der sich von der aktuellen Zeile bis zum Dateiende erstreckt! Bereichsdefinitionen können Sie auch als Zeichenketten-Attribut im Eingabefeld der Steuerzeile angeben. Bei einem RANGE-Bereich geben Sie z.B. die Zeilennummern der ersten und letzten Zeile des gewünschten Zeilenbereichs an und trennen die Nummern durch einen Bindestrich.

In Ihrem Fall bietet sich eine sehr einfache Bereichsangabe an: Die erste Zeile ist bereits durch die Markierung der aktuellen Schreibmarkenposition (das grafische Zeichen) gekennzeichnet. Diese Zeilennummer brauchen Sie dem MAXed also nicht erst anzugeben; er nimmt automatisch die aktuelle Position an Stelle der fehlenden Bereichsgrenze innerhalb des Zeichenketten-Attributs. Für die zweite Bereichsgrenze, das Dateiende, gibt es außerdem ein spezielles Symbol, den Punkt.

- ▷ Damit verkürzt sich das Zeichenketten-Attribut für den RANGE-Bereich auf: - .  
Sie geben direkt im Anschluß ein trennendes Leerzeichen ein und tippen ein zweites Kommandoattribut, den ERASE-Puffernamen ein: voge11

```
Bis in den Himmel 'nein...
█
Der Kuckuck, der der Grasmück
  So gern ins Nestchen heckt
Und lacht darob mit arger Tück
  Und manchen Ehmann neckt...
```

>-. vogel1 █ File abs2.2 Ln 173 Cl 3

▷ Jetzt fehlt nur noch der Aufruf des Kommandos: <ERASE>

Der "Kuckuckgesang" ist verschwunden und im ERASE-Puffer mit dem Namen *vogel1* gelandet. Wenn was schiefgelaufen ist, können Sie die letzten textverändernden Kommandos (also auch Löschkommandos) in umgekehrter Reihenfolge wieder durch mehrmaliges <UNDO> rückgängig machen und bei einem vorhergehenden Bearbeitungsstand erneut aufsetzen.

Der "Kuckuck" im ERASE-Puffer *vogell* kann noch warten, die "Schwalbe" ist jetzt an der Reihe:

- ▷ Verwenden Sie <MINUSLINE> und <UPARROW>, um die Schreibmarke auf die erste Spalte der Leerzeile vor dem "Schwalbengesang" zu setzen. Sie probieren jetzt eine Bereichsdefinition eines Rechtecks (RECTANGLE). Dieses Rechteck soll den gesamten "Gesang" umfassen. Der Anfang einer Kommandoattributierung ist immer gleich: <ENTER> und danach: <DOWNARROW>

Der MAXed meldet in der Steuerzeile einen RANGE! Denn die Definition eines RANGE-Bereichs mit der Schreibmarke beginnt genauso wie die eines RECTANGLE-Bereichs.

- ▷ Sie bewegen die Schreibmarke mit <DOWNARROW> nach unten auf die letzte Zeile des Textes.

Solange Sie sich nur in vertikaler Richtung bewegen, bleibt der MAXed bei seiner Auffassung, es könnte ein RANGE-Bereich werden. Erst, wenn Sie eine horizontale Bewegung ausführen, steht es für den MAXed definitiv fest: Das muß eine RECTANGLE-Definition werden.

- ▷ Bewegen Sie die Schreibmarke in der letzten Zeile nach rechts mit <RIGHTARROW>.

Sofort ändert sich auch die Meldung in der Steuerzeile zu "RECTANGLE". Um den gesamten Text dieses "Gesangs" mit dem Rechteck zu umfassen, müßten Sie jetzt eigentlich die Schreibmarke so weit nach rechts bewegen, bis sich der Text aller Zeilen des Rechtecks nur noch links von der Spalte der Schreibmarke befindet.

In Ihrem Fall steht wiederum eine einfachere Methode zur Verfügung: Sie haben die Schreibmarke bereits nach rechts bewegt, sodaß unzweifelhaft ein RECTANGLE-Bereich vorliegt. Wenn Sie die Schreibmarke nun wieder nach links auf die Ausgangsspalte bewegen, signalisieren Sie dem MAXed, daß Sie den ganzen Text rechts bis zum Zeilenende der längsten Zeilen des "Gesangs" als Rechteck haben wollen:

- ▷ Sie bewegen die Schreibmarke also wieder an die Ausgangsspalte zurück mit <LEFTARROW>.

```

Um einen wild zerrissenen Mädchenkranz.
Ein Zauber soll des Liebsten Zunge binden:
Wir wollen still den Weg zur Laube finden. [Alle ab!b
█
Die Schwalbe, die den Sommer bringt,
Der Spatz, der Zeisig fein,
Die Lerche, die sich lustig schwingt
█ Bis in den Himmel 'nein...

```

RECTANGLE Anfang bei Ln 168 Cl 3

File abs2.2

Ln 172 Cl 3

- ▷ Es fehlt nur noch die Eingabe des ERASE-Puffernamens und der Aufruf des Kommandos: `vogel2 <ERASE>`

Nun zu den Stellen, wo die beiden "Gesänge" wieder eingefügt werden sollen.

- ▷ Sie setzen die Schreibmarke auf die Zeile nach *Zettel* (*singt*):  
`<ENTER> 79 <GOTO>`  
 Sie fügen den Inhalt des ERASE-Puffers *vogel1*, ein RANGE-Bereich, vor der aktuellen Zeile ein: `<ENTER> vogel1 <RESTOR>`

Der "Kuckucksgesang" würde sicher etwas stärker zur Geltung kommen, wenn Sie ihn nach rechts einrückten. Dazu müssen Sie Leerzeichen einfügen, und zwar vor jeder Zeile die gleiche Anzahl. Man könnte also sagen: Ein Rechteck aus Leerzeichen!

- ▷ Sie positionieren die Schreibmarke mit <DOWNARROW> auf die erste Textspalte der ersten Zeile des "Gesanges". Sie definieren ein Rechteck, das 4 Zeilen und 2 Tabulatorsprünge umfaßt mit:

```
<ENTER> <DOWNARROW> <DOWNARROW> <DOWNARROW>
<TAB> <TAB>
```

Sie füllen dieses Rechteck mit Leerzeichen aus, um den darunterliegende Text im Einfüge-Modus INSERT nach rechts zu verschieben mit <SPACE>

Wie Sie sehen, können Sie mit SPACE nicht nur Leerzeilen einfügen (bzw. eine Zeile mit einer Leerzeile überschreiben), sondern einen beliebigen RANGE-Bereich, RECTANGLE-Bereich oder STREAM-Bereich mit Leerzeichen ausfüllen! Das Kommando SPACE setzen Sie nochmals ein:

- ▷ Sie setzen die Schreibmarke mit UPARROW in die Leerzeile vor der *erwachenden Titania* und fügen einen RANGE-Bereich von fünf Zeilen ausschließlich aus Leerzeichen ein: <ENTER> 5 <SPACE>

Sie haben den RANGE-Bereich einfach durch eine Zahl angegeben. Diese Zahl stellt keine Bereichsgrenze in Form einer Zeilennummer dar, sondern die Anzahl der Zeilen ab der aktuellen Zeile.

Ein Rechteck (RECTANGLE) können Sie wesentlich "flexibler" plazieren als einen Zeilenbereich (RANGE). Sie positionieren die Schreibmarke vor dem Einfügen eines Puffers mit einem RECTANGLE-Inhalt immer an jene Stelle im Text, an der die linke obere Ecke des Rechtecks liegen soll.

- ▷ Sie bewegen die Schreibmarke um eine Tabulatorposition nach rechts: <TAB> <TAB>  
Jetzt fügen Sie das Rechteck des "Kuckucks" ein:  
<ENTER> voge12 <RESTOR>

Einige Zeilen tiefer erhalten die vier *Elfen* der *Titania* einen zusätzlichen Rollentext.

- ▷ Sie bewegen die Schreibmarke um 3 Bildschirmseiten weiter:  
<ENTER> 3 <PLUSPAGE>

An dieser Stelle dürfen die vier sich bereits kurz äußern. Sie können Ihnen durchaus noch zwei weitere Worte zumuten. Die nachstehenden Zeilen bieten sich geradezu an, zur nochmaligen Verwendung kopiert zu werden:

- ▷ Sie positionieren die Schreibmarke auf der Leerzeile über der *Ersten Elfe* und kopieren den RANGE-Bereich, der sich über acht Zeilen ab der aktuellen erstreckt in einen benannten PICK-Puffer namens *achten*:

```
<ENTER> 8 achten <PICK>
```

Nun bewegen Sie Dateiausschnitt und Schreibmarke um 14 Zeilen weiter nach oben:

```
<ENTER> 14 <RVLINEFEED>
```

Die Schreibmarke müßte jetzt in der Zeile direkt über dem Ausdruck von *Allen vier Elfen* zu stehen kommen. Fügen Sie die *achten* hier ein mit <PUT>

Den Namen des PICK-Puffers brauchten Sie nicht einzugeben?! Das liegt daran, daß der MAXed bei jedem Kopierkommando mit PICK den kopierten Textbereich immer in den Standard-PICK-Puffer schreibt, ob Sie nun den Textbereich gleichzeitig in einen benannten PICK-Puffer schreiben oder nicht. Da Sie seit dem Kopieren der *Elfen* in die *achten* kein anderes PICK-Kommando ausgeführt haben, steht der RANGE-Bereich immer noch im "Durchgangs-Puffer" Standard-PICK-Puffer zur Verfügung.

Es wäre doch zu niveaulos, wenn die vier Geister nur Heilswünsche äußern dürften. Das streichen Sie an dieser Stelle:

- ▷ Sie setzen die Schreibmarke auf das *H* des *Heil* der ersten *Elfe* und definieren einen RECTANGLE-Bereich: <ENTER>

Sie bewegen die Schreibmarke mit <DOWNARROW> bis zur Zeile der vierten *Elfe*, definieren mit <RIGHTARROW> <LEFTARROW> das Rechteck und löschen das ganze mit <ERASE>.

Der neue Text, den Sie der ersten *Elfe* aufbürden lautet lediglich: Hier! <RETURN> <RETURN> <ENDLINE>

Die zweite *Elfe* wird da schon mehr gefordert: Und ich!

Um nicht zuviel Aufwand zu treiben (stellen Sie sich vor, Sie müßten sich für jede *Elfe* was einfallen lassen! Nicht auszudenken!), wenden Sie einen altbewährten Trick an, der nicht nur bei Theater-Chören angewandt wird: Das Stilmittel der Wiederholung!

- ▷ Sie setzen die Schreibmarke auf das *U* von *Und* und kopieren den Zeilenrest in den Standard-PICK-Puffer: <ENTER> <PICK>

Sie "versorgen" die restlichen zwei *Elfen* jeweils durch die Kommandofolge: <DOWNARROW> <DOWNARROW> <PUT>

Damit lassen Sie es erst mal bewenden. Die Dateiversion im aktuellen Fenster wollen Sie unter einem anderen Namen abspeichern:

- ▷ Sie speichern die aktuelle Datei unter dem Namen *waldprobe* ab:  
<ENTER> *waldprobe* <SAVE>

Weil der MAXed beim Beenden die aktuelle Dateiversion nicht auch noch in die Datei *abs2.2* abspeichern soll, geben Sie folgendes Kommando:

- ▷ Sie verlassen den MAXed ohne automatisches Abspeichern der aktuellen Dateiversionen mit dem Kommando QUIT. Dazu geben Sie ein:  
<ENTER> *quit* <ENTER>

## 2.3 Beispielsitzung 3 - Pseudo-Dateien und Eingabefeld

- ▷ Sie rufen den MAXed diesmal von Ihrem *HOME*-Verzeichnis aus auf: m

Sofern unter Ihrer Kennung an dieser Datensichtstation mit dem MAXed bereits gearbeitet wurde, blendet er die Datei ein, die er zuletzt in seinem aktuellen Fenster hatte. Wenn Sie also gewissenhaft geübt haben, kann es durchaus sein, daß z.B. die Datei *abs2.2* aus der vorangegangenen Beispielsitzung wieder am Bildschirm erscheint. Und das obwohl diese Datei sich vermutlich nicht in Ihrem *HOME*-Verzeichnis befindet! Der MAXed merkt sich nämlich immer den absoluten Pfadnamen. Übrigens befinden sich dann Dateiausschnitt und Schreibmarke noch an derselben Position, wie Sie sie zuletzt hinterlassen haben.

Ihre Beispieldateien stehen jedenfalls noch in einem Unterverzeichnis Ihres *HOME*-Verzeichnisses, das wir *max\_training* nennen wollen. Der Name der im folgenden zu bearbeitenden Datei wird Ihnen momentan noch nicht verraten, sodaß Sie sich erst mal den ganzen Inhalt des Verzeichnisses auflisten lassen. Nicht, daß Sie jetzt den MAXed wieder verlassen, um nachzuschauen! Das können Sie mit dem MAXed schneller und besser:

- ▷ Sie lassen sich den Inhalt Ihres Dateiverzeichnisses mit den Beispieldateien auflisten:  
<ENTER> \$HOME/max\_training <EDIT>

Wenn Sie den richtigen Pfadnamen angegeben haben, blendet Ihnen der MAXed eine seiner Pseudo-Dateien ein, die Pseudo-Datei EDITFILES. Der Name steht wieder in der Mitte der Steuerzeile. Unter der Voraussetzung, daß Sie die voranstehenden Beispielsitzungen wie angegeben durchgeführt haben und in diesem Verzeichnis keine weiteren Dateien angelegt haben, präsentiert Ihnen der MAXed die Pseudo-Datei EDITFILES mit folgender Liste:

```
/usr/alf
/usr/alf/max_training/abs2.1
/usr/alf/max_training/abs2.1.bak
/usr/alf/max_training/abs2.2
/usr/alf/max_training/abs2.3
/usr/alf/max_training/abs2.4
/usr/alf/max_training/abs2.5
/usr/alf/max_training/abs2.6
/usr/alf/max_training/waldprobe
/usr/alf/max_training/zettel
/usr/alf/max_training/zettel.bak
```

MAEHL: Datei und gebe &lt;EDIT&gt;

File max\_training/\* Ln 1

Cl 1

Bevor Sie die Liste in Angriff nehmen, noch ein Wort zum letzten MAXed-Kommando: Die Verwendung der Umgebungsvariablen *HOME* wird Sie vielleicht überrascht haben, den MAXed jedenfalls nicht. Er "kennt" alle gültigen Umgebungsvariablen, wenn Sie sie im Eingabefeld der Steuerzeile angeben. Bevor Sie ein attributiertes MAXed-Kommando ausführen, inspiziert der MAXed nämlich Ihre eingegebenen Zeichenketten-Attribute im Eingabefeld. Er sucht hierbei nach Zeichenketten, mit denen er eine sogenannte Textersetzung vornehmen kann. Grob gesagt hält er Ausschau nach den Zeichen Dollar \$, Klammeraffe @ und Prozent %, da er direkt hinter diesen Zeichen einen definierten Namen einer Umgebungsvariablen, eines Parameters für Textersetzung bzw. eines PICK-Puffers vermutet.

Das attributierte MAXed-Kommando wird also immer erst nach eventuellem Expandieren der Zeichenketten-Attribute ausgeführt. Schauen Sie sich doch die letzte Kommandoattributierung nochmal an. Das eingegebene Zeichenketten-Attribut ist noch im Eingabepuffer des Eingabefeldes zwischengespeichert!

- ▷ Sie holen die zuletzt eingegebene Kommandoattributierung ins Eingabefeld zurück mit: <ENTER> <BACKSPACE>

Wenn Sie – wenigstens für das Betriebssystem – unter dem Namen *alf* bekannt sein sollten und Sie Ihre Beispieldateien im Unterverzeichnis *max\_training* angelegt haben, könnte jetzt im Eingabefeld folgendes stehen: `/usr/alf/max_training`

Die Umgebungsvariable hat der MAXed also korrekt durch ihren Wert ersetzt. Doch wenden Sie sich wieder der Pseudo-Datei EDITFILES zu:

- ▷ Wie kommen Sie wieder aus dem Eingabefeld heraus? In diesem Fall so, wie Sie hineingekommen sind: <ENTER>

In EDITFILES sind alle Dateinamen des zuvor angegebenen Verzeichnisses inklusive des Namens des übergeordneten Dateiverzeichnisses aufgelistet. Wählen Sie sich eine Datei aus!

- ▷ Bewegen Sie die Schreibmarke auf den Namen der Beispieldatei *abs2.3*. Der MAXed fordert Sie in der Steuerzeile auf: <EDIT>

Die Beispieldatei *abs2.3* wird ins aktuelle Fenster geholt. In der Szene, die Sie sich eingeblendet haben, artikulieren die Beteiligten einige ihrer Sorgen, die die Aufführung der *höchst kläglichen Komödie und dem höchst grausamen Tod des Pyramus und der Thisbe* bei Hofe des *Theseus* betreffen. Sie als verantwortlicher Textautor müssen den angesprochenen, "problematischen" Textstellen selbstverständlich nachgehen und diese nötigenfalls "entschärfen". Doch zunächst gilt es, eine Bestandsaufnahme zu machen. Sie werden also die aufgetauchten Befürchtungen in einer Datei sammeln und auch die geäußerten Lösungsvorschläge notieren.

Sie gehen die Szene durch und finden auch sogleich den ersten Sorgenpunkt in Zeile 18, den *Zettel* vorbringt: Die Gewaltanwendung des *Pyramus* gegen sich selbst!

- ▷ <ENTER> 18 <GOTO>

Sie setzen sich hier eine (unsichtbare) Marke, um die Stelle später schnell wiederzufinden:

- ▷ Sie setzen an der aktuellen Position der Schreibmarke die (unbenannte) Standard-Marke: <LABEL>

Das zweite Problem taucht wenige Zeilen später auf: *Schnauz* macht auf die Empfindlichkeit höfischer Damen gegen Raubtiere aufmerksam!

- ▷ <ENTER> 19 <LINEFEED>

Sie setzen auch hier eine Marke. Da Sie dieselbe Marke nicht für zwei Positionen gleichzeitig verwenden können, vergeben Sie einen Namen (IDENTIFIER): p2

- ▷ <ENTER> p2 <LABEL>

Auch das dritte Problem, das *Schnock* anspricht, läßt nicht lange auf sich warten: Mondschein muß gewährleistet sein! Diesmal wollen Sie ein Stichwort aus dem Text als Markenname verwenden, da solche Namen häufig aussagekräftiger sind und Ihnen das Eintippen eines Namens erspart:

- ▷ <PLUSPAGE> <PLUSLINE>  
<ENTER> 5 <DOWNARROW>

Sie bewegen die Schreibmarke mit mehrmaligem <WORDRIGHT> auf das Stichwort *Mond*.

Sie setzen eine Marke an der aktuellen Position, die den Namen des Stichwortes trägt, indem Sie das Kommando LABEL mit der Bereichsangabe für das aktuelle Wort, nämlich mit dem Sonderzeichen Komma attributieren. Der MAXed holt sich den IDENTIFIER schon selbst:

- ▷ <ENTER> , <LABEL>

Nun müssen Sie eine neue Datei anlegen, in der Sie die Probleme sammeln. Sie wollen sie *probleme* nennen.

<ENTER> max\_training/probleme <EDIT> <EDIT>

Für eine erfolgversprechende Lösungsuche ist eine geordnete Darstellung erste Voraussetzung. Wenigstens soll das Textbild wieder einen ordentlich Eindruck machen. Dazu setzen Sie erst den linken Formatrand, an dem Sie den Text dieser Datei linksbündig beginnen wollen, und zwar auf Spalte 3:

- ▷ Sie bewegen die Schreibmarke in die Spalte 3 und geben <MARGIN>. Mit <RETURN> setzen Sie die Schreibmarke in die nächste Zeile an den linken Formatrand.

Die Überschriften sollen jedoch am linken Fensterrand beginnen! Der MAXed stellt sich auch ausgefallenen Wünschen nicht in den Weg. Er beläßt es bei der Warnung, indem er die Spaltennummer in der Steuerzeile invers darstellt:

- ▷ Sie springen an den Zeilenanfang mit <STARTLINE> und geben ein:  
Aufgetauchte Probleme bezüglich der kläglichen Komödie:  
<RETURN> <RETURN>

An der aktuellen Position setzen Sie wieder die Standard-Marke:  
<LABEL>

Keine Bange! Die zuvor gesetzte Standard-Marke existiert immer noch. Der MAXed hält sich nämlich für jede geöffnete Datei einen eigenen Satz an Marken!

- ▷ Sie lassen einige Zeilen frei: <ENTER> 5 <RETURN>  
In dieser Zeile schreiben Sie die zweite Überschrift: <STARTLINE>  
Vorgetragene Einfälle zur Behebung der aufgetretenen Probleme:  
<RETURN> <RETURN>

An der aktuellen Position setzen Sie eine weitere Marke. Ihnen kommt dafür der Name *einfall* in den Sinn: <ENTER> einfall <LABEL>

Jetzt wäre es günstig, wenn Sie beide Dateien gleichzeitig auf dem Bildschirm betrachten könnten! Sie legen einfach ein weiteres Fenster an:

- ▷ Sie positionieren die Schreibmarke etwa in der Mitte des aktuellen Dateiausschnitts, indem Sie die Schreibmarke z.B. auf Zeile 12 setzen: <ENTER> 12 <GOTO>  
Sie unterteilen das aktuelle Fenster horizontal an der aktuellen Zeile und richten unterhalb ein neues Fenster ein: <HWINDOW>

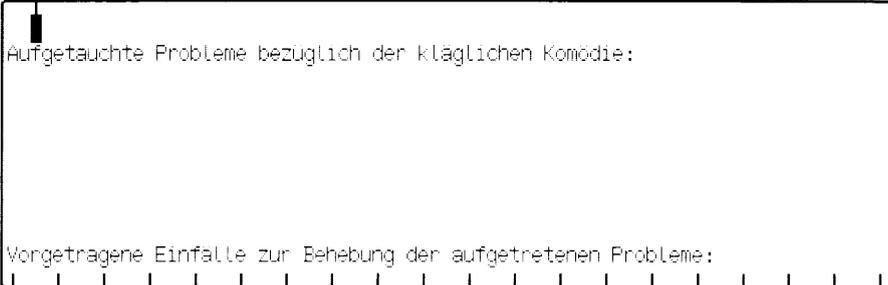
```

.....
Aufgetauchte Probleme bezüglich der kläglichen Komödie:

Vorgetragene Einfälle zur Behebung der aufgetretenen Probleme:

.....

```



```

Mode: INSERT      ..aining/probleme  Ln 1  Cl 3

```

Das untere Fenster wird zum aktiven Fenster: Das sehen Sie daran, daß die Umrandung des Fensters aus einer durchgezogenen Linie besteht und Sie die Schreibmarke hier zum Arbeiten bewegen können. Das obere, passive Fenster ist in der rechten unteren Ecke mit einer Nummer, seiner Fenster- nummer, gekennzeichnet.

Da Sie dem Kommando `HWINDOW` keinen Dateinamen (`IDENTIFIER`) als Kommandoattribut angegeben haben, blendet der `MAXed` dieselbe Datei nochmals ein.

- ▷ Sie wechseln im aktuellen Fenster die aktuelle Datei mit `<ENTER> @b/abs2.3 <EDIT>`

Dieselbe Textstelle wird eingeblendet, die Sie zuvor im oberen, jetzt passiven Fenster eingeblendet hatten, bevor Sie dort zur Datei *probleme* wechselten.

Die zuletzt eingegebene Kommandoattributierung hat folgenden Hintergrund: Der MAXed hatte beim routinemäßigen Kontrollieren Ihres Zeichenketten-Attributs im Eingabefeld den Klammeraffen @ entdeckt, dem ein *b* folgte. Diese beiden Zeichen verwandelte er sogleich in einen Pfadnamen, nämlich in den Pfadanteil des vollständigen Pfadnamens der aktuellen Datei. Und dies war zu dieser Zeit die Datei */usr/alf/max\_training/probleme*. Der Dateiname *abs2.3* erhielt auf diese Weise den gleichen Kopf: */usr/alf/max\_training/abs2.3*.

Ohne die Angabe des Pfadnamens hätten Sie *abs2.3* nicht im Fenster eingeblendet bekommen, da das aktuelle Dateiverzeichnis des MAXed Ihr *HOME*-Verzeichnis ist, aus dem Sie ihn gestartet haben!

Die Fenster wechseln Sie übrigens mit dem Kommando <CHWINDOW>. Probieren Sie es aus! Sie werden feststellen, daß sich die Angaben in der Steuerzeile auf die jeweils aktuelle Datei, d.h. diejenige Datei, die im aktiven Fenster eingeblendet ist, beziehen.

Doch wieder zurück zu den anstehenden Aufführungsproblemen! Zuerst wollen Sie sich die aufgetauchten Sorgenpunkte notieren. Sie bleiben im unteren Fenster, in dem die Datei *abs2.3* eingeblendet ist.

- ▷ Sie springen an die Position der Standard-Marke mit:  
<ENTER> <GOTO>  
Sie kopieren den ganzen Absatz in den PICK-Puffer *totmachen* mit:  
<ENTER> \$ *totmachen* <PICK>

Das Sonderzeichen Dollar \$ steht für den gesamten aktuellen Absatz, der durch Leerzeilen begrenzt ist. Wenige Zeilen weiter unten hat *Zettel* schon eine Lösung des Problems parat. Sie wollen *Zettels* "Kernsatz" kopieren.

- ▷ <PLUSLINE> <PLUSLINE>  
Bewegen Sie die Schreibmarke auf das *S* von *Schreibt*. Dann kopieren Sie den Satz als STREAM-Bereich:  
<ENTER> <RETURN> <RETURN> <UPEND> <WORDLEFT> <WORDLEFT>  
Sie geben den IDENTIFIER *prolog1* als PICK-Puffername ein und kopieren den STREAM-Bereich: *prolog1* <PICK>

Sie wechseln das Fenster und kopieren Problem und Lösung in die Datei *probleme*:

```

▷ <CHWINDOW>

   <ENTER> <GOTO>
   <ENTER> totmachen <PUT>
   , <ENTER> einfall <GOTO>
   <ENTER> prolog1 <PUT>
   <CHWINDOW>

```

Jetzt haben Sie doch glatt wieder vergessen, wie die zweite Marke heißt, die Sie in dieser Datei gesetzt haben, um die gefundenen Textstellen schneller aufzuspüren! Aber der MAXed protokolliert alle verschiedenartigen IDENTIFIER, und so können Sie sich seine Notizen jederzeit wieder in seinen Pseudo-Dateien ansehen!

```

▷ Sie lassen sich die Pseudo-Datei LABELS mit der Liste aller existierenden Marken der aktuellen Datei im aktuellen Fenster einblenden:
   <ENTER> @ <GOTO>

```

Die benannten Marken und die unbenannte Standard-Marke sind untereinander aufgelistet. Zur Erinnerung ist jeweils die Zeile mit der markierten Markenposition angefügt! Sie wählen sich die Marke *p2* aus (Sie erinnern sich an die Vorgehensweise in der Pseudo-Datei EDITFILES?):

```

▷ Sie setzen die Schreibmarke auf die Zeile mit dem Namen p2 und drücken <GOTO>.
   Sie kopieren die aktuelle Zeile in den PICK-Puffer loewe:
   <ENTER> loewe <PICK>

```

Schnauz – nicht dumm – weiß gleich darauf Rat:

```

▷ <ENTER> 9 <LINEFEED>
   Sie kopieren die zwei Zeilen: <ENTER> 2 prolog2 <PICK>

```

Das letzte Problem nehmen Sie auch gleich mit:

```

▷ <ENTER> mond <GOTO>
   Sie kopieren die aktuelle Zeile in den PICK-Puffer mond:
   <ENTER> mond <PICK>

```

Die Lösung findet sich gleich 8 Zeilen weiter:

```

▷ <ENTER> 8 <LINEFEED>
   Sie kopieren den aktuellen Absatz: <ENTER> $ fenster <ERASE>

```

Das war wohl die falsche Taste. Aber Sie wissen ja, daß auch gelöschte Textbereiche gespeichert werden.

- ▷ Sie machen die letzte Textveränderung (das Löschkommando) wieder rückgängig: <UNDO>  
Sie wechseln wieder das Fenster und springen auf die Problemsammlung:  
<CHWINDOW> <ENTER> <GOTO>

Auch für die PICK-Puffer stellt der MAXed eine Pseudo-Datei zur Verfügung. Darauf greifen Sie sofort zurück:

- ▷ Sie blenden die Pseudo-Datei PICKBUFFERS ein: <ENTER> @ <PUT>  
Sie setzen die Schreibmarke auf die Zeile mit dem Namen *loewe* und drücken <PUT>.

Das zweite Problem ist nach dem ersten eingefügt. Das dritte folgt nach einem <RETURN> sogleich. Die Pseudo-Datei PICKBUFFERS ist jetzt Alternativ-Datei, was Ihnen den Aufruf erleichtert:

- ▷ Mit <EDIT> blenden Sie wieder PICKBUFFERS ein, wählen den PICK-Puffer *mond* aus und fügen das dritte Problem nach dem zweiten ein: <PUT>

Sie springen zu den Lösungsvorschlägen mit  
<ENTER> einfall <GOTO>.

Nach zweimaligem <RETURN> kopieren Sie den PICK-Puffer *prolog2* auf die gleiche Weise ein. Danach setzen Sie die Schreibmarke mit weiteren zwei <RETURN> wieder ans Textende.

Das *fenster* ist in der Liste nicht aufzufinden! Ohne den Mondenschein sieht es allerdings etwas "düster" aus für die sommernächtliche Komödienveranstaltung! Sie bringen etwas "Licht" in die Sache, indem Sie sich erinnern, daß Sie das "*fenster*" ja gar nicht als PICK-Puffer, sondern als ERASE-Puffer angelegt hatten. Dafür hat der MAXed natürlich eine eigene Pseudo-Datei:

- ▷ Sie blenden sich die Pseudo-Datei ERASEBUFFERS mit den Namen aller angelegten ERASE-Puffer ein: <ENTER> @ <RESTOR>

Sie kopieren sich die "Fenster-Idee" in Ihre Datei *probleme*, indem Sie mit der Schreibmarke auf den ERASE-Puffer *fenster* zeigen und <RESTOR> eingeben.

Auch wenn Sie den Puffer nicht mehr aufgefunden hätten, wäre der gelöschte Absatz noch zu retten gewesen: Der MAXed führt nämlich einen großen "Abfall-Behälter" in Form einer weiteren Pseudo-Datei namens RESTORE. Schauen Sie da doch mal nach:

- ▷ Sie blenden sich die Pseudo-Datei RESTORE ein: <ENTER> <EDIT>

"Stöbern" Sie hier mal durch Bewegen von Fenster und Schreibmarke "herum". Sie finden alle Texte wieder, die Sie in irgendeiner Weise im Laufe der Sitzung verändert haben, auch den "versehentlich" gelöschten Absatz.

Zum Abschluß würde Sie sicherlich noch interessieren, welche Dateien im Verlauf der bisherigen Sitzung angelegt bzw. geöffnet wurden:

- ▷ Sie blenden sich die Pseudo-Datei OPENFILES mit der Liste aller vom MAXed temporär gehaltenen Dateien ein: <ENTER> @ <EDIT>

Sie erkennen die Namen von Pseudo-Dateien, die vom MAXed lediglich für die aktuelle MAXed-Sitzung am Leben erhalten werden, an den spitzen Klammern. Jene Dateien, die Sie während der Sitzung verändert haben, sind invers dargestellt.

- ▷ Sie positionieren die Schreibmarke auf die Zeile mit dem Dateinamen *probleme* und speichern diese Datei ab mit <SAVE>.

Auch die Datei *abs2.3* "leuchtet" Ihnen entgegen! Diese wollen Sie keinesfalls in einem versehentlich veränderten Zustand abspeichern:

- ▷ Sie setzen die Schreibmarke auf die Zeile mit dem Namen *abs2.3* und holen sich die Datei ins aktuelle Fenster mit: <EDIT>

Es fällt Ihnen hier ebenfalls auf, daß der Dateiname in der Steuerzeile in warnender Inversdarstellung anzeigt, daß in dieser Datei seit Sitzungsbeginn Änderungen vorgenommen wurden.

- ▷ Sie setzen die aktuelle Datei wieder auf den Stand bei Sitzungsbeginn mit <ENTER> - <EDIT>

Alle Veränderungen in dieser Datei sind aufgehoben. Deshalb erhält auch die Anzeige des Dateinamens wieder ihre normale Darstellung zurück.

Das zweite Fenster benötigen Sie jetzt nicht mehr:

- ▷ Sie löschen das zuletzt erzeugte Fenster mit <ENTER> <HWINDOW>

Das alte Fenster kann wieder den ganzen MAXed-Bildschirm einnehmen. Als letztes wollen Sie in Ihrer Datei *probleme* noch eine Überschrift einfügen und einen Datumsstempel setzen:

- ▷ Sie blenden die Pseudo-Datei OPENFILES wieder ein mit <EDIT>  
Sie wählen sich *probleme* aus und blenden die Datei ein: <EDIT>

Sie verschieben das Fenster nach oben mit <MINUSLINE> und bewegen die Schreibmarke auf das Wort *Einfälle*. Sie kopieren dieses in den PICK-Puffer *p*: <ENTER>, *p* <PICK>

Dann setzen Sie die Schreibmarke in der Zeile 2 auf das Wort *Probleme* und kopieren es in den Standard-PICK-Puffer: <PICKWORD>

Mit <HOME> gelangen Sie in die linke obere Fensterecke in der ersten Zeile. Nachdem Sie mit <SPACE> eine Leerzeile eingefügt haben, schreiben Sie die Überschrift. Dazu nehmen Sie die Textersetzung des MAXed zu Hilfe:

```
<ENTER> 'Datei @f mit %%n und %p%n' <PUT>
```

Für das Kommando PUT ist diese Zeichenkette kein IDENTIFIER, da sie in Hochkommata gesetzt ist. Der Parameter für Textersetzung *@f* gibt den Pfadnamen der aktuellen Datei aus, %% bzw. %p% werden durch den Inhalt des Standard-PICK-Puffers bzw. des PICK-Puffers *p* ersetzt. Das aktuelle Datum kann der MAXed auch selber herausfinden und durch einen Parameter für Textersetzung ausgeben:

- ▷ Sie springen ans Zeilenende mit <ENDLINE> und geben <TAB>  
Hier fügen Sie das Datum ein: <ENTER> '@d' <PUT>
- ▷ Sie beenden die MAXed-Sitzung mit <ENTER> <EXIT>

Dieses Kommando veranlaßt den MAXed, mit dem Speicherplatz auf der Festplatte "sparsamer" umzugehen: Die zusätzlichen *.bak*-Dateien als Sicherungskopien der "Vor-Sitzungs-Zustände" aller veränderten Dateien werden gestrichen!



Sie haben dem MAXed beim Aufruf als Dateiangabe einen IDENTIFIER mitgegeben, der das Sonderzeichen Stern \* enthält. Damit sprechen Sie in diesem Fall mehrere Dateinamen an, die mit der Zeichenkette *abs* beginnen. Alle Dateinamen, die zu diesem IDENTIFIER passen, sind nun in der Pseudo-Datei zur Auswahl aufgelistet. In dieser Beispielsitzung werden Sie davon lediglich die Dateien *abs2.4* und *abs2.5* benötigen. Sie gedenken, diese Auswahlliste zu einem späteren Zeitpunkt nochmals zu verwenden, und streichen aus Übersichtsgründen alle anderen Dateinamen wieder heraus.

- ▷ Sie bewegen die Schreibmarke auf die entsprechenden Zeilen, um alle Dateinamen außer *abs2.4* und *abs2.5* mit <ERASE> zu löschen.

Nun gehen Sie an die "dauerhaften" Änderungen Ihrer "unsterblichen" Bühnentexte in der Datei *abs2.5*.

- ▷ Sie positionieren die Schreibmarke auf den Dateinamen *abs2.5* und blenden diese Datei im aktuellen Fenster ein: <EDIT>.

Die 2. *Szene des IV. Akts* erscheint vor Ihren Augen. Sie überprüfen zunächst, ob noch weitere Szenen in dieser Datei vorhanden sind. Da das Suchen zeitaufwendig sein kann, geben Sie diese Arbeit an den MAXed ab:

- ▷ Sie lassen den MAXed in Vorwärtsrichtung nach dem Wort *szene* suchen, indem Sie ein Suchkommando aufrufen:  
<ENTER> *szene* <PLUSFIND>

Schneller als Sie es vermutlich erwartet hatten, wurde der MAXed auch schon fündig: Gleich in der ersten Zeile entdeckte er etwas Passendes: Das Wort *Szene*. Die Schreibmarke sitzt auf dem ersten Zeichen der gefundenen Textstelle. Er nahm dabei offensichtlich keine Rücksicht auf Groß- und Kleinschreibung! Das erleichtert Ihnen also i.a. die Eingabe. In diesem Fall sind Sie natürlich noch nicht viel weiter gekommen, da Sie selbstverständlich nach einer weiteren Szene suchen. Also nochmal das ganze Suchkommando eingeben? Nein:

- ▷ Sie wiederholen das letzte Suchkommando: <PLUSFIND>

Der MAXed hat sich Ihr zuletzt eingegebenes Suchmuster, auch aktuelles PATTERN genannt, gemerkt und wiederverwendet. Damit hat er bereits die erste Zeile der nächsten Szene gefunden. Die letzte Zeile der vorausgehenden Szene versehen Sie gleich mal mit einem Merker:

- ▷ Sie bewegen die Schreibmarke um eine Zeile nach oben mit `<UPARROW>` und setzen an dieser Position der Schreibmarke die Standard-Marke: `<LABEL>`  
Sie springen mit der Schreibmarke wieder an den Dateianfang: `<GOTO>`

”Jede Szene sollte in eine eigene Datei” lautet einer Ihrer Grundsätze, weshalb Sie die vorausgehende Szene schon mal in einen PICK-Puffer zwischenspeichern:

- ▷ Sie kopieren den Zeilenbereich (RANGE) von der aktuellen Zeile (momentan die erste Zeile) bis zur Zeile mit der unbenannten Standard-Marke in den PICK-Puffer namens *szene\_IV2*:  
`<ENTER> -! szene_IV2 <PICK>`

Das Ausrufungszeichen ! repräsentiert die Position der Standard-Marke. Die Datei für diese Szene möchten Sie wie den PICK-Puffer benennen. Weshalb also nochmal neu eintippen, wenn der Name noch im Eingabepuffer gespeichert ist!

- ▷ Sie holen die letzte Kommandoattributierung aus dem Eingabepuffer ins Eingabefeld der Steuerzeile zurück: `<ENTER> <BACKSPACE>`  
Sie setzen die Schreibmarke an den Anfang des Eingabefeldes: `<UPARROW>`

Sie löschen die nicht benötigte Bereichsangabe -! :

`<DELCHAR> <DELCHAR>`

Sie legen die neue Datei *szene\_IV2* an: `<EDIT> <EDIT>`

Sie könnten jetzt die Szene ganz einfach mit `<PUT>` in die neue Datei kopieren, denn sie ist momentan auch noch im Standard-PICK-Puffer zwischengespeichert. Auch wäre es ein leichtes für Sie den Inhalt des Eingabepuffers ein weiteres Mal zu verwerten. Aber Abwechslung muß sein:

- ▷ Sie kopieren den Inhalt des PICK-Puffers *szene\_IV2* an die aktuelle Dateiposition: `<ENTER> sz* <PUT>`

Auch Puffernamen können Sie also mit Sonderzeichen abkürzen. Der MAXed hat zum eingegebenen IDENTIFIER unter den momentan existierenden PICK-Puffernamen genau einen passenden gefunden, nämlich *szene\_V2*, und sogleich den Inhalt dieses Puffers in die aktuelle Datei kopiert.

Die eingeblendete Szene bedarf noch einer Überarbeitung. Welchen Textstellen Sie wie den "letzten Schliff" geben, haben Sie sich zwischen den anstrengenden Probenterminen im Athener Stadtwald in der Datei *abs2.4* notiert. Sie blenden sich diese Notizen geschickterweise in einem zweiten Fenster ein:

- ▷ Sie positionieren die Schreibmarke auf die Spalte 48:

```
<ENTER> ,48 <GOTO>
```

Sie teilen das aktuelle Fenster an der aktuellen Spaltenposition vertikal und erzeugen rechts davon ein neues Fenster, in das die Datei *abs2.4* eingeblendet wird:

```
<ENTER> abs2.4 <VWINDOW>
```

Das neue Fenster ist nun das aktuelle Fenster und Sie greifen gleich den ersten Korrekturhinweis auf: Der Begriff *heißer Föhn* erscheint Ihnen – zumindest für Ihre Szene – etwas zu modisch, weshalb Sie ihn durch den altbewährten Begriff *Phönix* ersetzen wollen.

- ▷ Sie setzen die Schreibmarke mit <DOWNARROW> auf den Anfang des Wortes *heißer*.

Sie kopieren die beiden Wörter in den Standard-PICK-Puffer:

```
<ENTER> <WORDRIGHT> <WORDRIGHT> <LEFTARROW> <PICK>
```

Sie wechseln wieder in das Fenster "Null": <CHWINDOW>

Sie geben das Suchkommando: <ENTER> % <PLUSFIND>

Bei den Suchkommandos MINUSFIND und PLUSFIND hat das Prozent-Zeichen eine Sonderbedeutung: Es steht für den Standard-PICK-Puffer, wird jedoch bei der obigen Form der Kommandoattributierung nicht im Eingabefeld der Steuerzeile durch den Pufferinhalt ersetzt. Der MAXed sucht nach einer Textstelle, die mit dem Pufferinhalt identisch ist.

Doch wenden Sie sich wieder der gefundenen Textstelle zu:

- ▷ Sie löschen die beiden Wörter: <ERASEWORD> <ERASEWORD>

Sie fügen den neuen Begriff ein: Phönix

Wie in der Datei im rechten Fenster notiert, sind zwei Regieanweisungen in der Szene falsch plaziert und müssen ausgetauscht werden. Damit Sie nicht zwei Suchkommandos einzugeben brauchen, "packen" Sie beide Suchmuster in ein einziges PATTERN!

- ▷ Sie geben das Suchkommando mit zwei alternativen Suchmustern ein:  
`<ENTER> "schnock geht|zettel geht" <PLUSFIND>`

Mehrere alternative Suchmuster trennen Sie in einem PATTERN durch den senkrechten Strich. Die Anführungszeichen sind wieder notwendig, da sich innerhalb des PATTERNs Leerzeichen befinden: Damit der MAXed einzelne Zeichenketten-Attribute auseinanderhalten kann, darf ein Zeichenkettenattribut keine Leerzeichen enthalten, oder es muß durch Anführungszeichen begrenzt sein.

Prompt hat der MAXed auch schon die eine der beiden Textstellen gefunden. Da diese offensichtlich vertauscht sind, löschen Sie zunächst die eine:

- ▷ Sie löschen die aktuelle Zeile und speichern Sie im Standard-ERASE-Puffer: `<ERASE>`  
 Sie suchen die zweite Textstelle: `<PLUSFIND>`

Das zuletzt eingegebene PATTERN hat sich der MAXed selbstverständlich wieder gemerkt, sodaß Sie es nicht nochmals eingeben mußten, und hat damit die zweite Textstelle ebenfalls identifiziert. Der Austausch der beiden Textstellen ist jetzt für den MAXed (und für Sie) einfacher zu erledigen:

- ▷ Sie tauschen die aktuelle Zeile mit der zuvor gelöschten Textstelle (deren Inhalt sich im Standard-ERASE-Puffer befindet) aus mit `<EXCHANGE>`.

Das war schon alles! Gleichzeitig hat der MAXed die ehemals an dieser Stelle befindliche Textstelle an der Stelle in Ihrer Szene wieder eingefügt, an der Sie gerade eben die Zeile löschten. Wenn Sie sich überzeugen wollen, schauen Sie nach:

- ▷ Sie suchen rückwärts nach einer zum aktuellen PATTERN passenden Textstelle: `<MINUSFIND>`

Wie Sie hierdurch bemerkt haben, arbeiten die ansonsten so gegensätzlich orientierten Suchkommandos PLUSFIND und MINUSFIND "Hand in Hand", wenn es sich um das aktuelle PATTERN, d.h. das zuletzt eingegebene PATTERN handelt!

Die nächste Textkorrektur ist von existenzieller Bedeutung: Sie wollen sich und Ihre Mit-Schauspieler bei der Aufführung des Stückes nicht unter Wert verkaufen! Die *fünf Batzen* im Text ersetzen Sie durch den Betrag von *sechs Batzen*. Dabei werden Sie sicherheitshalber den ganzen Text durchgehen:

- ▷ Sie springen mit der Schreibmarke an den Dateianfang: <GOTO>  
Sie suchen nach dem nächsten Auftreten von *fünf*:  
<ENTER> fünf <PLUSFIND>

Sie werden dem bescheidenen Bälgenflicker *Flaut* den "richtigen" Wert unterschieben. Da Sie schon ahnen, daß dies nicht die einzige Stelle mit der "preislichen Untertreibung" ist, wenden Sie das wiederverwendbare Ersetzungskommando an:

- ▷ Sie ersetzen die gefundene Textstelle *fünf* durch die Ersetzungszeichenkette *sechs*: <ENTER> sechs <REPLACE>  
Sie lassen den MAXed die nächste passende Textstelle aufspüren:  
<PLUSFIND>  
Sie ersetzen die gefundene Textstelle mit der zuvor eingegebenen Ersetzungszeichenkette: <REPLACE>

Der MAXed ist nämlich in der Lage, sich neben dem aktuellen PATTERN für das Kommando REPLACE zusätzlich die aktuelle Ersetzungszeichenkette zu merken!

- ▷ Sie suchen und ersetzen mit <PLUSFIND> und <REPLACE> weitere passenden Textstellen, bis Sie der MAXed mit folgendner Meldung beruhigt:  
\*\*\*\* PATTERN nicht gefunden.

Die Ausdrucksvariante *Sapperlot* müssen Sie noch unbedingt durch das vornehmere *Sapperment* ersetzen!

- ▷ Sie springen mit der Schreibmarke wieder an den Dateianfang:  
<GOTO>  
Sie suchen nach dem Wort *sapperlot*:  
<ENTER> sapperlot <PLUSFIND>

Diesmal ging es daneben. An der Groß- oder Kleinschreibung kann es nicht liegen. Aber vielleicht taucht der Begriff innerhalb eines Wortes auf! Denn standardmäßig geht der MAXed davon aus, daß ein eingegebenes Suchmuster ein ganzes Wort darstellt! Sie nehmen jetzt an, daß *Sapperlot* einen Wortanfang bildet. Sie zeigen dem MAXed mit dem Sonderzeichen spitze Klammern < und > an, wo die gesuchte Zeichenkette vom restlichen Wort, dem sogenannten Wortkontext, abgetrennt sein könnte. Bei einem vermuteten Wortanfang bedeutet dies, daß Sie das rechte Ende durch die schließende spitze Klammer abschließen:

- ▷ Sie holen das eingegebene PATTERN aus dem Eingabepuffer ins Eingabefeld zurück: <ENTER> <BACKSPACE>  
Sie fügen am Ende die schließende spitze Klammer an: >

Genau genommen wollen Sie ja nur die Einfärbung *lot* des *Sapperlot* durch die günstigere Tönung *ment* ersetzen. Sie trennen das *lot* vom linken Wortkontext *sapper* ab, indem Sie dazwischen die öffnende spitze Klammer einfügen:

- ▷ Sie bewegen die Schreibmarke im Eingabefeld auf das *l* mit viermaligem <LEFTARROW> und fügen die öffnende spitze Klammer ein: <  
Sie probieren es mit dem neuen PATTERN nochmal: <PLUSFIND>

Nun hat der MAXed das *Sapperlot* auch schon entdeckt. Da Sie ihm außerdem mit der öffnenden spitzen Klammer bedeutet haben, daß der linke Teil *sapper* nur Wortkontext sein soll, hat er die Schreibmarke auf den Anfang der gesuchten Zeichenkette, nämlich das *l* gesetzt. Da Sie die Zeichenkette *lot* innerhalb des Wortes *Sapperlotsjunge* durch die Kontextklammern klar abgegrenzt haben, können Sie sie durch ein Ersetzungskommando ausschneiden und stattdessen die Ersetzungszeichenkette *ment* einsetzen:

- ▷ Sie ersetzen die gefundene Zeichenkette *lot* durch *ment*:  
<ENTER> *ment* <REPLACE>

Einen Orthographie-Fehler wollen Sie noch beheben: Das *Fänomen* ist Ihnen als *Phänomen* wesentlich angenehmer. Alle Vorkommnisse dieses "fänomenalen" Wortanfangs in der Szene sollen durch das *Ph* ersetzt werden. Das kann das Kommando REPLACE in einem Durchgang für Sie erledigen, wenn Sie es mit PATTERN, Ersetzungszeichenkette und der Bereichsangabe versehen:

- ▷ Sie beginnen die Kommandoattributierung mit dem PATTERN:  
<ENTER> \F>\änomen

Damit trennen Sie wiederum die gesuchte Zeichenkette *F* vom rechten Wortkontext *änomen*. Zum Zeichen, daß Sie beim Buchstaben *F* die Großschreibung und beim Buchstaben *ä* die Kleinschreibung berücksichtigt sehen wollen, haben Sie diese mit jeweils einem vorausgehenden Gegenstrich ausgestattet.

- ▷ Nach dem PATTERN geben Sie ein trennendes Leerzeichen und anschließend die Ersetzungszeichenkette ein: Ph  
Zum Schluß geben Sie nach einem weiteren trennenden Leerzeichen das Zeichenketten-Attribut für den Bereich (die gesamte aktuelle Datei) an, in dem die Ersetzungen vorgenommen werden sollen:  
. <REPLACE>

```

[Schnock geht.
Schnock: Meisters, der Herzog kommt eben vom
vier andere Herren und Damen mehr sind verhei
gegangen wäre, so wären wir alle gemachte Leu
Flaut: O Lieber Sappermentsjunge, Zettel! So
des Tags für Lebenszeit verloren. Er konnte s
entgehn - und wenn ihm der Herzog nicht sechs
Pyramus gegeben hätte, will ich mich hängen u
sechs Batzen des Tags für den Pyramus, oder g
[Zettel geht.
Zettel: Wo sind die Buben? Wo sind die Herzen
Squenz: Zettel! O allertrefflichster Tag! geb
Zettel: Meisters, ich muß Mundendinge reden,
was! Denn wenn ichs euch sage, bin ich kein e
alles sagen, just wie es sich zutrug!
Alt: Neu:
heißen Fohn Phonix
Austauschen:
Schnock geht Zettel geht
Alt: Neu:
fünf Batzen sechs Batzen
Sapperlot Sapperment
Fänomen Phänomen
Schnock geht Schnock kommt
Zettel geht Zettel kommt
Momentan Laeuft: REPLACING File szene_IV2 Ln 38 Cl 25

```

Das Kommando REPLACE stellt mit der Bereichsdefinition durch ein Zeichenketten-Attribut insofern eine Ausnahme dar, daß in diesem Fall das Zeichenketten-Attribut für den Bereich als letztes von drei Kommandoattributen eingegeben werden muß. Bei anderen MAXed-Kommandos geben Sie grundsätzlich alle Formen der Bereichsdefinition immer als erstes Kommandoattribut ein.

- ▷ Sie kontrollieren die Ersetzungen nach, indem Sie ein Suchmuster eingeben: <ENTER> @rä>

Der Parameter für Textersetzung *@r* wird im Eingabefeld durch die aktuelle Ersetzungszeichenkette *Ph* ersetzt.

- ▷ Sie positionieren die Schreibmarke wieder am Dateianfang mit <GOTO>.

Die letzte Notiz im rechten Fenster weist Sie darauf hin, daß die beiden Regieanweisungen, die Sie zuvor bereits ausgetauscht haben, auch noch falsch sind: Die Akteure müssen an diesen Stellen des Stücks selbstverständlich auftreten und nicht abgehen! Das PATTERN zum Auffinden dieser Regieanweisungen haben Sie zuvor schon mal eingegeben. Als aktuelles PATTERN ist es leider nicht mehr gespeichert. Trotzdem werden Sie es nicht nochmal eingeben müssen: Sie rufen stattdessen die Pseudo-Datei FINDHISTORY auf:

- ▷ Sie blenden die Pseudo-Datei FINDHISTORY im aktuellen Fenster ein:  
<ENTER> @ <PLUSFIND>

"szene"	Alt:	Neu:
"schnock geht zettel geht"	heißen Föhn	Phönix
"fünf"		
"sapperlot"	Austauschen:	
"sapper<lot>"	Schnock geht	Zettel geht
"Phä"		
	Alt:	Neu:
	fünf Batzen	sechs Batzen
	Sapperlot	Sapperment
	Fänomen	Phänomen
	Schnock geht	Schnock kommt
	Zettel geht	Zettel kommt

WAEHLE PATTERN und gebe <+FIND>

File <FINDHISTORY>

Ln 6

Cl 1

Der MAXed hat hier alle PATTERN, die Sie in der aktuellen Sitzung beim Aufruf den Suchkommandos PLUSFIND und MINUSFIND eingegeben haben, für Sie gesammelt! Sie brauchen nur noch auszuwählen und das gewünschte Suchkommando aufrufen. Die Funktionsweise dieser Pseudo-Datei bezeugt wiederum die gute Zusammenarbeit der beiden Suchkommandos: Sie teilen sich die Pseudo-Datei sowohl beim Abspeichern ihrer verwendeten PATTERN als auch beim Wiederverwenden eines dieser PATTERN.

- ▷ Sie bewegen die Schreibmarke auf die Zeile mit dem PATTERN "schnock geht|zettel geht"  
Sie rufen anschließend eines der beiden Suchkommandos auf, im Augenblick sinnvollerweise dasjenige für die Suche in Vorwärtsrichtung: <PLUSFIND>

Der MAXed hat die erste Stelle sogleich gefunden, sodaß Sie ein Ersetzungskommando eingeben können:

- ▷ <ENTER> geht kommt <REPLACE>

Da Sie hierbei keine Bereichsangabe gemacht haben, versucht der MAXed im Rest der aktuellen Zeile eine zum angegebenen PATTERN *geht* passende Textstelle zu finden. Die erste, die er entdeckt, ersetzt er durch die angegebene Ersetzungszeichenkette *kommt*.

- ▷ Sie suchen die nächste Textstelle: <PLUSFIND>

Sie stellen fest, das aktuelle PATTERN, das zuvor vom Suchkommando PLUSFIND verwendet wurde, ist nun durch das PATTERN *geht*, das Sie soeben beim Ersetzungskommando REPLACE eingegeben haben, aktualisiert worden.

- ▷ Sie wiederholen das letzte Ersetzungskommando: <REPLACE> und beenden die aktuelle Sitzung: <EXIT>

## 2.5 Beispielsitzung 5 - Komplexes Suchen und Ersetzen

Ihnen wurde aufgetragen, sich um die endgültige Besetzung des *Rüpelspiels* bei der herzoglichen Hochzeitsfeier zu kümmern. Dazu müssen Sie sich erst einen Überblick verschaffen, wieviele Rollen es letztendlich zu besetzen gilt.

- ▷ Sie rufen den MAXed im Verzeichnis mit Ihren Beispieldateien auf:  
m abs2.5

Die erste Szene in dieser Datei haben Sie bereits in der vorigen Beispielsitzung versorgt. Die beiden letzten Szenen speichern Sie nun auch noch in separaten Dateien ab. Sie wissen vielleicht noch, daß Sie nur nach dem Wort *Szene* suchen müssen, um den Beginn der nächsten Szene zu finden. Wie es der Zufall so will, findet sich dieses Wort bereits am Ende der ersten Zeile.

- ▷ Sie springen mit <ENDLINE> ans Zeilenende und definieren einen STREAM-Bereich, der genau das Wort *Szene* umfaßt:  
<ENTER> <WORDLEFT>  
Sie lassen den MAXed nach einem dazu identischen STREAM-Bereich suchen: <PLUSFIND>

```

IV. AKT - 2. SZENE █
[Athen: Stube in Squenzens Hause

[Squenz, Flaut, Schnauz und Schlucker kommen.

Squenz: Habt ihr nach Zettels Hause geschickt? Ist er noch nicht zu
Haus gekommen?

Schlucker: Man hört nichts von ihm. Ohne Zweifel ist er transportiert!

Flaut: Wenn er nicht kommt, so ist das Stück zum Henker! Es geht nicht
vor sich, nicht wahr?

Squenz: Es ist nicht möglich. Ihr habt keinen Mann in ganz Athen außer
ihm, der kapabel ist, den Pyramus herauszubringen.

Flaut: Nein; er hat schlechterdings den besten Witz von allen
Handwerksleuten in Athen.

Squenz: Ja, der Tausend! und die beste Person dazu! Und was eine süße
PLUSFIND: STREAM                               File abs2.5           Ln 1           Cl 15

```

- ▷ Sie setzen auf der ersten Zeile der ersten Szene des fünften Aktes eine Marke namens *m1*: <ENTER> m1 <LABEL>
- Sie suchen nochmals nach der nächsten Szene mit <PLUSFIND>
- Sie bewegen die Schreibmarke um eine Zeile nach oben: <UPARROW>
- Sie setzen hier nochmals eine Marke: <ENTER> m2 <LABEL>
- Sie kopieren die voranstehende Szene: <ENTER> m1-m2 part1 <PICK>

Sie haben damit den RANGE-Bereich, der durch die Zeilen mit den Marken *m1* und *m2* begrenzt ist, in den PICK-Puffer *part1* kopiert. Jetzt kommt noch die nachstehende Szene dran:

- ▷ Sie springen ans Dateieende: <ENTER> . <GOTO>

Die Schreibmarke steht offensichtlich am Ende der zweiten Szene des fünften Akts. Sie konnten sich eben davon überzeugen, daß auch diese Szene in ihrer ersten Zeile das Wort *Szene* enthält. Mit diesem Wissen können Sie beim Kopieren eine raffinierte Bereichsangabe machen:

- ▷ Sie kopieren die letzte Szene der Datei:

```
<ENTER> "szene"- part2 <PICK>
```

Für die erste Bereichsgrenze des RANGE-Bereichs haben Sie das PATTERN "*szene*" angegeben. Das veranlaßt den MAXed, in Rückwärtsrichtung nach der ersten passenden Textstelle zu suchen und an der entsprechenden Position die erste Bereichsgrenze festzulegen. Da die Angabe für die zweite Bereichsgrenze fehlt, nimmt er die aktuelle Schreibmarkenposition dazu her.

- ▷ Sie legen eine neue Datei an: <ENTER> premiere <EDIT> <EDIT>  
Sie wollen hier den Inhalt eines der erstellten PICK-Puffer einkopieren:

```
<ENTER> p* <PUT>
```

```
█  
PART1 RN:  V. AKT - 1. SZENE  
PART2 RN:  V. AKT - 2. SZENE
```

WAEHLE BUFFER und gebe <PUT>

File <PICKBUFFERS>

Ln 1

Cl 1

Den PICK-Puffernamen haben Sie mit dem Sonderzeichen Stern \* abgekürzt. Der MAXed prüfte, ob einer der existierenden PICK-Puffernamen zu diesem IDENTIFIER paßt. Dabei hat er gleich zwei Kandidaten gefunden: *p1* und *p2*. Die Entscheidung, welcher von beiden zu nehmen ist, überläßt er schon Ihnen, indem er sie in die Pseudo-Datei PICKBUFFERS einliest und anschließend diese zur Auswahl präsentiert. Im Anschluß an jeden PICK-Puffernamen und die Angabe seiner Bereichsart (hier: *RN* für RANGE) wird die erste Zeile des jeweiligen Pufferinhalts ebenfalls angezeigt. Ein Blick darauf erleichtert es Ihnen, den richtigen Puffer zu identifizieren.

- ▷ Sie bewegen die Schreibmarke auf den Puffernamen *part1* und kopieren dessen Inhalt in die aktuelle Datei: <PUT>  
 Sie erstellen eine weitere Datei: <ENTER> *epilog* <EDIT> <EDIT>  
 Sie blenden die Pseudo-Datei PICKBUFFERS (diesmal mit den Namen aller momentan existierenden PICK-Puffer) ein: <ENTER> @ <PUT>  
 Sie wählen den PICK-Puffer *part2* aus und geben: <PUT>

Sie haben bis jetzt schon mehrere Dateien bearbeitet. Sie speichern alle veränderten Dateien auf die Festplatte ab mit einem Kommando:

- ▷ <ENTER> <SAVE>

Der MAXed gibt Ihnen links unten in der Steuerzeile nacheinander die Namen der Dateien bekannt, die er abspeichert. Jetzt packen Sie die "statische Erfassung" der zu besetzenden Rollen des *Rüpelspiels* an:

- ▷ Sie blenden sich die Datei *premiere* ein: <ENTER> *premiere* <EDIT>

Zunächst wollen Sie die zweiseitige Einrückung des Textanfangs entfernen. Dazu ersetzen Sie in jeder Zeile die beiden Leerzeichen am Zeilenanfang durch die leere Ersetzungszeichenkette, die Sie durch zwei direkt aufeinanderfolgende Anführungszeichen angeben.

Bei der Eingabe des Suchmusters stellen Sie den Zeilenanfang, an dem die gesuchte Zeichenkette beginnen soll, durch das Zeichen Dach ^ dar, das Sie dem restlichen Suchmuster, den zwei Leerzeichen, voranstellen.

- ▷ <ENTER> "^ " " " .<REPLACE>

Eine Aufführung vor dem fürstlichen Publikum wird für gewöhnlich durch "Trompetenklänge" eröffnet. Das muß deshalb auch in diesem *Rüpelstück* vorgesehen sein und wird sicherlich am Anfang der Zeile, mit der die Aufführung beginnt, als Regieanweisung stehen!

- ▷ Sie lassen den MAXed nach den Trompeten "fahnden":  
<ENTER> ^\[trompete> <PLUSFIND>

Der MAXed war bei der Suche erfolgreich, obwohl die gefundene Textstelle nicht mit einem Gegenschrägstrich beginnt!? Die Regieanweisung ist lediglich durch eine vorausgehende öffnende eckige Klammer kenntlich gemacht! Des Rätsels Lösung ist: Innerhalb eines PATTERNS haben Sonderzeichen, das sind alle Zeichen außer den alphanumerischen Zeichen und dem Unterstrich, wie z.B. das Dach oder die spitzen Klammern i.a. eine Sonderbedeutung. Die eingegebene eckige Klammer sollte jedoch bei der Suche keine Sonderbedeutung besitzen, sondern nur dasselbe Zeichen (eben eine öffnende eckige Klammer am Zeilenanfang) darstellen. Zu diesem Zweck mußte die Sonderbedeutung der Klammer entwertet werden. Das erreichen Sie dadurch, daß Sie das Sonderzeichen Gegenschrägstrich direkt voranstellen. Sie erinnern sich: Den Gegenschrägstrich haben Sie bereits verwendet um damit Buchstaben zu kennzeichnen, bei denen der MAXed die Groß- oder Kleinschreibung berücksichtigen soll.

*Der Prolog tritt auf:* Wenn er schon auftreten muß, dann doch wenigstens am Anfang des Stücks, um das Publikum in die Handlung einzuführen und die Akteure vorzustellen! Nach seinem Abgang überläßt er die Bühne den übrigen Schauspielern. Sie wollen sich also an eben dieser Textstelle des Abgangs nach den Namen der weiteren Personen des Stücks umsehen.

In der Regieanweisung für den Abgang des *Prologs* wird der Name des *Prologs* vorkommen und das Wörtchen oder die Silbe "ab". Sein Abgang wird vermutlich wie seine Vorrede entsprechend aufwendig gestaltet sein. Sie werden also auch in der Regieanweisung eine geraume, jedoch schwer einschätzbare Anzahl ziemlich beliebiger Worte bzw. Zeichen zwischen dem Auftreten des Namens *Prolog* und seinen unwideruflichen *ab*-Gang einkalkulieren müssen. Im ersten Moment erscheint es schwierig, dem MAXed diesen "Unsicherheitsfaktor" verständlich zu machen. Und doch bedarf es nur einer winzigen zusätzlichen Angabe: Sie müssen hierzu nur wissen, daß in einem Suchmuster das beliebig häufige Auftreten eines Zeichens in einer gesuchten Zeichenkette, d.h. keinmal oder mehrmals in direkter Folge, durch einen nachfolgenden Stern \* dargestellt wird, während in einem Suchmuster als "Platzhalter" für ein beliebiges Zeichen der Punkt anzugeben ist. Jetzt brauchen Sie diese beiden Sonderzeichen für PATTERN nur noch zu kombinieren, um eine unbekannte Anzahl beliebiger, zusammenhängender Zeichen darzustellen:

- ▷ Sie suchen nach dem nächsten Auftreten einer Zeichenkette, die mit *Prolog* beginnt und nach beliebig vielen Zeichen die Silbe *ab* enthält:  
<ENTER> \Prolog.\*ab> <PLUSFIND>

Die vermutete, unbekannte Größe entpuppt sich als die Aufzählung der Namen weiterer "Personen" des Stücks, die zusammen mit dem *Prolog* abgehen, nachdem dieser mit dem *der Länge und Breite nach in wohlgeordneten Zeilen* erstatteten Einführungsmonolog endet. Im Abschnitt darüber finden Sie sicherlich auch die Namen der übrigen Akteure. Sie suchen jedoch noch nach einem Weg, die Zählerarbeit auf den MAXed abzuwälzen.

Die Namen der Akteure sind in den Zeilen vor ihrem jeweiligen Sprech-einsatz in Großbuchstaben geschrieben. Dies ist ein hervorstechendes Unterscheidungsmerkmal gegenüber den anderen Zeilenanfängen! Diese Zeilen müßten lediglich geschickt durchnummeriert werden. Es bietet sich an, die abgekürzten Rollennamen in Klammern in einem Ersetzungskommando durch eine laufende Nummer zu ersetzen. Sie müßten nur folgendes beachten: Der MAXed darf nicht wahllos durchnummerieren, da ja die einzelnen Akteure mehrere Einsätze haben, sondern nur beim Auftauchen eines neuen Namens weiterzählen.

- ▷ Sie beginnen die Eingabe des Ersetzungskommandos mit der Konstruktion des PATTERNS, des Suchmusters für die besagten Zeilenanfänge:

```
<ENTER> "^ [A-Z] {3,10} \( ([A-Z, a-z, ö]+ \. ) . > "
```

Einige Bestandteile des Suchmusters kennen Sie bereits: Die Anführungszeichen geben Sie an, weil dazwischen ein Leerzeichen vorkommt. Die beiden Gegenschrägstriche entwerten jeweils das nachfolgende Sonderzeichen (die öffnende runde Klammer und den Punkt), weil es nur für sich selbst stehen soll. Die spitzen Klammern trennen bekanntlich den inneren Teil des Suchmusters vom umgebenden Kontext ab, d.h. bei der Suche wird die Schreibmarke innerhalb der gefundenen Textstelle auf die Stelle gesetzt, die zum Anfang dieses inneren Teils des Suchmusters paßt. Beim Ersetzen wird nur der innere Teil ersetzt, während der Kontext unverändert bleibt. Das Dach am Anfang steht wieder für den Zeilenanfang.

Betrachten Sie jetzt die bisher unbekanntenen Teile des Suchmusters. Es handelt sich um zwei Einheiten,  $[A-Z]\{3,10\}$  und  $[A-Z,a-z,ö]+$ . Die beiden Teile lassen sich wiederum in zwei Einzelteile aufspalten, in einen Ausdruck, der durch eckige Klammern begrenzt ist, und einen nachfolgenden sogenannten Operator.

Die Bedeutung der Ausdrücke mit den eckigen Klammern läßt sich leicht entschlüsseln: Sie stehen jeweils für ein einzelnes Zeichen in der gesuchten Textstelle und enthalten eine Auswahl von Zeichen, von denen eines das passende Zeichen sein muß.  $A-Z$  steht dabei für den Zeichenbereich vom  $A$  bis zum  $Z$ . Folglich muß das dazu passende Zeichen in der gesuchten Textstelle ein Großbuchstabe sein.  $A-Z,a-z,ö$  steht analog für einen Großbuchstaben ( $A-Z$ ) oder ( $,$ ) für einen Kleinbuchstaben ( $a-z$ ) oder ( $,$ ) für das Zeichen  $ö$ . Innerhalb einer solchen Auswahl-Menge wird grundsätzlich zwischen Groß- und Kleinschreibung unterschieden!

Operatoren beziehen sich immer auf den Ausdruck, der direkt vorangeht, hier also die Auswahl-Mengen in eckigen Klammern. Operatoren legen lediglich die Anzahl der zusammenhängenden Zeichen in der gesuchten Textstelle fest, die zum vorausgehenden Ausdruck passen. Sie kennen bereits einen Operator! Richtig: der Stern `*` bedeutet kein oder mehrere Zeichen, die zum vorausgehenden Ausdruck passen. Recht ähnlich ist der Operator Plus `+`: Er bedeutet ein oder mehrere passende Zeichen. Schon etwas genauer ist der Operator `{3,10}`: Er bedeutet zwischen drei und zehn passende Zeichen.

Nun ist klar, wie die Textstelle aussehen muß, die zum eingegebenen PATTERN paßt: Sie beginnt am Zeilenanfang (`^`) mit einem Wort aus Großbuchstaben, das zwischen drei und zehn Zeichen lang ist (`[A-Z]{3,10}`). Darauf folgt nach einem Leerzeichen eine öffnende runde Klammer und anschließend wiederum ein Wort aus Groß- oder Kleinbuchstaben, das mindestens ein Zeichen lang ist (`[A-Z,a-z,ö]+`). Dieses Wort wird noch durch einen Punkt abgeschlossen. Die Übereinstimmung mit den Zeilenanfängen, die die Namen der agierenden Akteure enthalten ist frappierend!

▷ Sie geben jetzt nach einem trennenden Leerzeichen die Ersetzungszeichenkette ein: `Spieler\{-2d,0,1,!}`

Es gibt auch in Ersetzungszeichenketten einige Sonderzeichen mit Sonderbedeutung. Sie sollten deshalb auch hier alle Sonderzeichen entwerten, die Klartext darsellen. Bei der eingegebenen Ersetzungszeichenkette verlangt lediglich ein Ausdruck eine nähere Betrachtung: `(-2d,1,!,)`. Es handelt sich hierbei um den besagten Zähler! Die Angaben in den geschweiften Klammern legen sein Ausgabe-Format und die Zählweise fest: Das Ausgabeformat `-2d` besagt, daß die auszugebende Zahl als Dezimalzahl (`d`) dargestellt wird, daß für ihre Ausgabe ein Feld von zwei Zeichen (`2`) freigehalten wird und daß sie in diesem Feld linksbündig (`-`) ausgegeben wird. Begonnen wird mit der Null (`0`), erhöht wird immer um eins (`1`). Dies geschieht jedoch nur dann (`!`), wenn während des Ersetzungskommandos eine passende Textstelle gefunden wird, die sich von den bisher gefundenen unterscheidet!

```

Zog seinen Dolch heraus und sich das Herz zerbrach.
Was noch zu sagen ist, das wird - glaubt mir fürwahr! -
Euch Mondschein, Wand und Löw und das verliebte Paar
Der Läng und Breite nach, solange sie hier verweilen,
Erzählen, wenn Ihr wollt, in wohlgeremten Zeiten.
[ Prolog, Thisbe, Löwe und Mondschein ab

Theseus: Mich nimmt wunder, ob der Löwe sprechen wird.

Demetrius: Kein Munder, gnädiger Herr: ein Löwe kanns wohl, da so viele
Esel es tun!

WAND (Wa. N.N.):
In dem besagten Stück es sich zutragen tut,
Daß ich, Thom Schnauz genannt, die Wand vorstelle gut.
Und eine solche Wand, wovon Ihr solltet halten,
Sie sei durch einen Schlitz recht durch und durch gespalten,
Wodurch der Pyramus und seine Thisbe fein
Oft flüsternten fürwahr ganz leis und insgeheim.
Der Mörtel und der Kalk und dieser Stein tut zeigen,
Daß ich bin diese Wand, ich wills Euch nicht verschweigen;

```

```

eter\{-2d,0,1,}

```

```

File premiere

```

```

Ln 208

```

```

Cl 3

```

- ▷ Sie müssen jetzt nur noch nach einem trennenden Leerzeichen das Zeichenketten-Attribut für den Bereich eingeben, nämlich Punkt für die ganze Datei und das Ersetzungskommando abschicken:
 

```

. <REPLACE>

```

In kürzester Zeit haben Sie alle Namen der agierenden Akteure durchnummeriert, beginnend beim *Prolog* mit der Nummer 0 (den zählen Sie nicht voll mit) und endend beim Mondschein mit der Nummer 5. Sie schauen sich an, wie die Ersetzungszeichenketten ausgewertet und in eine gefundene Textstelle eingebaut wurden:

- ▷ Sie holen die Kommandoattributierung aus dem Eingabepuffer ins Eingabefeld zurück mit <ENTER> <BACKSPACE>. Sie löschen mit <BACKSPACE> alle Zeichen bis einschließlich dem Gegenschrägstrich \. Sie geben wieder ein abschließendes Anführungszeichen ein und finden die Textstellen mit mehrmaligem <PLUSFIND>.

```

PYRAMUS (Spieler-2 N.N.):
O Kuß mich durch das Loch von dieser garst'gen Wand!

THISBE (Spieler-3 N.N.):
Mein Kuß trifft nur das Loch, nicht deiner Lippen Rand.

PYRAMUS (Spieler-2 N.N.):
Willst du bei Nickels Grab heut nacht mich treffen an?

THISBE (Spieler-3 N.N.):
Seis Lebend oder tot, ich komme, wenn ich kann.

WAND (Spieler-1 N.N.):
So hab ich Wand nunmehr mein Part gemacht gut,
Und nun sich also Wand hinwegbegeben tut.
[Wand, Pyramus und Thisbe ab

Theseus: Nun ist die Moral weg zwischen den beiden Nachbarn!

Demetrius: Das ist nicht mehr als billig, gnädiger Herr, wenn Wände
Ohren haben!

```

Mode: INSERT      File premiere      Ln 291      Cl 1

Der MAXed hat für Sie errechnet, daß fünf Schauspieler und ein monologisierender *Prolog* benötigt werden. Die Besetzung liegt nun in Ihren Händen! Wie Sie feststellen mußten, wird sich die Premieren-Aufführung doch erheblich vom anfänglichen Probenkonzept unterscheiden (siehe Beispielsitzung 1). Die äußeren Umstände, z.B. fehlendes Equipment wie Gipswand und Vollmond-Beleuchtung und nicht zuletzt die Konzessionen an den sensiblen, herrschaftlichen Geschmack in Form des beschwichtigenden *Prologs* zwingen Sie sogar zu erheblichen Umbesetzungen!

Wenigstens die beiden Handlungsträger *Pyramus* und *Thisbe* bleiben. *Klaus Zettels* Part als *Pyramus* und *Franz Flauts* Part als *Thisbe* stehen unverrückbar fest. Sie werden die Zeichenkette *N.N.* ("nicht nominiert") durch deren Namen ersetzen:

- ▷ Sie ersetzen bei den *Pyramus*-Stellen die Zeichenkette *N.N.* durch den Namen *Zettels*. Sie identifizieren diese Stellen durch die Rollennummer 2:

```
<ENTER> "2 < n\.n\.\>\):;" "\: Zettel\, Part \?2"
<REPLACE>
```

Neu ist bei diesem Ersetzungskommando lediglich das Zeichen Semikolon ; am Ende des Suchmusters, es steht für das Zeilenende, und die Tatsache, daß natürlich auch Ersetzungszeichenketten in Anführungszeichen gesetzt werden müssen, wenn sie Leerzeichen enthalten. Sie besetzen die anderen Rollen ebenfalls, werden dabei selbstverständlich nicht jedesmal wieder die ganze Eingabe neu schreiben, sondern die letzte Kommandoattributierung aus dem Eingabepuffer ins Eingabefeld zurückholen und nur die Rollennummer im Suchmuster und den Namen in der Ersetzungszeichenkette ändern! Sie fahren also fort mit Flaut's *Thisbe*:

- ▷ <ENTER> <BACKSPACE>

Sie ersetzen die Rollennummer 2 durch 3:

```
<LEFTARROW> <LEFTARROW> <LEFTARROW> <BACKSPACE> 3
<BACKTAB> <BACKTAB> <LEFTARROW>
```

Sie ersetzen *Zettel* durch *Flaut*:

```
<DELCHAR><DELCHAR><DELCHAR><DELCHAR><DELCHAR><DELCHAR>
```

Flaut

Sie springen an den Anfang des Eingabefeldes mit <UPARROW>

Sie ersetzen die Rollennummer 2 durch 3:

```
<RIGHTARROW> <DELCHAR> 3
```

Sie schicken das Ersetzungskommando ab: <REPLACE>

Auf die gleiche Weise besetzen Sie den *Schnock* mit der Rolle 4 (*Löwe*) *Schnauz* mit der Rolle 1 (*Wand*), *Schlucker* mit der Rolle 5 (*Mond*) und *Squenz* mit der Rolle 0 (*Prolog*).

Sie haben sich sicher schon gefragt, was es mit dem *Part* ? auf sich hat! Damit haben Sie die Vorbereitungen für die letzte noch ausstehende Aktion getroffen: Das Durchnummerieren der einzelnen Einsätze ("Parts") für jede Rolle, um einen Eindruck für die Belastung des jeweiligen Schauspielers zu bekommen! Nach Ihrem bis jetzt erworbenen Know-How für Ersetzungskommandos benötigen Sie zum Gelingen kaum weitere Information:

- ▷ Sie ersetzen das Fragezeichen durch einen Zähler der Rolleneinsätze:

```
<ENTER> "Part <\?[0-5]>\)\:;" {d,1,1,=} . <REPLACE>
```

Die Ersetzungszeichenkette besteht diesmal nur aus einem Zähler: Die Zahl wird als Dezimalzahl (*d*) in einem genau passend großen Feld ausgegeben, ihr Anfangswert beträgt *1* und der Zähler wird um *1* hochgezählt. Alle identischen Textstellen, die zum Suchmuster gefunden werden, erhalten sozusagen einen eigenen Zähler; denn der Zähler erhält bei einer gefundenen Textstelle einen um 1 höheren Wert als denjenigen, den die zuletzt identisch vorgefundene Textstelle zugewiesen bekommen hatte (=).

Nach diesen komplizierten Mustern haben Sie sich eine Ruhepause verdient:

- ▷ <EXIT>

## 2.6 Beispielsitzung 6 - Suchen, Sammeln und Gestalten

Sie sollen dem Regisseur *Squenz* einen Gefallen erweisen: Über die Verwendung einer überarbeiteten Fassung des Stückes und der daraus resultierenden Neubesetzung einiger Rollen (siehe Beispielsitzung 5) konnte *Squenz* leider noch nicht alle betroffenen Akteure informieren. Er trug Ihnen deshalb auf, dem Darsteller der *Wand*, den er eines schwachen Gedächtnisses verdächtigt, einen Auszug des Stückes mit seinen Rollentexten zusammenzustellen und ihm eiligst zuzuschicken, damit er seine Texte "bis zur Generalprobe in den Kopf kriegt". Sie konnten *Squenz* zusichern, daß Sie die betroffenen Textstellen mit Hilfe Ihres Textverarbeitungsprogramms schnell finden und zusammenstellen würden. Es würde auch keinen großen Aufwand für Sie bedeuten, die wichtigen Stichwörter durch Zeichenattribute so hervorzuheben, daß dem Darsteller die Punkte, auf die er sich konzentrieren muß, sofort auffielen.

- ▷ Sie rufen den MAXed mit der Angabe eines neuen Dateinamens auf:  
`m wandrolle`  
 Wie es der MAXed verlangt, bestätigen Sie das Anlegen der neuen Datei:  
`<EDIT>`  
 Sie setzen die Formatränder: `<ENTER> 5 60 <MARGIN>`

Da Sie ein paar einleitende Worte im Fließtext einzugeben gedenken, schalten Sie den Eingabe-Modus auf LEXICAL um:

- ▷ `<ENTER> 1 <INSERTMODE>`

Damit haben Sie gleichzeitig den Silbentrenner des MAXed aktiviert, der beim Zeilenumbruch am rechten Formatrand Wörter trennen wird.

- ▷ Sie beginnen mit der Eingabe des Brieftextes:

```
<RETURN> <RETURN>
Herrn Thom Schnauz
- Kesselflicker -
Wohnhaft in Athen, Waldrand
<RETURN> <RETURN>
Lieber Thom!
<RETURN> <RETURN>
```

Bei der Eingabe folgender Zeilen werden Sie das <RETURN> nur an den Stellen betätigen, wo es angegeben ist. Der MAXed wird selbständig den automatischen Zeilenumbruch auslösen, wenn Zeilen über den rechten Formatrand hinauswachsen, und sein Silbentrenner wird selber entscheiden, welche Wörter er trennt. Sie brauchen sich zunächst überhaupt nicht darum kümmern.

- ▷ Sie geben also nach zwei führenden Leerzeichen weiter ein:

```

    Ich habe das Vergnügen, Sie für die Rolle der "Wand"
in der spaßhaften Tragödie "Ein kurz langweil'ger Akt
vom jungen Pyramus und Thisbe seinem Lieb." einzuteilen!
<RETURN> <RETURN> <RETURN>
<TAB> Hinweis: Die Aufführung <RETURN>
findet am Abend des nächsten Neumondes im Palast des
Herzogs statt. <RETURN>
<TAB> (So Seine Gnaden das Stück zu hören belieben.) <RETURN>
<RETURN>
    Ich bitte Sie eindringlich, Ihre Rollentexte (siehe
Anhang) bis zur Generalprobe einzustudieren! <RETURN>
<RETURN>
gez.
```

Die eingegebenen Zeilen werden Sie im Blocksatz formatieren:

- ▷ Sie bewegen die Schreibmarke in den ersten Absatz und formatieren diesen: <ENTER> <FORMAT>

Beim Aufruf des nicht-attribuierten Formatierkommandos <FORMAT> wird automatisch der gesamte aktuelle Absatz, ein RANGE-Bereich, der von Leerzeilen begrenzt ist, formatiert. Der Abschnitt war aufgrund des automatischen Zeilenumbruchs bereits im Flattersatz geschrieben. Durch das Formatieren im Blocksatz wurde der Text der einzelnen Zeilen nur noch an den Formaträndern ausgerichtet.

- ▷ Bewegen Sie die Schreibmarke in die erste Zeile des zweiten Absatzes und formatieren Sie den gesamten Rest des Textes:  
<ENTER> -. <FORMAT>

Wie Sie feststellen, hat der MAXed jeden Absatz getrennt formatiert. Dabei hat er die Einrückungen des Textes übernommen. Innerhalb des zweiten Abschnitts hat er außerdem die letzte Zeile, die sich durch ihre Einrückung von der vorausgehenden Zeile abhebt, ebenfalls getrennt formatiert.

Ihr *gez.* am linken Formatrand in der letzten Zeile macht sich besser in der Mitte der Zeile. Sie positionieren die Schreibmarke also auf diese Zeile und zentrieren den Text dieser Zeile zwischen den beiden Formaträndern:

▷ <CENTER>

Einen Absender sollten Sie selbstverständlich am Anfang der Datei einfügen:

▷ Sie springen mit <RETURN> zunächst an den linken Formatrand und danach an den Dateianfang: <GOTO>

Sie wollen Ihr "Inkognito" wahren und werden deshalb ein Pseudonym für den Absender verwenden. Dieses kopieren Sie sich am schnellsten aus einer anderen Datei.

▷ Sie blenden sich die Datei mit dem Pseudonym ein:  
<ENTER> abs2.6 <EDIT>

Der Namenszug, den Sie sich abkopieren wollen, ist im aktuellen Fenster rechts oben eingeblendet. Sie werden sich diesen als Rechteck (RECTANGLE) kopieren:

▷ Durch <WORDRIGHT> setzen Sie die Schreibmarke auf das *W* von *Wilhelm*. Sie definieren den RECTANGLE-Bereich durch Bewegen der Schreibmarke: <ENTER>  
<DOWNARROW> <DOWNARROW> <DOWNARROW> <RIGHTARROW> <LEFTARROW>  
Sie geben noch einen IDENTIFIER für den PICK-Puffer ein:  
abs <PICK>  
Sie blenden wieder die Alternativ-Datei ein: <EDIT>  
Sie fügen das Rechteck vor dem bisherigen Text ein:  
<ENTER> abs + <PUT>

Durch die Angabe des Pluszeichens haben Sie das Kommando PUT veranlaßt, vor dem Schreiben des PICK-Pufferinhalts an der aktuellen Schreibmarkenposition so viele Leerzeilen einzufügen, wie das gespeicherte Rechteck an Zeilen umfaßt. Ansonsten wären keine neuen Zeilen eingefügt worden, sondern das Rechteck hätte beim Einfügen (im Einfüge-Modus LEXICAL!) den Text der folgenden Zeilen nach rechts verschoben.

Das Datum sollte am Briefanfang nicht fehlen:

▷ Sie bewegen die Schreibmarke ans Zeilenende mit  
<ENDLINE> <RIGHTARROW>.  
Sie lassen den MAXed über den Parameter für Textersetzung @d das aktuelle Datum einfügen: <ENTER> '@d' <PUT>.

Das Datum wollen Sie am rechten Formatrand ausrichten. Um das Kommando JUSTIFY geschickt einsetzen zu können treffen Sie ein paar Vorbereitungen:

- ▷ Sie speichern die aktuellen Formatgrenzen in der MAXed-Variable *m*:  
`<ENTER> m=@m <ENTER>`  
 Sie bewegen die Schreibmarke mit `<WORDLEFT>` an den Anfang des vorausgehenden Wortes und setzen hier vorübergehend den linken Formatrand: `<MARGIN>`  
 Jetzt richten Sie den Text der aktuellen Zeile zwischen den beiden Formaträndern aus: `<JUSTIFY>`  
 Danach stellen Sie wieder die alten Formatgrenzen ein:  
`<ENTER> $m <MARGIN>`

```

Wilhelm Schüttelbinne                               13.12.89
- Schauspieler und
  Dramatiker -
London, Globus-Theater

Herrn Thom Schnauz
- Kesselflicker -
Wohnhaft in Athen, Waldrand

Lieber Thom!

  Ich habe das Vergnügen, Sie für die Rolle der "Wand"
in der spaßhaften Tragödie "Ein kurz langweil'ger Akt
vom jungen Pyramus und Thisbe seinem Lieb." einzutei-
len!

  Hinweis: Die Aufführung findet am Abend des näch-
sten Neumondes im Palast des Herzogs statt.
(so Seine Gnaden das Stück zu hören belieben.)

```

Mode: LEXICAL File wandrolle Ln 1 Cl 13

Sie positionieren nun das Fenster so, daß die letzte Zeile der Datei direkt unter dem oberen Fensterrand zu stehen kommt:

▷ <ENTER> @#,1 <HOME>

Der Parameter für Textersetzung @ # wurde im Eingabefeld vor der Kommandoausführung durch die momentane Anzahl der Zeilen der aktuellen Datei ersetzt. Das Kommando HOME bewegte die angegebene Dateiposition an die linke obere Fensterecke.

▷ Sie beginnen einen neuen Absatz: <RETURN> <RETURN>  
 Sie tippen ein: Anhang:  
 Sie geben nochmals zweimal <RETURN> ein.

Im folgenden werden Sie die Textstellen mit den Einsätzen der *Wand* zusammenstellen. Zur Kennzeichnung, daß dies Ausschnitte aus dem Stück sind, werden Sie ihnen jeweils Leerzeilen, die drei Punkte einschließen, voranstellen:

▷ Sie geben drei Punkte ein: . . .  
 Sie geben <RETURN> ein. Sie kopieren die drei letzten Zeilen in den PICK-Puffer *punkt*:  
 <ENTER> <UPARROW> <UPARROW> punkt <PICK>  
 Sie geben nochmals <RETURN> ein.

Da auch Sie momentan überfragt sind, an welchen Stellen in welchen Dateien sich sämtliche Textstellen befinden, die den Darsteller der *Wand* betreffen, müssen Sie alle Beispieldateien danach durchsuchen! Sie haben in den beiden vorausgegangenen Beispielsitzungen Erfahrung im Umgang mit PATTERN und Suchkommandos gesammelt. Es wäre aber etwas mühsam, nacheinander jede einzelne Ihrer Beispieldateien durchzusuchen. Das können Sie dem MAXed überlassen, der das nach Aufruf eines einzigen Kommandos automatisch ausführt:

▷ Sie lassen den MAXed alle Dateien, deren Namen zur IDENTIFIER-Angabe *abs\** passen, nach dem Suchmuster <*wand*> durchsuchen:  
 <ENTER> "<wand>" abs\* <GOTO>

Beim Kommando GOTO müssen Sie ein Suchmuster grundsätzlich in Anführungszeichen setzen. Der MAXed meldet in der Steuerzeile:

\*\*\*\* Momentan laeuft: FILE SCAN

Dann blendet der MAXed das Ergebnis seiner Suche ein:

```
abs2.2:45:Schnauz: O Zettel! du bist verwandelt! Was seh ich an dir?
abs2.3:77:in der großen Stube eine Wand haben; denn Pyramus und Thisbe, sagt
abs2.3:78:Historie, redeten durch die Spalte einer Wand miteinander.
abs2.3:80:Schnock: Ihr bringt mein Leben keine Wand hinein! ... Was sagst du,
abs2.3:83:Zettel: Einer oder der andre muß Wand vorstellen; und laßt ihn ein
abs2.3:85:bißchen Mörtel an sich haben, um Wand zu bedeuten; und laßt ihn sei
abs2.5:96:Und ihrer aller Sinn, zugleich verwandelt,
abs2.5:239:[Pyramus, Thisbe, Wand, Mondschein und Löwe treten als stumme Pers
abs2.5:247:Die Wand, die garst'ge Wand, die ihre Lieb tät scheiden.
abs2.5:267:Euch Mondschein, Wand und Löw und das verliebte Paar
abs2.5:277:WAND (Wa. N.N.):
abs2.5:279:Daß ich, Thom Schnauz genannt, die Wand vorstelle gut.
abs2.5:280:Und eine solche Wand, wovon Ihr solltet halten,
abs2.5:285:Daß ich bin diese Wand, ich wills Euch nicht verschweigen;
abs2.5:291:Demetrius: Das ist das witzigste Wandloch, das je den Mund auftat!
abs2.5:293:Theseus: Pyramus geht auf die Wand los. Stille!
abs2.5:300:Und du, o Wand, o süß und Liebenswerte Wand,
abs2.5:302:Du Wand, o Wand, o süß und Liebenswerte Wand!
abs2.5:304:(Wand hält die Finger in die Höhe)
abs2.5:305:Hab Dank, du gute Wand! der Himmel lohn es dir!
abs2.5:307:O böse Wand, durch die ich nicht seh meine Zier,
```

WAHLE Zeile und gebe <GOTO> File <SCANLIST> Ln 1 Cl 1

In der Pseudo-Datei SCANLIST sind zeilenweise alle zum verwendeten Suchmuster passenden Textstellen aus den angegebenen Dateien aufgelistet. Am Zeilenanfang ist jeweils die Herkunft der betreffenden Textzeile angegeben: Dateiname und Zeilennummer. Nicht alle Textstellen betreffen die Rolleneinsätze der *Wand*, da das PATTERN <Wand> die Zeichenkette *wand* in beliebigem Wortkontext finden.

Die Schreibmarke steht auf der ersten Zeile. Zum Test werden Sie sich jetzt vom MAXed direkt auf diese Textstelle in der angegebenen Datei befördern lassen:

▷ <GOTO>

Sie müssen nun die Textstellen, die Sie für Ihren Zweck benötigen, aufsammeln, d.h. in PICK-Puffer kopieren. Für Sammelaktionen hält der MAXed den `_PICK`-Puffer, den Puffer mit besonderen Eigenschaften, bereit: Bei jedem PICK-Kommando, bei dem Sie explizit in den Standard-PICK-Puffer kopieren, d.h. keinen IDENTIFIER als Ziel angeben, wird der kopierte Textbereich gleichzeitig in den `_PICK`-Puffer geschrieben. Er überschreibt dabei jedoch nicht den bisherigen Inhalt, sondern wird am Ende des Puffers angehängt. Dieser Puffer wird nur dann überschrieben, wenn Sie seinen Inhalt mit PUT wieder in den Text geschrieben haben und einen Textbereich erneut mit PICK explizit in den Standard-PICK-Puffer kopieren.

Da der `_PICK`-Puffer vermutlich bereits Text enthält, löschen Sie ihn zunächst, damit der MAXed ihn wieder automatisch neu anlegt und von Anfang beschreibt:

▷ Sie blenden die Pseudo-Datei PICKBUFFERS ein: <ENTER> @ <PUT>  
 Sie bewegen die Schreibmarke auf den Namen des `_PICK`-Buffers (ein Unterstrich). Sie löschen diesen PICK-Puffer: <ERASE>  
 Sie blenden sich die zuvor erstellte Pseudo-Datei SCANLIST wieder ein:  
 <ENTER> " <GOTO>

Sie wählen sich aus dieser Liste die erste Textstelle heraus, die Ihnen brauchbar erscheint: Dies ist die Regieanweisung für den Auftritt von *Pyramus, Thisbe, Wand*, usw. in der Datei *abs2.5*:

▷ Sie bewegen die Schreibmarke mit <DOWNARROW> in diese Zeile und springen mit der Schreibmarke an die zugehörige Textstelle in der Datei *abs2.5*: <GOTO>  
 Sie kopieren diese Zeile in den Standard-PICK-Puffer und damit gleichzeitig in den `_PICK`-Puffer: <PICK>

Wie bereits erwähnt, werden im `_PICK`-Puffer die weiteren zu kopierenden Textstellen am Ende angehängt. Um die nächste Textstelle von der ersten optisch abzusetzen, werden Sie zuvor die Zeile mit den drei Punkten anfügen. Diese haben Sie bereits im `PICK`-Puffer *punkt* gespeichert:

- ▷ Sie hängen eine Kopie des Inhalts des `PICK`-Puffers *punkt* ans Ende des `_PICK`-Puffers an:  
`<ENTER> punkt >> _ <PICK>`

Mit Hilfe der schließenden spitzen Klammern innerhalb der Kommandoattribution des Kommandos `PICK` führen Sie lediglich eine Kopieraktion zwischen zwei `PICK`-Puffern aus. Wenn Sie nur eine dieser Klammern angegeben hätten, wäre das Ziel (der `_PICK`-Puffer) mit dem Inhalt der Quelle (der `PICK`-Puffer *punkte*) ganz überschrieben worden!

- ▷ Sie blenden wieder die Alternativ-Datei ein: `<EDIT>`

In der Pseudo-Datei `SCANFILES` wählen Sie als nächstes die Zeile aus, in der die erste Zeile des ersten Sprechensatzes der *Wand*, beginnend mit *WAND* (*Wa. n.n.*); zitiert wird:

- ▷ Sie bewegen die Schreibmarke auf die erste Zeile mit diesem Zitat und springen in die Datei *abs2.5*: `<GOTO>`  
 Sie kopieren den gesamten aktuellen Absatz: `<ENTER> $ <PICK>`

Das Sonderzeichen Dollar stellt das Zeichenkettenattribut für den aktuellen Absatz dar.

- ▷ Sie hängen wiederum die Zeile mit den Punkten an:  
`<ENTER> punkt >> _ <PICK>`  
 Sie blenden wieder die Pseudo-Datei `SCANFILES` ein: `<EDIT>`

Die nächste Textstelle ist die Regieanweisung für die *Wand*, die Finger in die Höhe zu halten:

- ▷ Sie bewegen die Schreibmarke auf die entsprechende Zeile und springen "ins Geschehen": `<GOTO>`

Um den Einsatz nicht zu verpassen, ist es äußerst wichtig, daß der Akteur seine Stichwörter kennt! Sie werden also den Text, den der vorausgehende Darsteller aufzusagen hat, dazukopieren:

- ▷ Sie kopieren den Anfang des aktuellen Absatzes bis zur aktuellen Zeile: <ENTER> \$- <PICK>  
Sie hängen wieder die Zeile mit den Punkten an mit  
<ENTER> punkt >> \_ <PICK>  
blenden die Pseudo-Datei SCANFILES ein: <EDIT>

Die *Wand* beendet ihren Auftritt mit einem letzten Sprechensatz, dessen Anfangszeile (*WAND (Wa. n.n.):*) sich ebenfalls in der Datei *abs2.5* befindet:

- ▷ Sie bewegen die Schreibmarke mit <PLUSLINE> und <DOWNARROW> in die Zeile, in der dieser Sprechensatz zitiert ist und springen in die "Szene": <GOTO>  
Sie kopieren den Text der *Wand* inklusive nachstehender Regieanweisung für den Abgang und dem vorausgehenden Versprechen der *Thisbe*:  
<ENTER> "tot"-\$ <PICK>

Die Zeichenkette in Anführungszeichen innerhalb der Bereichsdefinition interpretiert der MAXed als PATTERN, das eine vorausgehende, passende Textstelle als Bereichsgrenze festlegt. Sie haben bereits alle benötigten Texte aufgesammelt und werden den Inhalt des \_-PICK-Puffers in der Datei *wandrolle* einfügen:

- ▷ Sie blenden die Datei *wandrolle* ein: <ENTER> wandrolle <EDIT>  
Sie kopieren den Inhalt des \_-PICK-Puffers an die aktuelle Schreibmarkenposition: <ENTER> \_ <PUT>

Der Darsteller der *Wand* soll sich in den Textausschnitten schnell zurechtfinden. Sie werden sie also übersichtlich aufbereiten und mit Zeichenattributen versehen, die Sprechensätze, Regieanweisungen und Stichwörter hervorheben.

Zunächst werden Sie mit ALIGN den linken Textrand begradigen. Dazu darf der linke Textrand nicht über den linken Formatrand hinausragen.

Sie setzen also die Formatränder außer Kraft, indem Sie sie auf Spalte 1 setzen:

▷ <ENTER> 1 1 <MARGIN>

Jetzt richten Sie den gesamten Text von der aktuellen Zeile bis zum Dateiende an der aktuellen Spalte 5 aus:

▷ <ENTER> -. <ALIGN>  
Den linken Formatrand setzen Sie auch wieder an der aktuellen Position: <MARGIN>

Die aktuelle Zeile ist etwas zu lang. Sie werden den Text in zwei Zeilen unterbringen:

▷ Sie springen auf das *P* von *Personen* mit  
<ENDLINE> <WORDLEFT> <WORDLEFT> <WORDLEFT>  
und schieben den Zeilenrest so in eine neue Zeile, daß der Text an Spalte 6 beginnt: <ENTER> <RETURN> <RIGHTARROW> <SPACE>

Die Angaben in Klammern zu Beginn der Sprechensätze werden Sie löschen.

▷ Sie bewegen die Schreibmarke direkt hinter das nächste Wort *WAND* mit  
<ENTER> 5 <RETURN>  
<ENTER> 4 <RIGHTARROW>

Sie löschen den Zeilenrest mit <ENTER> <ERASE>  
Sie geben einen Doppelpunkt und ein Leerzeichen ein und hängen die nachfolgende Zeile am Ende der aktuellen Zeile an:  
<ENTER> <ERASE>

Sie stellen fest, daß der Text der aktuellen Zeile nicht mit dem passenden Reim endet! Die letzten beiden Wörter gehören in die folgende Zeile.

▷ Sie bewegen die Schreibmarke auf das *D* des *Daß* und schieben den Zeilenrest in die nächste Zeile: <WRAP>

Nun bewegen Sie die Schreibmarke direkt hinter das Wort *PYRAMUS* und geben ein: <ENTER> <RETURN> <ERASE>

Es fehlt nur noch der Doppelpunkt und ein Leerzeichen. Verfahren Sie mit der nachfolgenden Stelle nach *WAND* genauso. Dann bewegen Sie die Schreibmarke wieder nach oben auf die erste Zeile der Regieanweisung für den Auftritt der *Personen*: <ENTER> 31 <GOTO>

Regieanweisungen werden Sie vom übrigen Text abheben, indem Sie ihnen die Zeichenattribute "Unterstrichen" und "Inversdarstellung" hinzufügen.

- ▷ Sie blenden zur Veränderung der aktuellen Einstellung der Zeichenattribute in der Steuerzeile das ATTRIBUTE-Feld ein:  
<ENTER> @ <SETATTRIB>

Die Schreibmarke ist in der Lücke zwischen den zwei Blöcken mit den aktuellen Einstellungen der einzelnen Zeichenattribute positioniert. Links daneben ist eine Meldung eingeblendet: *Waehle* <SETATTRIB>.

- ▷ Sie bewegen die Schreibmarke nach links: <LEFTARROW>

Die Meldung ändert sich zu: *Gewaehlt: reversed*

Mit einer *l* ist an dieser Position das Zeichenattribut "Inversdarstellung" bereits auf "gesetzt" eingestellt. Sie belassen diese Einstellung und bewegen die Schreibmarke weiter nach links:

- ▷ <LEFTARROW> <LEFTARROW>

Nach der Meldung zu schließen, haben Sie gerade die Einstellungsposition für das Zeichenattribut *underscore* (Unterstrichen) erreicht. Durch eine *0* ist es auf "nicht-gesetzt" eingestellt.

- ▷ Sie ändern diese Einstellung und tippen ein: 1  
Sie speichern die Einstellungen des ATTRIBUTE-Feldes ab und aktivieren gleichzeitig diese aktuelle Einstellung der Zeichenattribute:  
<SETATTRIB>

Wenn Sie jetzt neuen Text eintippen sollten, erscheinen die Zeichen mit den Zeichenattributen, die in der aktuellen Einstellung auf "gesetzt" eingestellt sind. Sie wollen jedoch dem Text der aktuellen und der folgenden Zeile nachträglich die "gesetzten" Zeichenattribute *reversed* und *underscore* hinzufügen:

- ▷ <ENTER> <DOWNARROW> <ATTRIBUTE>

```

gez.

Anhang:

...

Pyramus, Thisbe, Wand, Mondschein und Löwe treten als stumme
Personen auf.

...

WAND: In dem besagten Stück es sich zutragen tut,
Daß ich, Thom Schnauz genannt, die Wand vorstelle gut.
Und eine solche Wand, wovon Ihr solltet halten,
Sie sei durch einen Schlitz recht durch und durch gespalten,
Wodurch der Pyramus und seine Thisbe fein
Oft flüsternten fürwahr ganz leis und insgeheim.
Der Mörtel und der Kalk und dieser Stein tut zeigen,
Daß ich bin diese Wand, ich wills Euch nicht verschweigen;
Und dies die Spalte ist, zur Linken und zur Rechten,
Wodurch die Buhler zwei sich täten wohl besprechen.

```

Mode: LEXICAL Att: 0101 0000 File wandrolle Ln 31 Cl 11

Den Beginn des nächsten Sprechensatzes, die Zeichenkette *WAND:*, wollen Sie ebenfalls hervorheben.

- ▷ Sie bewegen die Schreibmarke auf die *WAND:* <ENTER> 5 <RETURN>  
 Sie definieren einen einzeiligen STREAM, der die Zeichenkette *WAND:* einschließt: <ENTER> <WORDRIGHT> <RIGHTARROW>  
 Sie fügen die "gesetzten" Zeichenattribute hinzu: <ATTRIBUTE>

Aus Gründen der Übersichtlichkeit halten Sie es für günstiger, hier nur das Zeichenattribut *reversed* zu belassen.

- ▷ Sie blenden wieder das ATTRIBUTE-Feld ein: <ENTER> @ <SETATTRIB>  
 Sie bewegen die Schreibmarke nach links:  
 <LEFTARROW> <LEFTARROW> <LEFTARROW>  
 Sie ändern die Einstellung von *underscore* auf "nicht-gesetzt" (0):  
 <DOWNARROW> <DOWNARROW>

Die Einstellung an der aktuellen Schreibmarkenposition können Sie nämlich auch mit <DOWNARROW> und (in "Gegenrichtung") mit <UPARROW> ändern.

- ▷ Sie aktivieren die veränderte Einstellung: <SETATTRIB>  
Sie definieren wieder den STREAM-Bereich und fügen hinzu bzw. entfernen in ihm Zeichenattribute gemäß der aktuellen Einstellung:  
<ENTER> <WORDRIGHT> <RIGHTARROW> <ATTRIBUTE>

Das Zeichenattribut *underscore* wurde wieder entfernt, da es in der aktuellen Einstellung auf "nicht-gesetzt" eingestellt ist. Dem nächsten Sprechereinsatz wollen Sie dieselben Zeichenattribute verpassen.

- ▷ Sie definieren einen STREAM-Bereich als Suchmuster, indem Sie die letzte Kommandoattributierung aus dem Eingabepuffer ins Eingabefeld zurückholen und das Suchkommando starten:  
<ENTER> <BACKSPACE> <PLUSFIND>

Der MAXed meldet unerwarteterweise: STREAM nicht gefunden  
Der Grund ist: Da die aktuelle Einstellung der Zeichenattribute aktiviert ist, untersucht der MAXed beim Textvergleich, ob auch alle Zeichenattribute außer jenen, die auf "nicht-berücksichtigen" (\*) eingestellt sind, gleich vorhanden sind.

- ▷ Sie deaktivieren wieder die aktuelle Einstellung: <SETATTRIB>  
Sie probieren nochmals das Suchkommando: <PLUSFIND>  
Sie aktivieren wieder die aktuelle Einstellung: <SETATTRIB>  
Sie fügen das Zeichenattribut *reversed* an der aktuellen Textstelle hinzu: <ENTER> <WORDRIGHT> <RIGHTARROW> <ATTRIBUTE>

Sie springen an die nächste Regieanweisung mit <RETURN> <RETURN>  
Sie setzen das Zeichenattribut *underscore* in der aktuellen Einstellung und aktivieren diese gleichzeitig:

<ENTER> *underscore* <SETATTRIB>

Sie fügen die Zeichenattribute in der aktuellen Zeile hinzu:  
<ATTRIBUTE>

Der Darsteller der *Wand* muß sich für diesen Sprechensatz unbedingt seine Stichworte merken: den Text der vorausgehenden Zeile, den *Thisbe* spricht! Damit die *Wand* diese Stelle nicht "verschläft", werden Sie sie invers und blinkend darstellen:

- ▷ Sie bewegen die Schreibmarke nach oben: <ENTER> 4 <UPARROW>  
In der aktuellen Einstellung setzen Sie gleichzeitig *blinking* auf 1, *underscore* auf 0, aktivieren diese Einstellung und fügen die gesetzten Zeichenattribute auch noch der aktuellen Zeile hinzu:  
<ENTER> <DOWNARROW> <UPARROW> b,u <ATTRIBUTE>

Die Namen der Zeichenattribute haben Sie durch ihre Anfangsbuchstaben ersetzt. Vor dem Komma stehen die Namen derjenigen Zeichenattribute, die in der aktuellen Einstellung auf 1 gesetzt werden sollen, nach dem Komma diejenigen, die auf 0 gesetzt werden sollen.

Wenn Sie diesen Text ausdrucken sollten, wird das Zeichenattribut *blinking* vom Druckkommando *maxlpr* für Ihren Drucker in Bereitschaft umgewandelt. Damit der Text der aktuellen Zeile noch aufs papier paßt, werden Sie ihn in zwei Zeilen unterbringen:

- ▷ Sie bewegen die Schreibmarke mit WORDRIGHT auf das *i* des *ich* und schieben den Zeilenrest so in eine neue Zeile, daß der Text am linken Formatrand beginnt: <ENTER> <SPACE>
- ▷ Sie suchen die vorausgehende Regieanweisung:  
<ENTER> finger <MINUSFIND>

Auch hier darf die *Wand* das Spreizen der Finger nicht vergessen, wenn *Pyramus* nach dem Spalt in der Wand Ausschau hält!

- ▷ Sie bewegen den Dateiausschnitt nach oben: <MINUSLINE>  
Sie bewegen die Schreibmarke in der vorausgehenden Zeile auf das *d* des *daß* und schieben den Zeilenrest in eine neue Zeile:  
<ENTER> <SPACE>  
Sie fügen in den beiden folgenden Zeilen die Zeichenattribute gemäß der aktuellen Einstellung hinzu: <ENTER> 2 <ATTRIBUTE>

Schließlich versehen Sie die Regieanweisung mit den Zeichenattributen *underscore* und *reversed*:

- ▷ Sie springen mit der Schreibmarke auf die Regieanweisung:

<PLUSFIND>

Sie fügen die genannten Zeichenattribute hinzu:

<ENTER> 1 u r, <ATTRIBUTE>

Da nach dem Komma nichts angegeben ist, stellt der MAXed automatisch alle nicht genannten Zeichenattribute auf 0 ein.

Für den Ausdruck soll der Anhang auf einer neuen Seite beginnen. Deshalb fügen Sie in der Datei das entsprechende Steuerzeichen für den Drucker ein:

- ▷ Sie springen auf die erste Zeile des Anhangs:

<ENTER> anhang <MINUSFIND>

Sie springen auf Spalte 1: <STARTLINE>

Sie fügen das Steuerzeichen ein: <CODE>L

Damit haben Sie dem Darsteller der *Wand* alle erforderlichen Informationen zusammengestellt.

- ▷ Sie beenden die Sitzung: <ENTER> <EXIT>

Um den Brief auszudrucken, verwenden Sie das MAXed-Dienstprogramm *maxlpr*, da dieses die Zeichenattribute für den Drucker entsprechend umwandelt. Für den Laserdrucker 9022 würde das Druckkommando so lauten:

```
maxlpr -22 wandrolle
```

1

2

3

4

### 3 Wie arbeitet der MAXed? - Grundlagen

— In den beiden Kapiteln 3 und 4 finden Sie eine Zusammenstellung der einzelnen Bestandteile des MAXed und deren Arbeitsweise.

Dieses Kapitel soll Ihnen das "Strickmuster" des "MAXimalen editors" verdeutlichen und Sie in die Handhabung des großen Funktionsumfangs einführen.

—

—

—

—

### 3.1 Die Dateihaltung des MAXed

Der MAXed arbeitet grundsätzlich nur auf Kopien von Dateien. Wenn Sie eine Datei laden und Textveränderungen vornehmen, hält der MAXed diese Änderungen für die Dauer der MAXed-Sitzung in einer temporären Datei fest. Im Fenster des MAXed sehen Sie einen Ausschnitt der temporär existierenden Dateiversion. Das "Original", die Datei auf der Festplatte, bleibt unangetastet, bis Sie die temporäre Dateiversion auf die Festplatte abspeichern. Dazu sichern Sie eine geladene Datei explizit oder lassen sie beim Beenden des MAXed automatisch auf die Festplatte schreiben (siehe Abschnitt 3.2 Starten und Beenden des MAXed, Abschnitt 3.3.1 Einblenden und Sichern einer Datei, Abschnitt 3.4.1 Wechseln zwischen Dateien).

Beim Anlegen einer neuen Datei erstellt der MAXed zunächst eine neue, leere Datei auf der Festplatte. Dann blendet er die Kopie dieser leeren Datei ein. Ihre Texteingaben setzen auch in diesem Fall auf einer temporären Dateiversion auf (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

Sobald Sie in einer geladenen Datei die erste Änderung vornehmen, stellt der MAXed automatisch die Zugriffsrechte der Original-Datei auf der Festplatte neu ein: Diese Datei, deren Inhalt nach wie vor unverändert ist, darf jetzt nur noch gelesen werden. Dadurch verhindert der MAXed, daß die Original-Datei versehentlich durch einen weiteren schreibenden Zugriff verändert wird, während Sie auf der zugehörigen temporären Version arbeiten (siehe Bild 3-1).

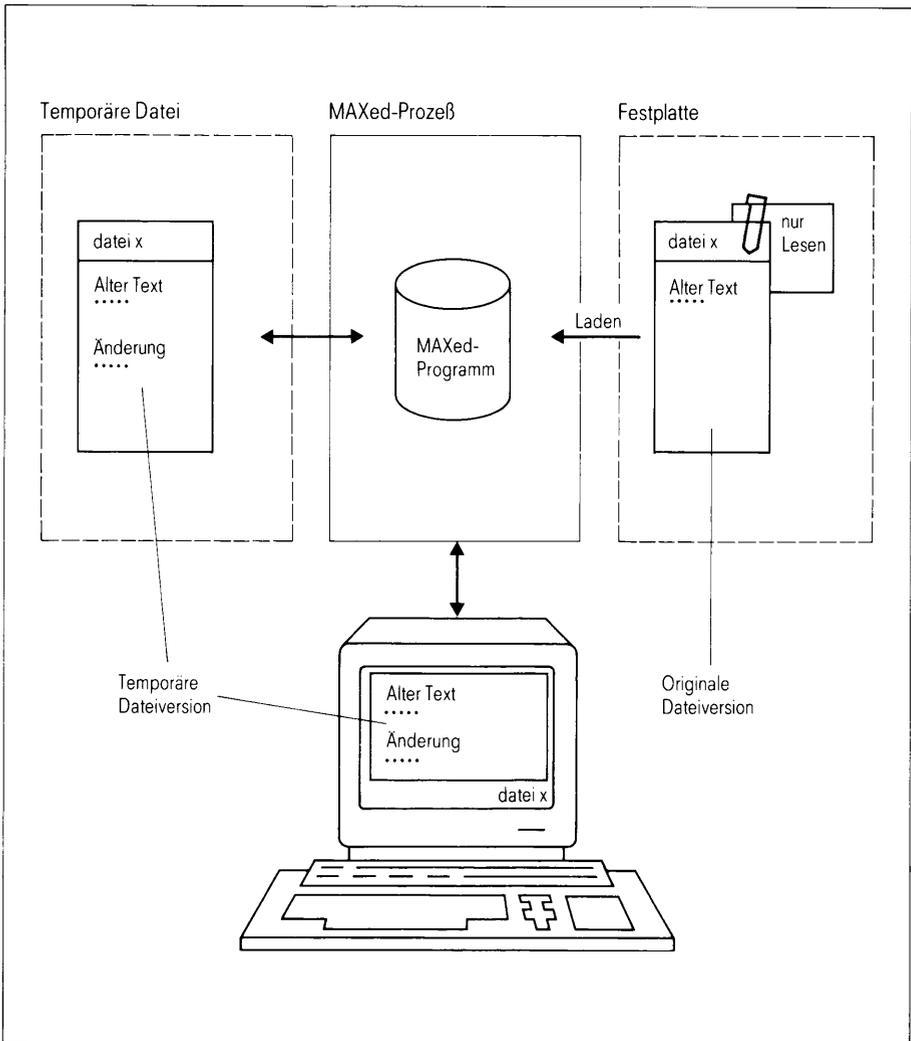


Bild 3-1 temporäre/originale Dateiversionen nach einer Änderung

Beim ersten Sichern einer geladenen Datei erstellt der MAXed zuerst eine Sicherungskopie der Original-Datei auf der Festplatte: Diese seit Sitzungsbeginn nicht veränderte Dateiversion wird in einer zweiten Datei gleichen Namens mit der angehängten Endung *.bak* abgespeichert. Dann wird die geladene Datei abgespeichert: Die Original-Datei wird aktualisiert und erhält wieder ihre ursprünglichen Zugriffsrechte zurück, solange sie der eingblendeten temporären Dateiversion entspricht (siehe Bild 3-2).

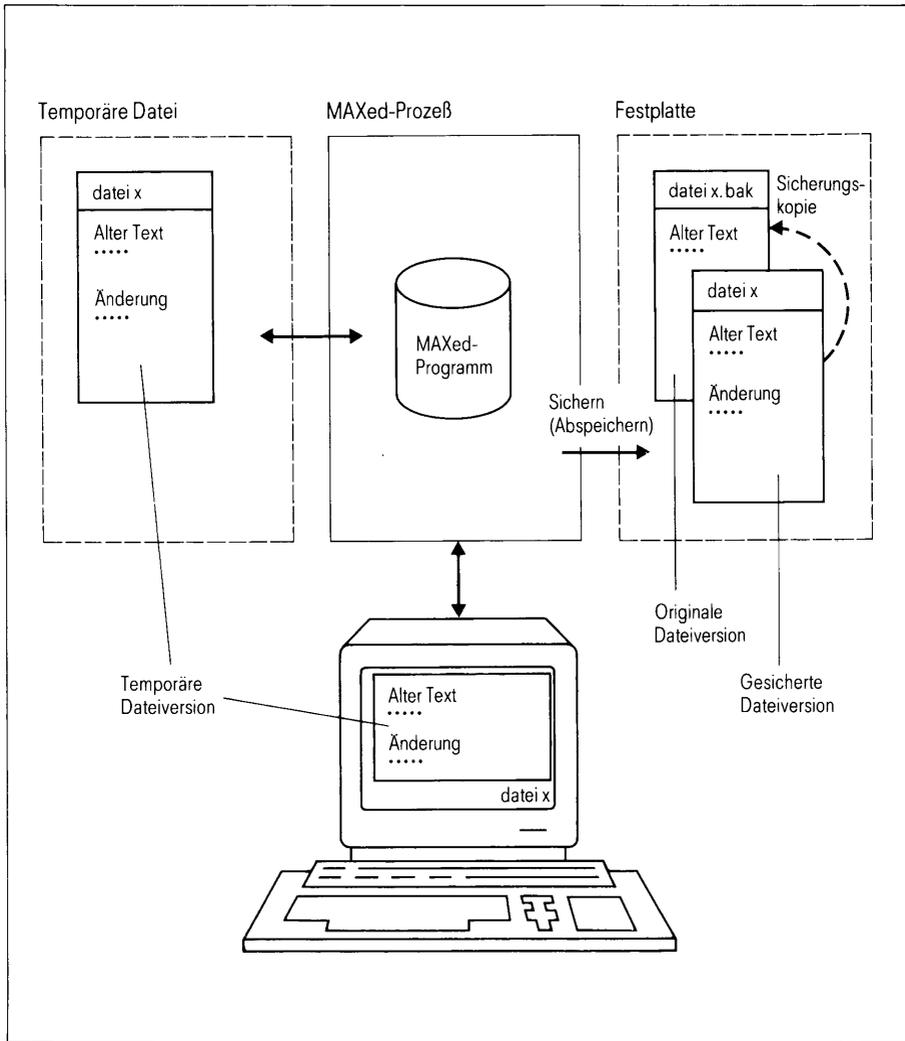


Bild 3-2 temporäre/gesicherte Dateiversionen nach dem Sichern

Wenn Sie weitere Änderungen an der geladenen Datei vornehmen, führt dies der MAXed wiederum nur in der temporären Dateiversion aus. Bei neuerlichem Abspeichern auf die Festplatte bringt der MAXed lediglich die Datei mit dem Original-Namen auf den aktuellen Stand. Die Sicherungskopie der .bak-Datei bleibt unverändert erhalten.

Falls Sie mit den Änderungen in der temporären Dateiversion nicht zufrieden sind, können Sie während der MAXed-Sitzung jederzeit wieder auf einer der gespeicherten Dateiversionen neu aufsetzen. Dazu bringen Sie die eingblendete temporäre Datei auf den Stand bei Sitzungsbeginn bzw. beim letzten Sichern (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei). Im Bild 3-2 entspricht dies dem Inhalt der Datei *dateix.bak* bzw. der Datei *dateix* auf der Festplatte.

Beim Beenden des MAXed wird die temporäre Dateiversion standardmäßig auf der Festplatte gespeichert und die Version bei Sitzungsbeginn in eine *.bak*-Datei geschrieben, falls dies nicht schon zuvor durch ein Sichern erfolgt ist (siehe Abschnitt 3.2 Starten und Beenden des MAXed).

#### Anmerkung

Falls sich Ihre Datensichtstation während einer MAXed-Sitzung unerwartet "aufhängen" sollte und Sie den MAXed nicht mehr ordnungsgemäß beenden können, ist Ihre temporäre Dateiversion noch zu retten: Lassen Sie den MAXed nach dem Neustart die letzte Sitzung rekonstruieren (siehe Abschnitt 5.2 MAXed-Sitzung rekonstruieren). Der MAXed wird hierbei auf der Dateiversion mit dem Original-Namen aufsetzen, nachdem er von dieser Datei eine zusätzliche Sicherungskopie erstellt hat: eine Datei gleichen Namens mit der angehängten Endung *.sav*.

## 3.2 Starten und Beenden des MAXed

### Den MAXed starten

Sie starten den MAXed aus der SINIX-Shell durch folgendes Kommando (siehe Abschnitt 5.1: maxed – Den MAXed aufrufen):

```
m
```

Beim Aufruf dieses Kommandos können Sie auch verschiedene Schalter eingeben und eine Dateiangabe machen:

```
m [schalter] datei
```

Der MAXed startet mit dem Einblenden der Versionsnummer, präsentiert den MAXed-Bildschirm und blendet eine Datei ein (siehe unten und Abschnitt 3.5 Der Bildschirm des MAXed):

- Wenn Sie beim Aufruf einen Dateinamen angeben, versucht der MAXed diese Datei zu laden und einzublenden. Existiert die angegebene Datei nicht, meldet sich der MAXed mit der Aufforderung, diese Datei neu anzulegen (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).
- Wenn Sie keine Dateiangabe machen, versucht der MAXed diejenige Datei zu laden, die beim Beenden der letzten MAXed-Sitzung unter Ihrer Benutzerkennung und an Ihrer Datensichtstation eingebildet war. Dazu liest er die sogenannte Status-Datei ein, die einige Informationen über den Sitzungszustand am Ende der letzten MAXed-Sitzung enthält. Falls die Status-Datei existiert, erhalten Sie den zuletzt eingebildeten Dateiausschnitt wieder. Außerdem restauriert der MAXed den vorigen Zustand verschiedener Einstellungen und Puffer (siehe Abschnitt 5.1: maxed – Den MAXed aufrufen).

- Wenn der MAXed beim Start keine Datei zum Laden findet, blendet er ersatzweise eine eigene Hilfsdatei ein. Im deutschen Konversationsmodus erhalten Sie folgende Meldung (siehe Abschnitt 4.1.10 Die Sprach-Modi):

```
Sie sehen diesen Text,  
  
* weil beim MAXed-Aufruf keine zu editierende Datei angegeben war  
  oder  
* weil MAXed die angegebene Datei nicht finden konnte  
  oder  
* weil die angegebene Datei keine Textdatei ist.  
  
Fuer weitere Informationen geben Sie bitte  
  <HELP>  
oder  
  <ENTER> help <ENTER>  
ein.  
  
File <UNDEFINED> Ln 1 Cl 1
```

Bild 3-3 Startmeldung bei nicht angegebener Datei

## Den MAXed beenden

Sie beenden den MAXed mit dem Kommando EXIT oder QUIT. Dabei legen Sie fest, ob bzw. in welcher Weise die temporäre Version einer geladenen Datei auf der Festplatte abgespeichert wird (siehe Kapitel 6: EXIT, QUIT).

Der MAXed speichert geladene Dateien ab und erstellt Sicherungskopien in *.bak*-Dateien (siehe Abschnitt 3.1 Die Dateihaltung des MAXed) mit <EXIT>

Der MAXed legt beim Abspeichern keine Sicherungskopien an mit <ENTER> <EXIT>

Der MAXed beendet sich ohne Abspeichern geladener Dateien mit <QUIT>

In letzterem Fall gehen jene Änderungen an geladenen Dateien verloren, die Sie nicht während der MAXed-Sitzung explizit gesichert haben (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

### Anmerkung

Ein weiterer Tip für den Notfall: Wenn Ihre Datensichtstation während einer MAXed-Sitzung "nichts" mehr von Ihrer Tastatur annehmen will, jedoch der Rechner noch läuft, schicken Sie dem MAXed-Prozeß von einer anderen Datensichtstation aus das Signal *SIGTERM*. Der MAXed wird dieses abfangen und die Sitzung regulär beenden, indem er selbständig das Kommando <EXIT> ausführt.

### 3.3 Arbeiten mit der aktuellen Datei

Mit dem MAXed haben Sie die Möglichkeit, mehrere Dateien zu laden und zwischen diesen hin- und herzuschalten (siehe Abschnitt 3.4 Arbeiten mit mehreren Dateien). Sie können jedoch immer nur eine Datei gleichzeitig mit der Schreibmarke bearbeiten, nämlich die gerade aktuelle Datei. Sie ist im aktuellen Fenster des MAXed-Bildschirms eingeblendet, und die Schreibmarke befindet sich innerhalb des sichtbaren Dateiausschnitts (siehe Abschnitt 3.5 Der Bildschirm des MAXed).

#### 3.3.1 Einblenden und Sichern einer Datei

##### Datei im aktuellen Fenster einblenden

Sie blenden während einer MAXed-Sitzung mit dem Kommando EDIT eine Datei ins aktuelle Fenster (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIKATOR, Kapitel 6: EDIT):

```
<ENTER> dateiname <EDIT>
```

Alternativ können Sie hier auch das folgende Kommando geben:

```
<ENTER> dateiname <RETURN>
```

Der MAXed lädt vor dem Einblenden die angegebene Datei von der Festplatte, wenn sie existiert und in der aktuellen MAXed-Sitzung nicht bereits geladen ist (siehe Abschnitt 3.1 Die Dateihaltung des MAXed).

##### Datei neu anlegen

Um eine neue Datei anzulegen, gehen Sie zunächst genauso wie beim Laden einer bereits existierenden Datei vor (siehe oben und Abschnitt 3.2 Starten und Beenden des MAXed). Der MAXed überprüft die Existenz der angegebenen Datei auf der Festplatte. Wenn er sie nicht findet, gibt er folgende Meldung in der Steuerzeile aus (siehe Abschnitt 3.5 Der Bildschirm des MAXed):

```
<EDIT> zum Anlegen von: dateiname
```

Der MAXed fordert Sie hiermit auf, das Anlegen einer neuen Datei durch nochmaliges Drücken der Kommandotaste <EDIT> zu bestätigen. Auf diese Weise wird Ihnen nochmals Gelegenheit gegeben, den eingetippten Dateinamen zu überprüfen. Nach Ihrer Bestätigung legt der MAXed eine neue, leere Datei unter dem Namen *dateiname* auf der Festplatte an und blendet sie im aktuellen Fenster ein (siehe Abschnitt 3.1 Die Dateihaltung des MAXed).

Sie brechen das Anlegen einer neuen Datei ab, indem Sie statt <EDIT> die Taste <ENTER> drücken.

### Datei sichern

Mit dem Kommando SAVE speichern Sie den Inhalt der aktuellen Datei auf die Festplatte ab (siehe Kapitel 6: SAVE). Sie aktualisieren dadurch die zugehörige gespeicherte Dateiversion mit den Änderungen, die Sie an der eingeblendeten temporären Dateiversion vorgenommen haben. Sie können den Inhalt der aktuellen Datei aber auch in eine andere Datei abspeichern.

Sie sichern die Änderungen an der aktuellen Datei auf die Festplatte mit <SAVE>

Sie sichern den Inhalt der aktuellen Datei in einer anderen Datei auf der Festplatte ab mit

<ENTER> *dateiname* <SAVE>

Wenn die Datei *dateiname* noch nicht existiert, legt der MAXed eine neue Datei mit diesem Namen an und speichert den Inhalt der aktuellen Datei unter dem Namen *dateiname* auf die Festplatte ab. Falls die Datei *dateiname* bereits existiert, gibt er folgende Meldung in der Steuerzeile aus (siehe Abschnitt 3.5 Der Bildschirm des MAXed):

<SAVE> ueberschreibt: *dateiname*

Der MAXed fordert Sie hiermit auf, das Überschreiben dieser Datei durch nochmaliges Drücken der Kommandotaste <SAVE> zu bestätigen.

Sie brechen das Überschreiben der Datei ab, indem Sie statt <SAVE> die Taste <ENTER> drücken.

### Datei auf einen früheren Versionsstand zurücksetzen

Wenn Sie mit den Änderungen an der aktuellen Datei nicht zufrieden sind oder die aktuelle Datei versehentlich verändert haben, können Sie die aktuelle Datei auf einen früheren Stand der Bearbeitung zurücksetzen (siehe Abschnitt 3.1 Die Dateihaltung des MAXed).

Sie setzen die aktuelle Datei wieder auf den Stand beim letzten Sichern mit `<ENTER> ^ <EDIT>`

Sie setzen die aktuelle Datei wieder auf den Stand bei Sitzungsbeginn mit `<ENTER> - <EDIT>`

Ziehen Sie jedoch zuvor folgendes in Betracht: Die seither vorgenommenen Änderungen gehen unrettbar verloren, wenn Sie den Inhalt der aktuellen Datei nicht zuvor unter einem anderen Dateinamen gesichert haben (siehe oben).

### Anmerkung

Wenn Sie bei der Bearbeitung Ihrer Datei auf einem früheren Stand der aktuellen MAXed-Sitzung wieder aufsetzen wollen und nicht gleich alle Änderungen seit dem letzten Abspeichern bzw. seit Sitzungsbeginn verwerfen wollen, verwenden Sie das Kommando `UNDO`. Mit diesem Kommando haben Sie die Möglichkeit, die bisherigen Änderungen in der MAXed-Sitzung schrittweise oder in einem definierten Umfang zurückzuspielen, bis Sie einen gewünschten Stand ihrer bisherigen Arbeit erreichen (siehe Kapitel 6: `UNDO`, Abschnitt 4.1.9 Der Zurücksetzen-Modus).

### 3.3.2 Bewegen der Schreibmarke

Sie können die Schreibmarke innerhalb des eingeblendeten Dateiausschnitts im aktuellen Fenster auf jede Stelle bewegen (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Dazu stehen Ihnen eine ganze Reihe von Kommandos zur Verfügung, die Bewegungen oder Sprünge verschiedener Länge in verschiedene Richtungen erlauben. Außerdem können Sie direkt auf relative Positionen im eingeblendeten Dateiausschnitt aber auch auf beliebige absolute Positionen in der aktuellen Datei springen. In letzterem Fall verschiebt sich der Dateiausschnitt gegebenenfalls automatisch mit, sodaß die Schreibmarke immer sichtbar bleibt.

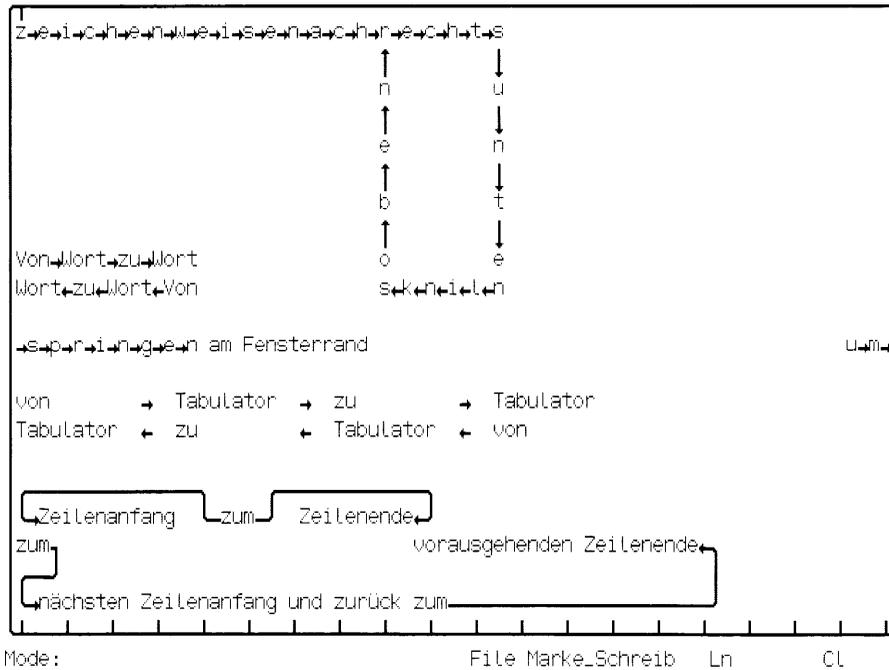


Bild 3-4 Bewegen der Schreibmarke innerhalb des Fensters

Sie bewegen die Schreibmarke

- um eine Zeilen- bzw. Spaltenposition in einer der vier Himmelsrichtungen mit den ARROW-Kommandos weiter (siehe Kapitel 6: DOWNARROW, LEFTARROW, RIGHTARROW, UPARROW)
- um eine Tabulatorposition in horizontaler Richtung mit den TAB-Kommandos weiter (siehe Abschnitte 3.5 Der Bildschirms des MAXed, 4.3.1 Formatränder, Kapitel 6: BACKTAB, TAB).

#### *Anmerkung*

Beachten Sie bei diesen beiden Kommandogruppen, daß die Schreibmarke, wenn Sie sie auf bzw. über einen Fensterrand bewegen, standardmäßig auf den gegenüberliegenden Fensterrand derselben Zeile bzw. Spalte umspringt und sich bei weiteren Kommandos von dort aus weiterbewegt. Diese Arbeitsweise ist häufig vorteilhaft zum schnellen Positionieren. Sie können sie aber auch umstellen (siehe Abschnitt 4.1.4 Der Fenster-Modus).

Sie bewegen die Schreibmarke

- wortweise in horizontaler Richtung mit den WORD-Kommandos weiter (siehe Kapitel 6: WORDLEFT, WORDRIGHT).
- an den Anfang der nächsten Zeile bzw. ans Ende der vorherigen Zeile mit den Kommandos RETURN bzw. UPEND (siehe Abschnitt 4.1.3 Der Zeilenschaltungs-Modus, Kapitel 6: RETURN, UPEND).

Durch Verstärken eines der oben genannten Bewegungskommandos können Sie mehrere gleiche Bewegungsschritte in derselben Richtung mit einem MAXed-Kommando ausführen (siehe Abschnitt 3.7 Die Kommandoverstärkung).

#### **Beispiel**

```
<ENTER> 42 <RIGHTARROW>
```

Sie bewegen die Schreibmarke auf einen Schlag um 42 Zeichen nach rechts.

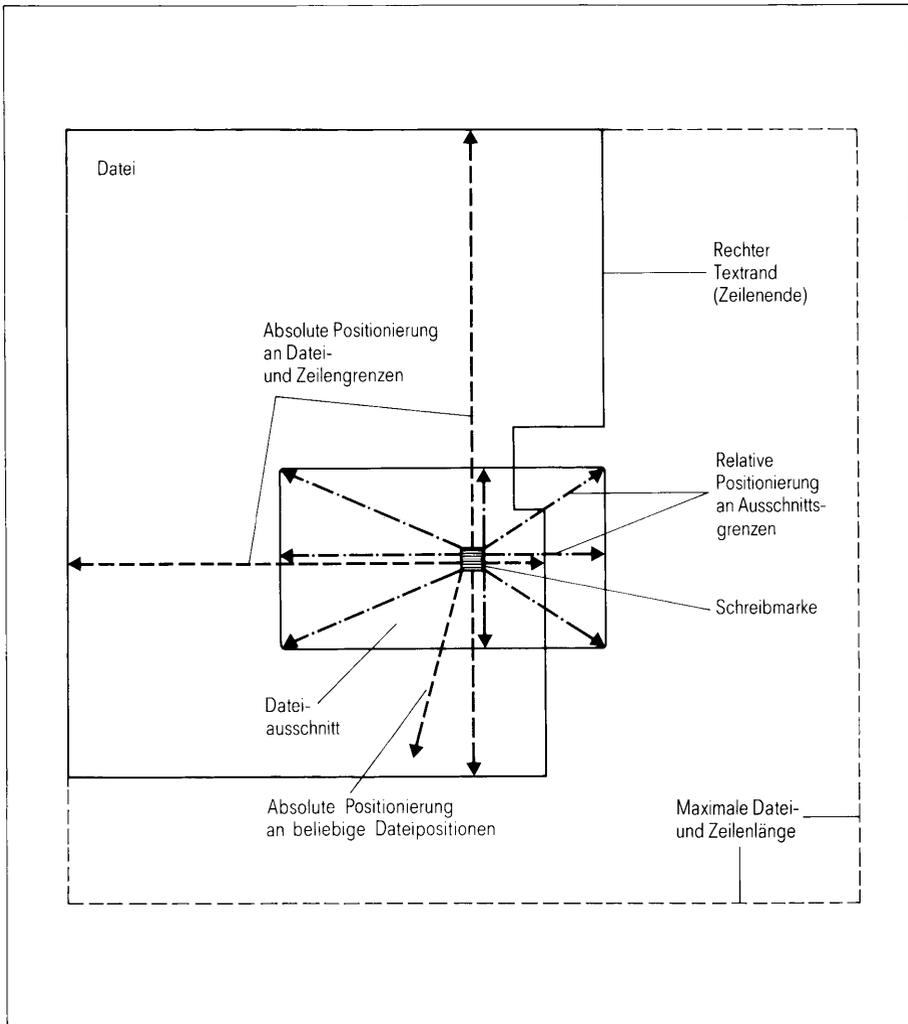


Bild 3-5 Sprung der Schreibmarke an relative und absolute Positionen

Mit dem Kommando HOME können Sie die Schreibmarke innerhalb des eingblendeten Dateiausschnitts an Positionen am Rand und an den Ecken des aktuellen Fensters setzen (siehe Kapitel 6: HOME).

Sie springen mit der Schreibmarke

- in die linke obere Ecke des aktuellen Fensters mit  
<HOME>
- in horizontaler Richtung an den linken bzw. rechten Rand des aktuellen Fensters mit  
<ENTER> <LEFTARROW> <HOME> bzw.  
<ENTER> <RIGHTARROW> <HOME>
- in vertikaler Richtung an den oberen bzw. unteren Rand des aktuellen Fensters mit  
<ENTER> <UPARROW> <HOME> bzw.  
<ENTER> <DOWNARROW> <HOME>
- in die rechte obere bzw. linke untere bzw. rechte untere Ecke des aktuellen Fensters mit  
<ENTER> <RIGHTARROW> <UPARROW> <HOME> bzw.  
<ENTER> <LEFTARROW> <DOWNARROW> <HOME> bzw.  
<ENTER> <RIGHTARROW> <DOWNARROW> <HOME>

Mit den Kommandos `STARTLINE` und `ENDLINE` positionieren Sie die Schreibmarke innerhalb der aktuellen Zeile auf der ersten Spalte bzw. direkt hinter der letzten beschriebenen Spalte. Sie springen an den Zeilenanfang bzw. an das Zeilenende mit

<STARTLINE> bzw.  
<ENDLINE>

Der Dateiausschnitt im aktuellen Fenster bewegt sich gegebenenfalls mit (siehe Kapitel 6: `ENDLINE`, `STARTLINE`).

Mit dem Kommando `GOTO` können Sie die Schreibmarke auf jede beliebige Stelle innerhalb der Grenzen der aktuellen Datei absolut positionieren (siehe Kapitel 6: `GOTO`). Der Dateiausschnitt im aktuellen Fenster bewegt sich gegebenenfalls mit.

Sie positionieren die Schreibmarke

- auf die erste Zeile der aktuellen Datei mit `<GOTO>`
- auf die letzte Zeile der aktuellen Datei mit `<ENTER> . <GOTO>`
- an eine beliebige Zeile und Spalte der aktuellen Datei mit `<ENTER> [zeile] [, [spalte]] <GOTO>`

### Beispiel

`<ENTER> 4242,42 <GOTO>`

Die Schreibmarke wird auf Zeile 4242 und Spalte 42 positioniert.

### 3.3.3 Verschieben des Dateiausschnitts

Sie können jeden Ausschnitt der aktuellen Datei im aktuellen Fenster einblenden, indem Sie den Dateiausschnitt in horizontaler Richtung von der ersten Spalte bis zur maximalen Zeilenlänge sowie in vertikaler Richtung von der ersten Zeile bis zur maximalen Anzahl der Zeilen einer im MAXed geladenen Datei verschieben (siehe Anhang: Maximalwerte bei der Dateigröße). Es sind aber auch diagonale Verschiebungen auf absolute Dateipositionen möglich. Die Schreibmarke wird gegebenenfalls mitbewegt, sodaß sie immer im eingeblendeten Dateiausschnitt sichtbar bleibt. Zum Verschieben des Dateiausschnitts stehen Ihnen wiederum MAXed-Kommandos mit verschiedener Schrittweite oder mit absoluter Positionierung innerhalb der Datei zur Verfügung.

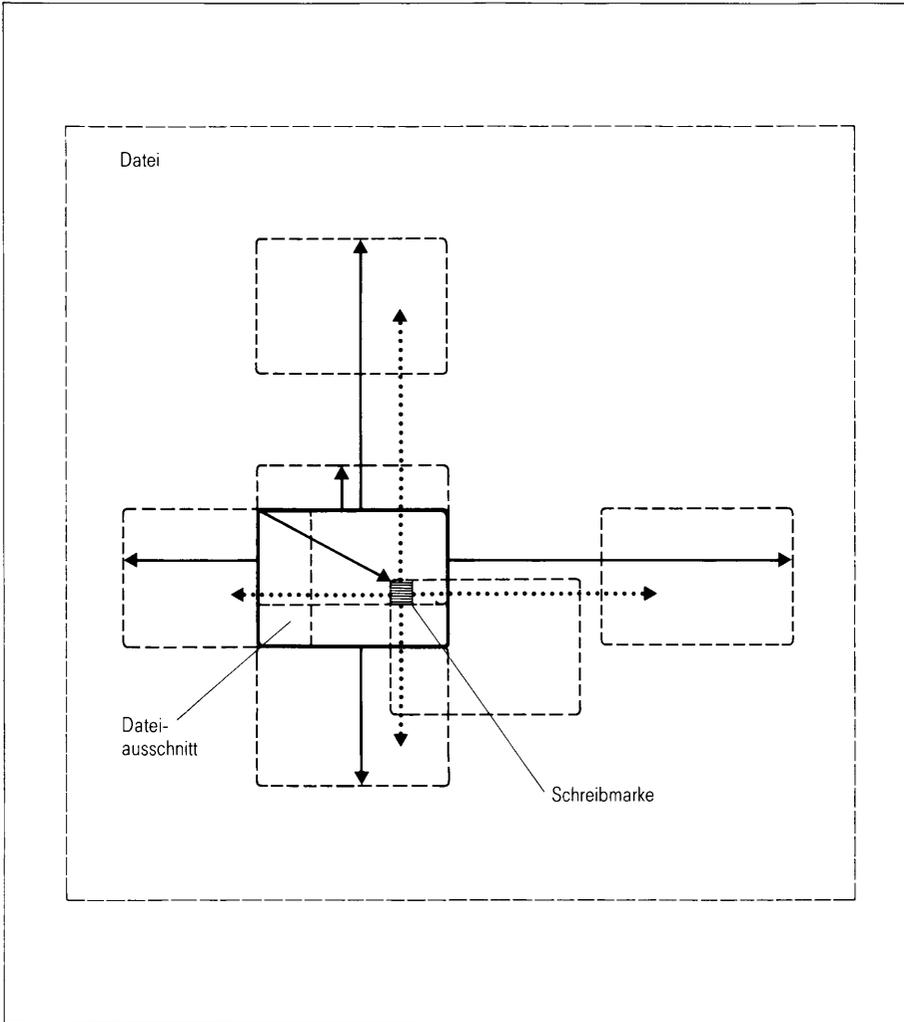


Bild 3-6 Verschieben des Dateiausschnitts

Mit den LINE-Kommandos verschieben Sie den Dateiausschnitt im aktuelle Fenster in vertikaler Richtung um eine oder mehrere Zeilen (siehe Kapitel 6: LINEFEED, RVLINFEED, MINUSLINE, PLUSLINE).

Sie verschieben den Dateiausschnitt

- um eine Zeile nach unten bzw. nach oben mit

<LINEFEED> bzw.  
<RVLINFEED>

Die Schreibmarke bewegt sich in dieselbe Richtung mit, bleibt also relativ zum Fensterrand stehen.

- um etwa ein Drittel der Fensterhöhe nach unten bzw. nach oben mit

<PLUSLINE> bzw.  
<MINUSLINE>

Die Schreibmarke bewegt gegebenenfalls in dieselbe Richtung mit, sodaß sie immer sichtbar bleibt.

Mit den PAGE-Kommandos blättern Sie in der aktuellen Datei in vertikaler Richtung um ganze Seiten weiter (siehe Kapitel 6: MINUSPAGE, PLUSPAGE). Sie verschieben den Dateiausschnitt um eine ganze Fensterhöhe nach unten bzw. nach oben mit

<PLUSPAGE> bzw.  
<MINUSPAGE>

Die Schreibmarke bewegt sich in dieselbe Richtung mit, bleibt also relativ zum Fensterrand stehen.

Durch Verstärken eines der oben genannten Verschiebe-Kommandos können Sie mehrere gleiche Fensterverschiebungen in derselben Richtung mit einem MAXed-Kommando ausführen (siehe Abschnitt 3.7 Die Kommandoverstärkung).

### Beispiel

<ENTER> 42 <PLUSPAGE>

Sie blättern auf einen Schlag um 42 Seiten nach unten.

Mit den `WIN`-Kommandos verschieben Sie den Dateiausschnitt im aktuellen Fenster in horizontaler Richtung um mehrere Spalten (siehe Kapitel 6: `WINL`, `WINR`).

Sie verschieben den Dateiausschnitt

- um etwa ein Drittel der Fensterbreite nach links bzw. nach rechts mit

`<WINL>` bzw.  
`<WINR>`

- um  $n$  Spalten nach links bzw. nach rechts mit

`<ENTER> n <WINL>` bzw.  
`<ENTER> n <WINR>`

Die Schreibmarke bewegt gegebenenfalls in dieselbe Richtung mit, sodaß sie immer sichtbar bleibt.

Mit dem Kommando `HOME` können Sie den Dateiausschnitt im aktuellen Fenster in diagonaler Richtung auf eine beliebige absolute Position innerhalb den Grenzen der aktuellen Datei verschieben (siehe Kapitel 6: `HOME`).

Sie verschieben den Dateiausschnitt so, daß

- die Position der Schreibmarke in der linken oberen Ecke des aktuellen Fensters steht, mit

`<ENTER> <HOME>`

- die Position Zeile *zeile* Spalte *spalte* der aktuellen Datei in der linken oberen Ecke des aktuellen Fensters steht, mit

`<ENTER> [zeile] [, [spalte]] <HOME>`

Die Schreibmarke wird gleichzeitig immer an die linke obere Ecke bewegt.

### Beispiel

`<ENTER> ,42 <HOME>`

Die Spalte 42 der aktuellen Zeile wird in die linke obere Fensterecke geschoben.

### 3.4 Arbeiten mit mehreren Dateien

Sie können während einer MAXed-Sitzung mehrere Dateien laden (siehe Abschnitte 3.2 Starten und Beenden des MAXed, 3.3.1 Einblenden und Sichern einer Datei). Der MAXed hält alle von Ihnen geladenen Dateien bis zum Sitzungsende ständig geöffnet. Dadurch stehen diese Dateien jeweils in ihrem aktuellen Arbeitszustand für Sie bereit. Sie können zwischen den geöffneten Dateien beliebig hin- und herschalten.

#### 3.4.1 Wechseln zwischen Dateien

Sie wechseln zwischen verschiedenen Dateien mit dem Kommando EDIT. Der Kommandoaufruf zum Umschalten zu einer bereits geöffneten Dateien ist identisch mit jenem zum Laden einer Datei:

```
<ENTER> dateiname <EDIT> oder  
<ENTER> dateiname <RETURN>
```

Eine bereits geöffnete Datei finden sie mit den Einstellungen und mit der Schreibmarkenposition vor, wie sie zuletzt eingeblendet war.

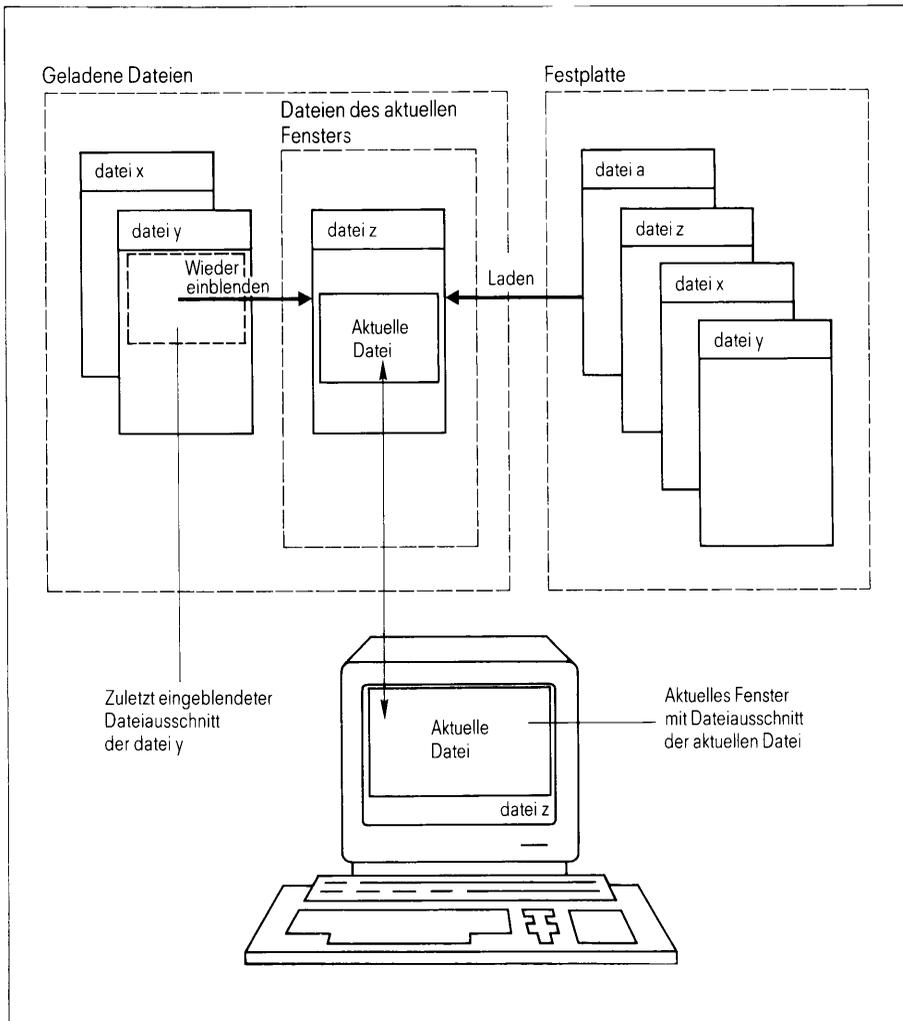


Bild 3-7 Wechseln von der aktuellen zu einer anderen Datei

Der MAXed merkt sich jeweils die Datei, die zuletzt im aktuellen Fenster eingeblendet war, als Alternativ-Datei. Sie schalten um zur Alternativ-Datei mit `<EDIT>`

Damit tauschen Sie die aktuelle Datei und die Alternativ-Datei untereinander aus.

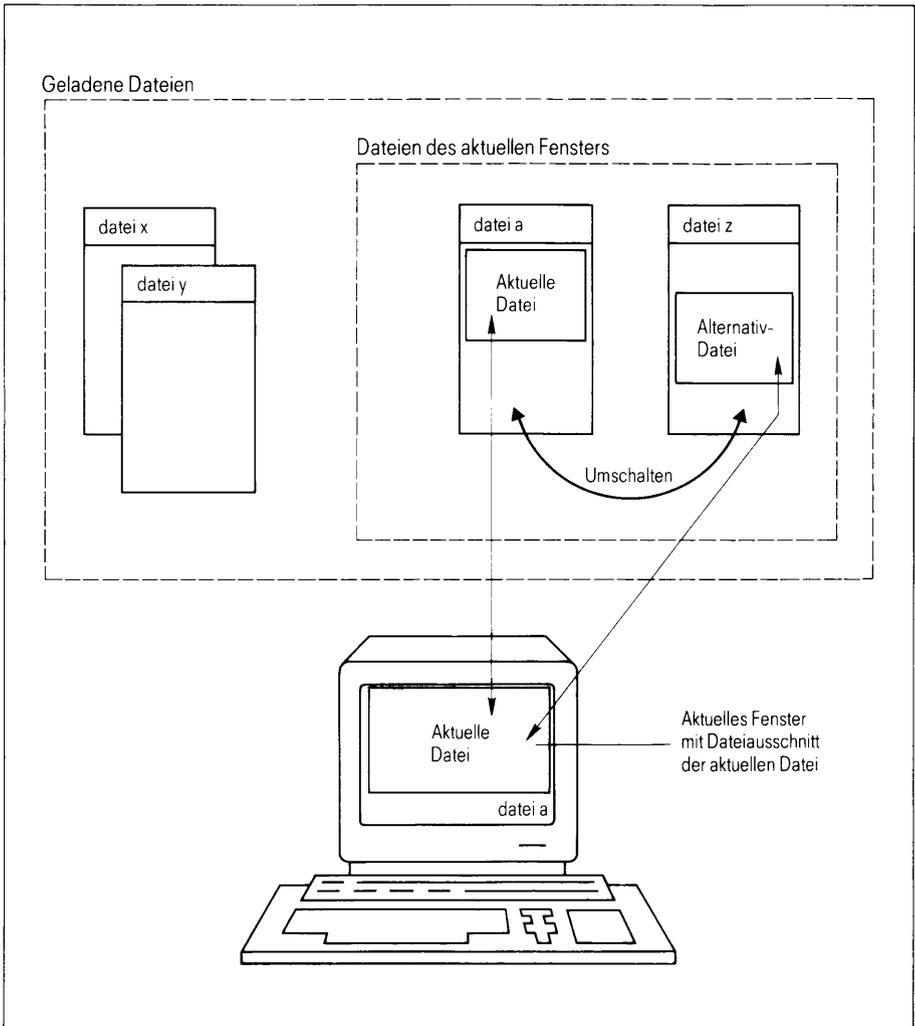


Bild 3-8 Umschalten von der aktuellen zur Alternativ-Datei

### Anmerkung

Es besteht auch die Möglichkeit, die aktuelle Datei nochmals im aktuellen Fenster einblenden zu lassen, z.B. mit folgendem Kommando (siehe Abschnitt 3.6.3 Sonderzeichen für die Kommandoattributierung):

<ENTER> @f <EDIT>

Da die aktuelle Datei nun auch als Alternativ-Datei existiert, können Sie mit <EDIT> zwischen zwei unabhängigen Schreibmarkenpositionen in derselben Datei hin- und herspringen.

### 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten

Der MAXed bietet Ihnen die Möglichkeit, mehrere Fenster auf Ihrem Bildschirm anzulegen. Jedes Fenster kann einen beliebigen Ausschnitt einer beliebigen Datei zeigen. In verschiedenen Fenstern dürfen Sie auch verschiedene Ausschnitte ein und derselben Datei einblenden.

Grundsätzlich ist immer nur eines der angelegten Fenster aktiv. Das aktive Fenster ist das aktuelle Fenster, das durch eine durchgezogene Umrangungslinie kenntlich gemacht ist. Darin ist ein Ausschnitt der aktuellen Datei eingeblendet (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Die Schreibmarke befindet sich im aktiven Fenster, und Sie arbeiten innerhalb des Fensterrands genauso wie bei einem einzigen Fenster auf dem Bildschirm.

```

Anfang der datei1          Anfang der datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei1 datei1 datei1 datei1 Ende der datei2
datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1 datei1
datei1 datei1 datei1 datei1 Ende der datei1
datei1 datei1 datei1 datei1
Mode: INSERT                      File datei1                      Ln 3                      Cl 3
    
```

Bild 3-9 Mehrere Dateiausschnitte auf dem Bildschirm

Mit den Kommandos `HWINDOW` bzw. `VWINDOW` teilen Sie das aktuelle Fenster horizontal bzw. vertikal in zwei Fenster (siehe Abschnitt 3.5 Der Bildschirm des MAXed, Kapitel 6: `HWINDOW`, `VWINDOW`). In beiden Fällen wird das neu erzeugte Fenster zum aktuellen Fenster. Die Fenster werden in der Reihenfolge ihrer Erstellung durchnummeriert.

Sie teilen das aktuelle Fenster an der aktuellen Position der Schreibmarke

- horizontal und erzeugen unterhalb ein neues Fenster mit

`<HWINDOW>`

- vertikal und erzeugen rechts davon ein neues Fenster mit

`<VWINDOW>`

Im neu erstellten, aktuellen Fenster ist die bisher aktuelle Datei nochmals eingeblendet. Sie können im aktuellen Fenster die Datei wechseln (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei). Erstellen eines Fensters und Wechseln der Datei läßt sich aber auch in einem Kommando ausführen:

Sie teilen das aktuelle Fenster und lassen sich eine andere Datei im neuen Fenster einblenden mit

`<ENTER> dateiname <HWINDOW>` bzw.  
`<ENTER> dateiname <VWINDOW>`

Sie wechseln in ein anderes, bereits bestehendes Fenster mit dem Kommando `CHWINDOW` (siehe Kapitel 6: `CHWINDOW`). Das bisher aktuelle Fenster wird deaktiviert und ein anderes Fenster wird als aktuelles Fenster aktiviert.

Sie wechseln in das Fenster

- mit der nächsthöheren Nummer oder ersatzweise in jenes mit der kleinsten Nummer mit

`<CHWINDOW>`

- mit einer bestimmten Nummer *nummer* mit

`<ENTER> nummer <CHWINDOW>`

Sie löschen das zuletzt erzeugte Fenster mit der höchsten Nummer mit

<ENTER> <HWINDOW> oder  
<ENTER> <VWINDOW>

Das "jüngste" Fenster wird entfernt. Das "zweitjüngste" Fenster, von dem das "jüngste" abgetrennt worden war, erhält wieder seine alte Größe und wird zum aktuellen Fenster.

Jedem Fenster, in dem Sie mindestens einmal die Datei gewechselt haben, ist wiederum die jeweils zuletzt eingeblendete Datei als Alternativ-Datei zugeordnet (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien). Der MAXed lädt allerdings ein und dieselbe Datei nur einmal, auch wenn Sie sie in mehreren Fenstern einblenden. Das hat zur Folge, daß die Änderungen an einer Datei im aktuellen Fenster, die noch in einem der momentan passiven Fenster sichtbar ist, in diesem zweiten Fenster ebenfalls nachgezogen werden.

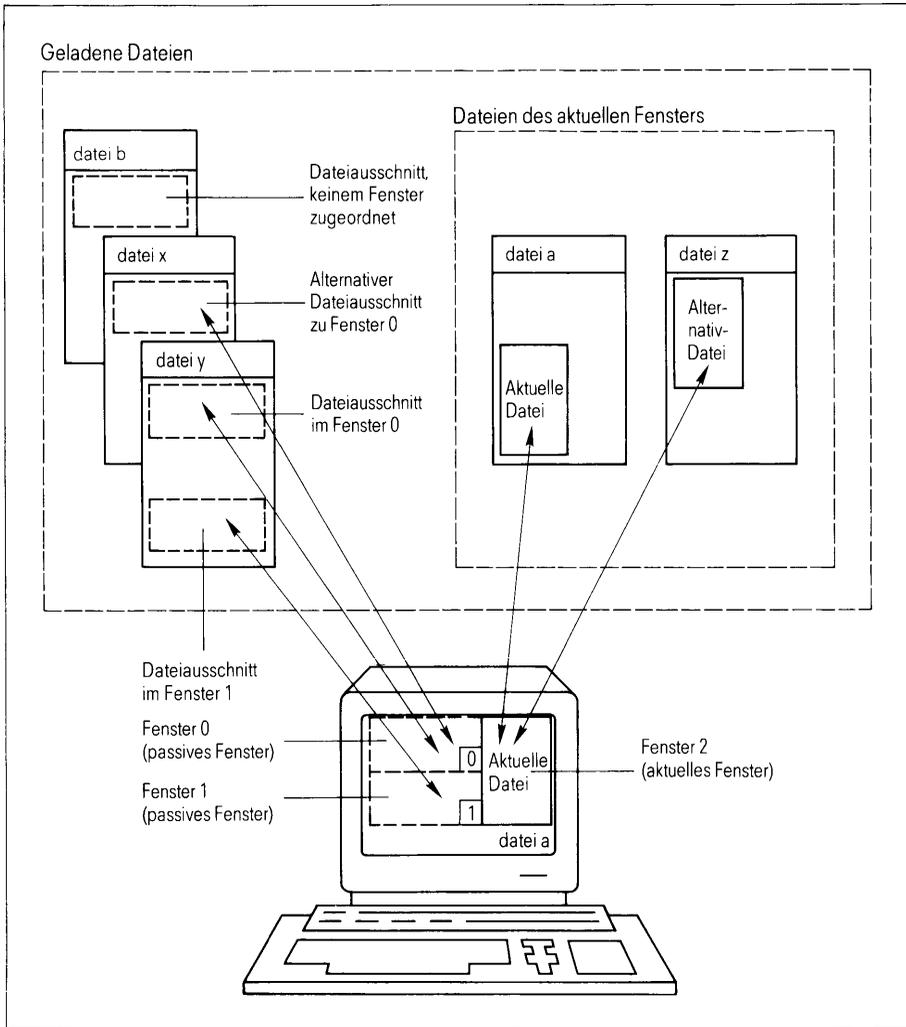


Bild 3-10 Zuordnung geöffneter Dateien zu den einzelnen Fenstern

### 3.5 Der Bildschirm des MAXed

Der MAXed-Bildschirm ist in den Fensterbereich und in die Steuerzeile unterteilt, die sich am unteren Rand des Bildschirms befindet (siehe Abschnitt 3.5.2 Die Steuerzeile). Der MAXed-Bildschirm belegt beispielsweise beim *Alpha-Terminal 97801*, das für die Abbildungen in diesem Handbuch verwendet wurde, den maximalen Bereich von 24 Zeilen und 79 Spalten.

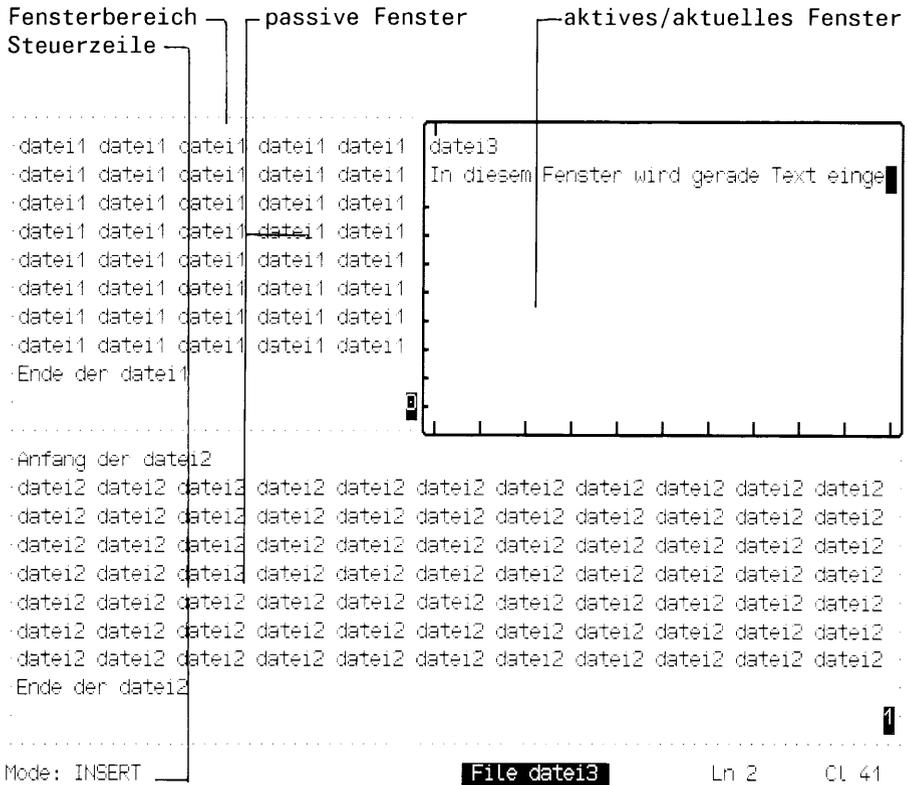


Bild 3-11 MAXed Bildschirm mit mehreren Fenstern

Im einfachsten Fall besteht der Fensterbereich nur aus einem Fenster, dem aktuellen Fenster, das damit die maximale Größe eines MAXed-Fensters erhält (siehe Abschnitt 3.5.1 Das aktuelle Fenster): Beim oben genannten *Alpha-Terminal* bedeutet dies einen maximalen Dateiausschnitt von 21 Zeilen und 77 Spalten.

Der Fensterbereich kann aber auch in bis zu 16 Fenster unterteilt werden (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten). Die Fenster werden in der Reihenfolge ihrer Erstellung mit dezimalen Ziffern, das sind aufsteigend die Zahlen 0 bis 9 und Buchstaben a bis f, durchnummeriert.

Grundsätzlich ist immer nur eines der angelegten Fenster aktiv. Das aktive Fenster ist das aktuelle Fenster, das durch eine durchgezogene Umrangungslinie kenntlich gemacht ist. Die Schreibmarke befindet sich im aktiven Fenster, und Sie arbeiten innerhalb des Fensterrands genauso wie bei einem einzigen Fenster auf dem Bildschirm. Die Angaben in der Steuerzeile beziehen sich immer auf das momentan aktuelle Fenster.

Sie können den unteren Teil des MAXed-Bildschirm mit einem Bereich überlagern, der die CLI-Subshell des MAXed anzeigt (siehe Abschnitt 4.5.1 Wechseln in die CLI-Sub-Shell, Kapitel 6: CLI):

```

.....
datei1 datei1 datei1 datei1 datei1 ter wurde gerade Text eingegeben.
datei1 datei1 datei1 datei1 datei1
Ende der datei1
.....
Anfang der datei2
datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2
datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2 datei2
.....
MAXed S C H L A E F T           File datei3           Ln 1           Cl 49
$ echo Hier ist die Shell
Hier ist die Shell
$ █

```

Bild 3-12 Gleichzeitige Anzeige von MAXed-Bildschirm und MAXed-Subshell

Auf diese Weise wird gleichzeitig der MAXed-Bildschirm angezeigt, während Sie in der Subshell arbeiten, und umgekehrt. Beim Umschalten zwischen den beiden Bereichen wird der bisher aktive Bereich deaktiviert und die Schreibmarke wechselt in den neu aktivierten Bereich. Die Unterteilung des Bildschirms in die zwei Bereiche bleibt jedoch erhalten, bis Sie die Anzeige der Subshell explizit löschen (siehe Kapitel 6: CLI).

**Anmerkung**

Falls Ihr Bildschirm die Darstellung verschiedener Zeilenlängen zuläßt, können Sie diese mit dem MAXed-Kommando <CHLINELEN> umschalten (siehe Kapitel 6: CHLINELEN).

### 3.5.1 Das aktuelle Fenster

Im aktuellen Fenster des MAXed-Bildschirms ist ein Ausschnitt der aktuellen Datei eingublendet.

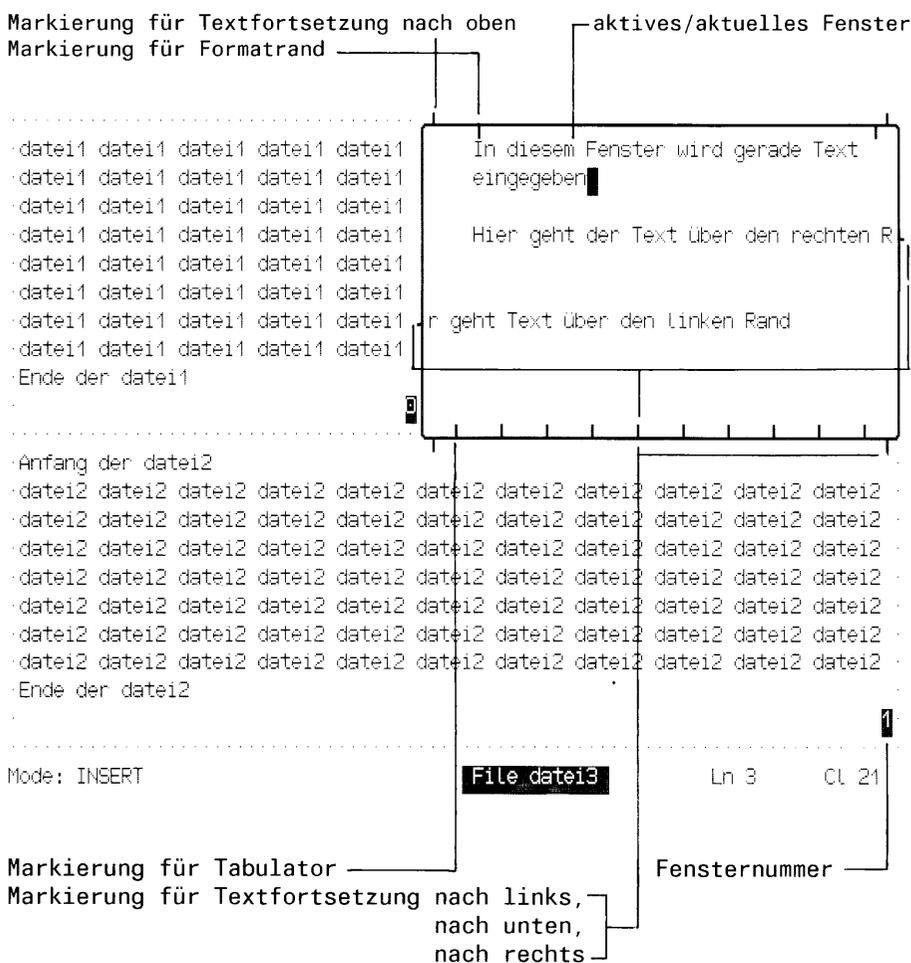


Bild 3-13 Das aktuelle Fenster des MAXed-Bildschirms

Sie erkennen das aktuelle Fenster an der durchgezogenen Umrandung im Gegensatz zu den unterbrochenen Umrandungslinien der passiven Fenster. Bei den passiven Fenstern wird jeweils am rechten unteren Rand deren Nummer angezeigt. Die Umrandungslinie des aktuellen Fensters enthält einige Merkmale, die Ihnen Auskunft über die Umgebung des sichtbaren Dateiausschnitts sowie über Formatrand- und Tabulatoreinstellungen geben (siehe Abschnitte 4.3.1 Tabulatoren, 4.3.2 Formatränder):

#### Am oberen Rand

wird links und rechts durch kleine senkrechte Markierungsstriche nach oben angezeigt, ob auch oberhalb des sichtbaren Fensterausschnitts in der Datei Zeilen vorhanden sind. Dies ist immer dann der Fall, wenn das Fenster nicht am Anfang der bearbeiteten Datei steht, also bereits in vertikaler Richtung nach unten verschoben wurde (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts). Durch kleine nach unten zeigende senkrechte Markierungsstriche erkennen Sie, an welcher Spalte der linke bzw. rechte Formatrand definiert ist (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN).

#### Am unteren Rand

wird links und rechts durch kleine senkrechte Markierungsstriche nach unten angezeigt, ob auch unterhalb des sichtbaren Fensterausschnitts in der Datei Zeilen vorhanden sind. Durch kleine nach unten zeigende senkrechte Markierungsstriche erkennen Sie, an welchen Spalten Tabulatoren definiert sind (siehe Abschnitt 4.3.1 Tabulatoren, Kapitel 6: SRTAB).

#### Am linken Rand

werden durch einen kleinen waagrechten Markierungsstrich nach links jene Zeilen markiert, in denen der Text über den linken Fensterrand hinausgeht. Dies tritt dann auf, wenn Sie das Fenster bereits in horizontaler Richtung nach rechts verschoben haben (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

#### Am rechten Rand

werden durch einen kleinen waagrechten nach rechts zeigenden Markierungsstrich jene Zeilen markiert, in denen der Text über den rechten Rand hinausgeht.

### 3.5.2 Die Steuerzeile

Die Steuerzeile befindet sich am unteren Rand des MAXed-Bildschirms. Sie enthält einige Informationen zur momentan aktuellen Datei, d.h. derjenigen Datei, die gerade im aktuellen Fenster eingeblendet ist (siehe Abschnitt 3.5.1 Das aktuelle Fenster). Auf der Steuerzeile finden Sie von links nach rechts drei Felder vor:

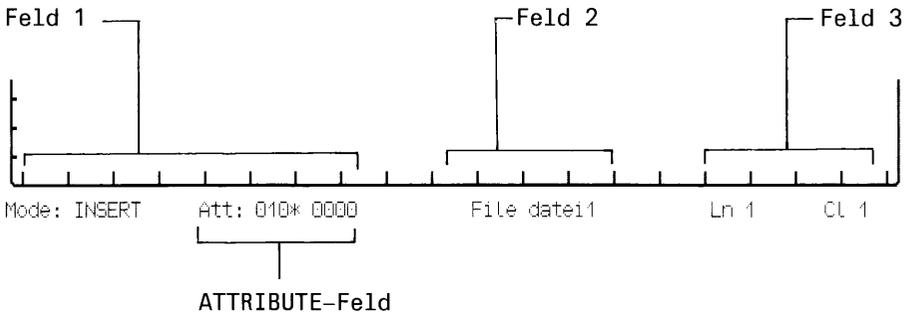


Bild 3-14 Die Steuerzeile des MAXed-Bildschirms

**Feld 1: Mode: eingabe-modus**

An dieser Stelle wird der momentan gültige Eingabe-Modus angezeigt (siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: INSERTMODE). Sie können im wesentlichen zwischen den zwei alternativ verfügbaren, grundsätzlichen Modi zur Texteingabe umschalten: Einfüge-Modus und Überschreibe-Modus. Standardmäßig ist der Einfüge-Modus INSERT eingestellt.

*Beispiel*

```
<ENTER> i <INSERTMODE>
```

Damit stellen Sie den Standard-Einfüge-Modus INSERT ein.

```
<INSERTMODE>
```

Mit diesem Kommando schalten Sie zwischen dem alternativ verfügbaren Einfüge- und Überschreibe-Modus um.

**Feld 2: File dateiname**

In Feld 2 steht normalerweise der Name der aktuellen Datei (In Bild 3-1 ist UNDEFINED eingeblendet, da noch keine Datei geladen ist). Sobald Sie in dieser Datei Änderungen vornehmen, wird das Feld 2 invers dargestellt (siehe Abschnitt 4.4 Die Zeichenattribute). Wenn Sie die aktuelle Datei wieder auf den Versionsstand bei Sitzungsbeginn zurücksetzen, wird Feld 2 wieder normal dargestellt (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

**Feld 3: Ln zeile C1 spalte**

Hier wird die aktuelle Schreibmarkenposition in der aktuellen Datei durch Zeilen- und Spaltennummer angezeigt. Steht die Schreibmarke links des linken bzw. auf oder rechts des rechten Formatrandes, so wird die Spaltenangabe invers dargestellt (siehe Abschnitt 4.3.2 Formatränder).

**Das ATTRIBUTE-Feld: ATT: iubr gxxx**

Dieses Feld wird zwischen Feld 1 und 2 angezeigt, wenn Sie die aktuelle Einstellung der Zeichenattribute aktiviert haben oder mit der Schreibmarke in das ATTRIBUTE-Feld springen, um die aktuelle Einstellung zu verändern (siehe Abschnitt 4.4 Die Zeichenattribute, Kapitel 6: ATTRIBUTE, SETATTRIB). In letzterem Falle blendet der MAXed im Feld 1 gleichzeitig eine Meldung ein.

### 3.6 Prinzipielles zur Attributierung von Kommandos

Sie schicken ein MAXed-Kommando grundsätzlich durch Drücken der zugehörigen Kommandotaste bzw. Tastenkombination ab (siehe Referenzkarte).

Viele der MAXed-Kommandos können Sie ausführen, indem Sie lediglich die Kommandotasten drücken, ohne weitere Angaben zu machen. Hierbei wird im allgemeinen die voreingestellte Standard-Ausführung der Kommandos ausgelöst (siehe Abschnitt 3.12 Voreingestellte Kommandoattribute).

#### Beispiel 1

<EDIT>

holt die Alternativ-Datei ins aktuelle Fenster (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien).

#### Beispiel 2

<ENTER> <EDIT>

Wenn Sie diese beiden Kommandotasten wie angegeben nacheinander drücken, wird die Pseudo-Datei RESTORE im aktuellen Fenster eingeblendet (siehe Abschnitt 4.2 Die Pseudo-Dateien).

### 3.6.1 Das Format der Kommandoattributierung

Wenn Sie genauere Angaben zu einem MAXed-Kommando machen müssen, z.B. den Namen einer bestimmten Datei, so geben Sie diese Angabe dem Kommando in Form einer Kommandoattributierung mit.

Jede Attributierung eines Kommandos leiten Sie durch Drücken der Kommandotaste <ENTER> ein. Die Schreibmarke springt ins Eingabefeld der Steuerzeile, und Sie können die Kommandoattribute eingeben (siehe Abschnitt 3.5.2 Die Steuerzeile). Danach lösen Sie das attributierte MAXed-Kommando durch Drücken der zugehörigen Kommandotaste <KOMMANDO> aus oder brechen die Eingabe ab, z.B. mit dreimal Betätigen der Taste *ESC*.

Ein attributiertes Kommando hat im allgemeinen die Form:

```
<ENTER> [bereich] [argument [argument ...]] <KOMMANDO>
oder
<ENTER> [n] [argument [argument ...]] <KOMMANDO>
```

Folgende optionale Attribute dienen der Spezifizierung von MAXed-Kommandos:

#### bereich

der Bereich kann durch Bewegen der Schreibmarke oder durch ein Zeichenketten-Attribut in der Steuerzeile eingegeben werden (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut).

#### argument

ein Argument in Form einer Zeichenkette kann sein:

- ein IDENTIFIER, also ein Name für eine Datei, einen Puffer, eine Marke, einen Makro oder eine MAXed-Steuerdatei (siehe Abschnitt 3.8 Der IDENTIFIER als Kommandoattribut);
- ein PATTERN zur Darstellung von Suchmustern (siehe Abschnitt 3.10 PATTERN als Kommandoattribut);
- eine Ersetzungszeichenkette, (siehe Abschnitt 3.11 Die Ersetzungszeichenkette als Kommandoattribut, Kapitel 6: REPLACE);

- ein Parameter für spezielle Varianten oder Einstellungen von MAXed-Kommandos (siehe Abschnitt 6.1 Die MAXed-Kommandos);
  - ein SINIX-Kommando (siehe Abschnitt 4.5 Die SINIX-Schnittstellen)
- n eine ganze Zahl als Verstärkungsfaktor (siehe Abschnitt 3.7 Die Kommandoverstärkung) zur mehrfachen Ausführung eines Kommandos.

### Anmerkung

- Wenn Sie mehrere Zeichenketten-Attribute eingeben, müssen Sie diese durch Leerzeichen trennen und die oben angegebene Reihenfolge einhalten (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Ein Zeichenketten-Attribut, das Leerzeichen enthält, müssen Sie in Anführungszeichen "" setzen.

#### Beispiel

```
<ENTER> attribut_1 "att ribut_2" <KOMMANDO>
```

Damit attributieren Sie ein MAXed-Kommando mit den zwei Zeichenketten-Attributen *attribut\_1* und *att ribut\_2*.

IDENTIFIER und Bereichsdefinitionen dürfen grundsätzlich weder Leerzeichen enthalten noch in Anführungszeichen stehen (siehe Abschnitt 3.8 Der IDENTIFIER als Kommandoattribut, Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut).

- Wie durch die obige Kommando-Syntax angedeutet ist, können Sie MAXed-Kommandos, die mit Bereichen arbeiten, nicht verstärken und umgekehrt verstärkbare Kommandos nicht mit Bereichsdefinitionen attributieren.
- Die einzige Ausnahme bezüglich der allgemeinen Reihenfolge der Attribute findet sich beim Kommando REPLACE (siehe Kapitel 6: REPLACE). Falls Sie hierbei *bereich* als Zeichenkette eingeben, müssen Sie diese als letztes Attribut angeben.

**Beispiel**

```
<ENTER> datei 4242 42 <zweiundvierzig> <EDIT>
```

holt die Datei mit dem Namen *datei* ins aktuelle Fenster, setzt die Schreibmarke in Zeile 4242 auf Spalte 42 und sucht vorwärts nach einer Textstelle, die die Zeichenkette *zweiundvierzig* enthält (siehe Kapitel 6: EDIT, Abschnitt 3.10 PATTERN als Kommandoattribut)

**Eine Alternative zum Abschicken von MAXed-Kommandos**

Alternativ zum Drücken der zugehörigen Kommandotaste können Sie ein MAXed-Kommando auch folgendermaßen abschicken:

```
<ENTER> [bereich] [argument [argument ...]] KOMMANDO <ENTER>  
oder  
<ENTER> [n] [argument [argument ...]] KOMMANDO <ENTER>
```

Sie geben den Kommandonamen als letztes Kommandoattribut im Eingabefeld ein und starten das Kommando mit einem abschließenden <ENTER>. Hierbei sind auch eindeutige Abkürzungen des Kommandonamens zulässig.

**Beispiel 1**

```
<ENTER> ED <ENTER>
```

holt die Alternativ-Datei ins aktuelle Fenster. *ED* ist eine eindeutige Abkürzung des MAXed-Kommandos EDIT. Die angegebene Kommandosequenz ist identisch mit <EDIT>

**Beispiel 2**

```
<ENTER> ED <ENTER>
```

Sie geben vor dem Namen(skürzel) des MAXed-Kommandos EDIT noch andere Zeichen ein (hier ein Leerzeichen): Das hat die Bedeutung eines einleitenden <ENTER> vor Aufruf des Kommandos <EDIT>. Sie blenden mit dem angegebenen Kommando die Pseudo-Datei RESTORE ins aktuelle Fenster. Die angegebene Kommandosequenz ist identisch mit:

```
<ENTER> <EDIT>
```

**Beispiel 3**

```
<ENTER> datei 4242 42 <zweiundvierzig> ed <ENTER>
```

Sie blenden die Datei *datei* ins aktuelle Fenster. Die vor dem Namen(skürzel) des MAXed-Kommandos EDIT eingegebenen Zeichenketten stellen die Kommandoattribute dar. Die angegebene Kommandosequenz ist identisch mit dem weiter oben beschriebenen Kommando <ENTER> datei 4242 42 <zweiundvierzig> <EDIT>

### 3.6.2 Das Eingabefeld und der Eingabepuffer

Im Eingabefeld auf der Steuerzeile können Sie Ihre Eingaben mit Hilfe einiger MAXed-Kommandos korrigieren. Dazu stehen Ihnen folgende Kommandos zur Verfügung:

- Der eingestellte Eingabe-Modus ist auch im Eingabefeld der Steuerzeile wirksam (siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: INSERTMODE). Im Einfüge-Modus werden eingegebene Zeichen in den bereits vorhandenen Text eingefügt und im Überschreibe-Modus OVERWRITE überschreiben sie den darunterliegenden Text. Sie können im Eingabefeld durch Drücken von <INSERTMODE> zwischen den alternativ verfügbaren Eingabe-Modi umschalten. Diese Änderung des Eingabe-Modus ist dann nur temporär für das Eingabefeld gültig!
- Sie löschen das Zeichen links der Schreibmarkenposition mit <BACKSPACE> und das Zeichen unter der Schreibmarke mit <DELCHAR> (siehe Kapitel 6: BACKSPACE, DELCHAR).
- Sie können die Schreibmarke zeichenweise, wortweise und tabulatorweise bewegen (siehe Kapitel 6: LEFTARROW, RIGHTARROW, WORDLEFT, WORDRIGHT, TAB, BACKTAB).
- Sie springen an den Anfang des Eingabefeldes mit <UPARROW> oder <STARTLINE> und wieder an das Ende des eingegebenen Textes mit <DOWNARROW> oder <ENDLINE> (siehe Kapitel 6: ENDLINE, STARTLINE).

#### Achtung

Bei der Verwendung der ARROW- und TAB-Kommandos während der Eingabe im Eingabefeld der Steuerzeile sollten Sie folgende Sonderfälle beachten:

- Verstärkung eines ARROW- oder TAB-Kommandos mit einer positiven ganzen Zahl  $n$  (siehe Abschnitt 3.7 Die Kommandoverstärkung): <ENTER>  $n$  <KOMMANDO>
- Einstellen eines Fenster-Modus mit den ARROW-Kommandos, wobei *modus* die Buchstaben  $c$ ,  $h$ ,  $m$ ,  $n$  oder  $v$  sein kann (siehe Abschnitt 4.1.4 Der Fenster-Modus): <ENTER> *modus* <UPARROW>

Wenn Sie genau eines der Kommandoattribute *n* bzw. *modus* im Eingabefeld eingegeben haben, lösen Sie mit Betätigen einer ARROW- oder TAB-Kommandotaste eines der oben genannten Kommandos aus! Dasselbe Verhalten ergibt sich, wenn Sie Zeichenketten eingegeben haben, die durch Textersetzung zu genau einem der genannten Kommandoattribute ausgewertet werden (siehe 3.6.4 Textersetzung im Eingabefeld). Dabei besteht jedoch keine ernsthafte Gefahr für Ihre Dateibearbeitung.

#### Die Länge des Eingabefeldes:

Die maximale Länge des Eingabefeldes entspricht etwa der Zeilenlänge in einer MAXed-Datei (siehe Anhang: Maximalwerte bei der Dateigröße). Die Steuerzeile gibt nämlich nur einen Ausschnitt des Eingabefeldes frei! Wenn Sie bei der Eingabe einer Kommandoattributierung den rechten Rand des Eingabefeldes erreichen, springt die Schreibmarke an den Anfang der MAXed-Steuerzeile zurück. Ihr bisher eingegebener Text wird nach links aus dem sichtbaren Bereich des Eingabefeldes hinausgeschoben. Am Anfang des Eingabefeldes erscheint ein Bindestrich, der Ihnen dies signalisiert. Sie können mit der Eingabe Ihrer Zeichenketten-Attribute fortfahren oder die Schreibmarke wieder mit den oben beschriebenen Kommandos im Eingabefeld rückwärts bewegen.

#### Meldung der aktuellen Bereichsdefinition:

Wenn Sie einen Bereich durch Bewegen der Schreibmarke definieren und danach noch Zeichenketten-Attribute eingeben, wird die angezeigte Meldung der Bereichsdefinition aus dem Eingabefeld in den Eingabepuffer geschoben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke, siehe unten: Verwenden des Eingabepuffers). Die Zeichenketten-Attribute, die Sie danach eingeben, beginnen wieder am Anfang des Eingabefeldes. Die Meldung zu dieser Form der Bereichsdefinition ist kein Zeichenketten-Attribut und kann deshalb nicht korrigiert werden.

## Verwenden des Eingabepuffers

Der MAXed speichert immer die zuletzt im Eingabefeld der Steuerzeile eingegebenen Kommandoattribute in einem zugehörigen Eingabepuffer. Sie können die gespeicherte Kommandoattributierung wieder aus dem Eingabepuffer ins Eingabefeld holen mit <ENTER> <BACKSPACE>

Nach Bedarf korrigieren Sie die eingeblendete Kommandoattributierung und verwenden Sie für ein weiteres MAXed-Kommando, das Sie anschließend durch Drücken der entsprechenden Kommandotaste auslösen.

Geschicktes Positionieren der Schreibmarke:

Wie weiter oben beschrieben, wird auch die Meldung zu einer Bereichsdefinition durch Bewegen der Schreibmarke im Eingabepuffer zwischengespeichert (gegebenenfalls zusammen mit den nachfolgend eingegebenen Zeichenketten-Attributen). Unter der Voraussetzung, daß diese Bereichsdefinition mehr als eine Zeile umfaßt, können Sie sie dazu benutzen, um mit der Schreibmarke ans andere Ende des definierten Bereichs zu springen. Sie holen die Meldung zu einer zuvor eingegebenen Bereichsdefinition in das Eingabefeld zurück mit <ENTER> <BACKSPACE> (siehe oben)

Gegebenenfalls müssen Sie <BACKSPACE> mehrmals betätigen bis alle zusätzlich eingegebenen Zeichenketten-Attribute gelöscht sind. Sie setzen die Schreibmarke ans andere Ende des definierten Bereichs mit <GOTO>

### Anmerkung

Wenn Sie als Bereichsdefinition einen einzeiligen STREAM verwenden, interpretiert der MAXed die oben genannte Kommandofolge als folgendes Kommando: <ENTER> "PATTERN" <GOTO>

Der einzeiligen STREAM *bereich* dient anstelle von PATTERN als Suchmuster zur Erstellung der Pseudo-Datei SCANLIST (siehe Abschnitt 4.2 Die Pseudo-Dateien, Kapitel 6: GOTO).

**Beispiel**

Sie kopieren mit dem Kommando PICK einen Textbereich und definieren ihn durch Bewegen der Schreibmarke (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER):

```
<ENTER> <PLUSPAGE> <PLUSLINE> <DOWNARROW> <DOWNARROW> <PICK>
```

Nach Eingabe des Kommandos befindet sich die Schreibmarke wieder am Beginn des Textbereichs. Sie wollen aber am anderen Ende weiterarbeiten. Genau dahin gelangen Sie aber mit

```
<ENTER> <BACKSPACE> <GOTO>
```

### 3.6.3 Sonderzeichen für die Kommandoattributierung

Alle nicht-alphanumerischen Zeichen mit Ausnahme des Unterstrichs `_` haben bei Zeichenketten-Attributen in der Regel eine Sonderbedeutung:

- Zum einen gibt es einen festen Satz von Sonderzeichen für Bereichsdefinitionen, der für alle MAXed-Kommandos, die mit Bereichen arbeiten, gültig ist (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut).
- Ein zweiter Satz von Sonderzeichen steht Ihnen für die Eingabe von Suchmustern zur Verfügung (siehe Abschnitt 3.10 PATTERN als Kommandoattribut), wobei die Sonderzeichen zur Erstellung von regulären Ausdrücken verwendet werden.
- Ein dritter Satz von Sonderzeichen ist speziell für Ersetzungszeichenketten gültig (siehe Abschnitt 3.11 Die Ersetzungszeichenkette als Kommandoattribut, Kapitel 6: REPLACE).
- Die vier Sonderzeichen Klammeraffe `@`, Dollar-Zeichen `$`, Prozent-Zeichen und Gegenhochkomma ``` können Sie in allen Zeichenketten-Attributen verwenden, um damit eine Textersetzung auszulösen (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).
- Daneben gibt es einige Sonderzeichen, die je nach Kommando eine spezifische Bedeutung haben (siehe Abschnitt 6.1 Die MAXed-Kommandos).

### 3.6.4 Textersetzung im Eingabefeld

Nach dem Abschicken eines MAXed-Kommandos, jedoch noch vor dessen Ausführung, expandiert der MAXed die eingegebenen Zeichenketten-Attribute. Dabei lösen folgende vier Sonderzeichen in Verbindung mit passenden Zeichenketten innerhalb der eingegebenen Zeichenketten-Attribute eine Textersetzung aus:

- Parameter für Textersetzung mit dem Klammeraffen-Zeichen @
- MAXed-Variablen mit dem Dollar-Zeichen \$
- PICK-Puffernamen mit dem Prozent-Zeichen %
- SINIX-Kommandoaufrufe mit dem Zeichen Gegenhochkomma `

Sie können diese Sonderzeichen mit den zugehörigen Zeichenketten prinzipiell überall im Eingabefeld der Steuerzeile einsetzen. Beim Expandieren werden sie durch ihre Werte, Inhalte bzw. Ausgaben ersetzt. Dies ist im folgenden näher beschrieben.

#### Parameter für Textersetzung

Parameter für Textersetzung sind MAXed-interne Variablen, die zu verschiedenen aktuellen Parameter-Einstellungen ausgewertet werden. Sie können sie überall im Eingabefeld der Steuerzeile durch einen einzeiligen Text ersetzen lassen. Parameter für Textersetzung bestehen grundsätzlich aus einem einzelnen Zeichen. Um die Textersetzung auszulösen, stellen Sie diesem Zeichen einen Klammeraffen @ direkt voran. Durch einen vorausgehenden Gegenschrägstrich \ entwerten Sie jedoch das Sonderzeichen (siehe unten: Entwertung und Auswertung in der Steuerzeile).

Ein Wort oder eine beliebige Zeichenkette liefern folgende Parameter:

- @*n* mit *n* als positiver ganzer Zahl wird während des Ablaufs bzw. der Definition eines Makros durch den *n*-ten Parameter des Makros ersetzt (siehe Abschnitt 4.7 Arbeiten mit Makros) Bei einem LEARN- oder MACRO-Kommando können Sie bis zu acht Makro-Parameter als Kommandoattribute angeben (siehe Kapitel 6: LEARN, MACRO). Auf die zu Beginn eingegebenen aktuellen Makro-Parameter *P1* bis *P8* können Sie bis zur Beendigung des Makros durch die Verwendung der formalen Makro-Parameter @*1* bis @*8* zurückgreifen: Im Eingabefeld der Steuerzeile wird ein formaler Parameter @*n* durch den zugehörigen aktuellen Parameter *Pn* ersetzt.

- ⌋
- Ⓐ, wird durch das Wort an der aktuellen Position der Schreibmarke in der aktuellen Datei ersetzt. Falls die Schreibmarke nicht unter einem "sichtbaren" Zeichen steht, wird durch die leere Zeichenkette ersetzt.
  - Ⓐ; wird durch den Text der Zeile, auf der die Schreibmarke momentan positioniert ist, ersetzt. Falls die aktuelle Zeile leer ist, wird durch die leere Zeichenkette ersetzt.
  - Ⓐp ([p]attern) wird durch das zuletzt eingegebene PATTERN ersetzt (siehe Abschnitt 3.10 PATTERN als Kommandoattribut, Kapitel 6: PLUSFIND, MINUSFIND, REPLACE). Falls kein Suchmuster definiert ist, wird die leere Zeichenkette ausgegeben.
  - Ⓐ" wird durch die zuletzt durch ein PLUSFIND- oder MINUSFIND-Kommando gefundene Textstelle oder ersatzweise durch die leere Zeichenkette ersetzt (siehe Kapitel 6: MINUSFIND, PLUSFIND).
  - Ⓐr ([r]eplace) wird durch die zuletzt eingegebene Ersetzungszeichenkette ersetzt (siehe Abschnitt 3.11 Die Ersetzungszeichenkette als Kommandoattribut, Kapitel 6: REPLACE). Falls keine Ersetzungszeichenkette definiert ist, wird die leere Zeichenkette ausgegeben.

### Beispiel 1

<ENTER> Ⓐ, <EDIT>

⌋ Der MAXed versucht, die Datei ins aktuelle Fenster zu holen, deren Name identisch ist mit dem Wort an der aktuellen Position der Schreibmarke, (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien).

### Beispiel 2

<ENTER> Ⓐr <PLUSFIND>

⌋ Im Anschluß an ein Ersetzungskommando suchen Sie in Vorwärtsrichtung nach dem Auftreten der Ersetzungszeichenkette.

Einen Datei- oder Benutzernamen liefern folgende Parameter:

- ④u ([u]ser) wird durch die Kennung des Benutzers ersetzt.
- ④v ([v]ersion) wird durch die Versionsbezeichnung des MAXed ersetzt.
- ④f ([f]ile) wird durch den vollständigen Pfadnamen der Datei im aktuellen Fenster ersetzt. Ist eine Pseudo-Datei im aktuellen Fenster, wird *@f* nur durch ihren einfachen Namen ersetzt (Ausnahme: Pseudo-Datei HELP).
- ④e ([e]nding) wird durch den relativen Namen der aktuellen Datei ohne Pfadanteil ersetzt.
- ④b ([b]eginning) wird durch den bloßen Pfadanteil des vollständigen Pfadnamens der aktuellen Datei ersetzt. Ist eine Pseudo-Datei im aktuellen Fenster, wird durch die leere Zeichenkette ersetzt (Ausnahme: Pseudo-Datei HELP).

### Beispiel

```
<ENTER> @b/datei <EDIT>
```

Sie holen die Datei *datei* ins aktuelle Fenster, die sich im selben Dateiverzeichnis wie die momentan aktuelle Datei befindet.

Ein einzelne Zahl liefern folgende Parameter:

- ① **@n** ([n]ummer) wird durch die das aktuelle Fenster identifizierende dezimale Nummer ersetzt (siehe Abschnitt 3.5 Der Bildschirm des MAXed).
- ② **@x** wird durch die Nummer des erfolgreichen ([x]-ten) Suchmusters aus den alternativen Suchmustern des verwendeten PATTERN ersetzt, zu dem eine Textstelle gefunden wurde, als Sie damit zuletzt ein PLUSFIND- oder MINUSFIND-Kommando aufrufen (siehe Kapitel 6: MINUSFIND, PLUSFIND): Als erweiterter regulärer Ausdruck kann ein PATTERN mehrere Suchmuster, durch den senkrechten Strich getrennt, enthalten. Diese werden von links nach rechts mit 1 beginnend durchnummeriert. Falls zuletzt nur ein Suchmuster angegeben war, wird durch die 1 ersetzt. Falls noch keine Textstelle gefunden wurde, wird durch leere Zeichenkette ersetzt.
- ③ **@#** wird durch die Anzahl der Zeilen der aktuellen Datei ersetzt.
- ④ **@\_** wird durch die Länge (Spaltenanzahl) der aktuellen Zeile ersetzt

### Beispiel

Sie wollen die Zeilenlängen mehrerer Dateien in einer Datei *datei* auflisten. Diese Arbeit soll durch ein Makro rationalisiert werden. Sie erstellen das Makro *makro* im Lern-Modus und geben ihm den Namen und die Zeilenanzahl der aktuellen Datei als Makro-Parameter mit (siehe Abschnitt 4.1.11 Der Lern-Modus):

```
<ENTER> makro @e @# <LEARN>
```

Sie wechseln in die Datei *datei*, schreiben die Werte der beiden Parameter ans Dateiende, blenden die Alternativ-Datei wieder ein und beenden die Makrodefinition:

```
<ENTER> datei <EDIT>
<ENTER> . <GOTO>
<ENTER> '@1 hat so viele Zeilen: @2' <PUT>
<EDIT>
<LEARN>
```

Bei der nächsten Datei im aktuellen Fenster, die Sie erfassen wollen, genügt folgendes Kommando:

```
<ENTER> makro @e @# <MAKRO>
```

Zeilen- und Spaltennummern, getrennt durch ein Leerzeichen, liefern folgende Parameter:

- @m (lmargin) wird durch die Spaltennummern (*l r*), an denen der linke und der rechte Formatrand im aktuellen Fenster definiert sind, ersetzt (siehe Abschnitt 4.3.2 Formatränder).
- @s (s|ize) wird durch Höhe und Breite (*h b* in Zeilen und Spalten) des aktuellen Fensters ersetzt.
- @c (c|ursor) wird durch Zeilen- und Spaltennummer (*z s*) der aktuellen Schreibmarkenposition im aktuellen Fenster ersetzt.
- @l (l|abel) wird durch Zeilen- und Spaltennummer (*z s*) der Position, an der die Standard-Marke in der Datei im aktuellen Fenster gesetzt ist, ersetzt (siehe Kapitel 6: GOTO, LABEL). Falls die Standard-Marke nicht definiert ist, wird *l l* ausgegeben.
- @w (w|indow) wird durch Zeilen- und Spaltennummer (*z s*) der linken oberen Ecke des aktuellen Fensterausschnitts ersetzt.
- @{ wird durch Zeilen- und Spaltennummer (*z s*) des Anfangs der zuletzt durch ein PLUSFIND- oder MINUSFIND-Kommando gefundenen Textstelle oder ersatzweise durch die leere Zeichenkette ersetzt (siehe Kapitel 6: MINUSFIND, PLUSFIND).
- @} wird durch Zeilen- und Spaltennummer (*z s*) der Position direkt nach dem Ende der zuletzt durch ein PLUSFIND- oder MINUSFIND-Kommando gefundenen Textstelle oder ersatzweise durch die leere Zeichenkette ersetzt (siehe Kapitel 6: MINUSFIND, PLUSFIND).

**Beispiel**

Sie wollen in der Datei *datei* die Formatränder genauso setzen wie in der aktuellen Datei. Sie erstellen ein Makro *makro* (zum weiteren Gebrauch) im Lern-Modus und geben den Namen der Datei *datei* und die Spaltennummern der aktuellen Formatränder als aktuelle Makro-Parameter ein:

```
<ENTER> makro datei @m <LEARN>
<ENTER> @1 <EDIT>
<ENTER> @2 @3 <MARGIN>
<LEARN>
```

(Die beiden Spaltennummern ersetzen *@m*, bestehen aus zwei Wörtern und werden deshalb den beiden formalen Makro-Parametern *@2* und *@3* zugewiesen!).

Die Einstellung eines Betriebs-Modus des MAXed liefern folgende Parameter:

- @i            ([i]nsermode) wird durch den aktuell eingestellten Eingabe-Modus *O*, *I*, *W*, *L* oder *S* ersetzt (siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: INSERTMODE).
- @^            wird durch den aktuell eingestellten Zeilenschaltungs-Modus *N*, *E*, *S* oder *T* ersetzt (siehe Abschnitt 4.1.3 Der Zeilenschaltungs-Modus).
- @>            wird durch den aktuell eingestellten Fenster-Modus *N*, *C*, *V*, *H* oder *M* ersetzt (siehe Abschnitt 4.1.4 Der Fenster-Modus).
- @\*            wird durch den aktuell eingestellten Such-Modus *\**, *<*, *>* oder *< >* ersetzt (siehe Abschnitt 4.1.5 Der Such-Modus).
- @g            (lan[g]uage) wird durch die eingestellte Sprache *De* oder *En* ersetzt. Davon ist nur die Spracheinstellung für den Silbentrenner und die Datumsausgabe (siehe unten *@d*) betroffen, nicht jedoch der Konversations-Modus! (siehe Abschnitt 4.3.3 Flatter- und Blocksatz, 4.1.10 Die Sprach-Modi).

Zeit und Datum liefern folgende Parameter:

- @d ([d]atum) wird durch das aktuelle Datum (deutsch: *tt.mm.jj* oder englisch: *dd-MON-yyyy*, siehe oben: @g) ersetzt.
- @t wird durch die aktuelle Zei[t] (deutsch wie englisch: *hh:mm.ss*) ersetzt.

### Beispiel

Sie wollen in die rechte obere Ecke des aktuellen Fensters das aktuelle Datum schreiben und die Schreibmarke danach wieder an die momentan aktuelle Position setzen:

```
<ENTER> @f <EDIT>
```

Die aktuelle Datei wird nochmals eingeblendet und existiert nun auch als Alternativ-Datei. Mit <EDIT> springen Sie jetzt zwischen zwei aktuellen Schreibmarkenpositionen in der Datei hin und her (siehe Kapitel 6: EDIT). Sie setzen die Schreibmarke an den rechten oberen Fensterand, schreiben dort das aktuelle Datum und blenden wieder die Alternativ-Datei mit der alternativen Schreibmarkenposition ein (siehe Kapitel 6: HOME):

```
<HOME> / Sprung in die linke obere Ecke
<ENTER> <RIGHTARROW> <HOME> / Sprung in die rechte obere Ecke
<ENTER> 10 <LEFTARROW> / Bewegung um 10 Spalten nach links
<ENTER> '@d' <PUT> / Ausgeben des aktuellen Datums
<EDIT> / Einblenden der Alternativ-Datei
```

## Ersetzen von MAXed-Variablen

MAXed-Variablen sind in der Handhabung ihrer Definition und der Auswertung Shell-Variablen recht ähnlich. Die Variablen aus der Umgebung des MAXed (also i.a. jene der Shell, aus der Sie den MAXed aufgerufen haben) sind während der MAXed-Sitzung ebenfalls gültig. Variablennamen können Sie überall im Eingabefeld der Steuerzeile durch ihren Wert ersetzen lassen. Um diese Textersetzung auszulösen, stellen Sie einem Variablennamen ein Dollar-Zeichen \$ direkt voran. Durch einen vorausgehenden Gegenschrägstrich \ entwerten Sie jedoch das Sonderzeichen (siehe unten: Entwertung und Auswertung in der Steuerzeile).

Wenn Sie Variablennamen innerhalb von Zeichenketten-Attributen zur Textersetzung verwenden, schließen Sie sie aus Eindeutigkeitsgründen durch je ein weiteres, optionales Dollar-Zeichen nach rechts ab:

```
$variable  
bzw.  
[text]$variable$[text]
```

Sie definieren während der MAXed-Sitzung eine MAXed-Variable im Eingabefeld der Steuerzeile:

```
<ENTER> variable=wert <ENTER>
```

Die Namen von MAXed-Variablen bestehen aus einem Wort, d.h. einer zusammenhängenden Zeichenkette aus alphanumerischen Zeichen. Dabei unterscheidet der MAXed sehr wohl zwischen Groß- und Kleinbuchstaben! Anführungszeichen werden vom MAXed automatisch gesetzt, wenn *wert* Leerzeichen enthält.

Während der MAXed-Sitzung sind alle Umgebungsvariablen so lange gültig, wie Sie diese Variablen nicht durch eine Neudefinition einer gleichnamigen MAXed-Variablen überdecken.

Diejenigen Variablen, die Sie während der MAXed-Sitzung definieren, werden vom MAXed in der Pseudo-Datei VARIABLES gespeichert (siehe Abschnitt 4.2.16 Die Pseudo-Datei VARIABLES). Sie blenden die Pseudo-Datei VARIABLES im aktuellen Fenster ein mit

```
<ENTER> = <ENTER>
```

Sie können Variablen und ihre Werte auch direkt in dieser Pseudo-Datei anlegen bzw. verändern.

### Anmerkung

Alle MAXed-Variablen werden automatisch in die Subshells des MAXed exportiert (siehe Abschnitt 4.5 Die SINIX-Schnittstellen).

### Beispiel 1

Sie merken sich den vollständigen Pfadnamen der aktuellen Datei in der Variablen *variable*: `<ENTER> variable=@f <ENTER>`

Sie holen sich die Pseudo-Datei EDITFILES mit einer Auswahlliste von Dateinamen ins aktuelle Fenster, wählen sich eine Datei aus und blenden diese im aktuellen Fenster mit `<EDIT>` ein (siehe Abschnitt 4.2 Die Pseudo-Dateien). Nach getaner Arbeit wollen Sie nochmals die früher bearbeitete Datei begutachten. Sie blenden sich diese im aktuellen Fenster ein (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien): `<ENTER> $variable <EDIT>`

### Beispiel 2

Sie wollen sich die linke obere Ecke des aktuellen Fensters merken, um das aktuelle Fenster später wieder an die momentane Position zu bewegen:

```
<ENTER> variable=@w <ENTER>
```

Damit speichern Sie die linke obere Fensterposition in der Variablen *variable*. Zu einem späteren Zeitpunkt positionieren Sie das aktuelle Fenster wieder an die gleiche Zeilen- und Spaltenposition (siehe Kapitel 6: HOME):

```
<ENTER> $variable <HOME>
```

**Beispiel 3**

Sie wollen eine eigene Konfigurationsdatei für den MAXed zu erstellen (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Das Bibliotheksverzeichnis des MAXed haben Sie in der Variablen MAXLIB abgespeichert:

```
MAXLIB=/usr/lib/MAXed
```

Sie holen sich die Quelldatei der Standard-Konfigurationsdatei ins aktuelle Fenster:

```
<ENTER> $MAXLIB/$TERM.cf1 <EDIT>
```

Sie kopieren sich den Inhalt dieser Datei in eine eigene Datei, indem Sie den Inhalt der aktuellen Datei unter anderem Namen abspeichern (siehe Kapitel 6: SAVE). Sie wollen die neue Datei in Ihrem HOME-Verzeichnis abspeichern. Der Name soll lediglich eine neue Endung bekommen: Ihre Benutzerkennung. Dazu greifen Sie auf Umgebungsvariablen zurück: HOME, TERM und USER. Um sicher zu gehen, testen Sie Ihre Namensgebung erst aus, indem Sie die Eingabe auf der Steuerzeile auswerten lassen (siehe unten: Entwertung und Auswertung auf der Steuerzeile):

```
<ENTER> = $HOME/$TERM$USER <ENTER>
```

In der Steuerzeile könnte z.B. folgendes angezeigt sein:

```
ARG: /usr/a1f/97801USER
```

Jedenfalls stellen Sie fest, daß die Endung (die Variable USER) von der Textersetzung offensichtlich unberührt blieb: Der MAXed erkannte das letzte Dollar-Zeichen als optionalen rechten Abschluß der Variablen TERM! Jetzt geben Sie das "richtige" Kommando ein:

```
<ENTER> $HOME/$TERM$$USER <SAVE>
```

Diese Datei können Sie gleich anschließend zur Bearbeitung im aktuellen Fenster einblenden lassen:

```
<ENTER> <BACKSPACE> <EDIT>
```

## Ersetzen von PICK-Puffernamen

PICK-Puffer dienen der Zwischenspeicherung beliebiger Textbereiche (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut). Sie definieren sie während der MAXed-Sitzung mit dem Kommando PICK oder als sogenannte Textbausteine in MACRO-Bibliotheken (siehe Kapitel 6: PICK, Abschnitt 4.6 Arbeiten mit Textbausteinen). PICK-Puffernamen können Sie überall im Eingabefeld der Steuerzeile durch den Inhalt des PICK-Puffers ersetzen lassen (siehe Abschnitt 3.8.3 Der PICK-Puffernamen als IDENTIFIER). Um diese Textersetzung auszulösen, stellen Sie einem PICK-Puffernamen ein Prozent-Zeichen % direkt voran. Durch einen vorausgehenden Gegenschrägstrich \ entwerten Sie jedoch das Sonderzeichen (siehe unten: Entwertung und Auswertung in der Steuerzeile).

Wenn Sie PICK-Puffernamen (IDENTIFIER) innerhalb von Zeichenketten-Attributen zur Textersetzung verwenden, schließen Sie sie aus Eindeutigkeitsgründen durch je ein weiteres, optionales Prozent-Zeichen nach rechts ab:

```
%[IDENTIFIER]  
bzw.  
[text] %[IDENTIFIER] %[text]
```

Wenn Sie keinen IDENTIFIER angeben, wird der Standard-PICK-Puffer verwendet (siehe Abschnitt 3.8.3 Der PICK-Puffernamen als IDENTIFIER).

### Anmerkung

Beim Verwenden eines PICK-Puffers innerhalb einer Kommandoattributierung ist jedoch zu beachten, daß der Inhalt des PICK-Puffers grundsätzlich als einzeliger STREAM ausgegeben wird (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut). Dies ist unabhängig vom Bereichstyp des PICK-Puffers, da Zeilenende-Zeichen bei der Ersetzung auf der Steuerzeile ausgelassen werden.

**Beispiel**

```
<ENTER>    puffer <PICK>
```

Das Wort an der aktuellen Position der Schreibmarke ist durch das Komma als Bereich definiert und wird in den PICK-Puffer *puffer* kopiert (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut, Kapitel 6: PICK).

```
<ENTER> %puffer%.c <EDIT>
```

Der MAXed versucht, die Datei mit dem Namen, der sich zusammensetzt aus dem Wort, das im Puffer *puffer* gespeichert ist, und der Endung *.c*, ins aktuelle Fenster zu holen (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien).

**Ersetzen von SINIX-Kommandoaufrufen**

Analog zur Vorgehensweise auf der Shell-Ebene wird vor Ausführen eines einbettenden MAXed-Kommandos eine in Gegenhochkommata stehende Zeichenkette als Aufruf eines SINIX-Kommandos interpretiert, das in einer Subshell ausgeführt wird. Die Zeichenkette inklusive Hochkommata wird durch die Ausgabe des SINIX-Kommandos ersetzt (siehe Abschnitt 4.5 Die SINIX-Schnittstellen). SINIX-Kommandoaufrufe können Sie überall im Eingabefeld der Steuerzeile durch ihre Standard-Ausgabe ersetzen lassen. Um diese Textersetzung auszulösen, stellen Sie einem SINIX-Kommandoaufruf ein Gegenhochkomma ` direkt voran. Durch einen direkt vorausgehenden Gegenschrägstrich \ entwerten Sie das Sonderzeichen (siehe unten: Entwertung und Auswertung in der Steuerzeile).

Wenn Sie SINIX-Kommandoaufrufe innerhalb von Zeichenketten-Attributen zur Textersetzung verwenden, schließen Sie sie aus Eindeutigkeitsgründen durch je ein weiteres, optionales Gegenhochkomma nach rechts ab:

```
[SINIX-kommando]
bzw.
[text]`[SINIX-kommando]`[text]
```

Wenn Sie keinen SINIX-Kommandoaufruf angeben, nimmt der MAXed an, daß in der aktuellen Zeile der Aufruf eines SINIX-Kommandos steht, das er ausführen soll, um seine Standard-Ausgabe an der Stelle des Gegenhochkommata (bzw. der beiden aufeinanderfolgenden Gegenhochkommata) in der Kommandoattributierung einzufügen.

### Anmerkung

Die Ausgabe besteht grundsätzlich aus einem einzeiligen STREAM (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut), da Zeilenende-Zeichen bei der Ersetzung auf der Steuerzeile ausgelassen werden.

### Beispiel

```
<ENTER> `echo $HOME`/* .c <EDIT>
```

Der MAXed läßt zunächst das SINIX-Kommando `echo $HOME` von der Shell in einem Subprozeß ausführen und ersetzt es durch seine Ausgabe: das *HOME*-Verzeichnis des Benutzers. Danach versucht der MAXed, alle Dateinamen aus dem *HOME*-Verzeichnis, die mit *.c* enden, ins aktuelle Fenster zu holen (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien).

Übrigens: Da die Shell-Variable *HOME* i.a. auch dem MAXed als Umgebungsvariable bekannt ist, läßt sich obiges MAXed-Kommando vereinfachen durch:

```
<ENTER> $HOME/* .c <EDIT>
```

### Entwertung und Auswertung im Eingabefeld

Beim Aufrufen eines MAXed-Kommandos, das mit Zeichenketten attribuiert ist, versucht der MAXed im Eingabefeld überall, wo es möglich ist, Textersetzungen vorzunehmen. Dabei werden alle auftretenden Parameter für Textersetzung, MAXed-Variablen, PICK-Puffernamen und SINIX-Kommandoaufrufe zusammen mit den zugehörigen Sonderzeichen von links nach rechts durch die Werte, Inhalte bzw. Ausgaben ersetzt.

Sie verhindern Textersetzungen im Eingabefeld der Steuerzeile, indem Sie die Sonderzeichen @, \$, ` und & durch einen vorausgehenden Gegenschrägstrich \ entwerten. Die entwerteten Sonderzeichen werden nach der Expandierung im Klartext ausgegeben.

### Anmerkung

Die Auswirkungen des *et*-Zeichen & auf den MAXed bleiben vorerst ein ungelüftetes Geheimnis. Die Verwendung des Zeichens im Eingabefeld ohne den entwertenden Gegenschrägstrich sei nur Abenteurern und Freunden von "[e]xtra-[t]errestrischen" Meta-Zeichen gestattet!

Sie haben die Möglichkeit, beliebige Zeichenketten im Eingabefeld der Steuerzeile auswerten und nach der Expandierung in der Steuerzeile anzeigen zu lassen. Dabei wird kein MAXed-Kommando ausgelöst, sodaß Sie diesbezüglich jedes Risiko vermeiden (siehe unten: Anmerkung). Sie lösen die Auswertung einer Zeichenkette im Eingabefeld aus mit

<ENTER> = beliebige zeichenkette(n) <ENTER>

Alle vorhandenen Parameter für Textersetzung, Namen von Variablennamen, PICK-Puffernamen und SINIX-Kommandoaufrufe werden zusammen mit den zugehörigen Sonderzeichen durch ihre Werte, Inhalte bzw. Ausgaben ersetzt. Die expandierte Zeichenkette wird danach auf der Steuerzeile angezeigt: **ARG:** expandierte zeichenkette(n)

Die expandierte Zeichenkette ist außerdem im Eingabepuffer gespeichert (siehe Abschnitt 3.6.2 Eingabefeld und Eingabepuffer). Sie können sie wieder abrufen mit <ENTER> <BACKSPACE>

**Anmerkung**

- Wenn alphanumerische (Teil-)Zeichenketten, die den nicht entwerteten Sonderzeichen \$, % oder ` nachfolgen bzw. von den Sonderzeichen \$, % oder ` eingeschlossen werden, nicht als Variablenname, PICK-Puffername bzw. SINIX-Kommandoaufruf definiert sind, so ersetzt sie der MAXed zusammen mit den voranstehenden bzw. einschließenden Sonderzeichen durch die leere Zeichenkette. Dies geschieht bei einzelnen und paarweisen Prozentzeichen bzw. Gegenhochkommata genauso, wenn der Standard-PICK-Puffer bzw. der SINIX-Kommandoaufruf der aktuellen Zeile nicht definiert ist.
- In Anbetracht des obigen Sachverhalts erscheint es sinnvoll, zur Auswertung im Anschluß an das am Anfang des Eingabefeldes eingegebene Gleichheitszeichen immer noch ein Leerzeichen einzutippen: Falls Sie dies unterlassen und die nachfolgende Zeichenkette durch die leere Zeichenkette ersetzt wird, blendet der MAXed aufgrund des vorliegenden Kommandoattributs, das Gleichheitszeichen die Pseudo-Datei VARIABLES im aktuellen Fenster ein (siehe Abschnitt 4.2 Die Pseudo-Dateien).
- Für die Wiederverwendung einer ausgewerteten Zeichenkette im Eingabepuffer als Kommandoattributierung ist es notwendig, das führende Gleichheitszeichen zu entfernen. Wenn Sie schon zur Auswertung das Gleichheitszeichen weglassen, kann durch das abschließende <ENTER> als unerwünschter Nebeneffekt ein MAXed-Kommando ausgelöst werden! (siehe Abschnitt 3.6.1 Das Format der Kommandoattributierung: Eine Alternative zum Abschicken von MAXed-Kommandos). Bei entsprechender Erfahrung werden Sie abschätzen können, ob dies eintreten kann. Dann steht Ihnen der Einsatz dieser "schnellen" Variante zur Auswertung und Wiederverwendung einer Zeichenkette frei. Eine etwaige Fehlermeldung des MAXed  
\*\*\*\* Keine Funktion fuer diese Eingabe.  
werden Sie gelassen zur Kenntnis nehmen.

**Beispiel 1**

<ENTER> = @f <ENTER>

Der Pfadname der aktuellen Datei wird im Eingabefeld ausgegeben und im Eingabepuffer gespeichert.

**Beispiel 2**

Tippen Sie folgenden Text in eine Zeile ihrer Datei:

Kennung:@u, HOME-Verzeichnis:\$HOME, aktuelle Datei:@f

Kopieren Sie diese Zeile in den Standard-PICK-Puffer mit  
<PICK>

Lassen Sie sich den Inhalt des Standard-PICK-Puffers auf der Steuerzeile ausgeben mit

<ENTER> = % <ENTER>

Die Zeichenkette ist dafür nun wirklich zu lang geworden. Sie setzen die Schreibmarke nochmals ins Eingabefeld und holen sich die letzte Eingabe aus dem Eingabepuffer zurück mit

<ENTER> <BACKSPACE>

Sie springen an den Anfang des Eingabefeldes mit

<UPARROW>

Sie müßten jetzt den Anfang folgender Zeichenkette sehen:

= Kennung:@u, HOME-Verzeichnis:\$HOME, aktuelle Datei:@f

Sie sehen, der Inhalt wurde nicht rekursiv weiter ersetzt. Sie erzwingen eine weitere Ersetzung der (zugegeben schon recht langen) Zeichenkette, indem Sie wieder <ENTER> betätigen:

Schauen Sie sich dies nochmals an mit <ENTER> <BACKSPACE>!

**Beispiel 3**

Sie legen sich eine neue Datei an:

```
<ENTER> hitchcock <EDIT> <EDIT>
```

Geben Sie folgendes Kommando ein:

```
<ENTER> 'aktuelle Datei:' <PUT> <RETURN>
```

```
<ENTER> '`pwd`\/@e' <PUT>
```

Bei der Auswertung der Zeichenkette im Eingabefeld wird eine Textersetzung durch die Entwertungen verhindert. Die Gegenschrägstriche haben ihre Schuldigkeit getan und werden eliminiert. In den beiden Zeilen müßte folgendes stehen:

```
aktuelle Datei:
```

```
`pwd`/@e
```

Sie setzen die Schreibmarke auf die erste dieser Zeilen und kopieren sich diese in den PICK-Puffer *pick\_puf1*:

```
<ENTER> pick_puf1 <PICK>
```

Die zweite Zeile kopieren Sie in den Standard-PICK-Puffer:

```
<DOWNARROW> <PICK>
```

Sie setzen die Schreibmarke auf eine der nächsten Zeilen und geben folgende Kommandos:

```
<ENTER> '%pick_puf%' <PUT>
```

```
<RETURN>
```

```
<ENTER> <BACKSPACE> <PUT>
```

Dies war nicht zweimal dasselbe Kommando, wie Sie sehen:

```
aktuelle Datei: `pwd`/@e
```

```
aktuelle Datei: /usr/wie_heiß_ich/hitchcock
```

### 3.7 Die Kommandoverstärkung

Einige MAXed-Kommandos können verstärkt werden, d.h. sie werden mehrmals hintereinander durch einen Kommandoaufruf ausgeführt. Als erstes Zeichenketten-Attribut geben Sie den Verstärkungsfaktor ein (siehe 3.6.1 Das Format der Kommandoattributierung). Das ist eine positive, ganze Zahl  $n$ , die angibt, wie oft das MAXed-Kommando hintereinander ausgeführt werden soll:

```
<ENTER> [n] [argument [argument ...]] <KOMMANDO>
```

Zu den Kommandos, die verstärkt werden können, gehören:

- alle Bewegungs-Kommandos außer GOTO, HOME, WORDLEFT, WORDRIGHT, UPEND, STARTLINE und ENDLIN (siehe Abschnitt 3.3.2 Bewegen der Schreibmarke, Abschnitt 3.3.3 Verschieben des Dateiausschnitts);
- einige weitere Kommandos, die nicht mit Bereichsdefinitionen, wie z.B. MACRO, PUT und RESTOR (siehe Abschnitte 3.9 Bereichsdefinition als Kommandoattribut, 6.1 Die MAXed-Kommandos).

#### Beispiel 1

```
<ENTER> 42 <RETURN>
```

Mit diesem Kommando bewegen Sie die Schreibmarke um 42 Zeilen in der eingeblendeten Datei weiter (siehe Kapitel 6: RETURN).

#### Beispiel 2

```
<ENTER> 42 makro <MACRO>
```

Sie lassen das Makro *makro* 42 mal ausführen. Der MAXed wird nur dann vorzeitig mit einer Fehlermeldung stoppen, wenn der Ausführung des soundsovielten Makros was im Wege stehen sollte. Jede einzelne Makroausführung beginnt in dem Sitzungszustand, in dem die jeweils vorausgehende endete (siehe Abschnitt 4.7 Arbeiten mit Makros, Kapitel 6: MACRO).

### 3.8 Der IDENTIFIER als Kommandoattribut

Ein IDENTIFIER ist ein Name, den Sie als Zeichenketten-Argument bei der Kommandoattributierung eingeben (siehe Abschnitt 3.6.1 Format der Kommandoattributierung). Der MAXed kennt IDENTIFIER für verschiedene Arten von Puffern:

- Dateinamen (siehe Abschnitt 3.8.1)
- ERASE-Puffername (siehe Abschnitt 3.8.2)
- PICK-Puffername (siehe Abschnitt 3.8.3)
- Markennamen (siehe Abschnitt 3.8.4)
- Makronamen (siehe Abschnitt 3.8.5)
- Namen von Steuerprogrammen (siehe Abschnitt 3.8.6)

Im wesentlichen bezeichnet ein IDENTIFIER einen Puffer, der Text oder andere Daten speichert. Ein Puffer wird vom MAXed im allgemeinen temporär gehalten, d.h. er existiert höchstens bis Sitzungsende. Dies gilt nicht für Benutzerdateien, die der MAXed im Dateisystem anlegt, und bestimmte Puffer, die Sie explizit in Dateien abspeichern können (siehe Abschnitte 3.1 Die Dateihaltung des MAXed, 4.6 Arbeiten mit Textbausteinen, 4.7 Arbeiten mit Makros).

#### Definition und Anwendung von IDENTIFIER

Sie können während der MAXed-Sitzung Puffer definieren, um Texte oder andere Daten zu speichern, und später wieder ihren Inhalt abrufen. Bestimmte Puffer können Sie über das Sitzungsende hinaus abspeichern oder explizit in Quelldateien erstellen und übersetzen lassen. Diese Puffer definieren Sie für eine andere MAXed-Sitzung von neuem, indem Sie sie nach Bedarf laden (siehe Abschnitte 4.6 Arbeiten mit Textbausteinen, 4.7 Arbeiten mit Makros).

Einen Puffer erstellen Sie mit dem zur Art des Puffers passenden MAXed-Kommando, d.h. Sie speichern unter einem IDENTIFIER den Puffer-Inhalt ab:

```
<ENTER> ... IDENTIFIER ... <ERSTELLEN-KOMMANDO>
```

Einen existierenden Puffer verwenden Sie wieder durch das zur Art des Puffers passende MAXed-Kommando:

<ENTER> ... IDENTIFIER ... <ANWENDEN-KOMMANDO>

Für die verschiedenartigen IDENTIFIER stehen folgende Kommandos zur Verfügung:

IDENTIFIER	<ERSTELLEN-KOMMANDO>	<ANWENDEN-KOMMANDO>
Dateiname	<EDIT> <EDIT>	<EDIT> bzw. <PUT>
ERASE-Puffername	<ERASE>	<RESTOR>
PICK-Puffername	<PICK>	<PUT>
Markenname	<LABEL>	<GOTO>
Makroname	<LEARN>	<MACRO>
Steuerdateiname	—	<MACRO>

Einige vordefinierte IDENTIFIER bestehen aus Sonderzeichen. Wenn Sie einen neuen IDENTIFIER definieren, gelten für die Namensvergabe – mit Ausnahme der Dateinamen – folgende Regeln:

- Ein IDENTIFIER muß aus einem Wort bestehen, d.h. eine zusammenhängende Zeichenkette, die sich aus einem oder mehreren alphanumerischen Zeichen und – als einzigem nicht-alphanumerischem Zeichen – dem Unterstrich \_ zusammensetzt.
- Das erste Zeichen eines IDENTIFIER muß ein Buchstabe oder ein Unterstrich sein.
- Der MAXed unterscheidet dabei nicht zwischen Groß- und Kleinbuchstaben.

Wenn Sie bei einer Puffer-Definition einen IDENTIFIER verwenden, der gleichlautend für einen gleichartigen Puffer bereits existiert, wird dieser dadurch überschrieben.

Es können jedoch gleichzeitig verschiedenartige IDENTIFIER mit gleichlautenden Namen existieren, da jeder Puffer-Art ein spezieller Satz von Kommandos zugeordnet ist. Lediglich bei den Kommandos MACRO bzw. PUT kann es wegen gleichnamigen Makros und Steuerdateien bzw. PICK-Puffern und Dateien zu Überschneidungen kommen. In diesen Fällen haben Makronamen Vorrang vor gleichnamigen Steuerprogrammen und PICK-Puffernamen Vorrang vor gleichnamigen Dateinamen.

Zu allen Arten von IDENTIFIER außer den Steuerdateinamen hält sich der MAXed eine eigene Liste mit allen in der aktuellen Sitzung geöffneten bzw. definierten IDENTIFIER. Sie können sich jede dieser Listen in speziellen, sogenannten Pseudo-Dateien einblenden lassen (siehe Abschnitt 4.2 Die Pseudo-Dateien):

<ENTER> ® <ANWENDEN-KOMMANDO>

Sie können sich hier durch Positionieren der Schreibmarke und nochmaliges Betätigen der zugehörigen Kommandotaste <ANWENDEN-KOMMANDO> einen IDENTIFIER zur sofortigen Anwendung auswählen.

In den folgenden Abschnitten sind die verschiedenen Arten von IDENTIFIER im Zusammenhang mit den MAXed-Kommandos, mit denen Sie sie definieren bzw. wiederverwenden können, beschrieben.

### 3.8.1 Der Dateiname als IDENTIFIER

Der MAXed arbeitet auf dem Dateisystem seiner Aufrufumgebung (siehe Abschnitte 3.1 Die Dateihaltung des MAXed, 3.3 Arbeiten mit der aktuellen Datei, 3.4 Arbeiten mit mehreren Dateien, Kapitel 6: EDIT).

Während der MAXed-Sitzung erstellen Sie eine neue Datei mit

```
<ENTER> dateiname <EDIT><EDIT>
```

Sie können eine Datei, die bestimmte Sonderzeichen enthält, nicht anlegen (siehe unten: Sonderzeichen der IDENTIFIER für Dateien). Ein IDENTIFIER mit solchen Sonderzeichen erzwingt einen Auswahlversuch des MAXed aus den bereits existierenden Dateinamen des betroffenen Dateiverzeichnisses. Außerdem sind die Sonderzeichen zur Textersetzung im Eingabefeld gültig (siehe Abschnitt 3.6.4).

Sie holen sich eine existierende Datei ins aktuelle Fenster mit

```
<ENTER> dateiname <EDIT>
```

Die Alternativ-Datei:

Der MAXed ordnet jedem Fenster – wenn möglich – außer der aktuell eingeblendeten Datei noch eine Alternativ-Datei zu. Das ist jene Datei, die zuvor im jeweiligen Fenster eingeblendet war. Das kann ggf. auch eine Pseudo-Datei sein (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie holen die Alternativ-Datei mit <EDIT> ins aktuelle Fenster zurück (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern).

#### *Anmerkung*

Sie können auch ein und dieselbe Datei gleichzeitig als aktuelle Datei und als Alternativ-Datei halten. Dazu geben Sie beispielsweise ein:

```
<ENTER> @f <EDIT>
```

Die aktuelle Datei wird im aktuellen Fenster nochmals eingeblendet und wird damit gleichzeitig Alternativ-Datei. Mit <EDIT> springen Sie jetzt zwischen zwei "alternativen" aktuellen Schreibmarkenpositionen in einer Datei hin und her.

Bei der Eingabe eines Dateinamens als Kommandoattribut gelten folgende grundsätzliche Regeln:

- Wenn Sie eine neue Datei anlegen, müssen Sie bei der Namensvergabe die SINIX-Konventionen für Dateinamen beachten. Hierbei können Sie die nachstehend aufgeführten Sonderzeichen für Dateinamen nicht verwenden.
- Sie können während der MAXed-Sitzung alle Dateien des Dateisystems adressieren, wenn Sie den absoluten Pfadnamen als Dateinamen verwenden. Relative Pfadnamen beziehen sich auf das aktuelle Dateiverzeichnis, also im allgemeinen jenes, von dem aus Sie den MAXed aufgerufen haben (siehe Abschnitt 5.1 Den MAXed aufrufen, Kapitel 6: DEFDIR).

### Sonderzeichen der IDENTIFIER für Dateien

Der MAXed bietet darüber hinaus die Möglichkeit, mit Auswahllisten für Dateinamen zu arbeiten, die entweder automatisch angelegt werden oder die Sie durch Angabe eines geeigneten IDENTIFIERS selbst erstellen:

```
<ENTER> IDENTIFIER <EDIT>
```

Die Auswahllisten werden in sogenannte Pseudo-Dateien eingelesen, die Ihnen für die Dauer der Sitzung zur Verfügung stehen (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie geben auf indirekte Weise die gewünschte Datei an, indem Sie eine dieser Pseudo-Dateien aufrufen bzw. erstellen und durch Positionieren der Schreibmarke mit anschließendem <EDIT> auswählen. Ein IDENTIFIER für Dateien kann folgende Sonderzeichen enthalten:

- Sie können mehrere, bereits existierende Dateien auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck für Dateinamen entsprechend den SINIX-Konventionen eingeben. Innerhalb dieser regulären Ausdrücke können Sie daher folgende Sonderzeichen verwenden:
  - ? steht für ein beliebiges einzelnes Zeichen aus der Menge von Zeichen für Dateinamen laut SINIX-Konventionen.
  - \* steht für eine Teil-Zeichenkette mit beliebiger Kombination beliebig vieler (auch null) Zeichen.

- [xyz] steht für genau eines der Zeichen aus der in den eckigen Klammern [ und ] angegebenen Menge xyz.
- [!xyz] steht für genau ein Zeichen, das jedoch nicht in der angegebenen Menge xyz enthalten sein darf.
- [a-z] steht für genau eines der Zeichen aus dem Ausschnitt a bis z des ASCII-Zeichensatzes.

Wenn genau ein existierender Dateiname zu einem eingegebenen regulären Ausdruck paßt, holt der MAXed diese Datei ins aktuelle Fenster. Wenn mehrere Dateinamen passen, erstellt der MAXed eine Liste dieser Dateien und schreibt diese Liste in die Pseudo-Datei EDITFILES (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste eine der Dateien auswählen.

#### *Ausnahme*

Einige einzelne Sonderzeichen haben für das Kommando EDIT Sonderbedeutungen als Kommandoattribute, die die o.g. regulären Ausdrücke für Dateinamen überlagern (siehe Kapitel 6: EDIT).

#### *Beispiel*

```
<ENTER> verz/*.c <EDIT>
```

Der MAXed erzeugt die Liste aller Dateinamen aus dem Unterverzeichnis *verz*, die mit *.c* enden, schreibt sie in die Pseudo-Datei EDITFILES und holt diese ins aktuelle Fenster. Wenn genau ein Dateiname paßt, blendet er gleich diese Datei ein.

- Sie können mehrere Dateinamen gleichzeitig angeben, indem Sie sie durch jeweils ein Komma (ohne Leerzeichen!) trennen. Der MAXed erstellt die Liste aller passenden Dateinamen und präsentiert diese in der Pseudo-Datei EDITFILES im aktuellen Fenster.

#### *Beispiel*

```
<ENTER> datei1,datei2,.* <EDIT>
```

Der MAXed erzeugt die Liste aus den Dateien *datei1* und *datei2* sowie aller unsichtbaren Dateinamen aus dem aktuellen Dateiverzeichnis, also jene, die mit Punkt *.* beginnen.

- Sie können den Namen eines Dateiverzeichnisses eingeben. Hierbei gelten neben den oben angegebenen regulären Ausdrücken noch weitere Sonderzeichen für vordefinierte Dateiverzeichnisnamen:

- steht für das aktuelle Dateiverzeichnis.
- .. steht für das Dateiverzeichnis, das dem aktuellen übergeordnet ist.

Der MAXed erstellt die Liste aller sichtbaren Dateinamen des angegebenen Dateiverzeichnisses und präsentiert diese in der Pseudo-Datei EDITFILES im aktuellen Fenster.

### *Beispiel*

```
<ENTER> ../verz <EDIT>
```

Der MAXed erzeugt die Liste aller (sichtbaren) Dateinamen des "Geschwister-Verzeichnisses" ../verz und liest diese in die Pseudo-Datei EDITFILES ein.

- Der MAXed erstellt automatisch eine Liste aller im Verlauf der Sitzung geöffneten Dateien. Der IDENTIFIER
  - Ⓜ steht beim Kommando EDIT für die Pseudo-Datei OPENFILES mit der Liste aller von Ihnen im Verlauf der Sitzung geöffneten Dateien und weiterer momentan angelegten Pseudo-Dateien (siehe Abschnitt 4.2.3 Die Pseudo-Datei OPENFILES).

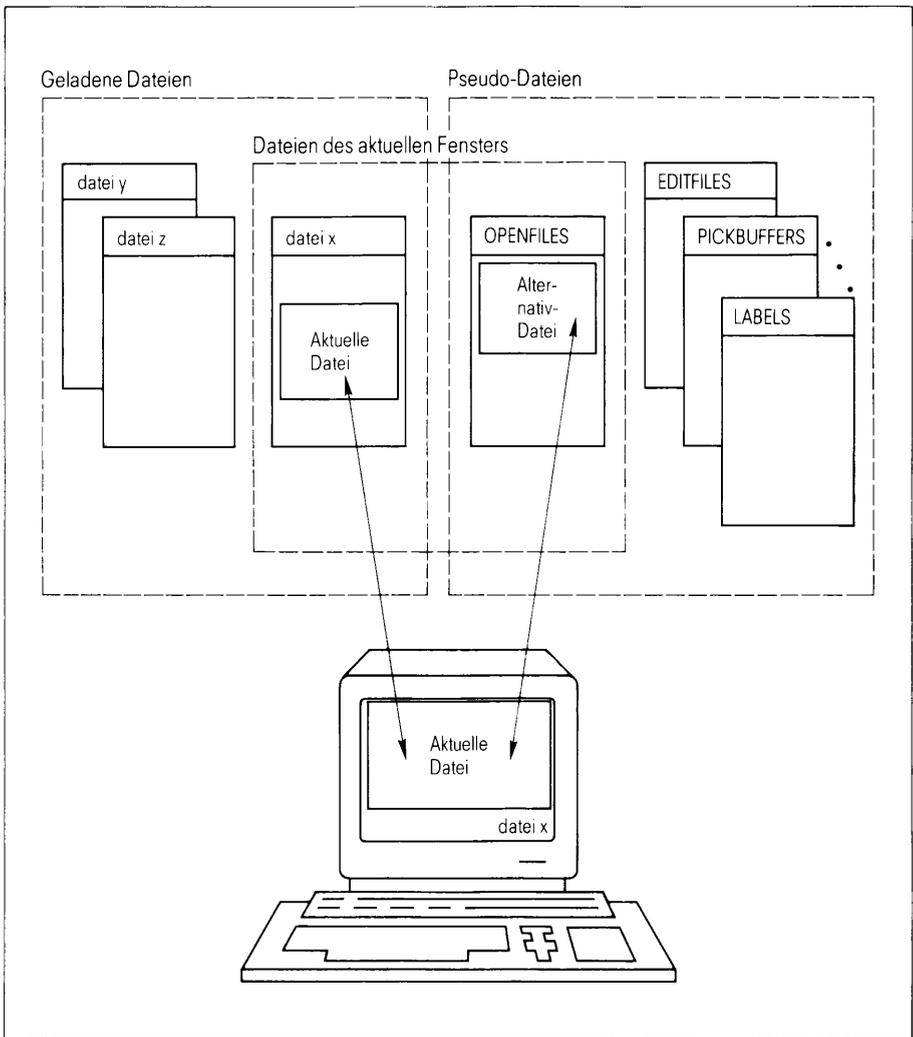


Bild 3-15 Geöffnete Dateien und Pseudo-Dateien

Bild 3-15 stellt beispielhaft dar, wie der MAXed in einer Sitzung verschiedene geladene Dateien und Pseudo-Dateien gleichzeitig nebeneinander hält. In der abgebildeten "Momentaufnahme" ist die Datei x im aktuellen Fenster eingeblendet, nachdem sie aus der Pseudo-Datei OPENFILES ausgewählt wurde. Diese Pseudo-Datei ist demnach gerade die Alternativ-Datei.

### Anmerkung

Bei Angabe eines Dateiverzeichnisnamens als IDENTIFIER werden folgende Dateinamen dieses Verzeichnisses in die zugehörige Pseudo-Datei eingelesen:

- der Name des übergeordneten Dateiverzeichnisses
- die sichtbaren Namen der gewöhnlichen Dateien
- die Namen der Unterverzeichnisse

Sollten Sie ein Dateiverzeichnis aus der Liste auswählen, wird für dieses erneut eine Auswahlliste erstellt, die den bisherigen Inhalt der Pseudo-Datei ersetzt. Auf diese Weise können Sie sich über mehrere Dateiverzeichnisse "durchwählen", bis Sie die gewünschte Datei finden.

### Kopieren eines gesamten Dateiinhalts

Sie können außerdem den gesamten Inhalt einer Datei an die aktuelle Schreibmarkenposition in der aktuellen Datei schreiben. Je nach Einfüge-Modus wird dieser Text eingefügt oder der darunterliegende Text ab der aktuellen Schreibmarkenposition überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Sie schreiben den Inhalt des mit IDENTIFIER bezeichneten PICK-Puffers oder der mit IDENTIFIER bezeichneten Datei an die aktuelle Schreibmarkenposition mit

```
<ENTER> IDENTIFIER <PUT>
```

Falls ein PICK-Puffername zu IDENTIFIER paßt, wird der Inhalt dieses Puffers an die aktuelle Schreibmarkenposition geschrieben bzw. die Pseudo-Datei PICKBUFFERS zuvor mit den passenden PICK-Puffernamen im aktuellen Fenster präsentiert (siehe Abschnitt 4.2 Die Pseudo-Dateien).

Wenn kein PICK-Puffername zu IDENTIFIER paßt, sucht der MAXed nach einem passenden Dateinamen.

Unter der Voraussetzung, daß sich kein PICK-Puffer angesprochen fühlt, können Sie mehrere existierende Dateien oder Dateiverzeichnisse auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck mit den oben aufgeführten Sonderzeichen eingeben. Wenn genau ein existierender Dateiname zu IDENTIFIER paßt, holt der MAXed diese Datei ins aktuelle Fenster. Wenn mehrere Dateinamen oder Dateiverzeichnisnamen passen, erstellt der MAXed eine Liste dieser Dateien und schreibt diese Liste in die Pseudo-Datei PUTFILES (siehe Abschnitt 4.2.5). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste eine der Dateien auswählen.

### 3.8.2 Der ERASE-Puffername als IDENTIFIER

ERASE-Puffer dienen dem Verschieben von Textbereichen (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut, Kapitel 6: ERASE). Ein gelöschter Textbereich wird in einem ERASE-Puffer gespeichert, sodaß Sie ihn später wieder an beliebiger Stelle in ihren bearbeiteten Dateien einfügen können (siehe Kapitel 6: RESTOR).

Sie löschen einen Textbereich und speichern ihn gleichzeitig in dem benannten ERASE-Puffer *erase\_puf* mit

```
<ENTER> [bereich] erase_puf <ERASE>
```

Sie kopieren den Inhalt des ERASE-Puffers *erase\_puf* an die aktuelle Schreibmarkenposition in der aktuellen Datei mit

```
<ENTER> erase_puf <RESTOR>
```

#### Anmerkung

Der Inhalt eines ERASE-Puffers wird abhängig vom eingestellten Eingabe-Modus in vorhandenem Text eingefügt, oder er überschreibt diesen (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Dabei ist der Bereichstyp des Pufferinhaltes dafür ausschlaggebend, welchen "Raum" der kopierte Text an der aktuellen Dateiposition einnimmt (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut: Bedeutung des Bereichstyps beim Kopieren).

**Beispiel**

<ENTER> erase\_puf <ERASE>

Damit löschen Sie die aktuelle Zeile und speichern sie gleichzeitig im ERASE-Puffer *erase\_puf*.

<ENTER> erase\_puf <RESTOR>

Sie fügen den Inhalt des ERASE-Puffers *erase\_puf* (die zuvor gelöschte Zeile) vor der aktuellen Zeile ein. Voraussetzung ist: als Eingabe-Modus ist ein Einfüge-Modus (z.B. INSERT) eingestellt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Falls der Überschreibe-Modus OVERWRITE eingestellt ist, wird die aktuelle Zeile durch die im ERASE-Puffer gespeicherte Zeile überschrieben.

**Der Standard-ERASE-Puffer:**

Neben den benannten ERASE-Puffern gibt es noch einen unbenannten, den Standard-ERASE-Puffer. Wenn Sie einen Textbereich löschen, wird dieser immer im Standard-ERASE-Puffer gespeichert, ob Sie nun einen IDENTIFIER (ERASE-Puffername) angeben oder nicht. Dadurch können Sie direkt im Anschluß an das Löschen eines Textbereichs in einen benannten ERASE-Puffer den Textbereich auch ohne Angabe des ERASE-Puffernamens wieder in den Text zu kopieren (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern).

**Beispiel**

<ENTER> <ERASE>

Sie löschen in der aktuellen Zeile den Zeilenrest von der aktuellen Position der Schreibmarke bis zum Zeilenende ohne das unsichtbare Zeilenende-Zeichen und speichern diesen Zeilenrest gleichzeitig im Standard-ERASE-Puffer. Falls die Schreibmarke hinter dem letzten Zeichen der aktuellen Zeile steht, wird das Zeilenende-Zeichen gelöscht und die nächste Zeile am Ende der aktuellen Zeile angehängt.

<RESTOR>

Sie fügen den Inhalt des Standard-ERASE-Puffers, den zuvor gelöschten Zeilenrest vor der aktuellen Spalte der Schreibmarke in der aktuellen Zeile ein. Voraussetzung ist: als Eingabe-Modus ist ein Einfüge-Modus (z.B. INSERT) eingestellt und Sie haben seit dem obigen ERASE-Kommando kein weiteres Löschkommando aufgerufen.

Bei der Eingabe eines ERASE-Puffernamens als Kommandoattribut gelten folgende grundsätzliche Regeln:

- Wenn Sie einen benannten Puffer anlegen, können Sie als Puffername eine beliebige Zeichenkette eingeben, die aus einem oder mehreren alphanumerischen Zeichen sowie als einzigem nicht-alphanumerischem Zeichen dem Unterstrich `_` bestehen kann und mit einem Buchstaben oder einem Unterstrich beginnt. Dabei unterscheidet der MAXed nicht zwischen Groß- und Kleinbuchstaben.
- Der MAXed legt den `_ERASE`-Puffer automatisch an, sobald Sie einmal das nicht-attributierte Kommando `<ERASE>` gegeben haben.

Der IDENTIFIER

- steht für den `_ERASE`-Puffer. Er hat die Funktion eines Kurzzeitgedächtnisses. Alle Textbereiche, die Sie durch das Kommando ERASE löschen und nur in den Standard-ERASE-Puffer kopieren, werden auch im `_ERASE`-Puffer gespeichert. Die gelöschten Textbereiche überschreiben dabei jedoch nicht jedesmal den bisherigen Pufferinhalt, sondern werden ans Ende des Puffers angehängt. Bei Verwenden dieses ERASE-Puffers kopieren Sie also die bisher durch `<ERASE>` gesammelten Textbereiche an die aktuelle Position der Schreibmarke in der aktuellen Datei.

Der `_ERASE`-Puffer sammelt so lange, bis Sie seinen Inhalt in den Text kopieren. Bei der nächsten Eingabe von `<ERASE>` wird der bisherige Inhalt gelöscht und die Sammlung beginnt von neuem.

*Beispiel*

Sie wollen 42 verstreute Zeilen löschen und gesammelt an einer anderen Textstelle wieder einfügen. Zunächst löschen Sie den `_ERASE`-Puffer in der Pseudo-Datei `ERASEBUFFERS` (siehe unten):

```
<ENTER> @ <RESTOR>
```

Die Schreibmarke positionieren Sie in der eingeblendeten Pseudo-Datei `ERASEBUFFERS` auf den Namen des `_ERASE`-Puffers, falls schon vorhanden, und löschen den Eintrag:

```
<ERASE>
```

Mit `<EDIT>` erhalten Sie wieder die Alternativ-Datei. Nun löschen Sie die 42 Textzeilen durch 42 `<ERASE>`-Kommandos. Zur Sicherheit kopieren Sie den Inhalt des `_ERASE`-Puffers in einen anderen benannten `ERASE`-Puffer (siehe unten: Kopieren von Pufferinhalten zwischen `ERASE`-Puffern):

```
<ENTER> _ > erase_puf <ERASE>
```

Sie fügen die Zeilen an anderer Stelle wieder ein:

```
<ENTER> _ <RESTOR>
```

oder

```
<ENTER> erase_puf <RESTOR>
```

### Sonderzeichen der IDENTIFIER für ERASE-Puffer

Der MAXed bietet darüber hinaus die Möglichkeit, mit Auswahllisten für ERASE-Puffernamen zu arbeiten, die entweder automatisch angelegt werden oder die Sie durch Angabe eines geeigneten IDENTIFIERS selbst erstellen:

<ENTER> IDENTIFIER <ERASE>

Die Auswahllisten werden in sogenannte Pseudo-Dateien eingelesen, die Ihnen für die Dauer der Sitzung zur Verfügung steht (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie geben auf indirekte Weise den gewünschten ERASE-Puffer an, indem Sie diese Pseudo-Datei aufrufen bzw. erstellen und durch Positionieren der Schreibmarke mit anschließendem <RESTOR> auswählen. Ein IDENTIFIER für ERASE-Puffer kann folgende Sonderzeichen enthalten:

– Sie können mehrere, bereits existierende ERASE-Puffer auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck mit folgenden Sonderzeichen eingeben:

? steht für ein beliebiges einzelnes Zeichen aus der Menge von Zeichen für IDENTIFIER.

\* steht für eine (Teil-)Zeichenkette mit beliebiger Kombination beliebig vieler (auch null) Zeichen.

[xyz] steht für genau eines der Zeichen aus der in den eckigen Klammern [ und ] angegebenen Menge xyz.

Wenn genau ein existierender ERASE-Puffernamen zu einem eingegebenen regulären Ausdruck paßt, kopiert der MAXed den Inhalt dieses ERASE-Puffers an die aktuelle Position der Schreibmarke im aktuellen Fenster. Wenn mehrere ERASE-Puffernamen passen, erstellt der MAXed eine Liste dieser IDENTIFIER und schreibt diese Liste in die Pseudo-Datei ERASEBUFFERS (siehe Abschnitt 4.2.6). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste einen der IDENTIFIER auswählen.

*Beispiel*

```
<ENTER> bereich1 p1 <ERASE>  
<ENTER> bereich2 p2 <ERASE>
```

Sie löschen damit zwei Textbereiche und speichern sie in den beiden benannten ERASE-Puffern *p1* und *p2*.

```
<ENTER> p* <RESTOR>
```

Der MAXed erzeugt die Liste aller ERASE-Puffernamen, die mit *p* beginnen (zumindest also *p1* und *p2*), schreibt sie in die Pseudo-Datei ERASEBUFFERS und blendet diese im aktuellen Fenster ein.

- Der MAXed erstellt automatisch eine Liste aller im Verlauf der Sitzung definierten benannten ERASE-Puffer. Der IDENTIFIER
  - Ⓜ steht beim Kommando RESTOR für die Pseudo-Datei ERASEBUFFERS mit der Liste aller von Ihnen im Verlauf der Sitzung definierten ERASE-Puffernamen und eventuell dem \_ für den automatisch angelegten \_-ERASE-Puffer (siehe Abschnitt 4.2 Die Pseudo-Dateien).

Neben dem \_-ERASE-Puffer hält sich der MAXed eine Pseudo-Datei namens RESTORE, die gelöschte Textbereiche sammelt (siehe Abschnitt 4.2.2 Die Pseudo-Datei RESTORE). Sie holen sich die Pseudo-Datei RESTORE ins aktuelle Fenster mit

```
<ENTER> <EDIT>
```

Diese Pseudo-Datei hat die Funktion eines Langzeitgedächtnisses, u.a. für alle jemals im Standard-ERASE-Puffer und im Standard-PICK-Puffer gespeicherten Textbereiche (siehe Abschnitt 3.8.3 Der PICK-Puffer als IDENTIFIER). Während der Sitzung können Sie sich aus dieser Datei z.B. wieder alle Texte kopieren, die Ihnen durch textverändernde Kommandos verloren gingen.

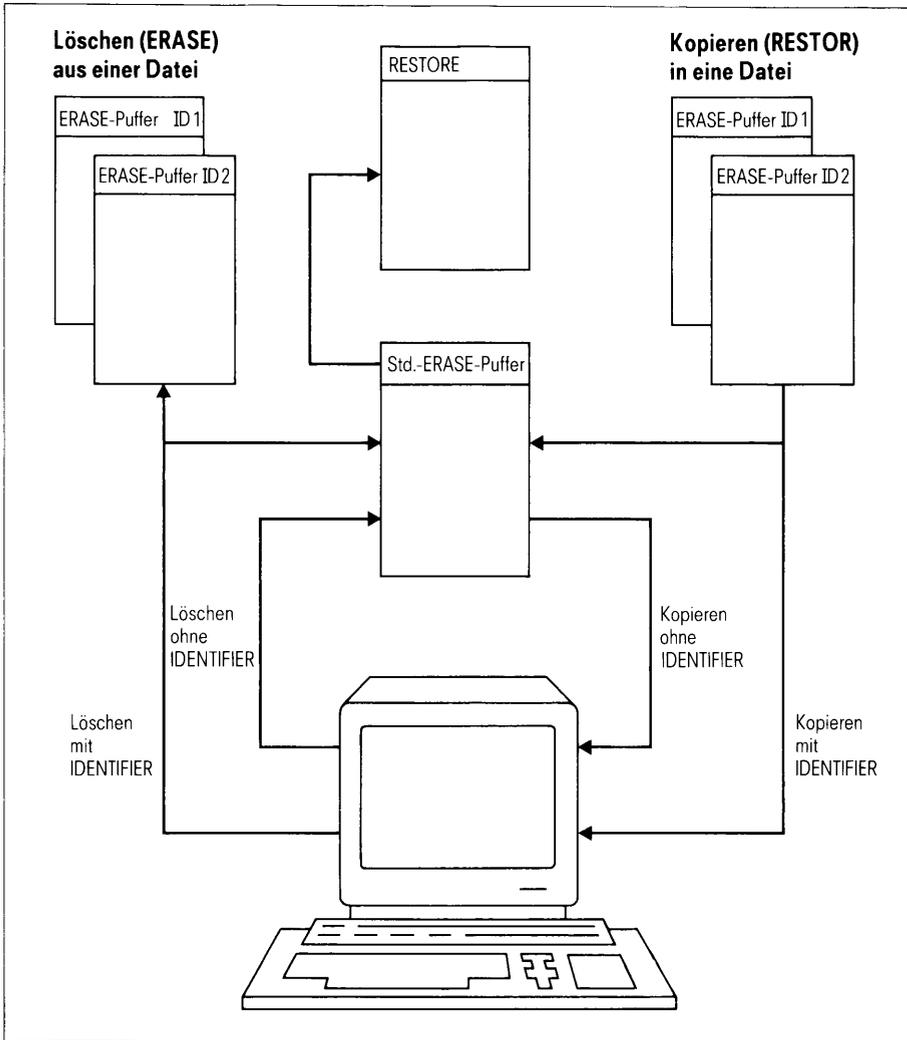


Bild 3-16 ERASE-Puffer und Pseudo-Datei RESTORE

Bild 3-16 veranschaulicht die Zusammenhänge verschiedener Puffer beim Löschen von Textbereichen und Wiederverwenden gelöschter Textbereiche: Der Standard-ERASE-Puffer und die Pseudo-Datei RESTORE werden grundsätzlich bei jedem Lösch- und Kopierkommando gefüttert. Die benannten ERASE-Puffer werden nur dann verwendet, wenn Sie explizit durch ihren IDENTIFIER angesprochen werden. Der `_ERASE`-Puffer dagegen wird ausschließlich mit jenem Textmaterial beladen, das beim Löschen nur im Standard-ERASE-Puffer, nicht aber in einen weiteren benannten ERASE-Puffer landete.

### Kopieren von Pufferinhalten zwischen ERASE-Puffern

Sie haben die Möglichkeit, den gespeicherten Textbereiche eines beliebigen ERASE-Puffers direkt in einen anderen ERASE-Puffer zu kopieren (siehe Kapitel 6: ERASE).

Sie kopieren den Inhalt des ERASE-Puffers `erase_puf1` in den ERASE-Puffer `erase_puf2` mit

```
<ENTER> [erase_puf1] > [erase_puf2] <ERASE>
```

Dabei wird der bisherige Inhalt von `erase_puf2` überschrieben!

Sie hängen eine Kopie des Inhalts von `erase_puf1` ans Ende des bisherigen Inhalts von `erase_puf2` mit

```
<ENTER> [erase_puf1] >> [erase_puf2] <ERASE>
```

### Anmerkung

Wie die Optionsklammern andeuten, können Sie die IDENTIFIER auch weglassen. Wenn kein Puffername angegeben ist, greift der MAXed auf den Standard-ERASE-Puffer zu.

### Austauschen zweier Textbereiche

Durch Kombination der MAXed-Kommandos ERASE und EXCHANGE tauschen Sie zwei Textbereiche in beliebigen Dateien aus (siehe Kapitel 6: EXCHANGE).

Sie löschen zunächst den einen Textbereich *bereich1* mit

```
<ERASE>  
bzw.  
<ENTER> bereich1 <ERASE>
```

Der gelöschte Textbereich ist im Standard-ERASE-Puffer zwischengespeichert. Sie tauschen den anderen Textbereich *bereich2* mit dem Inhalt des Standard-ERASE-Puffers, also *bereich1*, aus und fügen gleichzeitig *bereich2* an der Textstelle ein, an der Sie *bereich1* gelöscht haben:

```
<EXCHANGE>  
bzw.  
<ENTER> bereich2 <EXCHANGE>
```

### 3.8.3 Der PICK-Puffernamen als IDENTIFIER

PICK-Puffer dienen dem Kopieren von Textbereichen (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut, Kapitel 6: PICK). Ein kopierter Textbereich wird in einem PICK-Puffer gespeichert, sodaß Sie ihn später an beliebiger Stelle in ihren bearbeiteten Dateien einfügen können (siehe Kapitel 6: PUT).

Sie kopieren einen Textbereich in dem benannten PICK-Puffer *pick\_puf* mit

```
<ENTER> [bereich] pick_puf <PICK>
```

Sie kopieren den Inhalt des PICK-Puffers *pick\_puf* an die aktuelle Position der Schreibmarke in der aktuellen Datei mit

```
<ENTER> pick_puf <PUT>
```

#### Anmerkung

Der Inhalt eines PICK-Puffers wird abhängig vom eingestellten Eingabe-Modus in vorhandenem Text eingefügt, oder er überschreibt diesen (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Dabei ist der Bereichstyp des Pufferinhaltes dafür ausschlaggebend, welchen "Raum" der kopierte Text an der aktuellen Dateiposition einnimmt (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut: Bedeutung des Bereichstyps beim Kopieren).

#### Beispiel

```
<ENTER> pick_puf <PICK>
```

Damit kopieren Sie die aktuelle Zeile in den PICK-Puffer *pick\_puf*.

```
<ENTER> pick_puf <PUT>
```

Sie fügen den Inhalt des PICK-Puffers *pick\_puf* (die zuvor kopierte Zeile) vor der aktuellen Zeile ein. Voraussetzung ist: als Eingabe-Modus ist ein Einfüge-Modus (z.B. INSERT) eingestellt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Falls der Überschreibe-Modus OVERWRITE eingestellt ist, wird die aktuelle Zeile durch die im PICK-Puffer gespeicherte Zeile überschrieben.

Der Standard-PICK-Puffer:

Neben den benannten PICK-Puffern gibt es noch einen unbenannten, den Standard-PICK-Puffer. Wenn Sie einen Textbereich kopieren, wird dieser immer im Standard-PICK-Puffer gespeichert, ob Sie nun einen IDENTIFIER (PICK-Puffername) angeben oder nicht. Dadurch können Sie direkt im Anschluß an das Kopieren eines Textbereichs in einen benannten PICK-Puffer den Textbereich ohne Angabe des PICK-Puffernamens wieder in den Text kopieren (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern)

*Beispiel*

<ENTER> <PICK>

Sie kopieren den Rest der aktuellen Zeile von der aktuellen Position der Schreibmarke bis zum Zeilenende ohne das unsichtbare Zeilenende-Zeichen in den Standard-PICK-Puffer. Falls die Schreibmarke hinter dem letzten Zeichen der aktuellen Zeile steht, wird nur das Zeilenende-Zeichen in den Puffer kopiert.

<PUT>

Sie fügen den Inhalt des Standard-PICK-Puffers den zuvor kopierten Zeilenrest vor der aktuellen Spalte der Schreibmarke in der aktuellen Zeile ein. Voraussetzung ist: als Eingabe-Modus ist ein Einfüge-Modus (z.B. INSERT) eingestellt und Sie haben seit dem obigen PICK-Kommando kein weiteres Kopierkommando aufgerufen.

Bei der Eingabe von PICK-Puffernamen als Kommandoattribut gelten folgende grundsätzliche Regeln:

- Wenn Sie einen benannten Puffer anlegen, können Sie als Puffername eine beliebige Zeichenkette eingeben, die aus einem oder mehreren alphanumerischen Zeichen sowie als einzigem nicht-alphanumerischem Zeichen dem Unterstrich \_ bestehen kann und mit einem Buchstaben oder einem Unterstrich beginnt. Dabei unterscheidet der MAXed nicht zwischen Groß- und Kleinbuchstaben.

- Der MAXed legt den `_PICK`-Puffer automatisch an, sobald Sie einmal das nicht-attributierte Kommando `<PICK>` gegeben haben.

Der IDENTIFIER

- steht für den `_PICK`-Puffer. Er hat die Funktion eines Kurzzeitgedächtnisses. Alle Textbereiche, die Sie durch das Kommando `PICK` nur in den Standard-`PICK`-Puffer kopieren, werden auch im `_PICK`-Puffer gespeichert. Die kopierten Textbereiche überschreiben dabei jedoch nicht jedesmal den bisherigen Pufferinhalt, sondern werden ans Ende des Puffers angehängt. Bei Verwenden dieses `PICK`-Puffers kopieren Sie also die bisher durch `<PICK>` gesammelten Textbereiche an die aktuelle Position der Schreibmarke in der aktuellen Datei.

Der `_PICK`-Puffer sammelt so lange, bis Sie seinen Inhalt in den Text kopieren. Bei der nächsten Eingabe von `<PICK>` wird der bisherige Inhalt gelöscht und die Sammlung beginnt von neuem.

### Beispiel

Sie wollen diesmal die mysteriösen zweiundvierzig verstreuten Zeilen nur kopieren und gesammelt an einer anderen Textstelle wieder einfügen. Zunächst löschen Sie in der Pseudo-Datei `PICKBUFFERS` (siehe unten) den `_PICK`-Puffer:

```
<ENTER> @ <PUT>
```

Die Schreibmarke positionieren Sie in der eingeblendeten Pseudo-Datei `PICKBUFFERS` auf den Namen des `_PICK`-Puffers, falls schon vorhanden, und löschen den Eintrag:

```
<ERASE>
```

Mit `<EDIT>` erhalten Sie wieder die Alternativ-Datei. Nun kopieren Sie die 42 Textzeilen durch 42 `<PICK>`-Kommandos. Zur Sicherheit kopieren Sie den Inhalt des `_ERASE`-Puffers in einen anderen benannten `PICK`-Puffer (siehe unten: Kopieren von Pufferinhalten zwischen `PICK`-Puffern):

```
<ENTER> _ > pick_puf <PICK>
```

Sie fügen anschließend die Zeilen an anderer Stelle wieder ein:

```
<ENTER> _ <PUT>
```

oder

```
<ENTER> pick_puf <PUT>
```

### Sonderzeichen der IDENTIFIER für PICK-Puffer

Der MAXed bietet darüber hinaus die Möglichkeit, mit Auswahllisten für PICK-Puffernamen, die entweder automatisch angelegt werden oder die Sie durch Angabe eines geeigneten IDENTIFIERS selbst erstellen:

<ENTER> IDENTIFIER <EDIT>

Die Auswahllisten werden in sogenannte Pseudo-Dateien eingelesen, die Ihnen für die Dauer der Sitzung zur Verfügung steht (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie geben auf indirekte Weise den gewünschten PICK-Puffer an, indem Sie diese Pseudo-Datei aufrufen bzw. erstellen und durch Positionieren der Schreibmarke mit anschließendem <PUT> auswählen. Ein IDENTIFIER für PICK-Puffer kann folgende Sonderzeichen enthalten:

- Sie können mehrere, bereits existierende PICK-Puffer auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck mit folgenden Sonderzeichen eingeben:

? steht für ein beliebiges einzelnes Zeichen aus der Menge von Zeichen für IDENTIFIER.

\* steht für eine (Teil-)Zeichenkette mit beliebiger Kombination beliebig vieler (auch null) Zeichen.

[xyz] steht für genau eines der Zeichen aus der in den eckigen Klammern [ und ] angegebenen Menge xyz.

Wenn genau ein existierender PICK-Puffernamen zu einem eingegebenen regulären Ausdruck paßt, kopiert der MAXed den Inhalt dieses PICK-Puffers an die aktuelle Position der Schreibmarke im aktuellen Fenster. Wenn mehrere PICK-Puffernamen passen, erstellt der MAXed eine Liste dieser IDENTIFIER und schreibt diese Liste in die Pseudo-Datei PICKBUFFERS (siehe Abschnitt 4.2.7). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste einen der IDENTIFIER auswählen.

*Anmerkung*

Falls kein PICK-Puffername zu IDENTIFIER passen sollte, sucht der MAXed nach einem passenden Dateinamen und schreibt ggf. den gesamten Inhalt der zugehörigen Datei an die aktuelle Schreibmarkenposition bzw. erstellt dazu eine Auswahlliste, die in die Pseudo-Datei PUTFILES eingelesen wird (siehe Abschnitte 3.8.1 Der Dateiname als IDENTIFIER, 4.2.5 Die Pseudo-Datei PUTFILES).

*Beispiel*

```
<ENTER> bereich1 p1 <PICK>
<ENTER> bereich2 p2 <PICK>
```

Sie kopieren damit zwei Textbereiche in den beiden benannten PICK-Puffern *p1* und *p2*. Falls Sie übrigens die analoge Beispielvariante mit den ERASE-Puffern *p1* und *p2* in der aktuellen Sitzung ausprobiert haben sollten, werden Sie feststellen, daß sich die gleichnamigen IDENTIFIER nicht im geringsten stören (siehe Abschnitt 3.8.2 Der ERASE-Puffer als IDENTIFIER).

```
<ENTER> p* <PUT>
```

Der MAXed erzeugt die Liste aller PICK-Puffernamen, die mit *p* beginnen (zumindest also *p1* und *p2*), schreibt sie in die Pseudo-Datei PICKBUFFERS und blendet diese im aktuellen Fenster ein.

- Der MAXed erstellt automatisch eine Liste aller im Verlauf der Sitzung definierten benannten PICK-Puffer. Der IDENTIFIER

@ steht beim Kommando PUT für die Pseudo-Datei PICKBUFFERS mit der Liste aller von Ihnen im Verlauf der Sitzung definierten PICK-Puffernamen und eventuell dem \_ für den automatisch angelegten \_-PICK-Puffer (siehe Abschnitt 4.2.7 Die Pseudo-Datei PICKBUFFERS). PICK-Puffer können auch durch eine geladene MACRO-Bibliothek definiert sein (siehe Abschnitte 4.7 Arbeiten mit Textbausteinen, 5.1 Den MAXed aufrufen).

**Anmerkung**

Sie können definierte PICK-Puffer auch über das Sitzungsende hinaus in sogenannten MACRO-Bibliotheken abspeichern oder explizit in einer Quelldatei definieren und diese in eine MACRO-Bibliothek übersetzen (siehe Abschnitt 4.6 Arbeiten mit Textbausteinen). Sobald eine MACRO-Bibliothek bei einer MAXed-Sitzung geladen ist, stehen Ihnen die darin definierten PICK-Puffer (sprich: Textbausteine) zur Verfügung.

Neben dem \_-PICK-Puffer hält sich der MAXed eine Pseudo-Datei namens RESTORE, die gelöschte Textbereiche sammelt (siehe Abschnitt 4.2.2 Die Pseudo-Datei RESTORE). Sie holen sich die Pseudo-Datei RESTORE ins aktuelle Fenster mit

<ENTER> <EDIT>

Diese Pseudo-Datei hat die Funktion eines Langzeitgedächtnisses, u.a. für alle jemals im Standard-ERASE-Puffer und im Standard-PICK-Puffer gespeicherten Textbereiche (siehe Abschnitt 3.8.2 Der ERASE-Puffer als IDENTIFIER). Während der Sitzung können Sie sich aus dieser Datei z.B. wieder alle Texte kopieren, die Ihnen durch textverändernde Kommandos verloren gingen.

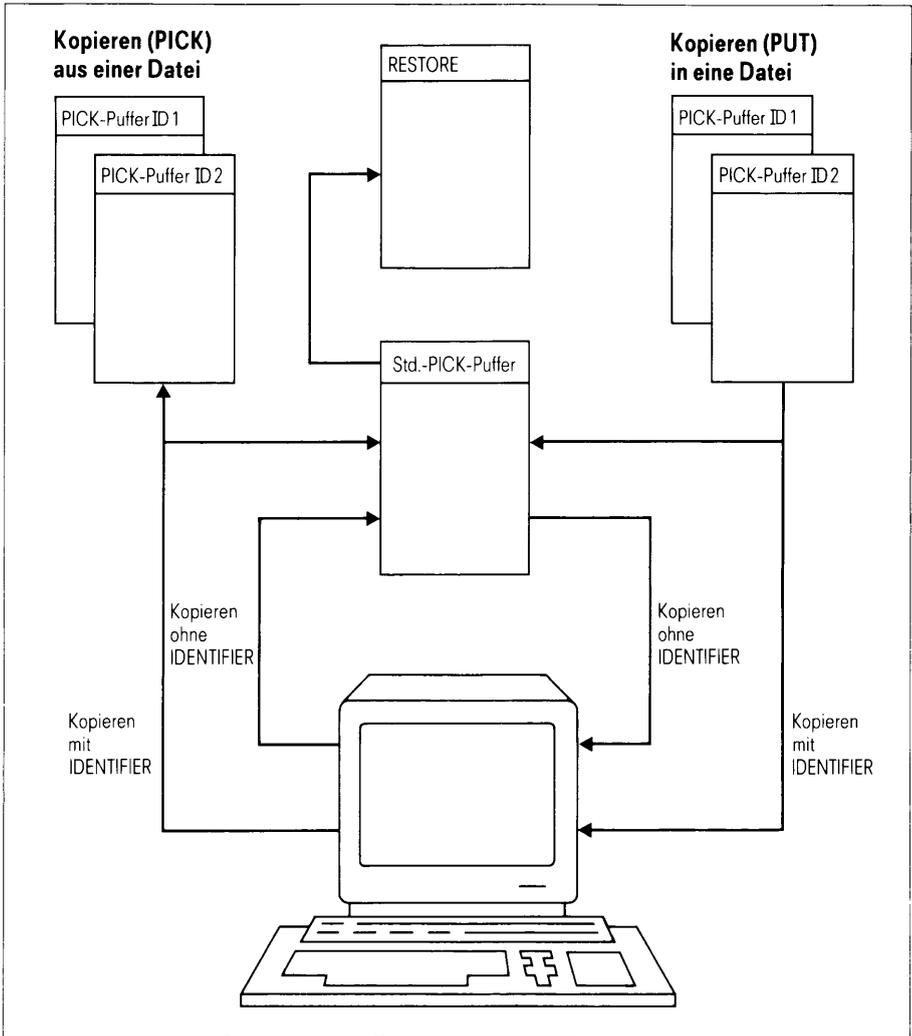


Bild 3-17 PICK-Puffer und Pseudo-Datei RESTORE

Bild 3-17 veranschaulicht die Zusammenhänge verschiedener Puffer beim Kopieren von Textbereichen: Der Standard-PICK-Puffer und die Pseudo-Datei RESTORE werden grundsätzlich bei jedem Kopierkommando gefüttert. Die benannten PICK-Puffer werden nur dann verwendet, wenn Sie explizit durch ihren IDENTIFIER angesprochen werden. Der `_`-PICK-Puffer dagegen wird ausschließlich mit jenem Textmaterial beladen, das beim Kopieren nur im Standard-PICK-Puffer, nicht aber in einem weiteren benannten PICK-Puffer landete.

### Kopieren von Pufferinhalten zwischen PICK-Puffern

Sie haben die Möglichkeit, den gespeicherten Textbereich eines beliebigen PICK-Puffers direkt in einen anderen PICK-Puffer zu kopieren (siehe Kapitel 6: PICK).

Sie kopieren den Inhalt des PICK-Puffers *pick\_puf1* in den PICK-Puffer *pick\_puf2* mit

```
<ENTER> [pick_puf1] > [pick_puf2] <PICK>
```

Dabei wird der bisherige Inhalt von *pick\_puf2* überschrieben!

Sie hängen eine Kopie des Inhalts von *pick\_puf1* ans Ende des bisherigen Inhalts von *pick\_puf2* mit

```
<ENTER> [pick_puf1] >> [pick_puf2] <PICK>
```

### Anmerkung

Wie die Optionsklammern andeuten, können Sie die IDENTIFIER auch weglassen. Wenn kein Puffername angegeben ist, greift der MAXed auf den Standard-PICK-Puffer zu.

### 3.8.4 Der Markenname als IDENTIFIER

Marken kennzeichnen absolute Positionen innerhalb von Dateien. Sie können Marken an beliebigen Stellen in Ihren geöffneten Datei setzen und diese Stellen später von einer beliebigen Position innerhalb derselben Datei direkt anspringen (siehe Kapitel 6: LABEL, GOTO). Dadurch ersparen Sie sich das Blättern und Suchen im Text. Marken sind dateispezifisch. Dadurch können Sie für jede geöffnete Datei einen eigenen Satz von Marken anlegen, wobei in verschiedenen Dateien durchaus gleiche Markennamen existieren dürfen, ohne sich zu beeinflussen.

Sie setzen an der Position der Schreibmarke in der aktuellen Datei eine (nicht sichtbare) Marke namens *marke* mit

```
<ENTER> marke <LABEL>
```

Sie springen mit der Schreibmarke von beliebiger Stelle einer Datei an die Position der zuvor gesetzten Marke *marke* mit

```
<ENTER> marke <GOTO>
```

Die Standard-Marke:

Neben den benannten Marken gibt noch eine unbenannte, die Standard-Marke, die unabhängig von den benannten Marken existiert. Sie können in jeder geöffneten Datei eine Standard-Marke mit <LABEL> setzen (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern). Innerhalb derselben Datei springen Sie mit der Schreibmarke an die Position der zuvor gesetzten Standard-Marke wieder zurück mit <ENTER> <GOTO>

Bei der Eingabe eines Markennamens als Kommandoattribut gilt folgende grundsätzliche Regel: Wenn Sie eine benannte Marke anlegen, können Sie als Markenname eine beliebige Zeichenkette eingeben, die aus einem oder mehreren alphanumerischen Zeichen sowie als einzigem nicht-alphanumerischem Zeichen dem Unterstrich \_ bestehen kann und mit einem Buchstaben oder einem Unterstrich beginnt. Dabei unterscheidet der MAXed nicht zwischen Groß- und Kleinbuchstaben.

### Sonderzeichen der IDENTIFIER für Marken

Der MAXed bietet darüber hinaus die Möglichkeit, mit Auswahllisten für Markennamen, die entweder automatisch angelegt werden oder die Sie durch Angabe eines geeigneten IDENTIFIERS selbst erstellen:

```
<ENTER> IDENTIFIER <LABEL>
```

Die Auswahllisten werden in sogenannte Pseudo-Dateien eingelesen, die Ihnen für die Dauer der Sitzung zur Verfügung steht (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie geben auf indirekte Weise die gewünschte Marke an, indem Sie diese Pseudo-Datei aufrufen bzw. erstellen und durch Positionieren der Schreibmarke mit anschließendem <GOTO> auswählen. Ein IDENTIFIER für Marken kann folgende Sonderzeichen enthalten:

- Sie können mehrere, bereits existierende Marken auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck mit folgenden Sonderzeichen eingeben:
  - ? steht für ein beliebiges einzelnes Zeichen aus der Menge von Zeichen für IDENTIFIER.
  - \* steht für eine (Teil-)Zeichenkette mit beliebiger Kombination beliebig vieler (auch null) Zeichen.
  - [xyz] steht für genau eines der Zeichen aus der in den eckigen Klammern [ und ] angegebenen Menge xyz.

Wenn genau ein für die aktuelle Datei existierender Markenname zu einem eingegebenen regulären Ausdruck paßt, springt die Schreibmarke direkt an die Stelle der gesetzten Marke. Wenn mehrere Markennamen passen, erstellt der MAXed eine Liste dieser IDENTIFIER und schreibt diese Liste in die Pseudo-Datei LABELS (siehe Abschnitt 4.2.8). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste einen der IDENTIFIER auswählen.

*Beispiel*

Sie positionieren die Schreibmarke an eine beliebige Stelle der aktuellen Datei und setzen dort die Marke *m1* mit

```
<ENTER> m1 <LABEL>
```

Sie bewegen die Schreibmarke innerhalb derselben Datei an eine andere Stelle und setzen dort die Marke *m2* mit

```
<ENTER> m2 <LABEL>
```

```
<ENTER> p* <GOTO>
```

Der MAXed antwortet mit einer Meldung in der Steuerzeile:

```
**** Kein LABEL definiert.
```

Sie versuchen es also nochmals mit

```
<ENTER> m* <GOTO>
```

Jetzt erzeugt der MAXed die Liste aller Markennamen, die mit *m* beginnen (zumindest also *m1* und *m2*), schreibt sie in die Pseudo-Datei LABELS und blendet diese im aktuellen Fenster ein.

- Der MAXed erstellt automatisch eine Liste aller im Verlauf der Sitzung für die aktuelle Datei definierten Marken. Der IDENTIFIER

@ steht beim Kommando GOTO für die Pseudo-Datei LABELS mit der Liste aller von Ihnen im Verlauf der Sitzung in der aktuellen Datei definierten Markennamen und eventuell der Standard-Marke (siehe Abschnitt 4.2.8 Die Pseudo-Datei LABELS). Marken, die bereits während der letzten MAXed-Sitzung in der aktuellen Datei gesetzt wurden, können auch durch Einlesen Status-Datei wieder definiert sein (siehe Abschnitt 5.1 Den MAXed aufrufen).

**Anmerkung**

Sie können gesetzte Marken auch wieder löschen. Dazu stehen Ihnen gleich zwei Möglichkeiten offen:

- Sie blenden die Pseudo-Datei LABELS ein und löschen den entsprechenden Eintrag mit <ERASE>.
- Sie setzen die Schreibmarke auf die Position der "nicht-mehr-erwünschten" Marke *marke*, z.B. mit

```
<ENTER> marke <GOTO>
```

Diese Marke löschen Sie:

```
<ENTER> <LABEL>
```

### 3.8.5 Der Makroname als IDENTIFIER

Makros sind Puffer, die Sequenzen von MAXed-Kommandos speichern. Beim Aufruf eines Makros arbeitet der MAXed die abgespeicherte Kommandofolge genau auf die gleiche Weise ab, als ob Sie diese Kommandos einzeln eingeben würden (siehe Kapitel 6: MACRO).

Parametrisierte Makros:

Jedoch müssen Makros nicht auf die simple, identische Wiederholung einer Kommandofolge beschränkt sein. Sie geben beim Aufruf eines parametrisierten Makros noch weitere Zeichenketten an, die das Makro bei seinen Aktionen als Makro-Parameter verwenden kann (siehe Abschnitt 4.7 Arbeiten mit Makros).

Sie erstellen Makros mit Hilfe des Kommandos LEARN in der Lernphase (siehe Abschnitt 4.1.11 Der Lern-Modus, Kapitel 6: LEARN). Sie schalten den Lern-Modus zur Definition des des benannten Makros *makro* ein mit

```
<ENTER> makro [parameter ...] <LEARN>
```

Von jetzt an werden alle Tastenanschläge aufgezeichnet! Sie beenden die Makroerstellung wieder mit

```
<LEARN>
```

Sie rufen ein definiertes Makro *makro* auf mit

```
<ENTER> makro <MACRO>
```

Die Ausführung des Makros beginnt an der aktuellen Schreibmarkenposition in der aktuellen Datei.

Neben den benannten Makros gibt es noch zwei unbenannte, das aktuelle Makro und das Standard-Makro (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern):

- Das aktuelle Makro ist das im Kommando-Puffer des Kommandos `MACRO` gespeicherte Makro. Im Kommando-Puffer speichert der MAXed immer das zuletzt durch das Kommando `LEARN` erstellte Makro oder das zuletzt ausgeführte, benannte Makro.
- Das Standard-Makro ist das Makro, das Sie zuletzt als unbenanntes Makro mit dem Kommando `LEARN` erstellt haben (siehe oben. Es wird in einem eigenen Kommando-Puffer gespeichert und existiert daher unabhängig von den anderen definierten Makros.

Bei der Eingabe eines Makronamens als Kommandoattribut gilt folgende grundsätzliche Regel:

Wenn Sie ein benanntes Makro anlegen, können Sie als Makroname eine beliebige Zeichenkette, die aus einem oder mehreren alphanumerischen Zeichen sowie als einzigem nicht-alphanumerischem Zeichen dem Unterstrich `_` bestehen kann und mit einem Buchstaben oder einem Unterstrich beginnt, eingeben. Dabei unterscheidet der MAXed nicht zwischen Groß- und Kleinbuchstaben.

### **Sonderzeichen der IDENTIFIER für Makros**

Der MAXed bietet darüber hinaus die Möglichkeit, mit Auswahllisten für Makronamen, die entweder automatisch angelegt werden oder die Sie durch Angabe eines geeigneten IDENTIFIERs selbst erstellen:

```
<ENTER> IDENTIFIER <MACRO>
```

Die Auswahllisten werden in sogenannte Pseudo-Dateien eingelesen, die Ihnen für die Dauer der Sitzung zur Verfügung steht (siehe Abschnitt 4.2 Die Pseudo-Dateien). Sie geben auf indirekte Weise das gewünschte Makro an, indem Sie diese Pseudo-Datei aufrufen bzw. erstellen und durch Positionieren der Schreibmarke mit anschließendem <MACRO> auswählen. Ein IDENTIFIER für Makros kann folgende Sonderzeichen enthalten:

- Sie können mehrere, bereits existierende Makros auf einmal ansprechen, wenn Sie als IDENTIFIER einen regulären Ausdruck mit folgenden Sonderzeichen eingeben:

? steht für ein beliebiges einzelnes Zeichen aus der Menge von Zeichen für IDENTIFIER.

\* steht für eine (Teil-)Zeichenkette mit beliebiger Kombination beliebig vieler (auch null) Zeichen.

[xyz] steht für genau eines der Zeichen aus der in den eckigen Klammern [ und ] angegebenen Menge xyz.

Wenn genau ein existierender Makroname zu einem eingegebenen regulären Ausdruck paßt, wird dieses Makro sofort an der Position der Schreibmarke in der aktuellen Datei ausgeführt. Wenn mehrere Makronamen passen, erstellt der MAXed eine Liste dieser IDENTIFIER und schreibt diese Liste in die Pseudo-Datei MACROS (siehe Abschnitt 4.2.10 Die Pseudo-Datei MACROS). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste einen der IDENTIFIER auswählen:

- Der MAXed erstellt automatisch eine Liste aller im Verlauf der Sitzung für die aktuelle Datei definierten Makros. Der IDENTIFIER

@ steht beim Kommando MACRO für die Pseudo-Datei MACROS mit der Liste aller von Ihnen im Verlauf der Sitzung in der aktuellen Datei definierten Makronamen (siehe Abschnitt 4.2.10 Die Pseudo-Datei MACROS). Makros, können auch durch eine geladene MACRO-Bibliothek definiert sein (siehe Abschnitte 4.7 Arbeiten mit Makros, 5.1 Den MAXed aufrufen).

**Anmerkung**

Sie können erstellte Makros auch über das Sitzungsende hinaus in sogenannten MACRO-Bibliotheken abspeichern oder explizit in einer Quelldatei definieren und diese in eine MACRO-Bibliothek übersetzen (siehe Abschnitt 4.7 Arbeiten mit Makros). Sobald eine MACRO-Bibliothek bei einer MAXed-Sitzung geladen ist, stehen Ihnen die darin definierten Makros zur Verfügung.

### 3.8.6 Der Name eines Steuerprogramms als IDENTIFIER

MAXed-Steuerprogramme sind ausführbare Programme, die eine MAXed-Sitzung kontrollieren. Wenn Sie ein Steuerprogramm aufgerufen haben, läuft es als eigener Prozeß im Hintergrund und kommuniziert mit dem MAXed-Prozeß. Sie erstellen MAXed-Steuerprogramme als C-Programme, die die Schnittstellen-Funktionen der MAXed-Bibliothek verwenden (siehe Abschnitt 4.8 Arbeiten mit Steuerprogrammen, Kapitel 6: MACRO).

Sie rufen das mit IDENTIFIER bezeichnetes Makro oder das mit IDENTIFIER bezeichnete, ausführbare Steuerprogramm auf mit

```
<ENTER> IDENTIFIER <MACRO>
```

Falls ein Makroname zu IDENTIFIER paßt, wird dieses ausgeführt bzw. die Pseudo-Datei MACROS zuvor mit den passenden Makronamen im aktuellen Fenster präsentiert (siehe Abschnitt 4.2.10 Die Pseudo-Datei MACROS).

Wenn kein Makroname zu IDENTIFIER paßt, sucht der MAXed nach einem passenden Dateinamen eines ausführbaren Steuerprogramms. Das Steuerprogramms läuft ab sofort parallel zum MAXed-Prozeß im Hintergrund.

Der MAXed arbeitet auf dem Dateisystem seiner Aufrufumgebung. Bei der Eingabe des Namens eines Steuerprogramms als Kommandoattribut gelten folgende Regeln:

- Sie müssen bei der Eingabe des Aufrufs des ausführbaren Steuerprogramms die SINIX-Konventionen beachten.
- Sie können während der MAXed-Sitzung alle Dateien des Dateisystems adressieren, wenn Sie den absoluten Pfadnamen als Dateinamen verwenden. Relative Pfadnamen beziehen sich auf das aktuelle Dateiverzeichnis, also im allgemeinen jenes von dem aus Sie den MAXed aufgerufen haben (siehe Abschnitt 5.1 Den MAXed aufrufen, Kapitel 6: DEFDIR). Außerdem sucht der MAXed ein Steuerprogramm noch in vordefinierten Dateiverzeichnissen (siehe Abschnitt 7.5 MAXed-Steuerprogramme).

### 3.9 Die Bereichsdefinition als Kommandoattribut

Viele der MAXed-Kommandos arbeiten mit Textbereichen. Wenn Sie Textteile löschen, kopieren oder formatieren wollen, geben Sie die Definition dieser Textbereiche als Kommandoattribute ein. Jedes MAXed-Kommando, das mit Textbereichen arbeitet, hat außerdem eine voreingestellte Bereichsdefinition, die immer dann verwendet wird, wenn Sie das Kommando nicht explizit mit einer Bereichsdefinition attributieren (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).

Sie kennzeichnen mit einer Bereichsdefinition einen zusammenhängenden Textpassage innerhalb der Datei im aktuellen Fenster. Eine Bereichsdefinition hat grundsätzlich drei Bestandteile:

- Position des Bereichsanfangs
- Position des Bereichsendes
- Bereichstyp

#### Die Bereichstypen

Der MAXed unterscheidet drei Bereichstypen:

- RANGE (Zeilenbereich)
- RECTANGLE (Rechteck)
- STREAM (Zeichenkette)

```

text text text text text text text text text text
RANGE RANGE
RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE
RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE RANGE
text text
text text text text text text text text text text text
text text RECTANGLE RECTANGLE RECTANGLE RECTANGLE text text
text text RECTANGLE RECTANGLE RECTANGLE RECTANGLE text text text
text text RECTANGLE RECTANGLE RECTANGLE RECTANGLE text text text text
text text RECTANGLE RECTANGLE RECTANGLE RECTANGLE text text
text text text text text text text text text text text
text text text text text text text text text text text
text text text text STREAM STREAM STREAM STREAM STREAM STREAM
STREAM STREAM STREAM STREAM STREAM STREAM STREAM STREAM STREAM
STREAM STREAM STREAM STREAM STREAM STREAM STREAM STREAM STREAM
STREAM STREAM STREAM STREAM text text text text
text text text text text text text text text text text
text text text text text text text text text text text

```

Mode: File Bereiche Ln Cl

Bild 3-18 Bereichstypen

Wie Bild 3-18 zeigt, haben die drei Bereichstypen folgende Eigenschaften:

Ein RANGE-Bereich

umfaßt immer ganze, aufeinanderfolgende Zeilen einschließlich der unsichtbaren Zeilenende-Zeichen. Die Bereichsgrenzen eines RANGE-Bereichs werden durch die Angabe der Positionen der ersten und der letzten Zeile festgelegt.

### Ein RECTANGLE-Bereich

ist ein beliebiger rechteckiger Textausschnitt. Die Bereichsgrenzen eines RECTANGLE-Bereichs werden durch die Angabe der Positionen zweier diagonal gegenüberliegender Eckpunkte festgelegt. Längs des oberen und des unteren angegebenen Eckpunktes werden die unsichtbaren, waagrechten Begrenzungslinien des RECTANGLE-Bereichs gezogen. Die senkrechten Begrenzungslinien verlaufen längs des linken angegebenen Eckpunktes und längs eines Zeichens links vom rechten angegebenen Eckpunkt. Zeilenende-Zeichen innerhalb der Bereichsgrenzen werden grundsätzlich ignoriert: Solche Zeilen, deren Enden links der rechten Begrenzungslinie liegen, werden so behandelt, als wären sie mit Leerzeichen aufgefüllt.

### Ein STREAM-Bereich

ist eine beliebige zusammenhängende Zeichenkette, die sich auch über mehrere Zeilen hinweg erstrecken kann. Die Bereichsgrenzen eines STREAM-Bereichs liegen auf dem Zeichen der angegebenen Anfangsposition und auf dem Zeichen links der angegebenen Endposition. Bei mehrzeiligen STREAM-Bereichen sind die Zeilenende-Zeichen innerhalb der Bereichsgrenzen mitenthalten.

### BLOCK- und RECORD-Bereiche sind

Sonderfälle von STREAM-Bereichen, die vordefinierte Bereichsgrenzen haben. Diese Bereichsgrenzen können Sie nur innerhalb von Zeichenketten-Attributen eingeben.

### Anmerkung

Beachten Sie, daß bei einem RECTANGLE-Bereich der rechte Eckpunkt ein Zeichen links der angegebenen rechten Bereichsgrenze liegt. Dadurch haben Sie die Möglichkeit, RECTANGLE-Bereiche, die einspaltig sind, zu definieren. Bei einem STREAM-Bereich liegt das rechte bzw. untere Ende ebenfalls auf dem Zeichen links der angegebenen Bereichsgrenze, sodaß Sie auch STREAM-Bereiche, die aus einem Zeichen bestehen, definieren können.

## Die Eingabe der Bereichsgrenzen

Es stehen Ihnen grundsätzlich zwei Verfahren zur Verfügung, um ein MAXed-Kommando mit einer Bereichsdefinition zu attributieren:

- Festlegen der Bereichs, indem Sie mit der Schreibmarke die Bereichsgrenzen im Text markieren (siehe Abschnitt 3.9.1 Die Bereichsdefinition durch Bewegen der Schreibmarke)
- Festlegen des Bereichs, indem Sie ein Zeichenketten-Attribut eingeben (siehe Abschnitt 3.9.2 Die Bereichsdefinition durch Zeichenketten-Attribut)

Sie können diese beiden Verfahrensweisen einer Bereichsattributierung auch dadurch kombinieren, daß Sie die eine Bereichsgrenze durch die aktuelle Position der Schreibmarke im Text und die andere Bereichsgrenze in Form eines Zeichenketten-Attributs festlegen. Dies ist in Abschnitt 3.9.2 näher beschrieben.

### Anmerkung

- Die Reihenfolge, in der Sie die Bereichsgrenzen bei der Kommandoattributierung angeben, muß nicht mit ihrer Aufeinanderfolge im Text übereinstimmen. Bevor der MAXed ein aufgerufenes Kommando ausführt, überprüft er die Reihenfolge der angegebenen Grenzen. Wenn diese nicht chronologisch aufeinanderfolgen, führt er das attributierte Kommando mit vertauschten Bereichsangaben durch. Chronologisch müssen Sie deshalb dann vorgehen, wenn Sie für die Bereichsangabe solche Sonderzeichen verwenden, die auf der Position der ersten Bereichsgrenze eine andere Bedeutung haben als auf der Position der zweiten Bereichsgrenze (siehe Abschnitt 3.9.2 Die Bereichsdefinition als Zeichenketten-Attribut).

Die Positionen der Bereichsgrenzen und der Bereichstyp bedingen sich gegenseitig:

- Innerhalb von Zeichenketten-Attributen werden diejenigen Angaben von Bereichsgrenzen, die Sie gleichlautend bei allen drei Bereichstypen verwenden können, dennoch unterschiedlich interpretiert; denn im Falle eines RANGE-Bereichs ist lediglich die Zeilennummer, die durch eine Bereichsgrenze festgelegt wird, von Bedeutung, während bei einem RECTANGLE- oder STREAM-Bereich die absolute Position (Zeilen- und Spaltennummer) ausschlaggebend ist.
- Bei einer Bereichsdefinition durch Bewegen der Schreibmarke wird umgekehrt der Bereichstyp allein dadurch festgelegt, wie Sie die Schreibmarke bewegen, um die Bereichsgrenzen zu markieren.

### **Die Bedeutung des Bereichstyps**

Bei jeder Bearbeitung eines Textteils, das größer als ein einzelnes Zeichen ist, liegt einer der drei oben genannten Bereichstypen vor. Selbst wenn Sie den zu bearbeitenden Bereich nicht als Kommandoattribut angeben sollten, so ist er doch unausgesprochen vordefiniert (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).

Am häufigsten werden Ihnen die Bereichstypen wohl beim Ausschneiden und Wiedereinsetzen sowie beim Kopieren von Textbereichen begegnen (siehe Abschnitte 3.8.2 Der ERASE-Puffername als IDENTIFIER, 3.8.3 Der PICK-Puffername als IDENTIFIER). Beim Wiedereinsetzen bzw. Kopieren eines Pufferinhalts in den Text einer Datei wendet der MAXed grundsätzlich folgende Verfahrensweise an (siehe Kapitel 6: PUT, RESTOR):

Abhängig vom aktuellen Eingabe-Modus wird der Pufferinhalt an der aktuellen Schreibmarkenposition eingefügt, oder er überschreibt den vorhandenen Text (siehe Abschnitt 4.1.1 Die Einfüge-Modi). Jener Textbereich, der beim Einsetzen eines Pufferinhalts verdrängt oder überschrieben wird, stimmt in Bereichstyp und Umfang (Anzahl Zeilen bzw. Spalten) mit dem Pufferinhalt überein. Für die verschiedenen Bereichstypen bedeutet dies:

- Ein Pufferinhalt mit Bereichstyp `RANGE` (Zeilenbereich) fügt sich vor der aktuellen Zeile ein oder überschreibt einen gleichgroßen Zeilenbereich ab der aktuellen Zeile.
- Ein Pufferinhalt mit Bereichstyp `RECTANGLE` (Rechteck) fügt sich mit seiner linken oberen Ecke an der aktuellen Schreibmarkenposition und mit seiner linken (unsichtbaren) Begrenzungslinie an der aktuellen Spalte ein oder überschreibt ein gleichgroßes Rechteck an der aktuellen Position.
- Ein Pufferinhalt mit Bereichstyp `STREAM` (Zeichenkette) fügt sich mit seinem Anfangszeichen an der aktuellen Schreibmarkenposition ein oder überschreibt ab dieser Position in folgender Form: die letzte Zeile des `STREAM`-Bereichs überdeckt gleichviele Zeichen der korrespondierenden letzten Zeile im Text. Ein mehrzeiliger `STREAM` ersetzt den gesamten Text zwischen der aktuellen Schreibmarkenposition und der korrespondierenden letzten Zeile.

### 3.9.1 Die Bereichsdefinition durch Bewegen der Schreibmarke

Eine Bereichsdefinition durch Bewegen der Schreibmarke geben Sie grundsätzlich als erstes bei der Attributierung eines MAXed-Kommandos ein. Positionieren Sie vor Beginn der Kommandoattributierung die Schreibmarke auf eine der beiden gewünschten Bereichsgrenzen und bewegen Sie im weiteren Verlauf der Attributierung die Schreibmarke zur anderen Bereichsgrenze. Dabei darf die zweite Bereichsgrenze durchaus außerhalb des momentan sichtbaren Textausschnitts im aktuellen Fenster liegen. Im Detail verfahren Sie folgendermaßen:

#### Die Definition eines RANGE-Bereichs

Eine RANGE-Definition erfolgt durch ausschließlich vertikale Bewegungen der Schreibmarke:

- Positionieren Sie die Schreibmarke vor Beginn der Attributierung auf der ersten oder der letzten Zeile des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein (siehe Abschnitt 3.6 Prinzipielles zur Attributierung von Kommandos). Sie sehen an der aktuellen Position der Schreibmarke im Text ein grafisches Zeichen als Markierung. Die Schreibmarke befindet sich im Eingabefeld der Steuerzeile. Geben Sie jedoch keinen Text ein.
- Bewegen Sie die Schreibmarke in vertikaler Richtung zur anderen Bereichsgrenze. Hierzu beginnen Sie mit einem der folgenden Kommandos:

<DOWNARROW> oder <UPARROW>

<PLUSPAGE> oder <MINUSPAGE> (siehe Kapitel 6)

Im Eingabefeld der Steuerzeile erscheint die Meldung

**RANGE** Anfang bei Ln zeile C1 spalte

Die Schreibmarke springt wieder in den Text zurück und bewegt sich dort, ausgehend von dem grafischen Zeichen, also der Markierung der aktuellen Schreibmarkenposition, entsprechend dem eingegebenen Kommando weiter. Ab jetzt stehen Ihnen alle Kommandos zum vertikalen Bewegen der Schreibmarke und vertikalen Verschieben des Dateiausschnitts zur Verfügung, um die Schreibmarke auf die andere Bereichsgrenze zu setzen (siehe Abschnitte 3.3.2 Bewegen der Schreibmarke, 3.3.3 Verschieben des Dateiausschnitts).

### Beispiel

<ENTER> <PLUSPAGE> <RVLINEFEED>

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines RANGE-Bereichs, der alle Zeilen von der Zeile mit der Markierung der aktuellen Schreibmarkenposition bis zu der momentanen Position der Schreibmarke im Text umfaßt: eine Bildschirmseite nach unten und wieder eine Zeile nach oben.

### Die Definition eines RECTANGLE-Bereichs

Eine RECTANGLE-Definition erfolgt durch Kombination von vertikalen und horizontalen Bewegungen der Schreibmarke:

- Positionieren Sie die Schreibmarke vor Beginn der Attributierung auf einen der linken Eckpunkte oder direkt hinter einen der rechten Eckpunkte des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke zuerst in vertikaler Richtung wie bei der Definition eines RANGE-Bereichs (siehe oben). Im Eingabefeld der Steuerzeile erscheint die Meldung  
**RANGE Anfang bei Ln zeile C1 spalte**  
Das soll Sie nicht irritieren. Ab jetzt stehen Ihnen alle Kommandos zum vertikalen und horizontalen Bewegen der Schreibmarke bzw. Verschieben des Fensters zur Verfügung. Bewegen Sie damit die Schreibmarke zum diagonal gegenüberliegenden Eckpunkt. Falls das ein linker Eckpunkt ist, setzen Sie die Schreibmarke auf diesen, falls das ein rechter Eckpunkt ist, setzen Sie die Schreibmarke auf die Position direkt dahinter.
- Bei der ersten horizontalen Bewegung ändert sich die Meldung in:  
**RECTANGLE Anfang bei Ln zeile C1 spalte**

### Beispiel

<ENTER> <PLUSPAGE> <TAB>

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines RECTANGLE-Bereichs, der seinen linken oberen Eckpunkt an der Markierung der aktuellen Schreibmarkenposition und seinen rechten unteren Eckpunkt ein Zeichen vor der momentanen Position der Schreibmarke im Text hat: eine Bildschirmseite nach unten und eine Tabulatorposition nach rechts.

### Die Definition eines STREAM-Bereichs

Eine STREAM-Definition erfolgt durch Kombination von horizontalen und vertikalen Bewegungen der Schreibmarke:

- Positionieren Sie die Schreibmarke vor Beginn der Attributierung auf das erste Zeichen oder direkt hinter das letzte Zeichen des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke zuerst in horizontaler Richtung. Hierzu beginnen Sie mit einem der folgenden Kommandos:  
<LEFTARROW> oder <RIGHTARROW>  
<WORDLEFT> oder <WORDRIGHT>  
<TAB> oder <BACKTAB>  
<RETURN> oder <UPEND>  
<WINL> oder <WINR> (siehe Kapitel 6)

Im Eingabefeld der Steuerzeile erscheint die Meldung  
**STREAM Anfang bei Ln zeile Cl spalte**

Ab jetzt stehen Ihnen alle Kommandos zum vertikalen und horizontalen Bewegen der Schreibmarke bzw. Verschieben des Fensters zur Verfügung, um die Schreibmarke auf die andere Bereichsgrenze zu setzen.

### Beispiel

```
<ENTER> <UPEND> <WORDLEFT>
```

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines STREAM-Bereichs, dessen Ende ein Zeichen vor der Markierung der aktuellen Schreibmarkenposition und dessen Anfang an der momentanen Position der Schreibmarke im Text liegt: ans Ende der vorherigen Zeile und ein Wort nach links.

### Die Eingabe weiterer Kommandoattribute

Nachdem Sie einen Bereich durch Bewegen der Schreibmarke definiert haben, können Sie weitere Kommandoattribute in Form von Zeichenketten im Eingabefeld der Steuerzeile eingeben. Sobald Sie das erste Zeichen eintippen, verschwindet die Meldung der aktuellen Bereichsdefinition (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). An der momentanen Position der Schreibmarke im Text erscheint ein zweites grafisches Symbol als Markierung und die Schreibmarke springt für die Eingabe an den Anfang des Eingabefeldes zurück.

### Spezielle RECTANGLE- und STREAM-Bereiche

Für RECTANGLE-Bereiche und einzeilige STREAM-Bereiche, deren Bereichsgrenzen links auf Spalte eins oder rechts hinter der letzten Spalte liegen sollen, steht Ihnen ein besonders einfaches Verfahren zur Verfügung (siehe Bild 3-19):

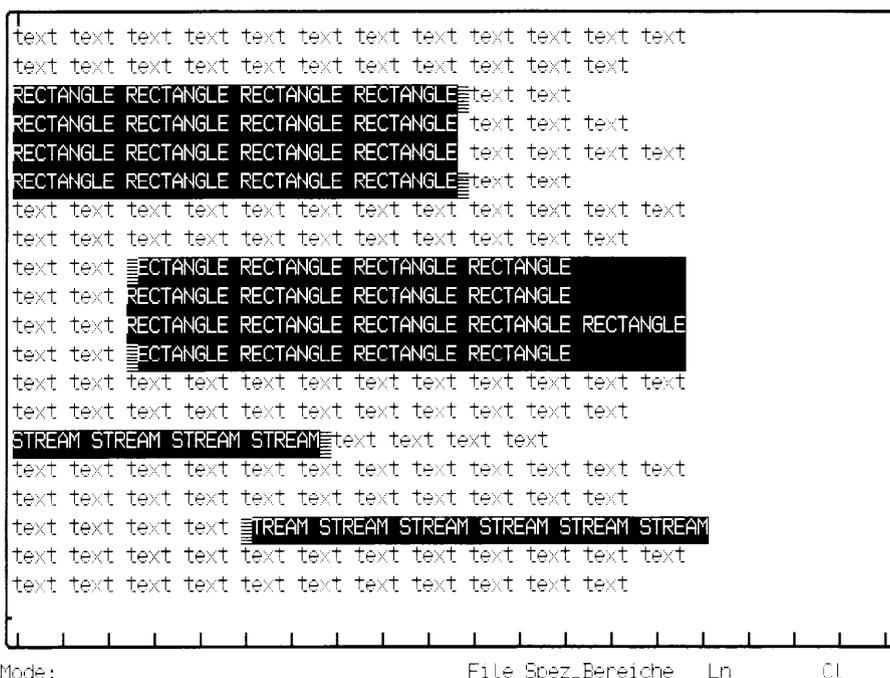


Bild 3-19 STREAM und RECTANGLE durch erste bzw. letzte Spalte begrenzt

Einen RECTANGLE-Bereich, der links durch die erste Spalte begrenzt sein soll, legen Sie so fest:

- Positionieren Sie die Schreibmarke vor Beginn der Kommandoattributierung direkt hinter einen der rechten Eckpunkte des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke zuerst in vertikaler Richtung (siehe oben: RANGE-Definition).

- Bewegen Sie die Schreibmarke im weiteren Verlauf einmal nach links in Richtung der gegenüberliegenden Begrenzungslinie.
- Bewegen Sie die Schreibmarke direkt hinter den zweiten rechten Eckpunkt in die Spalte der Ausgangsposition zurück.

Einen RECTANGLE-Bereich, der rechts durch die letzte Spalte der längsten Zeile begrenzt sein soll, legen Sie so fest:

- Positionieren Sie die Schreibmarke vor Beginn der Kommandoattributierung auf einen der linken Eckpunkte des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke zuerst in vertikaler Richtung (siehe oben: RANGE-Definition).
- Bewegen Sie die Schreibmarke im weiteren Verlauf einmal nach rechts in Richtung der gegenüberliegenden Begrenzungslinie.
- Bewegen Sie die Schreibmarke auf den zweiten linken Eckpunkt in die Spalte der Ausgangsposition zurück.

Einen einzeiligen STREAM-Bereich, der links durch die erste Spalte begrenzt sein soll, legen Sie so fest:

- Positionieren Sie die Schreibmarke vor Beginn der Kommandoattributierung direkt hinter das letzte Zeichen des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke mit einem der o.g. Kommandos einmal nach links in Richtung der anderen Bereichsgrenze und wieder zurück auf die Ausgangsposition.

Einen einzeiligen STREAM-Bereich, der rechts durch die letzte Spalte begrenzt sein soll legen Sie so fest:

- Positionieren Sie die Schreibmarke vor Beginn der Kommandoattributierung auf das erste Zeichen des gewünschten Bereichs.
- Leiten Sie die Kommandoattributierung durch <ENTER> ein.
- Bewegen Sie die Schreibmarke mit einem der o.g. Kommandos einmal nach rechts in Richtung der anderen Bereichsgrenze und wieder zurück auf die Ausgangsposition.

**Beispiel**

<ENTER> <LEFTARROW> <RIGHTARROW>

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines STREAM-Bereichs, der vom Anfang der aktuellen Zeile bis zu einem Zeichen vor der Markierung der Schreibmarkenposition reicht.

### 3.9.2 Die Bereichsdefinition durch Zeichenketten-Attribut

Eine Bereichsdefinition durch ein Zeichenketten-Attribut geben Sie bei allen MAXed-Kommandos, die mit Bereichen arbeiten, als erstes Attribut ein. Beim Kommando REPLACE wird diese Form der Bereichsangabe als letztes Kommandoattribut eingegeben. Die Bereichsgrenzen dürfen an beliebigen Positionen innerhalb der Datei im aktuellen Fenster liegen.

Sie geben die drei Bestandteile einer Bereichsangabe in einem Zeichenketten-Attribut in folgender Reihenfolge an:

- Argument für die erste Bereichsgrenze
- Symbol für den Bereichstyp RANGE -, RECTANGLE : oder STREAM ~
- Argument für die zweite Bereichsgrenze

Daneben gibt es einzelne Sonderzeichen, mit denen Sie vordefinierte Standard-Bereiche festlegen (siehe unten: Sonderzeichen für die Bereichsdefinition).

### Kombinieren von Zeichenketten-Attribut und Schreibmarkenposition

Sie kombinieren eine Bereichsdefinition durch Zeichenketten-Attribut mit der Bereichsangabe durch Bewegungen der Schreibmarke, indem Sie eines der beiden Argumente für die Bereichsgrenzen weglassen. Für das fehlende Argument wird ersatzweise die aktuelle Position im Text verwendet, die während der Kommandoattributierung durch ein grafisches Zeichen markiert ist (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegungen der Schreibmarke).

### Sonderzeichen für die Bereichsdefinition

Einige Sonderzeichen für die Bereichsdefinition durch ein Zeichenketten-Attribut bezeichnen bestimmte Texteinheiten oder Datei-Positionen in der aktuellen Datei:

- . (Punkt) steht für die gesamte aktuelle Datei bzw. deren Bereichsgrenzen
- \$ (Dollar-Zeichen) steht für den gesamten aktuellen Absatz bzw. dessen Bereichsgrenzen
- ; (Semikolon) steht für die aktuelle Zeile bzw. deren Bereichsgrenzen

- , (Komma) steht für das aktuelle Wort bzw. dessen Bereichsgrenzen
- ' (Hochkomma) steht für den gesamten aktuellen BLOCK bzw. dessen Bereichsgrenzen
- / (Schrägstrich) steht für den gesamten aktuellen RECORD bzw. dessen Bereichsgrenzen
- ! (Ausrufezeichen) steht für die Position der Standard-Marke

Alle diese Sonderzeichen mit Ausnahme des Ausrufezeichens können Sie als Einzelzeichen für eine Bereichsangabe verwenden. Aufgrund ihrer Beschaffenheit legen sie als Einzelzeichen bereits einen Bereichstyp und dessen Bereichsgrenzen fest.

Als Argument für eine der beiden Bereichsgrenzen bestimmen die obigen Sonderzeichen, abhängig vom Symbol für den Bereichstyp (-, : oder ~) und von der Stellung als erste oder Bereichsgrenze, eine konkrete Position in der aktuellen Datei (siehe Absatz 3.9 Die Bereichsdefinition als Kommandoattribut).

#### Anmerkung

- Ein Absatz ist eine zusammenhängende Texteinheit, die durch den Dateianfang, das Dateende oder Leerzeilen begrenzt ist.
- BLOCK und RECORD sind spezielle STREAM-Bereiche mit vordefinierten Bereichsgrenzen (siehe unten: BLOCK/RECORD als Spezialfall eines STREAM-Bereichs).
- Ein Wort ist eine einzeilige, zusammenhängende Zeichenkette aus alphanumerischen Zeichen, die von Zeilenanfang oder -ende, Leerzeichen oder Sonderzeichen begrenzt ist. Der MAXed betrachtet außerdem Sonderzeichen im Text als einzelne "Wörter".

**RANGE-Bereiche**

Die RANGE-Definition als Zeichenketten-Attribut hat die allgemeine Form

argument1-[argument2]  
 oder  
 [argument1]-argument2  
 oder  
 argument

Sie können hierbei folgende Argumente eingeben:

Angabe	als argument1	als argument2	als argument
n (positive Ganzzahl)	eine ganze Zahl für eine Zeilennummer	Eine ganze Zahl für eine Zeilennummer	Eine ganze Zahl: die Zeilenanzahl ab der akt. Zeile
.	Anfang (Zeile eins) der aktuellen Datei	Ende (letzte Zeile) der aktuellen Datei	Alle Zeilen der aktuellen Datei
\$	Erste Zeile des Ab- schnitts, in dem die Schreibmarke steht	Letzte Zeile des Ab- schnitts, in dem die Schreibmarke steht	Alle Zeilen des aktuellen Absatzs
;	Die aktuelle Zeile	Die aktuelle Zeile	—
IDENTIFIER	Die Zeile mit der durch IDENTIFIER angegebenen Marke	Die Zeile mit der durch IDENTIFIER angegebenen Marke	—
!	Die Zeile mit der Standard-Marke	Die Zeile mit der Standard-Marke	—
PATTERN	Die Zeile mit der ersten Textstelle, die zu PATTERN paßt (Rückwärtssuche)	Die Zeile mit der ersten Textstelle, die zu PATTERN paßt (Vorwärtssuche)	—
' (Hoch- komma)	Erste Zeile des BLOCKs, in dem die Schreibmarke steht	Letzte Zeile des BLOCKs, in dem die Schreibmarke steht	—

**Anmerkung**

- Marken werden durch das MAXed-Kommando LABEL gesetzt (siehe Abschnitt 3.8.4 Markenname als IDENTIFIER, Kapitel 6: LABEL).
- Ein PATTERN ist ein regulärer Ausdruck, der als Suchmuster verwendet wird (siehe Abschnitt 3.10 PATTERN als Kommandoattribut). Wenn Sie ein PATTERN als Argument für die erste Bereichsgrenze angeben, sucht der MAXed von der Markierung der aktuellen Schreibmarkenposition aus rückwärts nach der ersten Textstelle, die zu diesem PATTERN paßt und legt die erste Bereichsgrenze auf die Zeile mit dieser Textstelle. Entsprechend sucht er vorwärts, wenn Sie ein PATTERN als Argument für die zweite Bereichsgrenze verwenden und legt die zweite Bereichsgrenze auf das erste Zeichen dieser Textstelle.

**Beispiel**

<ENTER> -.

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines RANGE-Bereichs, der die Zeilen von der Markierung der aktuellen Schreibmarkenposition bis zum Dateiende umfaßt.

**RECTANGLE-Bereiche**

Die RECTANGLE-Definition als Zeichenketten-Attribut hat die allgemeine Form

argument1: [argument2]  
 oder  
 [argument1]: argument2

Sie können hierbei folgende Argumente eingeben:

Angabe	als argument1	als argument2
.	Anfang (Zeile 1, Spalte 1) der aktuellen Datei	Ende (letzte Zeile, Spaltenzahl der längsten Zeile des Bereichs) der aktuellen Datei
\$	Anfang des Absatzes (erste Zeile, Spalte 1), in dem die Schreibmarke steht	Ende des Absatzes (letzte Zeile, Spaltenzahl der längsten Zeile des Bereichs) in dem die Schreibmarke steht
;	Anfang der aktuellen Zeile	Ende der aktuellen Zeile
IDENTIFIER	Die Position der durch IDENTIFIER angegebenen Marke	Die Position der durch IDENTIFIER angegebenen Marke
!	Die Position der Std.-Marke	Die Position der Std.-Marke
PATTERN	Die Position der ersten Textstelle, die zum angegebenen PATTERN paßt (Rückwärtssuche)	Die Position der ersten Textstelle, die zum angegebenen PATTERN paßt (Vorwärtssuche)
'	Anfang des BLOCKs (erste Zeile, Spalte 1), in dem die Schreibmarke steht	Ende des Blocks (letzte Zeile, Spaltenzahl der längsten Zeile des Bereichs, in dem die Schreibmarke steht

**Anmerkung**

- Bei einer Angabe für *argument2*, die das Ende einer Zeile (;), eines Absatzes (\$) oder eines BLOCKs (') kennzeichnet, wird die rechte Begrenzungslinie auf diejenige Spalte gesetzt, an der die längste Zeile innerhalb des angegebenen RECTANGLE-Bereichs endet. Dadurch ist ausgeschlossen, daß Zeilenenden über die rechte Begrenzungslinie hinausragen.
- Wenn Sie ein PATTERN als Argument für die erste Bereichsgrenze angeben, sucht der MAXed rückwärts und setzt die erste Bereichsgrenze auf das erste Zeichen der nächsten passenden Textstelle. Entsprechend sucht er vorwärts, wenn Sie ein PATTERN als Argument für die zweite Bereichsgrenze verwenden, und legt die zweite Bereichsgrenze auf das erste Zeichen dieser Textstelle.

**Beispiel**

Sie setzen die Standard-Marke:

```
<ENTER> <LABEL>
```

Sie bewegen die Schreibmarke fünf Zeilen nach oben und drei Tabulatorpositionen nach rechts:

```
<ENTER> 5 <UPARROW>
```

```
<ENTER> 3 <TAB>
```

Sie beginnen eine Kommandoattributierung mit der Definition eines RANGE-Bereichs, der seine linke untere Ecke an der Position der Standard-Marke und seine rechte obere Ecke ein Zeichen vor der Markierung der aktuellen Schreibmarkenposition hat mit

```
<ENTER> ! :
```

**STREAM-Bereiche**

Die STREAM-Definition als Zeichenketten-Attribut hat die allgemeine Form

```
argument1 ~ [argument2]
oder
[argument1] ~ argument2
oder
argument
```

Sie können hierbei folgende Argumente eingeben:

Angabe	als argument1	als argument2	als argument
.	Anfang (Zeile 1, Spalte 1) der aktuellen Datei	Ende (letzte Zeile, letzte Spalte) der aktuellen Datei	—
\$	Anfang d. Absatzes (erste Zeile, Spalte 1), in dem die Schreibmarke steht	Ende des Absatzes (letzte Zeile, letzte Spalte), in dem die Schreibmarke steht	—
;	Anfang (Spalte 1) der aktuellen Zeile	Ende (letzte Spalte der aktuellen Zeile	Die aktuelle Zeile ohne Zeilenende-Z.
IDENTIFIER	Die Position der durch IDENTIFIER angegebenen Marke	Die Position der durch IDENTIFIER angegebenen Marke	—
!	Die Position der Standard-Märke	Die Position der Standard-Märke	—
PATTERN	Die Position der ersten Textstelle, die zu PATTERN paßt (Rückwärtssuche)	Die Position der ersten Textstelle, die zu PATTERN paßt (Vorwärtssuche)	—
, (Komma)	Anfang des Wortes, auf dem die Schreibmarke steht	Ende des Wortes, auf dem die Schreibmarke steht	Das Wort, auf dem die Schreibmarke steht

### Anmerkung

STREAM-Bereiche, die sich über mehrere Zeilen erstrecken, beinhalten auch die unsichtbaren Zeilenende-Zeichen der eingeschlossenen Zeilenenden. Einzeilige STREAM-Bereiche, wie z.B. bei Semikolon (;) als *argument*, enthalten keine Zeilenende-Zeichen. Bei bestimmten MAXed-Kommandos (siehe Kapitel 6) sind nur einzeilige STREAM-Bereiche als Kommandoattribute erlaubt.

### Beispiel

```
<ENTER> $~
```

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines STREAM-Bereichs, der von der ersten Spalte der ersten Zeile des Absatzes, in dem sich die Markierung der aktuellen Schreibmarkenposition befindet, bis ein Zeichen vor dieser Markierung reicht.

### BLOCK als Spezialfall eines STREAM-Bereichs

Ein BLOCK-Bereich kann nur als Zeichenketten-Attribut oder als Kombination von Zeichenketten-Attribut und aktueller Position der Schreibmarke definiert werden (siehe oben).

Die Bereichsgrenzen eines BLOCKs sind in der Konfigurationsdatei des MAXed (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen) festgelegt. Standardmäßig stellt ein BLOCK eine Texteinheit zwischen paarweise zusammengehörenden, öffnenden und schließenden Klammern dar.

In der Standard-Konfiguration des MAXed sind folgende Klammerpaare als Grenzen für BLOCK-Bereiche definiert:

- normale Klammern:       ( blockbereich )
- geschweifte Klammern:  { blockbereich }
- eckige Klammern:       [ blockbereich ]

Eine BLOCK-Definition bezieht sich immer auf die kleinste umgebende BLOCK-Struktur, innerhalb derer sich die Schreibmarke momentan befindet. Bevor der MAXed ein Kommando auf einem BLOCK-Bereich ausführt, überprüft er die Konsistenz der angesprochenen BLOCK-Struktur, also des kleinsten umgebenden BLOCKs. Falls die Klammerverschachtelung nicht korrekt ist, erhalten Sie auf der Steuerzeile eine Fehlermeldung. Wenn Sie nur Kommandos, die mit BLOCK-Bereichen arbeiten, ausführen, erhalten Sie die Konsistenz der angesprochenen BLOCK-Strukturen.

**Anmerkung**

Bei der Definition eines BLOCK-Bereichs können Klammern immer nur paarweise eingeschlossen sein, da der MAXed auf die Konsistenz der BLOCK-Struktur Wert legt.

Die BLOCK-Definition als Zeichenketten-Attribut hat die allgemeine Form

```
argument1 ~ [argument2]
oder
[argumen1] ~ argument2
oder
argument
```

Sie können hierbei folgende Argumente eingeben:

	als argument1	als argument2	als argument
' (Hochkomma)	Anfang des kleinsten umgebenden BLOCKs (Zeichen nach der öffnenden Klammer)	Ende des kleinsten umgebenden BLOCKs (Zeichen vor schließender Klammer)	Der gesamte, kleinste umgebende BLOCK (Klammern mit eingeschlossen)
'' (zwei Hochkommata)	—	—	Der gesamte, kleinste umgebende BLOCK (Klammern nicht mit eingeschlossen)

**Beispiel**

Die Schreibmarke befindet sich innerhalb eines Textes, in dem öffnende und schließende Klammern vorkommen. Sie setzen die Schreibmarke auf eine öffnende Klammer und geben ein:

```
<ENTER> ' ~ '
```

Damit beginnen Sie eine Kommandoattributierung mit der Definition eines BLOCK-Bereichs. Sie geben einen IDENTIFIER als zweites Attribut ein und schließen das Kommando ab durch:

```
puf1 <PICK>
```

Der MAXed sucht zuerst nach einem Klammerpaar, das die Markierung der aktuellen Schreibmarkenposition umgibt. Dabei überprüft er gleichzeitig die Klammerstruktur innerhalb dieses kleinsten umgebenden BLOCKs auf Konsistenz. Falls er keine weitere öffnende Klammer links der Markierung findet oder die dazu korrespondierende schließende rechte Klammer fehlt, gibt er eine Fehlermeldung aus. Wenn die Klammerstruktur korrekt ist, wird der gesamte Text innerhalb des umgebenden Klammerpaars (ohne die Klammern) in den PICK-Puffer *puf1* kopiert.

## RECORD als Spezialfall eines STREAM-Bereichs

Ein RECORD-Bereich kann nur als Zeichenketten-Attribut oder als Kombination von Zeichenketten-Attribut und aktueller Position der Schreibmarke definiert werden (siehe oben).

Die Bereichsgrenzen eines RECORDs sind in der Konfigurationsdatei des MAXed festgelegt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

In der Standard-Konfiguration des MAXed sind folgende Grenzen für RECORD-Bereiche definiert:

- Im Standard-Einfüge-Modus INSERT gilt  
der Anfang der aktuellen Zeile als RECORD-Anfang und  
das letzte Semikolon (;) der aktuellen Zeile als RECORD-Ende  
(siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: INSERT).
- Im LEXICAL-Modus dagegen gilt  
der vorausgehende Punkt bzw. der Dateianfang als RECORD-Anfang und  
der nachfolgende Punkt (.) als RECORD-Ende.

Eine RECORD-Definition bezieht sich immer auf die Umgebung der aktuellen Position der Schreibmarke. Dies kann beim alternativ verfügbaren Standard-Einfüge-Modus INSERT z.B. die Anweisung eines C-Quellprogramms bedeuten, auf der die Schreibmarke momentan positioniert ist, oder im LEXICAL-Modus der Satz, in dem die Schreibmarke steht (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Die RECORD-Definition als Zeichenketten-Attribut hat die allgemeine Form

```
argument1~[argument2]
oder
[argument1]~argument2
oder
argument
```

Sie können hierbei folgende Argumente eingeben:

	als argument1	als argument2	als argument
/ (Schrägstrich)	INSERT-Modus: Anfang der aktuellen Zeile; LEXICAL-Modus: Anfang des aktuellen Satzes (Zeichen nach dem letzten Punkt bzw. Dateianfang)	INSERT-Modus: das Zeichen vor dem letzten Semikolon der aktuellen Zeile; LEXICAL-Modus: Ende des aktuellen Satzes (inklusive der nächste Punkt)	INSERT-Modus: die gesamte aktuelle Anweisung (aktuelle Zeile vom Anfang bis zum letzten Semikolon); LEXICAL-Modus: der gesamte akt. Satz (inkl. Schlußpunkt)

### Anmerkung

- Beachten Sie, daß der MAXed bei eingestelltem Überschreibe-Modus die Festlegung des RECORD-Bereich entsprechend dem alternativ verfügbaren Einfüge-Modus vornimmt (siehe Abschnitt 4.1.1 Die Eingabemodi, Kapitel 6: INSERTMODE).
- Beachten Sie beim INSERT-Modus, daß ein RECORD immer der gesamte Text vom Zeilenanfang bis zum letzten Semikolon der aktuellen Zeile bedeutet, ungeachtet der Spalte, auf der die Schreibmarke momentan steht und ungeachtet weiterer Semikolons in der Zeile.
- Beachten Sie beim LEXICAL-Modus daß ein RECORD immer nur den Text bis zum nächsten Punkt (das muß kein Satzzeichen sein!) umfaßt.

### Beispiel

Die Schreibmarke befindet sich innerhalb eines Satzes, der mit Punkt abgeschlossen ist. Sie schalten als Eingabe-Modus den LEXICAL-Modus ein:

```
<ENTER> 1 <INSERTMODE>
```

Danach geben Sie ein:

```
<ENTER> / puf1 <ERASE>
```

Der MAXed löscht den gesamten RECORD-Bereich, in dem sich die Markierung der aktuellen Schreibmarkenposition befindet und kopiert ihn in den ERASE-Puffer *puf1*. In diesem Fall löscht er einen ganzen Satz inklusive des abschließenden Punktes.

### 3.10 PATTERN als Kommandoattribut

Ein PATTERN benutzen Sie als Muster, mit dem Sie dazu passende Textstellen in einer Datei suchen und durch eine Ersetzungszeichenkette ersetzen können (siehe Kapitel 6: MINUSFIND, PLUSFIND, Kapitel 6: REPLACE; siehe auch Kapitel 6: EDIT, ENTER, HWINDOW, PUT, VWINDOW).

Das aktuelle PATTERN

ist dasjenige PATTERN, das Sie zuletzt bei einem der oben genannten Such-Kommando eingegeben haben (siehe auch Anmerkung). Der MAXed greift auf das aktuelle PATTERN beim Aufruf eines dieser Kommandos zurück, wenn Sie dabei kein neues PATTERN eingeben (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen). Das aktuelle PATTERN wird in der Pseudo-Datei SETUP als Belegung des Parameters FINDPATTERN angezeigt, wo Sie es auch ändern können (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

#### *Anmerkung*

- Ein PATTERN, das Sie beim Kommando GOTO als Kommandoattribut zur Erstellung der Pseudo-Datei SCANLIST eingeben, setzt in der Pseudo-Datei SETUP den Parameter SCANPATTERN um. Das aktuelle PATTERN des Parameters FINDPATTERN wird erst dann durch dieses PATTERN ersetzt, wenn Sie aus SCANLIST mit <GOTO> eine gefundene Textstelle anspringen.
- Nach einem Formatierkommando FILL oder FORMAT im LEXICAL-Modus (siehe Abschnitte 4.3.3 Flatter- und Blocksatz, 4.1.1 Die Eingabe-Modi) wird das aktuelle PATTERN des Parameters FINDPATTERN (siehe oben) durch das Silbentrennzeichen – ersetzt, sodaß Sie sofort mit <PLUSFIND> oder <MINUSFIND> die Trennstellen auffinden können, wenn Sie die Trennkünste des MAXed kontrollieren wollen.

PATTERN als Bereichsargument:

Außerdem haben Sie die Möglichkeit, PATTERN bei Bereichsdefinitionen als Begrenzungsargumente zu verwenden (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut). Hierbei sucht der MAXed in der aktuellen Datei von der aktuellen Schreibmarkenposition aus nach der nächsten passenden Textstelle und legt diese als Bereichsgrenze fest.

## Passende Texteinheiten bei der Suche

Der MAXed unterscheidet folgende Texteinheiten, die Sie in einer Datei mit Hilfe eines PATTERNS suchen können:

### Ein Wort

ist eine Zeichenkette ohne Leerzeichen, die links und rechts durch den Zeilenanfang bzw. das Zeilenende, durch Leerzeichen oder Sonderzeichen begrenzt ist. Zu den Sonderzeichen zählen alle Zeichen außer den alphanumerischen Zeichen, dem Unterstrich `_` und dem Leerzeichen. Standardmäßig findet der MAXed zu einem gegebenen PATTERN die passenden Textstellen im Wortkontext.

### Eine Zeichenkette

ist eine Folge zusammenhängender, beliebiger Zeichen. Der MAXed kann zu geeigneten PATTERN alle Zeichenketten finden, die zwischen Anfang und Ende einer Zeile liegen.

### Anmerkung

- Ein Sonderzeichen innerhalb eines beliebigen Textes betrachtet der MAXed unabhängig vom Kontext auch als Wort.
- Weiterhin kann der MAXed auch verkettete Wörter, also eine beliebige Zeichenkette aus alphanumerischen Zeichen und Sonderzeichen, aber ohne Leerzeichen, standardmäßig als Texteinheit im Wortkontext finden.

### Beispiel 1

Mit dem Suchkommando

```
<ENTER> wort <PLUSFIND>
```

würden Sie im Standard-Such-Modus das Wort *wort* z.B. in folgenden Textzeilen finden (siehe unten: Verschiedene Einstellungen des Such-Modus):

```
Nur ein Wort noch,  
ein Sprich-Wort:  
Am Anfang war das WORT.
```

Mit obigem Kommando würden Sie jedoch in folgenden Textzeilen das Wort nicht mehr finden:

Er ist wortkarg;  
denn Worte sind wie Schall und Rauch.

Hierbei müßten Sie eine sogenannte Kontextangabe machen (siehe unten), z.B.:

<ENTER> wort> <PLUSFIND>

## Beispiel 2

Wenn Sie nach Sonderzeichen suchen, müssen Sie diese im Standard-Such-Modus mit einem vorausgehenden Gegenschrägstrich \ entwerfen. Mit dem Suchkommando

<ENTER> \! <PLUSFIND>

finden Sie das Ausrufezeichen z.B. in der Textzeile

Mach' keine Worte!

Auch in der folgenden Textzeile finden Sie damit jedes der drei Ausrufezeichen:

WORTE!- Nichts als Worte!!

Das Suchkommando

<ENTER> worte\! <PLUSFIND>

könnte in der eben genannten Textzeile immerhin noch zwei passende Textstellen finden. Bei der Textzeile

Nur Schlagworte!

und dem zuletzt genannten Suchkommando würde der MAXed jedoch nichts passendes entdecken. Dazu müßten Sie ihm erlauben, den Wortkontext bei der Suche zu vernachlässigen! Hierzu stellen Sie zuvor den Suchmodus entsprechend ein (siehe unten: Verschiedene Suchmodi) oder erweitern das Suchmuster explizit mit der entsprechenden Kontextangabe:

<ENTER> <worte\! <PLUSFIND>

## Der Aufbau eines PATTERNS

Ein PATTERN geben Sie im Eingabefeld der Steuerzeile als Kommandoattribut ein:

Im einfachsten Fall

besteht es aus einem Wort, und stellt damit ein im Klartext formuliertes Suchmuster für ein dazu passendes Wort in einer Datei dar. Bei der Suche berücksichtigt der MAXed allerdings nicht, ob die einzelnen Buchstaben groß oder klein geschrieben sind.

Im allgemeinen Fall

ist ein PATTERN eine Zeichenkette, die Sie in Anführungszeichen setzen müssen, wenn sie Leerzeichen enthält. Standardmäßig haben hierbei jedoch die Sonderzeichen eine spezielle Bedeutung, wenn Sie sie nicht explizit entwerten (siehe unten). Wenn Sie innerhalb eines PATTERNS Sonderzeichen einsetzen, müssen Sie gewisse Regeln beachten:

Ein PATTERN ist ein regulärer Ausdruck bzw. ein erweiterter regulärer Ausdruck. Unter Verwendung von Sonderzeichen erstellen Sie Suchmuster als Schablonen, mit denen Sie alle Textstellen finden können, die die angegebene spezifische Gemeinsamkeit haben.

Ein regulärer Ausdruck

kann sowohl ein einzelnes Zeichen als auch eine korrekt aufgebaute Zeichenkette sein. Dazu beachten Sie die Einsatzmöglichkeiten der unten aufgeführten Sonderzeichen. Innerhalb eines PATTERNS können Sie mit runden Klammern reguläre Ausdrücke gruppieren und mit sogenannten Operatoren versehen. Solche Gruppen von regulären Ausdrücken dürfen auch ineinander geschachtelt auftreten (siehe Abschnitt 3.10.1 Die Sonderzeichen des Standard-Suchmodus).

Ein erweiterter regulärer Ausdruck

besteht aus mehreren, alternativen Suchmustern, die, durch senkrechte Striche | getrennt, zu einem PATTERN zusammengefaßt sind (siehe Abschnitt 3.10.2 Erweiterte reguläre Ausdrücke).

### 3.10.1 Die Sonderzeichen des Standard-Suchmodus

Nachstehend sind alle Sonderzeichen aufgeführt, die standardmäßig innerhalb eines PATTERNs eine spezielle Bedeutung haben.

- \ der Gegenschrägstrich hat innerhalb eines PATTERN die Funktion eines Umschalters für das nachfolgende Zeichen:
- ein nachfolgendes Sonderzeichen (auch ein Gegenschrägstrich) wird entwertet, verliert also seine Sonderbedeutung. Entwerten Sie grundsätzlich alle Sonderzeichen innerhalb eines PATTERNs, die innerhalb der gesuchten Textstelle als Klartext auftreten.
  - ein nachfolgender Buchstabe erhält sein Merkmal als Groß- oder Kleinbuchstabe zurück. Standardmäßig wird bei der Suche nach einer passenden Textstelle Groß- oder Kleinschreibung nicht berücksichtigt.

”mu ster”

Anführungszeichen müssen ein PATTERN einschließen, das Leerzeichen enthält. Dies ist notwendig, damit das PATTERN vom MAXed als einzelnes Zeichenketten-Attribut erkannt werden kann (siehe 3.6.1 Das Format der Kommandoattributierung). Auch als erweiterter regulärer Ausdruck muß das gesamte PATTERN durch Anführungszeichen eingeschlossen sein, wenn es Leerzeichen enthält. Ein Anführungszeichen innerhalb eines PATTERN ist nur zulässig, wenn es durch den Gegenschrägstrich \ entwertet ist.

#### Beispiel 1

Zum Pattern `\a\)`  
paßt die Zeichenkette `a)`, jedoch nicht `A)`.

#### Beispiel 2

Zum Pattern `”Zu \”\a\)\””`  
passen die Zeichenketten `Zu ”a)”` und `zU ”a)”` usw.

```
linker_kontext < muster  
bzw.  
muster > rechter_kontext  
bzw.  
linker_kontext < muster > rechter_kontext
```

Mit Hilfe der öffnenden bzw. der schließenden spitzen Klammer können Sie in einem PATTERN den linken bzw. rechten Kontext angeben, in dem eine Zeichenkette steht, die zu dem Muster *muster* passen soll. Als Kontext dürfen Sie ein beliebiges Suchmuster für eine weitere Zeichenkette angeben, die sich direkt links bzw. rechts der gesuchten Zeichenkette befinden soll. Wenn Sie den linken bzw. den rechten Kontext nicht angeben – die öffnende spitze Klammer ist also das erste Zeichen bzw. die schließende spitze Klammer das letzte Zeichen eines Suchmusters – so ist der entsprechende Kontext der aufzufindenden Zeichenkette beliebig.

Sie müssen die spitzen Klammern insbesondere dann angeben, wenn die gesuchte Zeichenkette auch innerhalb eines Wortes gefunden werden soll. Sie sind aber auch dann sinnvoll, wenn Sie die aufzufindenden Textstellen näher beschreiben wollen, indem Sie zusätzlich die direkte Umgebung (den Kontext) in das PATTERN mit aufnehmen. Wenn der MAXed eine passende Textstelle findet, wird die Schreibmarke auf das erste Zeichen der zu *muster* passenden Teil-Zeichenkette gesetzt. Wenn Sie ein Ersetzungskommando ausführen, wird lediglich die zu *muster* passende Teil-Zeichenkette ersetzt (siehe Kapitel 6: REPLACE).

#### Beispiel 1

```
<er>  
paßt zu jeder Zeichenkette er mit beliebigem rechten und  
linkem Kontext (z.B. zweimal in Er-dbe-er-en ).
```

#### Beispiel 2

```
<er>ker  
paßt nur zu der Teilzeichenkette er , die z.B. in einem Wort  
K-er-ker mit dem rechten Kontext ker auftaucht.
```

*Beispiel 3*

`<h<er>b>`

passt nur zu der Teilzeichenkette *er*, die in einer Zeichenkette *h-er-b* auftaucht, wobei allerdings der weitere Kontext beliebig ist! (z.B. in *Sch-er-ben* oder *H-er-bst*).

*Beispiel 4*

`<ENTER> <\p\h> f . <REPLACE>`

Dieses Kommando ersetzt in der gesamten aktuellen Datei (.) alle vorkommenden *ph* (klein geschrieben!) durch *f* (siehe Abschnitte 3.9.2 Die Bereichsdefinition durch Zeichenkettenargument, 3.11 Die Ersetzungszeichenkette als Kommandoattribut).

Die beiden Sonderzeichen Dach und Semikolon stellen in einem PATTERN lediglich einen Kontext der aufzufindenden Textstellen dar: Zeilenanfang bzw. Zeilenende.

- ^ Ein Dach am Anfang eines regulären Ausdrucks bedeutet den Anfang einer Zeile. Ein Dach innerhalb eines Suchmusters ist nur sinnvoll, wenn es durch den Gegenschrägstrich \ entwertet ist.
- ; Ein Semikolon am Ende eines regulären Ausdruckes bedeutet das Ende einer Zeile. Ein Semikolon innerhalb eines Suchmusters ist nur sinnvoll, wenn es durch den Gegenschrägstrich \ entwertet ist.
- ^; Die Kombination von Dach und Semikolon ist das Suchmuster für eine Leerzeile. Genau genommen findet der MAXed die leere Zeichenkette zwischen Zeilenanfang und unsichtbarem Zeilenende-Zeichen. Die Schreibmarke wird dann an den Anfang einer gefundenen Leerzeile gesetzt.

*Beispiel 1*

`"^alpha bis omega;"`

paßt zu jeder Zeichenkette mit dem Text *alpha bis omega* (Buchstaben beliebig in Groß- oder Kleinschreibung), die vom Anfang bis zum Ende einer Zeile reicht.

*Beispiel 2*

Mit dem Ersetzungskommando

<ENTER> ^; | . <REPLACE>

löschen Sie alle Leerzeilen (^;) in der aktuellen Datei (.) (siehe Abschnitt 3.11 Die Ersetzungszeichenkette als Kommandoattribut, Kapitel 6: REPLACE). Hierbei bewirkt die Ersetzungszeichenkette (|) das Löschen des Kontextes: Zeilenende-Zeichen.

Die nachstehenden Ausdrücke sind innerhalb eines PATTERNS "Platzhalter" für einzelne Zeichen:

- Ein Punkt bedeutet ein beliebiges Zeichen mit Ausnahme des Zeilenende-Zeichens.
- Ein Bindestrich bedeutet das Zeichen direkt vor einem Silbentrennzeichen am Ende einer Zeil). Das Silbentrennzeichen wird vom MAXed selbst im LEXICAL-Modus erzeugt.
- \- Der entwertete Bindestrich bedeutet jeden Bindestrich außer dem Silbentrennzeichen. Auf diese Unterscheidung ist zu achten!

[*zeichenmenge*]

Eckige Klammern um eine Zeichenkette *zeichenmenge* bedeuten genau ein Zeichen dieser Menge von Zeichen und kein anderes. Groß- und Kleinbuchstaben werden innerhalb der eckigen Klammern immer unterschieden. Der Gegenschrägstrich hat innerhalb der eckigen Klammern nur Entwertungsfunktion für Sonderzeichen.

[*z1-z2*]

Befindet sich ein Bindestrich – in der Mitte zweier Zeichen *z1* und *z2* in eckigen Klammern, so wird dies als Ausschnitt *z1* bis *z2* des ASCII-Zeichensatzes interpretiert. Der ganze Ausdruck bedeutet dann genau: ein Zeichen aus dem Ausschnitt *z1* bis *z2* und kein anderes.

[z1-z2,z3-z4, ... ]

Sie können mehrere Abschnitte des ASCII-Zeichensatzes innerhalb eckiger Klammern angeben, indem Sie diese durch Kommata trennen. Der ganze Ausdruck bedeutet dann genau: ein Zeichen aus einem der angegebenen Abschnitte und kein anderes.

[ ^ zeichenmenge, ... , abschnitte, ... ]

Ist das erste Zeichen einer Zeichenspezifikation (Zeichenmengen oder Abschnitte, durch Kommata getrennt) in eckigen Klammern jedoch ein Dach ^, bedeutet der ganze Ausdruck genau ein beliebiges Zeichen mit Ausnahme der in der Klammer spezifizierten Zeichen und des Zeilenende-Zeichens.

#### Beispiel 1

[aAxY] bedeutet genau eines der Zeichen a, A, x oder Y.

#### Beispiel 2

[a-j] entspricht dem PATTERN [abcdefghij]

#### Beispiel 3

[a-c,K-N,1-4] entspricht dem PATTERN [abcKLMN1234]

Mit Hilfe der folgenden Sonderzeichen können Sie reguläre Ausdrücke innerhalb eines PATTERNs gruppieren und mit sogenannten Operatoren versehen. Diese Operatoren beziehen sich immer auf den direkt vorangehenden regulären Ausdruck: Das ist entweder ein einzelnes Zeichen oder eine geklammerte Zeichenkette. Ein Operator gibt an, wieviel Wiederholungen einer Zeichenkette, die Teil einer aufzufindenden Textstelle ist und zu dem angegebenen regulären Ausdruck paßt, in direkter Folge vorkommen. Geklammerte reguläre Ausdrücke können Sie auch ineinander verschachteln.

(reg\_ausdr)

Mit runden Klammern können Sie einen regulären Ausdruck als Einheit zusammenfassen. Die Klammern haben nur die Aufgabe, den regulären Ausdruck zu gruppieren, damit sich ein direkt folgender Operator auf die ganze geklammerte Einheit beziehen kann (siehe unten).

- \* Ein Stern, den Sie im Anschluss an einen regulären Ausdruck eingeben, bedeutet keine oder mehrere direkt aufeinander folgende Zeichenketten im Text, die zu dem angegebenen regulären Ausdruck passen.
- + Ein Plus-Zeichen, den Sie im Anschluss an einen regulären Ausdruck eingeben, bedeutet eine oder mehrere direkt aufeinander folgende Zeichenketten im Text, die zu dem angegebenen regulären Ausdruck passen.
- {n} Der Ausdruck mit *n* als ganzer Zahl zwischen 0 und 255, die einem regulären Ausdruck folgt, bedeutet genau *n* direkt aufeinander folgende Zeichenketten im Text, die zu dem angegebenen regulären Ausdruck passen.
- {min,} Der Ausdruck mit *min* als ganzer Zahl zwischen 0 und 255, die einem regulären Ausdruck folgt, bedeutet *min* oder mehr direkt aufeinander folgende Zeichenketten im Text, die zu dem angegebenen regulären Ausdruck passen.
- {min,max} Der Ausdruck mit *min* und *max* als ganze Zahlen zwischen 0 und 255, der einem regulären Ausdruck folgt, bedeutet wenigstens *min* und höchstens *max* direkt aufeinander folgende Zeichenketten im Text, die zu dem angegebenen regulären Ausdruck passen.

*Beispiel 1*

`h*(ui*)+`

Dieser Ausdruck entspricht dann u.a. den nachstehenden Zeichenketten: *u, uu, ..., ui, uiui, ..., uii, uiuiii, ..., hu, hhu, ..., huu, hhuu, ..., hui, hhui, ..., huiui, hhuiui, ...*

*Beispiel 2*

`(u*(ri+)){3}`

Dieser Ausdruck entspricht genau drei Folgen des regulären Ausdrucks im vorhergehenden Klammerpaar: *riri, riiri, ..., uriuri, uriuriuri, ..., uuriuuri, uuriuuriuuri, ...*

*Beispiel 3*

<ENTER> "[großklein]+ u.\* (sch){0,1}mächtig" <PLUSFIND>

Dieses Kommando durchsucht die aktuelle Datei vorwärts bis zum ersten Auftreten einer passenden Textstelle. Z.B. würden die Textstellen *groß und mächtig* oder *klein und schwächlich* gefunden.

**Anmerkung**

Beachten Sie, daß ein PATTERN, zu dem auch die leere Zeichenkette passen würde, nicht sinnvoll eingesetzt werden kann. Diese Situation tritt z.B. dann ein, wenn alle regulären Ausdrücke eines Suchmusters mit entsprechenden Operatoren (siehe oben) versehen sind, die auch die leere Zeichenkette als passende Zeichenkette zulassen. Die leere Zeichenkette paßt z.B. zu allen regulären Ausdrücken mit den Operatoren \* oder {0}.

### 3.10.2 Erweiterte reguläre Ausdrücke

Als erweiterter regulärer Ausdruck hat PATTERN die Form

```
reg_ausdr1|reg_ausdr2[|reg_ausdr3 ...]
```

Der senkrechte Strich trennt zwei alternative Suchmuster innerhalb eines PATTERNS, wobei jedes Suchmuster ein regulärer Ausdruck ist. Eine Zeichenkette, die mit *reg\_ausdr1* oder *reg\_ausdr2* usw. übereinstimmt, stimmt also mit PATTERN überein.

#### Beispiel 1

```
<ENTER> dick|doof <PLUSFIND>
```

Dieses Kommando durchsucht die aktuelle Datei vorwärts bis zum ersten Auftreten eines der Worte *dick* oder *doof*, wobei die einzelnen Buchstaben groß oder klein geschrieben sein dürfen.

#### Beispiel 2

```
" \- | \(|)"
```

Ein Suchkommando mit diesem PATTERN findet alleinstehende Bindestriche (\-) sowie alle öffnenden und schließenden runden Klammern ( oder ).

### 3.10.3 Textersetzung innerhalb von PATTERN

Selbstverständlich können Sie innerhalb eines PATTERNs Textersetzungen vornehmen lassen (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld). Im Eingabefeld der Steuerzeile bewirken die folgenden Sonderzeichen beim Auslösen eines Kommandos noch vor dessen Ausführung, daß die nachfolgenden bzw. eingeschlossenen (Teil-)Zeichenketten durch ihren Wert, ihren Inhalt bzw. ihre Ausgabe ersetzt werden:

- @ Der Klammeraffe geht einem Parameter für Textersetzung voraus.
- \$ Das Dollar-Zeichen geht dem Namen einer MAXed-Variablen voraus und schließt ihn optional ab.
- % Das Prozent-Zeichen geht einem PICK-Puffernamen voraus und schließt ihn optional ab. Den Standard-PICK-Puffer geben Sie durch % bzw. %% an (siehe Abschnitt 3.8.3 Der PICK-Puffernamen als IDENTIFIER).
- `SINIX-kommando` Gegenhochkomma schließen ein SINIX-Kommando ein (siehe Abschnitt 4.5 Die SINIX-Schnittstellen). Das zuletzt ausgeführte SINIX-Kommando können Sie wiederholt mit ` bzw. `` ausführen lassen.

#### Beispiel 1

```
<ENTER> <@r> <PLUSFIND>
```

Mit diesem Kommando suchen Sie (im Standard-Such-Modus) im Anschluß an ein Ersetzungskommando die dabei verwendete Ersetzungszeichenkette im Text. Wenn die Ersetzungszeichenkette allerdings Sonderzeichen enthält, sollten Sie diese erst entwerten, da sie sonst die entsprechenden Sonderbedeutungen innerhalb des PATTERNs bekommen:

```
<ENTER> = @r <ENTER>
```

Damit lassen Sie den Parameter im Eingabefeld erst auswerten, holen sich die expandierte Zeichenkette ins Eingabefeld zurück (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld):

```
<ENTER> <BACKSPACE>
```

Sie löschen das Gleichheitszeichen = und nehmen die entsprechenden Änderungen vor, bevor Sie <PLUSFIND> betätigen.

## Beispiel 2

Haben Sie zuletzt das PATTERN  
<h<er>b

eingetragen, so hat der Parameter @p momentan genau diesen Wert.  
Das PATTERN  
@pst

wird nach Eingabe eines Kommandos erweitert zu  
<h<er>bst

Damit finden Sie z.B. die Teilzeichenkette *er* im Wort *Späth-er-bst*.

## Anmerkung

Beachten Sie, daß ein durch Textersetzung expandiertes PATTERN zur Kommandoausführung u.U. Sonderzeichen, die für das PATTERN gültig sind, oder Leerzeichen erhalten kann. In letzterem Fall muß das ganze PATTERN in Anführungszeichen "" stehen (siehe oben).

### 3.10.4 Verschiedene Einstellungen des Such-Modus

Die oben beschriebene Arbeitsweise des MAXed bei Verwendung eines PATTERNs als Kommandoattribut entspricht der Arbeitsweise im Standard-Such-Modus. Dies ist der intelligenteste Such-Modus, den Sie einstellen können:

- Der MAXed sucht im Standard-Such-Modus passende Textstellen prinzipiell im Wortkontext, sofern Sie bei einem PATTERN nicht explizit den linken bzw. rechten Kontext angeben.
- Die nicht-alphanumerischen Zeichen haben i.a. eine Sonderbedeutung und müssen ggf. innerhalb eines PATTERNs entwertet werden.

Daneben können Sie noch vier weitere Such-Modi einstellen, die Ihnen für bestimmte Zwecke die Eingabe von PATTERN als Kommandoattribute erleichtern (siehe Abschnitt 4.1.5 Der Such-Modus).

### 3.11 Die Ersetzungszeichenkette als Kommandoattribut

Eine Ersetzungszeichenkette geben Sie beim Ersetzungskommando REPLACE beim Aufruf als Zeichenketten-Attribut an (siehe Kapitel 6: REPLACE). Alle Textstellen, die innerhalb des definierten Bereichs zum verwendeten PATTERN passen, werden durch die Ersetzungszeichenkette ersetzt (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).

#### Die aktuelle Ersetzungszeichenkette

ist diejenige Ersetzungszeichenkette, die Sie zuletzt bei einem Aufruf des MAXed-Kommandos REPLACE eingegeben haben. Der MAXed greift auf die aktuelle Ersetzungszeichenkette beim Aufruf des Kommandos REPLACE zurück, wenn Sie dabei keine neue Ersetzungszeichenkette eingeben (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen). Die aktuelle Ersetzungszeichenkette wird in der Pseudo-Datei SETUP als Belegung des Parameters REPLACESTRING angezeigt, in der Sie sie auch ändern können (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

Eine Ersetzungszeichenkette geben Sie im Eingabefeld der Steuerzeile als Kommandoattribut ein:

#### Im einfachsten Fall

besteht sie aus einem Wort, und stellt damit eine im Klartext formulierte Ersetzung für eine Textstelle dar, die zum verwendeten PATTERN paßt.

#### Im allgemeinen Fall

ist sie eine Zeichenkette, die Sie in Anführungszeichen setzen müssen, wenn sie Leerzeichen enthält. Hierbei haben jedoch einige Sonderzeichen eine spezielle Bedeutung, wenn Sie diese nicht explizit entwerten (siehe unten). Wenn Sie innerhalb einer Ersetzungszeichenkette Sonderzeichen einsetzen, müssen Sie gewisse Regeln beachten: Eine Ersetzungszeichenkette ist, ähnlich wie ein PATTERN, ein regulärer Ausdruck. Unter Verwendung von Sonderzeichen können Sie auch Ersetzungsmuster versehen mit sogenannten "Zählern" erstellen, durch die jene Textstellen, die zum angegebenen PATTERN passen, in der Reihenfolge ihres Auftretens versehen werden.

**Anmerkung**

Beachten Sie bei der Kommandoattributierung des Ersetzungskommandos REPLACE, daß die Bereichsdefinition durch Bewegen der Schreibmarke wie üblich als erstes eingegeben wird, während sie in Form eines Zeichenketten-Attributs als letztes angegeben werden muß!

**Beispiel**

Die Schreibmarke sitzt in der aktuellen Datei auf der ersten der drei folgenden Zeilen:

Nur ein Wort noch,  
ein Sprichwort:  
Am Anfang war das WORT.

Mit dem Ersetzungskommando

<ENTER> wort Bild 3 <REPLACE>

können Sie im Standard-Such-Modus in diesen drei Zeilen alle zum PATTERN wort passenden Textstellen durch die Ersetzungszeichenkette Bild ersetzen (siehe Abschnitt 4.1.5 Der Such-Modus). Das Resultat sieht dann so aus:

Nur ein Bild noch,  
ein Sprichwort:  
Am Anfang war das Bild.

Eine inhaltlich weniger aussagekräftige Variante, die jedoch verdeutlicht, wie im angegebenen RANGE-Bereich alle zu wort passenden Textstellen (nämlich zwei) entfernt und an ihrer Stelle die Ersetzungszeichenkette Bild eingefügt werden (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut). Die Ersetzung machen Sie danach wieder rückgängig mit

<UNDO>

Sowohl das PATTERN als auch die Ersetzungszeichenkette wollen Sie jetzt etwas komplizierter gestalten. Die Schreibmarke sitzt wieder auf der ersten der drei Zeilen, und Sie definieren den gleichen RANGE-Bereich durch Bewegen der Schreibmarke:

<ENTER> <DOWNARROW> <DOWNARROW> <DOWNARROW>

<wort weiß\-\-"NICHTS\-" <REPLACE>

Die Sonderzeichen Anführungszeichen " und Bindestrich – in der Ersetzungszeichenkette sind durch den Gegenschrägstrich entwertet, der ansonsten keine weitere Bedeutung hat. Das Resultat (obwohl dem

Original um "NICHTS" überlegen) weist jetzt immerhin drei Ersetzungen auf:

Nur ein weiß-"NICHTS" noch,  
ein Sprichweiß-"NICHTS":  
Am Anfang war das weiß-"NICHTS".



### 3.11.1 Die Sonderzeichen der Ersetzungszeichenkette

Nachstehend sind alle Sonderzeichen aufgeführt, die innerhalb einer Ersetzungszeichenkette bzw. als Ersetzungszeichenkette eine spezielle Bedeutung haben.

\ Der Gegenschrägstrich hat innerhalb einer Ersetzungszeichenkette die Funktion eines Umschalters für das nachfolgende Zeichen: Ein nachfolgendes Sonderzeichen (auch ein Gegenschrägstrich) wird entwertet, verliert also seine Sonderbedeutung. Entwerten Sie grundsätzlich alle Sonderzeichen, die innerhalb einer Ersetzungszeichenkette eine Sonderbedeutung haben und die Sie als Klartext eingeben möchten.



#### *Anmerkung*

Die Sonderzeichen, die Sie in Ersetzungszeichenketten entwerten sollten, um sie als Klartext eingeben zu können, umfassen drei Gruppen:

- die hier aufgeführten Sonderzeichen für Ersetzungszeichenketten: Gegenschrägstrich \, Anführungszeichen ", senkrechter Strich | und die geschweiften Klammern { und }
- die Sonderzeichen zur Textersetzung (siehe Abschnitt 3.11.2 Textersetzung innerhalb von Ersetzungszeichenketten)
- die Sonderzeichen für Bereichsdefinitionen (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut)



”er setzung”

Anführungszeichen müssen eine Ersetzungszeichenkette einschließen, die Leerzeichen enthält. Dies ist notwendig, damit das PATTERN vom MAXed als einzelnes Zeichenketten-Attribut erkannt werden kann (siehe 3.6.1 Das Format der Kommandoattributierung). Ein Anführungszeichen innerhalb einer Ersetzungszeichenkette ist nur zulässig, wenn es durch den Gegenschrägstrich \ entwertet ist.

”” Zwei Anführungszeichen, direkt hintereinander, als Ersetzungszeichenkette bedeuten die leere Zeichenkette. Eine zum verwendeten PATTERN gefundene Textstelle wird damit nur entfernt.

| Der senkrechte Strich als Ersetzungszeichenkette bedeutet ebenfalls die leere Zeichenkette, allerdings mit einer Besonderheit: Wird eine Zeile infolge der Ersetzungsoperation leer, so wird sie gelöscht.

#### *Beispiel*

```
<ENTER> ^anschrift\; | . <REPLACE>
```

Damit löschen Sie alle Zeilen der aktuellen Datei (.), die nur den Text *anschrift*: enthalten, wobei Groß- oder Kleinschreibung nicht berücksichtigt wird (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenkettenattribut).

#### *Anmerkung*

Da zu einem PATTERN nur Textstellen innerhalb (!) einer Zeile passen können, können Sie folglich das unsichtbare Zeilenende-Zeichen nicht ersetzen, geschweige denn durch Verwenden der leeren Ersetzungszeichenkette (””) entfernen (siehe Abschnitt 3.10 PATTERN als Kommandoattribut). Durch den senkrechten Strich | geben Sie jedoch dem MAXed zu verstehen, daß er alle Zeilen, in denen er eine Ersetzung mit der leeren Zeichenkette vornimmt, aufgrund derer sie zu Leerzeilen werden, gleich im Anschluß löschen soll.

Die Ersetzungszeichenfolge darf außerdem Ausdrücke enthalten, die bei einer Ersetzung im Text fortlaufende Zahlen ausgeben. Hierdurch können Sie ein oder mehrere Ersetzungsmuster für sogenannte "Zähler" in die Ersetzungszeichenkette einbauen:

{ausdruck}

Geschweifte Klammern schließen einen Ausdruck ein, der Ersetzungsmuster für einen "Zähler" darstellt und folgenden Aufbau hat:

```
{[format][,[anfangswert][,[inkrement][,[option]]]}
```

Der "Zähler" wird in den zu ersetzenden Text entsprechend der Angabe *format* eingefügt. Er wird zu Beginn des Ersetzungskommandos mit *anfangswert* initialisiert und nach jeder Ersetzung mit *inkrement* erhöht bzw. erniedrigt.

Die angegebene Reihenfolge der Klammerelemente müssen Sie bei der Eingabe einhalten. Wenn Sie für eines dieser Elemente nichts angeben, verwendet der MAXed dafür den Standard-Wert (siehe unten); Falls Sie alle nachfolgenden Spezifikationen weglassen, brauchen Sie ein Komma nicht anzugeben.

Die einfachste Art, einen Zähler zu spezifizieren, ist:  
{ } – was gleichbedeutend ist mit der Angabe {**d,1,1**}.

*format* gibt das Format an, in der ein "Zähler" ausgegeben werden soll. *format* hat folgende Syntax:

```
[-][0][n][basis]
```

Die einzelnen Elemente des Formats haben folgende Bedeutung:

**basis**

steht für die Basis der auszugebenden Zahl:

**d** geben Sie an für die Ausgabe einer Dezimalzahl

**o** geben Sie an für die Ausgabe einer Oktalzahl

**x** geben Sie an für die Ausgabe einer Sedezimalzahl

Wenn Sie die Basis nicht angeben werden standardmäßig Dezimalzahlen ausgegeben.

*Anmerkung*

Die Zahlen werden bei der Ausgabe jedoch nicht mit einer zusätzlichen Basis-Kennzeichnung versehen.

- n ist eine positive, ganze Zahl als Feldweite, also die Zeichenlänge, die für die Ausgabe reserviert wird. Wenn Sie die Feldweite nicht angeben, wird ein Feld mit genau der Zeichenlänge der jeweils auszugebenen Zahl reserviert.
- Der Bindestrich bewirkt eine linksbündige Ausrichtung der auszugebenden Zahlen innerhalb des Feldes. Wenn Sie den Bindestrich nicht angeben, wird rechtsbündig ausgerichtet.
- 0 bewirkt das Auffüllen des Feldes mit führenden Nullen, wenn eine auszugebende Zahl nicht die ganze Feldlänge beansprucht. Bei einer linksbündigen Ausgabe wird der nicht benötigte Rest des Feldes mit Leerzeichen ausgefüllt. Standardmäßig wird immer mit Leerzeichen aufgefüllt.

Die folgenden Elemente des Ausdrucks für einen Zähler bestimmen dessen Auswertung:

#### anfangswert

steht für eine Konstante, eine ganze Zahl, mit deren Wert der "Zähler" für die erste Ersetzung initialisiert wird. Dabei gilt:

- Eine Dezimalzahl geben Sie ohne weitere Kennzeichnung ein:  $n$
- Eine Oktalzahl geben Sie durch eine führende Null  $0$  ein:  $0n$
- Eine Sedezimalzahl geben Sie durch eine führende Null, gefolgt von dem Kleinbuchstaben  $x$  ein:  $0xn$
- Negativen Zahlen stellen Sie zusätzlich einen Bindestrich – als Minus-Zeichen voran:  $-n$

Wenn Sie keinen Anfangswert angeben, nimmt der MAXed standardmäßig den Wert 1 an.

#### inkrement

steht für eine Konstante, eine ganze Zahl, um deren Wert der "Zähler" nach jeder Ersetzung erhöht bzw. erniedrigt wird. Bei der Eingabe gilt gleiches wie bei *anfangswert*.

`option` steht für die Angabe eines besonderen "Anwendungszwecks":

Wenn Sie keine Option angeben, so wird der "Zähler" nach jedem Auftreten einer Zeichenkette, die zum PATTERN paßt, erhöht oder bei negativem Inkrement erniedrigt.

Mit den möglichen *options*-Angaben können Sie eine differenziertere "Zähler"-Ausgabe veranlassen:

- = steht für das Zählen von Wiederholungen; der "Zähler" wird nur dann weitergezählt, wenn der MAXed bei der Ersetzung einer zum verwendeten PATTERN passenden Zeichenkette feststellt, daß er eine identische Zeichenkette bereits seit dem Beginn dieser Ersetzungskommandos schon einmal ersetzt hat. Für jede Wiederholung einer identisch vorgefundenen Zeichenketten wird also einmal hoch- bzw. heruntergezählt.
- ! steht für das Zählen von verschiedenen gefundenen Zeichenkettenvarianten; der "Zähler" wird nur dann weitergezählt, wenn der MAXed bei der Ersetzung einer zum verwendeten PATTERN passenden Zeichenkette feststellt, daß er noch keine (!) hierzu identische Zeichenkette seit dem Beginn dieses Ersetzungskommandos ersetzt hat. Für jede neu entdeckte Variante einer passenden Zeichenketten wird also einmal hoch- bzw. heruntergezählt.

### 3.11.2 Textersetzung innerhalb von Ersetzungszeichenketten

Selbstverständlich können Sie auch innerhalb einer Ersetzungszeichenkette Textersetzungen vornehmen lassen (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld). Im Eingabefeld der Steuerzeile bewirken die folgenden Sonderzeichen beim Auslösen des Ersetzungskommandos noch vor dessen Ausführung, daß die nachfolgenden bzw. eingeschlossenen (Teil-)Zeichenketten durch ihren Wert, ihren Inhalt bzw. ihre Ausgabe ersetzt werden:

@ Der Klammeraffe geht einem Parameter für Textersetzung voraus.

\$ Das Dollar-Zeichen geht dem Namen einer Umgebungsvariablen des MAXed voraus und schließt ihn optional ab.

% Das Prozent-Zeichen geht einem PICK-Puffernamen voraus und schließt ihn optional ab. Den Standard-PICK-Puffer geben Sie durch % bzw. %% an (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER).

`SINIX-kommando`

Gegenhochkomma schließen ein SINIX-Kommando ein (siehe Abschnitt 4.5 Die SINIX-Schnittstellen). Das zuletzt ausgeführte SINIX-Kommando können Sie wiederholt mit ` bzw. `` ausführen lassen.

**Beispiel**

Sie bekommen eine Datei mit einem Text vorgesetzt, den Sie mit statistischen Anmerkungen zum Auftreten der Zeichenkette *wort* in allen Varianten versehen sollen:

```
1: Nur ein Wort noch,
2: ein Sprichwort:
3: Am Anfang war das WORT.
4: Mach' keine Worte!
5: "Worte"!- Nichts als Worte!!
6: Nur Schlagworte!
7: Er ist wortkarg;
8: denn Worte sind wie Schall und Rauch.
9: Und (so wortkarg war er)
10: eilte er wortlos von dannen.
```

Zunächst lassen Sie den MAXed durchzählen, wie oft die o.g. Zeichenkette auftritt. Sie setzen die Schreibmarke auf die erste Zeile und geben ein:

```
<ENTER> [a-z,A-Z]*wort[a-z,A-Z]* {} 10 <REPLACE>
```

Sie stellen fest, daß zu dem eingegebenen PATTERN passende Textstellen gefunden wurden (siehe Abschnitt 3.10 PATTERN als Kommandoattribut):

```
1: Nur ein 1 noch,
2: ein 2:
3: Am Anfang war das 3.
4: Mach' keine 4!
5: "5"!- Nichts als 6!!
6: Nur 7!
7: Er ist 8;
8: denn 9 sind wie Schall und Rauch.
9: Und (so 10 war er)
10: eilte er 11 von dannen.
```

Leider sind die passenden Textstellen bei der Aktion verloren gegangen. Also machen Sie das ganze wieder rückgängig mit <UNDO>. Sie wollen sich eine leicht abgeänderte "Sicherungskopie" in den Text setzen, sodaß sie durch das o.g. PATTERN nicht "entdeckt" wird! Sie positionieren die Schreibmarke am Anfang der ersten Zeile und erstellen dazu gleich ein Makro (siehe Abschnitt 4.7 Arbeiten mit Makros):

```
<LEARN>
<ENTER> [a-z,A-Z]*wort[a-z,A-Z]* <PLUSFIND>
<ENTER> '@" ' <PUT>
<WORDRIGHT>
<ENTER> <or<t> <PLUSFIND>
x <LEARN>
```

Die erste Zeile sieht vielversprechend aus:

1: Nur ein Wort Worxt noch,

Die restlichen Textstellen erledigen Sie in einem Durchgang:

```
<ENTER> 10 <MACRO>
```

Jetzt können Sie problemlos die überschüssigen Wortgebilde durch Anmerkungen ersetzen. In einer unbeschriebenen Zeile "entwickeln" Sie sich die Kommandoattributierung des nun folgenden Ersetzungs-kommandos:

```
[a-z,A-Z]*worxt[a-z,A-Z]* \<Sprüche{3d,0,,!}\> 10
```

Sie kopieren sich diese Zeile mit <PICK> in den Standard-PICK-Puffer. Danach setzen Sie die Schreibmarke wieder auf die erste Zeile des obigen Textes und attributieren das Ersetzungskommando mit dem Inhalt des Standard-PICK-Puffers:

```
<ENTER> % <REPLACE>
```

Die gesuchten Wörter sind nun mit Varianten-Nummern versehen:

1: Nur ein Wort <Sprüche 0> noch,

2: ein Sprichwort <Sprüche 1>:

3: Am Anfang war das WORT <Sprüche 2>.

4: Mach' keine Worte <Sprüche 3>!

5: "Worte <Sprüche 3>"!- Nichts als Worte <Sprüche 3>!!

6: Nur Schlagworte <Sprüche 4>!

7: Er ist wortkarg <Sprüche 5>;

8: denn Worte <Sprüche 3> sind wie Schall und Rauch.

9: Und (so wortkarg <Sprüche 5> war er)

10: eilte er wortlos <Sprüche 6> von dannen.

### 3.12 Voreingestellte Kommandoattribute

Bei vielen MAXed-Kommandos sind Kommandoattribute wie z.B. die Bereichsdefinition auf einen Standard-Wert voreingestellt. Zu einigen Kommandos gibt es auch einen aktuell belegten Puffer. Bei einigen anderen Kommandos werden Kommandoattribute aus der jeweils zuletzt eingegebene Kommandoattributierung zwischengespeichert. In den Fällen, wo Sie Kommandos mit Standard-Einstellungen, aktuellen Puffern oder zwischengespeicherten Kommandoattributen aufrufen wollen, brauchen Sie die entsprechenden Kommandoattribute nicht mehr einzugeben. Häufig können Sie Kommandos nicht-attributiert aufrufen, d.h. es genügt, wenn Sie die zugehörige Kommandotaste betätigen.

In den drei folgenden Abschnitten sind solche MAXed-Kommandos aufgelistet, die Ihnen durch Voreinstellungen die Kommandoattributierung erleichtern.

### 3.12.1 Kommandos mit konstanter Standard-Ausführung

Viele der MAXed-Kommandos haben eine konstante, voreingestellte Bereichsdefinition (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut).

#### Die aktuelle Zeile

Bei fehlender Bereichsattributierung greift der MAXed bei folgenden Kommandos grundsätzlich auf die aktuelle Zeile zu:

<ALIGN>

Verschiebt den Textanfang an die aktuelle Spaltenposition.

<ATTRIBUTE>

Setzt die voreingestellten Zeichenattribute.

<CENTER>

Zentriert den Text in der aktuellen Zeile. .

<COMPRESS>

Komprimiert den Text, d.h. entfernt überflüssige Leerzeichen.

<ERASE>

Löscht und speichert im Standard-ERASE-Puffer.

<ENTER> IDENTIFIER <ERASE>

Löscht und speichert im angegebenen ERASE-Puffer.

<EXCHANGE>

Tauscht die aktuelle Zeile mit dem zuvor gelöschten Textbereich aus.

<ENTER> <EXECUTE>

Führt jenes SINIX-Kommando asynchron aus, dessen Aufruf in der aktuellen Zeile steht.

<JUSTIFY>

Richtet den Text links- und rechtsbündig aus.

<LOWER>

Wandelt alle Großbuchstaben in Kleinbuchstaben um.

<PICK>

Kopiert in den Standard-PICK-Puffer.

<ENTER> IDENTIFIER <PICK>

Kopiert in den genannten PICK-Puffer.

<SPACE>

Überschreibt mit einer Leerzeile bzw. fügt eine Leerzeile ein.

<UPPER>

Wandelt alle Kleinbuchstaben in Großbuchstaben um.

### Der aktuelle Zeilenrest

Bei fehlender Bereichsattributierung greift der MAXed bei folgenden Kommandos grundsätzlich auf den Rest der aktuelle Zeile ab der aktuellen Spaltenposition der Schreibmarke zu:

<ENTER> <ATTRIBUTE>

Setzt die voreingestellten Zeichenattribute.

<ENTER> <COMPRESS>

Komprimiert den Text.

<ENTER> <DRAW>

Überschreibt mit einer durchgezogenen Linie.

<ENTER> <ERASE>

Löscht und speichert im Standard-ERASE-Puffer bzw. löscht das Zeilenende-Zeichen.

<ENTER> <EXCHANGE>

Tauscht den Zeilenrest mit dem zuvor gelöschten Textbereich aus.

<ENTER> <JUSTIFY>

Richtet den Text links- und rechtsbündig aus.

<ENTER> <LOWER>

Wandelt alle Großbuchstaben in Kleinbuchstaben um.

<ENTER> <PICK>

Kopiert in den Standard-PICK-Puffer.

<REPLACE>

bzw.

<ENTER> [PATTERN] [ersetzungszeichenkette] <REPLACE>

Führt an der nächsten passenden Textstelle innerhalb des Zeilerestes ein Ersetzungskommando aus, und verwendet ggf. das aktuelle PATTERN bzw. die zuletzt eingegebene Ersetzungszeichenkette, wenn diese nicht angegeben sind (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Attributen).

<ENTER> <SPACE>

Überschreibt mit Leerzeichen bzw. fügt ein neues unsichtbares Zeilenende-Zeichen ein.

<ENTER> <UPPER>

Wandelt alle Kleinbuchstaben in Großbuchstaben um.

## Das aktuelle Wort

Bei fehlender Bereichsattributierung greift der MAXed bei folgenden Kommandos grundsätzlich auf das Wort zu, auf dem die Schreibmarke aktuell positioniert ist bzw. auf eine zusammenhängende Zeichenkette der aktuellen Zeile:

<ENTER> <DEFDIR>

Stellt das aktuelle Dateiverzeichnis neu ein. Dabei wird als Name die erste zusammenhängende Zeichenkette in der aktuellen Zeile verwendet, falls diese den Pfadnamen eines existierenden Dateiverzeichnisses darstellt.

<ENTER> <MINUSFIND>

Sucht rückwärts nach einer Textstelle, die mit dem aktuellen Wort übereinstimmt.

<ENTER> <PLUSFIND>

Sucht vorwärts nach einer Textstelle, die mit dem aktuellen Wort übereinstimmt.

## Der aktuelle Absatz

Bei fehlender Bereichsattributierung greift der MAXed bei folgendem Kommando grundsätzlich auf den Absatz zu, in dem die Schreibmarke aktuell positioniert ist:

<ENTER> <FILTER>

bzw.

<ENTER> filter-programm [programm-parameter] <FILTER>

Wiederholt das zuletzt ausgeführte Filter-Programm bzw. führt das angegebene Filter-Programm aus. Dabei dienen alle Zeilen des aktuellen Absatzes als Ein- und Ausgabebereich (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Attributen).

## Der aktuelle Absatzrest

Bei fehlender Bereichsattributierung greift der MAXed bei folgenden Kommandos grundsätzlich auf den Rest des aktuellen Absatzes von der aktuellen Zeile bis Absatzende zu:

<FILL>

Formatiert im Flattersatz.

<ENTER> <FILL>

Formatiert im Flattersatz. Zusätzlich wird an der Position der Schreibmarke ein Trennvorschlag vermerkt.

<FILTER>

Wiederholt das zuletzt ausgeführte Filter-Programm, das die Zeilen des aktuellen Absatzes ab der aktuellen Zeile als Ein- und Ausgabebereich verwendet (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Attributen).

<FORMAT>

Formatiert im Blocksatz.

<ENTER> <FORMAT>

Formatiert im Blocksatz. Zusätzlich wird an der Position der Schreibmarke ein Trennvorschlag vermerkt.

## Das aktuelle Fenster

Bei fehlender Bereichsattributierung greift der MAXed bei folgenden Kommandos grundsätzlich auf einen vordefinierten Bereich innerhalb des aktuellen Fensters zu (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten):

<PRINT>

Druckt den Text des aktuellen Fensters aus.

<ENTER> <PRINT>

Druckt den Text ab der aktuellen Position der Schreibmarke bis zum Ende des aktuellen Fensters.

### 3.12.2 Kommandos mit Standard-Puffern

Einige der MAXed-Kommandos, die mit IDENTIFIER arbeiten, haben auch einen aktuell belegten, namenlosen Puffer, der bestimmte Daten aus dem letzten Aufruf des Kommandos speichert (siehe Abschnitt 3.8 Der IDENTIFIER als Kommandoattribut). Sie greifen auf einen aktuellen Puffer immer dann zu, wenn Sie einen Kommandoaufruf nicht mit einem IDENTIFIER attributieren.

#### Die Alternativ-Datei

Für das MAXed-Kommando EDIT wird zu jedem Fenster auf Ihrem MAXed-Bildschirm der Name der Alternativ-Datei gespeichert (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien, Kapitel 6: EDIT). Das ist diejenige Datei, die vor der momentan eingeblendeten Datei im jeweiligen Fenster eingeblendet war, sofern zuvor bereits eine andere Datei eingeblendet war. Beim Umschalten zur Alternativ-Datei geben Sie keinen IDENTIFIER an:

<EDIT>

### Der Standard-ERASE-Puffer

Für die MAXed-Kommandos ERASE und RESTOR wird derjenige Textbereich, den Sie bei der letzten Löschaktion aus einer Datei in einem ERASE-Puffer speicherten oder aus einem ERASE-Puffer wieder in eine Datei kopiert haben (oder umgekehrt), immer auch im Standard-ERASE-Puffer gespeichert (siehe Abschnitt 3.8.2 Der ERASE-Puffernamenname als IDENTIFIKATOR). Der Inhalt dieses Puffers bleibt bis zur nächsten Löscho- bzw. Kopieraktion mit ERASE bzw. RESTOR erhalten. Dann wird er durch den gerade gelöschten bzw. kopierten Text ersetzt. Der Standard-ERASE-Puffer dient als "Durchgangsschleuse" für alle weiteren benannten ERASE-Puffer.

Sie löschen die aktuelle Zeile bzw. den angegebenen Bereich und speichern sie bzw. ihn im Standard-ERASE-Puffer mit

```
<ERASE>  
bzw.  
<ENTER> [bereich] <ERASE>
```

Sie kopieren den Inhalt des Standard-ERASE-Puffers einmal bzw.  $n$ -mal in den Text ab der aktuellen Position der Schreibmarke mit

```
<RESTOR>  
bzw.  
<ENTER> n <RESTOR>
```

## Der Standard-PICK-Puffer

Für die MAXed-Kommandos PICK und PUT wird derjenige Textbereich, den Sie bei der letzten Kopieraktion aus einer Datei in einen PICK-Puffer kopiert haben (oder umgekehrt), immer auch im Standard-PICK-Puffer gespeichert (siehe Abschnitt 3.8.3 Der PICK-Puffernamen als IDENTIFIER). Der Inhalt dieses Puffers bleibt bis zur nächsten Kopieraktion mit PICK oder PUT erhalten. Dann wird er durch den neu kopierten Text ersetzt. Der Standard-PICK-Puffer dient als "Durchgangsschleuse" für alle weiteren benannten PICK-Puffer. Beim Kopieren mit dem Standard-PICK-Puffer geben Sie keinen IDENTIFIER an:

Sie kopieren die aktuelle Zeile bzw. den angegebenen Bereich in den Standard-PICK-Puffer mit

```
<PICK>  
bzw.  
<ENTER> [bereich] <PICK>
```

Sie kopieren den Inhalt des Standard-PICK-Puffers einmal bzw.  $n$ -mal in den Text ab der aktuellen Position der Schreibmarke mit

```
<PUT>  
bzw.  
<ENTER> n <PUT>
```

## Die Standard-Marke

Innerhalb einer Datei markieren Sie Positionen mit dem Kommando LABEL und springen an diese Marken wieder zurück mit dem Kommando GOTO (siehe Abschnitt 3.8.4 Der Markenname als IDENTIFIER). Dabei können Sie in jeder geöffneten Datei außer benannten Marken noch eine unbenannte, die Standard-Marke, setzen. Eine Standard-Marke existiert unabhängig sowohl von den benannten Marken derselben Datei als auch von weiteren Standard-Marken, die Sie in anderen geöffneten Dateien gesetzt haben. In der Pseudo-Datei LABELS wird für die Standard-Marke ein Eintrag ohne Namensangabe aufgelistet (siehe Abschnitt 4.2.8 Die Pseudo-Datei LABELS).

Sie setzen an der Position der Schreibmarke in der aktuellen Datei die Standard-Marke mit

```
<LABEL> 
```

Sie springen mit der Schreibmarke von beliebiger Stelle in derselben Datei an die Position der zuvor gesetzten Standard-Marke mit

```
<ENTER> <GOTO> 
```

## Das aktuelle Makro

Für das MAXed-Kommando `MACRO` wird immer das zuletzt mit `MACRO` ausgeführte oder mit Hilfe des Kommandos `LEARN` erstellte benannte `MACRO` im Kommando-Puffer zwischengespeichert (siehe Abschnitt 3.8.5 Der Makroname als IDENTIFIER). Der Inhalt dieses Puffers bleibt bis zur nächsten Definition bzw. Ausführung eines benannten Makros mit `LEARN` bzw. `MACRO` erhalten. Dann wird er durch das gerade erstellte bzw. ausgeführte Makro ersetzt. Der Kommando-Puffer dient als Durchgangsschleuse für alle weiteren benannten Makros.

Sie führen das aktuelle Makro einmal bzw.  $n$ -mal aus mit

```
<MACRO>  
bzw.  
<ENTER> n <MACRO>
```

In der Steuerzeile erscheint während der Ausführung die Meldung  
**MACRO:** makroname

### Anmerkung

Das aktuelle Makro wird nur bei der Definition des Standard-Makros durch das Kommando `LEARN` überschrieben, nicht bei der Ausführung des Standard-Makros.

## Das Standard-Makro

Für das Kommando `MACRO` ist außer dem aktuellen Makro (siehe oben) noch ein zweiter Kommando-Puffer vorhanden, der das Standard-`MACRO` enthält. Das ist das unbenannte Makro, das Sie in der aktuellen Sitzung mit `LEARN` definiert haben. Das Standard-Makro existiert unabhängig von benannten Makros (siehe Abschnitt 3.8.5 Der Makroname als `IDENTIFIER`).

Sie erstellen während der `MAXed`-Sitzung das Standard-Makro durch Aufzeichnen aller Aktionen, die zwischen dem einleitenden und abschließenden `<LEARN>` liegen:

```
<LEARN>
...
<LEARN>
```

Sie führen das Standard-Makro einmal aus mit

```
<ENTER> <MACRO>
```

In der Steuerzeile erscheint während der Ausführung die Meldung  
`MACRO: <DEFAULT>`

### 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen

Einige der MAXed-Kommandos speichern Kommandoattribute des jeweils letzten Kommandoaufrufs. Zur Wiederholung des letzten Aufrufs brauchen Sie keine oder nur einen Teil der Kommandoattribute mit anzugeben. Einige dieser aktuellen Kommandoattribute werden in der Pseudo-Datei SETUP angezeigt, wo Sie sie auch verändern können (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

#### Das aktuelle SINIX-Kommando

Sie führen während der MAXed-Sitzung ein SINIX-Kommando asynchron aus, indem Sie das MAXed-Kommando EXECUTE mit dem Kommandoaufruf des SINIX-Kommandos attributieren (siehe Abschnitt 4.5 Die SINIX-Schnittstellen, Kapitel 6: EXECUTE). Für das Kommando EXECUTE wird jeweils das Kommandoattribut (also der Aufruf des SINIX-Kommandos) gespeichert, mit dem es zuletzt ausgeführt wurde.

<EXECUTE>

Wiederholt das zuletzt ausgeführte EXECUTE-Kommando.

### Das aktuelle Filter-Programm

Sie führen während der MAXed-Sitzung ein Filter-Programm aus, indem Sie das MAXed-Kommando FILTER (siehe Abschnitt 4.5.3 Ausführen von Filter-Programmen, Kapitel 6: FILTER) mit dem Kommandoaufruf des Filter-Programms attributieren und einen RANGE-Bereich in der aktuellen Datei als Ein- und Ausgabebereich des Filter-Programms definieren. Für das Kommando FILTER wird jeweils das Kommandoattribut (also der Aufruf des Filter-Programms) gespeichert, mit dem es zuletzt ausgeführt wurde.

```
<FILTER>  
bzw.  
<ENTER> <FILTER>
```

Wiederholt das zuletzt ausgeführte FILTER-Kommando.

#### *Anmerkung*

Beachten Sie hierbei, daß die Bereichsdefinition beim letzten FILTER-Kommando nicht in der gespeicherten Kommandoattributierung enthalten ist, sondern standardmäßig vordefinierte Bereiche verwendet werden (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).

## Das aktuelle PATTERN

Sie können in einer Datei nach Textstellen suchen, indem Sie ein Such-Kommando mit einem PATTERN als Suchmuster attributieren (siehe Kapitel 6: MINUS-FIND, PLUSFIND, Kapitel 6: REPLACE; siehe auch Kapitel 6: EDIT, ENTER, HWINDOW, PUT, VWINDOW). Bei allen unten angeführten Kommandos, die mit PATTERN arbeiten, jedoch kein neues PATTERN als Zeichenketten-Attribut erhalten, greift der MAXed automatisch auf das aktuelle PATTERN zurück.

Das aktuelle PATTERN ist i.a. jenes PATTERN, das Sie zuletzt beim Aufruf eines Such-Kommandos als Kommandoattribut eingegeben haben (siehe Abschnitt 3.10 PATTERN als Kommandoattribut, siehe auch unten: Gespeichertes PATTERN bei FILE-SCAN). Bei eingestelltem LEXICAL-Modus belegt der MAXed das aktuelle PATTERN Im Anschluß an ein Formatierkommando mit dem Suchmuster für das Silbentrennzeichen (siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: FILL, FORMAT). In der Pseudo-Datei SETUP ist der Parameter FINDPATTERN mit dem aktuellen PATTERN belegt (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

### <MINUSFIND>

Führt ein Suchkommando mit dem aktuellen PATTERN in Rückwärtsrichtung aus.

### <PLUSFIND>

Führt ein Suchkommando mit dem aktuellen PATTERN in Vorwärtsrichtung aus.

### <REPLACE>

bzw.

<ENTER> [ersetzungszeichenkette] [bereich] <REPLACE>

Führt im angegebenen Bereich ein Ersetzungskommando mit dem aktuellen PATTERN und der angegebenen Ersetzungszeichenkette aus.

Wenn Sie keine Ersetzungszeichenkette angeben, wird die aktuelle Ersetzungszeichenkette verwendet (siehe unten).

Wenn Sie keinen Bereich definieren, wird der aktuelle Zeilenrest nach einer zum aktuellen PATTERN passenden Textstelle durchsucht (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).

## Die aktuelle Ersetzungszeichenkette

Sie können mit dem Kommando REPLACE in einer Datei innerhalb eines definierten Bereichs nach Textstellen suchen und gefundene Textstellen durch eine Ersetzungszeichenkette ersetzen (siehe Abschnitt 3.11 Die Ersetzungszeichenkette als Kommandoattribut, Kapitel 6: REPLACE). Wenn Sie dabei keine Ersetzungszeichenkette angeben, greift der MAXed auf diejenige zurück, die Sie zuletzt bei einem Aufruf dieses Kommandos eingegeben haben. In der Pseudo-Datei SETUP ist der Parameter REPLACE-STRING mit der aktuellen Ersetzungszeichenkette belegt (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

<REPLACE>

bzw.

<ENTER> [bereich] <REPLACE>

Führt ein Ersetzungskommando aus und verwendet dabei sowohl das aktuelle PATTERN als auch die aktuelle Ersetzungszeichenkette.

Wenn Sie keinen Bereich definieren, wird der aktuelle Zeilenrest nach einer zum aktuellen PATTERN passenden Textstelle durchsucht (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).

## Gespeichertes PATTERN bei FILE-SCAN

<ENTER> """ [IDENTIFIER] <GOTO>

Ruft die Funktion FILE-SCAN auf, die die Pseudo-Datei SCANLIST erstellt, und greift dabei auf dasjenige PATTERN zurück, das Sie zuletzt bei einem Aufruf dieses Kommandos eingegeben haben. In der Pseudo-Datei SETUP ist der Parameter SCANPATTERN mit dem PATTERN zu diesem Kommando belegt (siehe Kapitel 6: GOTO, Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

Das aktuelle PATTERN (siehe oben) wird durch den Wert von SCAN-PATTERN ersetzt, sobald Sie aus der Pseudo-Datei SCANLIST mit nochmaligem <GOTO> an eine der gefundenen Textstellen springen.

Wenn IDENTIFIER nicht angegeben ist, verwendet der MAXed ebenfalls die entsprechende Angabe vom letzten Aufruf dieses Kommandos oder ersatzweise die aktuelle Datei. In der Pseudo-Datei SETUP ist der Parameter SCANFILES mit dem aktuellen IDENTIFIER belegt (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

## 4 Wie arbeitet der MAXed? - Erweiterte Möglichkeiten

— Dieses baut auf Kapitel 3 "Grundlagen" auf. Es wird Ihnen verschiedene Einstellungsmöglichkeiten des MAXed aufzeigen und Ihren Umgang mit Kommandos in speziellen Einsatzgebieten vertiefen. Dabei sind die Abschnitte 4.6 "Arbeiten mit Textbausteinen", 4.7 "Arbeiten mit Makros" und 4.8 "Arbeiten mit Steuerprogrammen" als Hinweis auf komplexe Anwendungsmöglichkeiten des MAXed gedacht, denen Sie sich in fortgeschrittenem Studium zuwenden sollten.

—

—

—

—

## 4.1 Die Betriebs-Modi des MAXed

### 4.1.1 Die Eingabe-Modi

Der MAXed verfügt über zwei alternative Eingabe-Modi (siehe Kapitel 6: INSERTMODE):

- Einfüge-Modus
- Überschreibe-Modus OVERWRITE

Sie bestimmen, ob beim Schreiben und Kopieren in den bearbeiteten Dateien ein Text in bestehenden eingefügt wird oder ob dieser dabei überschrieben wird. Sie schalten zwischen den beiden alternativ verfügbaren Eingabe-Modi um mit

<INSERTMODE>

Außerdem unterscheidet der MAXed auch noch zwischen vier verschiedenen Einstellungsmöglichkeiten des Einfüge-Modus. Das beeinflusst teilweise den alternativ verfügbaren Überschreibe-Modus, indem der Silbentrenner des MAXed aktiviert wird und Leerzeichen beim Einfügen und Formatieren auf verschiedene Weise behandelt werden (siehe Abschnitte 4.1.10 Die Sprach-Modi, 4.3 Das Formatieren):

Varianten des Einfüge-Modus und ihre Unterscheidungsmerkmale	Auswirkung auf Überschreibe-Modus OVERWRITE
<b>INSERT-Modus:</b> Beim Schreiben und Einfügen von Textbereichen wird ein evtl. nachfolgender Text ohne Veränderung nach rechts bzw. nach unten weitergeschoben. Der automatische Zeilenumbruch erfolgt wortorientiert.	Der automatische Zeilenumbruch erfolgt wortorientiert. Ein RECORD-Bereich umfaßt einen einzeiligen STREAM von Zeilenanfang bis zum letzten Semikolon der aktuellen Zeile
<b>WORD-Modus:</b> Beim Schreiben und Einfügen von einzeiligen STREAM-Bereichen wird evtl. nachfolgender Text noch schrittweise zusammengesoben bis alle überflüssigen Leerzeichen entfernt sind. Der automatische Zeilenumbruch erfolgt wortorientiert.	Der automatische Zeilenumbruch erfolgt wortorientiert. Ein RECORD-Bereich umfaßt einen einzeiligen STREAM von Zeilenanfang bis zum letzten Semikolon der aktuellen Zeile

Varianten des Einfüge-Modus und ihre Unterscheidungsmerkmale	Auswirkung auf Überschreibe-Modus OVERWRITE
<b>LEXICAL-Modus:</b> Beim Schreiben und Einfügen von einzelnen STREAM-Bereichen wird evtl. nachfolgender Text noch schrittweise zusammengeschieben bis alle überflüssigen Leerzeichen entfernt sind. Der automatische Zeilenumbruch erfolgt durch den Silbentrenner des MAXed.	Der automatische Zeilenumbruch erfolgt silbenorientiert. Ein RECORD-Bereich umfaßt i.a. einen ganzen Satz, d.h. den STREAM-Bereich zwischen zwei Punkten (ersatzweise Dateianfang bzw. -ende).
<b>STREAM-Modus:</b> Beim Schreiben und Einfügen von einzelnen STREAM-Bereichen wird evtl. nachfolgender Text noch schrittweise zusammengeschieben bis sämtliche Leerzeichen entfernt sind. Der automatische Zeilenumbruch erfolgt wortorientiert.	Der automatische Zeilenumbruch erfolgt wortorientiert. Ein RECORD-Bereich umfaßt einen einzelnen STREAM von Zeilenanfang bis zum letzten Semikolon der aktuellen Zeile

Der aktuell eingestellte Eingabe-Modus wird am linken Rand der Steuerzeile angezeigt (siehe Abschnitt 3.5.2 Die Steuerzeile).

#### Anmerkung

- Die Auswirkung des Silbentrenners betrifft den automatischen Zeilenumbruch beim Schreiben über den rechten Formatrand und beim Formatieren mit Hilfe der Kommandos FILL und FORMAT (siehe Abschnitte 4.3.2 Formatränder, 4.3.3 Flatter- und Blocksatz).
- Eine RECORD-Bereichsdefinition geben Sie als Zeichenketten-Attribut ein (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut). Je nach Einstellung des alternativ verfügbaren Einfüge-Modus können Sie damit z.B. ganze Sätze oder Anweisungen in C-Quelldateien ansprechen.

Im Folgenden sind die verschiedenen Einstellungen der Eingabe-Modi mit ihren Auswirkungen auf das Verhalten des MAXed bei der Texteingabe und bei der Verwendung bestimmter Kommandos aufgeführt.

## Der Einfüge-Modus INSERT

Der INSERT-Modus (Standard-Einfüge-Modus) ist in der

Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Bei der Texteingabe und beim Kopieren von Pufferinhalten in den Text wird in den bereits bestehenden Text eingefügt. Der bisherige Text wird dabei unverändert nach rechts bzw. nach unten weitergeschoben. Sie schalten den INSERT-Modus (Standard-Einfüge-Modus) ein mit

<ENTER> i <INSERTMODE>

Im einzelnen hat diese Einstellung des Einfüge-Modus folgende Auswirkungen:

Aktionen/ Kommandos	INSERT-Modus (Standard-Einfüge-Modus)
Texteingabe durch Einzelzeichen	Eingegebene Zeichen werden an der Position der Schreibmarke eingefügt. Die Schreibmarke und mit ihr evtl. rechts davon vorhandener Text in der aktuellen Zeile werden nach rechts weitergeschoben.
<BACKSPACE>	Das Zeichen links der Schreibmarke wird gelöscht, falls diese nicht am Zeilenanfang steht. Die Schreibmarke und mit ihr evtl. rechts davon vorhandener Text in der aktuellen Zeile werden nach links verschoben.
Textbereiche einkopieren PUT, RESTOR	<ul style="list-style-type: none"> <li>- RANGE-Bereiche: Die Zeilen werden vor der aktuellen Zeile eingefügt. Evtl. nachfolgende Zeilen werden nach unten verschoben. (Ein Dateiinhalt wird als RANGE-Bereich betrachtet).</li> <li>- RECTANGLE-Bereich: Das Rechteck wird mit der linken oberen Ecke an der aktuellen Schreibmarkenposition eingefügt. Ein evtl. unter dem Bereich vorhandener Text wird in den jeweiligen Zeilen nach rechts verschoben</li> <li>- STREAM-Bereiche: Die Zeichenkette wird einschließlich der enthaltenen Zeilenendezeichen an der aktuellen Schreibmarkenposition eingefügt. Evtl. nachfolgender Text wird nach rechts und unten verschoben</li> </ul>

Aktionen/ Kommandos	INSERT-Modus (Standard-Einfüge-Modus)
RETURN	Im SPACE- oder EMPTY-Modus der Zeilenschaltung (>>>> Abschnitt 4.1.3 Der Zeilenschaltungs-Modus) wird vor der nächsten Textzeile eine Leerzeile eingefügt.
WRAP	Beim Umbrechen des Zeilenrestes (>>>> Abschnitt 4.3.5 Umbrechen und Zusammenfügen) wird dieser in der nächsten Zeile vor vorhandenem Text bzw. ab dem linken Formatrand eingefügt.
Automatischer Zeilenumbruch: Leertaste, RETURN	Wenn Sie bei der Texteingabe über den rechten Formatrand hinauskommen, lösen Sie durch Betätigen der Leertaste oder von <RETURN> den automatischen Zeilenumbruch aus. Am Zeilenende wird wortweise umbrochen und in der nächsten Zeile vor evtl. bereits vorh. Text bzw. ab dem linken Formatrand eingefügt. Ein bereits vorhandener Text wird nach rechts geschoben.
Textbereiche formatieren: FILL, FORMAT, CENTER	Wenn Sie einen Bereich Textbereich mit FILL oder FORMAT formatieren, wird er zwischen linkem und rechtem Formatrand im Flutter- oder Blocksatz ausgerichtet. Der Zeilenumbruch erfolgt wortweise. Nachfolgender oder vorausgehender Text wird grundsätzlich nicht verändert. Lediglich der definierte Textbereich wird durch seine formatierte Version ersetzt. Mit CENTER zentrieren Sie einen Textbereich zwischen den Formaträndern. Bei diesen Formatierkommandos werden alle überflüssigen Leerzeichen wie im LEXICAL-Modus (siehe unten) gelöscht!

## Der Überschreibe-Modus OVERWRITE

Im OVERWRITE-Modus überschreiben Sie bei der Texteingabe und beim Kopieren eines Pufferinhalts in einen Text den bereits bestehenden Text. Der OVERWRITE-Modus ist weitgehend unabhängig von dem alternativ verfügbaren Einfüge-Modus. Beim automatischen Zeilenumbruch am rechten Formatrand und beim Formatieren von Textbereichen bricht jedoch die Umbruchtechnik des alternativ verfügbaren Einfüge-Modus durch: Sei dies nun der Silbentrenner des MAXed beim LEXICAL-Modus oder der Wortumbruch bei den anderen Einstellungen des Einfüge-Modus (siehe Abschnitt 4.1.10 Die Sprach-Modi). Außerdem umfaßt ein RECORD beim LEXICAL-Modus i.a. einen Satz, während er sonst einen einzeligen STREAM vom Zeilenanfang bis zum letzten Semikolon der aktuellen Zeile, also i.a. eine Anweisung in einem C-Quellprogramm, darstellt. Sie schalten explizit den alternativ verfügbaren Überschreibe-Modus OVERWRITE ein mit

<ENTER> o <INSERTMODE>

Im einzelnen hat der OVERWRITE-Modus folgende Auswirkungen:

Aktionen/ Kommandos	Überschreibe-Modus (OVERWRITE-Modus)
Texteingabe durch Einzelzeichen	Eingegebene Zeichen überschreiben an der Position der Schreibmarke einen eventuell bereits bestehenden Text.
<BACKSPACE>	Das Zeichen links der Schreibmarke, falls diese nicht am Zeilenanfang steht, wird mit einem Leerzeichen überschrieben und die Schreibmarke auf diese Position gesetzt. Nachfolgender Text bleibt unverändert stehen
Textbereiche einkopieren: PUT, RESTOR	<ul style="list-style-type: none"> <li>- RANGE-Bereiche: Die Zeilen überschreiben ab der aktuellen Zeile eine gleichgroße Anzahl evtl. nachfolgender Zeilen. Ein Dateiinhalt wird auch als RANGE-Bereich betrachtet.</li> <li>- RECTANGLE-Bereich: Das Rechteck wird mit der linken oberen Ecke an der aktuellen Schreibmarkenposition in den Text kopiert. Ein evtl. in diesem Bereich bereits bestehender Text wird überschrieben.</li> <li>- STREAM-Bereiche: Die Zeichenkette wird einschließlich der enthaltenen Zeilenende-Zeichen an die aktuelle Schreibmarkenposition kopiert. Ein evtl. in diesem Bereich bereits bestehender Text wird überschrieben.</li> </ul>
RETURN	Im SPACE- oder EMPTY-Modus der Zeilenschaltung wird eine evtl. folgende Textzeile mit einer Leerzeile überschrieben (siehe Abschnitt 4.1.3 Der Zeilenschaltungs-Modus).
WRAP	Verhält sich entsprechend dem eingestellten Einfüge-Modus.
Automatischer Zeilenumbruch: Leertaste, RETURN	Wenn Sie bei der Texteingabe über den rechten Formatrand hinauskommen, lösen Sie durch Betätigen der Leertaste oder von <RETURN> den automatischen Zeilenumbruch aus. Am Zeilenende wird entsprechend dem eingestellten Einfüge-Modus umbrochen und in die nächste Zeile ab dem linken Formatrand geschoben. Ein evtl. bereits vorhandener Text wird dabei ab dem linken Formatrand überschrieben.
Textbereiche formatieren: FILL, FORMAT, CENTER	Verhält sich entsprechend dem alternativ verfügbaren Einfüge-Modus.

## Der Einfüge-Modus WORD

Im WORD-Modus wird einzugebender und einzukopierender Text in bereits bestehenden Text eingefügt. Der bisherige Text wird dabei nach rechts bzw. nach unten weitergeschoben. Dabei geht der MAXed jedoch davon aus, daß Zwischenräume zwischen zwei Wörtern oder auch zwischen zwei Sonderzeichen lediglich eines Leerzeichens bedürfen. Diese Einstellung macht sich u.a. dann bemerkbar, wenn Sie beim Schreiben einfügen oder einen einzeiligen STREAM-Bereich in einen bestehenden Text einkopieren: Der nachfolgende Text wird erst schrittweise komprimiert, bis alle überflüssigen Leerzeichen entfernt, bevor er als Ganzes nach rechts weitergeschoben wird. Sie schalten den WORD-Modus ein mit

<ENTER> w <INSERTMODE>

Im einzelnen hat diese Einstellung des Einfüge-Modus folgende Auswirkungen:

Aktionen/ Kommandos	WORD-Modus (Einfüge-Modus)
Texteingabe durch Einzelzeichen	Verhält sich im wesentlichen wie im INSERT-Modus, jedoch wird ein evtl. nachfolgender Text zunächst zusammengesoben: In den einzelnen Wort-Zwischenräumen werden schrittweise alle Leerzeichen bis auf eines entfernt.
<BACKSPACE>	Verhält sich wie im INSERT-Modus.
Textbereiche einkopieren	Verhält sich im wesentlichen wie im INSERT-Modus. Beim Einfügen einzelziger STREAM-Bereiche in bereits bestehenden Text wird der nachfolgende Text zunächst komprimiert.
RETURN	Verhält sich wie im INSERT-Modus.
WRAP	Verhält sich im wesentlichen wie im INSERT-Modus. Beim Einfügen des Zeilenrestes in die nächste Zeile wird ein evtl. nachfolgender Text jedoch zunächst komprimiert.
Automatischer Zeilenumbruch	Verhält sich wie im INSERT-Modus. Am Zeilenende wird wortweise umbrochen.
Textbereiche formatieren FILL, FORMAT, CENTER	Verhält sich im wesentlichen wie im INSERT-Modus. Am Zeilenende wird wortweise umbrochen. Jedoch werden nur mehrfach auftretende Leerzeichen als überflüssig betrachtet und gelöscht.

## Der Einfüge-Modus LEXICAL

Mit dem LEXICAL-Modus versucht der MAXed die Grundbedürfnisse beim Schreiben von Fließtext zu befriedigen: Der Silbentrenner des MAXed ist aktiviert und beim Einfügen in bereits bestehenden Text entfernt er wie im WORD-Modus die überflüssigen Leerzeichen (siehe Abschnitt 4.1.10 Die Sprach-Modi). Darunter fallen im LEXICAL-Modus auch jene vor Satzzeichen und Klammern mit Ausnahme der öffnenden Klammern. Sie schalten den LEXICAL-Modus ein mit

<ENTER> I <INSERTMODE>

Im einzelnen hat diese Einstellung des Einfüge-Modus folgende Auswirkungen:

Aktionen/ Kommandos	LEXICAL-Modus (Einfüge-Modus)
Texteingabe durch Einzelzeichen	Verhält sich im wesentlichen wie im INSERT-Modus, jedoch wird ein evtl. nachfolgender Text zunächst zusammengesoben: In den einzelnen Wort-Zwischenräumen werden schrittweise alle Leerzeichen bis auf eines entfernt.
<BACKSPACE>	Verhält sich wie im INSERT-Modus.
Textbereiche einkopieren	Verhält sich im wesentlichen wie im INSERT-Modus. Beim Einfügen einzeliger STREAM-Bereiche in bereits bestehenden Text wird der nachfolgende Text zunächst komprimiert.
RETURN	Verhält sich wie im INSERT-Modus.
WRAP	Verhält sich im wesentlichen wie im INSERT-Modus. Beim Einfügen des Zeilenrestes in die nächste Zeile wird ein evtl. nachfolgender Text jedoch zunächst komprimiert.
Automatischer Zeilenumbruch	Verhält sich wie im INSERT-Modus. Am Zeilenende wird nach Möglichkeit getrennt.
Textbereiche formatieren	Verhält sich wie im INSERT-Modus. Am Zeilenende wird getrennt!
Textbereiche formatieren FILL, FORMAT, CENTER	Verhält sich im wesentlichen wie im INSERT-Modus. Am Zeilenende wird jedoch getrennt! Außerdem werden alle überflüssigen Leerzeichen gelöscht

### Der Einfüge-Modus STREAM

Mit dem STREAM-Modus möchte der MAXed "experimentierfreudigen" Menschen und radikalen Gegnern von Leerzeichen ein Angebot zur Unterhaltung machen. Normale Texte werden Sie in diesem Eingabe-Modus nur schwerlich schreiben können. Jedoch ist die Erstellung primitiv-codierter Geheimtexte durchaus vorstellbar. Lange Worte – kurzer Sinn: Probieren Sie es selbst aus. Sie schalten den STREAM-Modus ein mit

<ENTER> s <INSERTMODE>

#### Anmerkung

- Sie können die o.g. Eingabe-Modi auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter EDITMODE mit dem entsprechenden Kommandoattribut (*o*, *i*, *l*, *w*, *s*) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Der Parameter für Textersetzung @i wird im Eingabefeld der Steuerzeile durch das Zeichen für den aktuellen Eingabe-Modus ersetzt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

### 4.1.2 Der Zeichnen-Modus

Mit Hilfe des Kommandos DRAW können Sie in der Datei im aktuellen Fenster Linien zeichnen (siehe Abschnitt 4.4.3 Zeichnen von Linien, Kapitel 6: DRAW). Sie schalten den Zeichnen-Modus ein mit

<DRAW>

Wenn Sie anschließend die Schreibmarke mit den vier folgenden ARROW-Kommandos bewegen, überschreiben Sie den darunterliegenden Text mit einer Linie (siehe Kapitel 6: LEFTARROW, RIGHTARROW, DOWNARROW, UPARROW):

<LEFTARROW>	zeichnet eine horizontale Linie nach links weiter.
<RIGHTARROW>	zeichnet eine horizontale Linie nach rechts weiter.
<DOWNARROW>	zeichnet eine vertikale Linie nach unten weiter.
<UPARROW>	zeichnet eine vertikale Linie nach oben weiter.

Den Zeichnen-Modus schalten Sie durch Betätigen einer beliebigen anderen Taste aus.

#### Anmerkung

- Wenn Sie keine Zeichenattribute aktiviert haben, wird beim Zeichnen der Linien die Einstellung GRAPHIC verwendet. Sonst kommen die eingestellten Zeichenattribute zum Zuge (siehe Abschnitt 4.4 Die Zeichenattribute, Kapitel 6: SETATTRIB).
- Wenn Sie beim Zeichnen mit den ARROW-Kommandotasten einen Richtungswechsel vornehmen, setzt der MAXed automatisch passende Ecken oder Verzweigungen ein.
- Beim Zeichnen wird das aktuelle Fenster mitbewegt, wenn Sie mit der Schreibmarke auf den Fensterrand gelangen. Dieses Verhalten während des Zeichnen-Modus ist unabhängig vom eingestellten Fenster-Modus (siehe Abschnitt 4.1.4 Der Fenster-Modus).

Sie stellen für den Zeichnen-Modus eine größere Linienstärke ein mit  
<ENTER> a <DRAW>

Sie stellen für den Zeichnen-Modus eine kleinere Linienstärke ein mit  
<ENTER> b <DRAW>

### Anmerkung

- Die Linienstärke ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Die Einstellung der Linienstärke bleibt erhalten, solange sie nicht wieder explizit umgestellt wird.

### 4.1.3 Der Zeilenschaltungs-Modus

Beim Korrigieren bzw. Ergänzen innerhalb von bereits bestehendem Text kann es hilfreich sein, wenn Sie einen geeigneten Zeilenschaltungs-Modus eingeschaltet haben, um sich das Einfügen von Leerzeilen zu ersparen oder um mit wechselnden Einrückungen besser zurecht zu kommen (siehe Kapitel 6: RETURN).

#### Der Zeilenschaltungs-Modus NORMAL

Diese Einstellung des Zeilenschaltungs-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Durch Betätigen der Kommandotaste <RETURN> während der Texteingabe wird die Schreibmarke an die Spaltenposition des linken Formatrandes in der nächsten Zeile bewegt (siehe Abschnitt 4.3.1 Tabulatoren, Kapitel 6: MARGIN). Wenn die Schreibmarke dabei auf der letzten Zeile des sichtbaren Ausschnitts im aktuellen Fenster steht, wird zusätzlich das Fenster um etwa ein Drittel seiner Höhe nach unten verschoben.

Sie stellen den Zeilenschaltungs-Modus NORMAL ein und führen das Kommando <RETURN> gleichzeitig einmal mit dieser Modus-Einstellung aus mit

```
<ENTER> n[ormal] <RETURN>
```

### Der Zeilenschaltungs-Modus SPACE

Im dieser Einstellung des Zeilenschaltungs-Modus wird die Arbeitsweise bei Einstellung NORMAL dadurch ergänzt, daß abhängig vom Eingabe-Modus vor dem Sprung der Schreibmarke in die neue Zeile eine Leerzeile in die Datei geschrieben wird: Im Einfüge-Modus wird diese Leerzeile vor der nächsten Zeile eingefügt. im Überschreibe-Modus OVERWRITE wird die nächste Zeile mit der Leerzeile überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Dadurch können Sie in bestehenden Text Zeilen einfügen bzw. überschreiben, ohne auf den nachfolgenden bzw. unterliegenden Text achten zu müssen.

Sie stellen den Zeilenschaltungs-Modus SPACE ein und führen das Kommando <RETURN> gleichzeitig einmal mit dieser Modus-Einstellung aus mit

```
<ENTER> s[pace] <RETURN>
```

### Der Zeilenschaltungs-Modus EMPTY

Diese Einstellung des Zeilenschaltungs-Modus unterscheidet sich von der Einstellung SPACE lediglich darin, daß nur dann Leerzeilen eingefügt werden, wenn der aktuellen Zeile nicht schon eine Leerzeile folgt. Sie haben beim Einfügen in bestehenden Text i.a. keine überschüssigen Leerzeilen zwischen der Position der Schreibmarke und dem nachfolgenden Text.

Sie stellen den Zeilenschaltungs-Modus EMPTY ein und führen das Kommando <RETURN> gleichzeitig einmal mit dieser Modus-Einstellung aus mit

```
<ENTER> e[mpty] <RETURN>
```

## Der Zeilenschaltungs-Modus TEXT

In dieser Einstellung des Zeilenschaltungs-Modus wird die Arbeitsweise bei der Einstellung NORMAL dadurch ergänzt, daß die Schreibmarke in der nächsten Zeile auf die Spalte der aktuellen Einrückung gesetzt wird. Das bedeutet konkret: Ist die nächste Zeile beschrieben, springt die Schreibmarke dort in die erste beschriebene Spalte. Ist die nächste Zeile leer, so springt die Schreibmarke dort in dieselbe Spalte, in der der Text der aktuellen Zeile beginnt. Voraussetzung ist allerdings, daß die entsprechende Spalte nicht links des linken Formatrandes liegt. Mit <RETURN> gelangen Sie höchstens bis zum linken Formatrand!

Sie stellen den Zeilenschaltungs-Modus TEXT ein und führen das Kommando <RETURN> gleichzeitig einmal mit dieser Modus-Einstellung aus mit

```
<ENTER> t[ext] <RETURN>
```

Sie können verschiedene Einstellungen des Zeilenschaltungs-Modus nach Bedarf kombinieren, sofern sich diese sinnvoll ergänzen.

### Anmerkung

- Sie können den Zeilenschaltungs-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter RETMODE mit dem entsprechenden Kommandoattribut (*n*, *e*, *s*, *t*) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Der Parameter für Textersetzung  $@^$  wird im Eingabefeld der Steuerzeile durch das Zeichen für den aktuellen Zeilenschaltungs-Modus ersetzt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

### Beispiel

```
<ENTER> e t <RETURN>
```

Das Kommando bewegt die Schreibmarke in der nächsten, ggf. neu einzufügenden Leerzeile auf die Spaltenposition, ab der in der aktuellen Zeile der Text beginnt.

#### 4.1.4 Der Fenster-Modus

Sie können die Umspring-Eigenschaft der Schreibmarke von einem Fensterrand zum gegenüberliegenden ganz oder teilweise abstellen. Von dieser Möglichkeit werden Sie Gebrauch machen, wenn sich der Dateiausschnitt beim Schreiben oder beim Bewegen der Schreibmarke mit den ARROW- oder TAB-Kommandos mitbewegen soll oder Sie mit mehreren Fenstern zugleich arbeiten, die Sie häufig wechseln müssen (siehe Abschnitte 3.3.2 Bewegen der Schreibmarke, 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten, Kapitel 6: BACKTAB, TAB, DOWNARROW, LEFTARROW, RIGHTARROW, UPARROW).

Die aktuelle Einstellung des Fenster-Modus beeinflusst die Arbeitsweise der Kommandos BACKTAB, TAB, DOWNARROW, LEFTARROW, RIGHTARROW und UPARROW in gleicher Weise: In der Standard-Einstellung des Fenster-Modus springt die Schreibmarke am Fensterrand in gewohnter Weise um und bewegt sich von dort um eine Zeilen-, Spalten- bzw. Tabulatorposition weiter. In einer anderen Einstellung überspringt sie den Rand zum anliegenden Fenster und bewegt sich dort um die entsprechenden Positionen weiter. Schließlich gibt es Einstellungen, bei der die Schreibmarke sich trotz Fensterrand um die entsprechende Position weiterbewegt, indem sich der Dateiausschnitt im aktuellen Fenster gleichzeitig um etwa ein Drittel der Fensterbreite in dieselbe Richtung verschiebt.

Zum Umstellen des Fenster-Modus können Sie eines der ARROW-Kommandos DOWNARROW, LEFTARROW, RIGHTARROW oder UPARROW verwenden. Die Wirkung bleibt dieselbe, die Schreibmarke wird dabei nicht weiterbewegt.

## Der Fenster-Modus NORMAL

Diese Einstellung des Fenster-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Wenn Sie mit der Schreibmarke durch eines der Kommandos BACKTAB, TAB, DOWNARROW, LEFTARROW, RIGHTARROW oder UPARROW auf bzw. über den Fensterrand bewegen, springt die Schreibmarke automatisch zum gegenüberliegenden Fensterrand in derselben Zeile bzw. Spalte um. Dadurch bleibt die Schreibmarke immer innerhalb des aktuell sichtbaren Ausschnitts. Diese Umspring-Eigenschaft können Sie auch zum schnellen Positionieren der Schreibmarke ausnutzen (siehe Kapitel 6: HOME).

Sie stellen den Fenster-Modus NORMAL ein mit

```
<ENTER> n[ormal] <ARROW-KOMMANDO>
```

## Der Fenster-Modus HORIZONTAL

Diese Einstellung des Fenster-Modus entspricht in vertikalen Bewegungen der Einstellung NORMAL. Für die horizontalen Bewegungen der Schreibmarke durch eines der o.g. Kommandos ist die Umspring-Eigenschaft der Schreibmarke abgestellt. Stattdessen verschiebt sich der Dateiausschnitt um etwa ein Drittel der Fensterbreite in dieselbe Richtung, in der sich die Schreibmarke danach weiterbewegen kann. Spätestens am Zeilenanfang, in seltenen Fällen auch an der rechten Zeilenbegrenzung geht es nicht mehr weiter (siehe Anhang: Maximalwerte bei der Dateigröße).

Sie stellen den Fenster-Modus HORIZONTAL ein mit

```
<ENTER> h[orizontal] <ARROW-KOMMANDO>
```

### Der Fenster-Modus VERTICAL

Diese Einstellung des Fenster-Modus entspricht in horizontalen Bewegungen der Einstellung NORMAL. Für die vertikalen Bewegungen der Schreibmarke durch eines der o.g. Kommandos ist die Umspring-Eigenschaft der Schreibmarke abgestellt. Stattdessen verschiebt sich der Dateiausschnitt um etwa ein Drittel der Fensterhöhe in dieselbe Richtung, in der sich die Schreibmarke danach weiterbewegen kann. Spätestens am Zeilenanfang, in seltenen Fällen auch an der unteren Dateibegrenzung geht es nicht mehr weiter (siehe Anhang: Maximalwerte bei der Dateigröße).

Sie stellen den Fenster-Modus VERTICAL ein mit

`<ENTER> v[ertical] <ARROW-KOMMANDO>`

### Der Fenster-Modus MOVE

Diese Einstellung des Fenster-Modus kombiniert die beiden Einstellungen HORIZONTAL und VERTICAL. Die Umspring-Eigenschaft der Schreibmarke ist für alle vier Himmelsrichtungen abgestellt. Stattdessen verschiebt sich der Dateiausschnitt in dieselbe Richtung, in der sich die Schreibmarke bewegt.

Sie stellen den Fenster-Modus MOVE ein mit

`<ENTER> m[ove] <ARROW-KOMMANDO>`

## Der Fenster-Modus CHANGEWINDOW

Diese Einstellung des Fenster-Modus ist für die Arbeit mit mehreren Fenstern gedacht. Die Umspring-Eigenschaft der Schreibmarke ist für alle vier Himmelsrichtungen abgestellt, solange am entsprechenden Fensterrand ein weiteres Fenster angrenzt. Die Schreibmarke überspringt den Rand und bewegt sich im nächsten Fenster weiter, das dabei zum aktuellen Fenster wird. Das Überschreiten des äußersten Fenstersrandes läßt der MAXed aus technischen Gründen nicht zu: In diesem Fall muß die Schreibmarke doch wieder umspringen, und zwar wie gewohnt zum gegenüberliegenden Rand des aktuellen Fensters.

Sie stellen den Fenster-Modus CHANGEWINDOW ein mit

```
<ENTER> c[hangewindow] <ARROW-KOMMANDO>
```

Für eine Kombination verschiedener Einstellungen des Fenster-Modus sieht der MAXed keine Möglichkeit.

### Anmerkung

- Sie können den Fenster-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter WINMODE mit dem entsprechenden Kommandoattribut (*n*, *c*, *h*, *v*, *m*) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Der Parameter für Textersetzung @ > wird im Eingabefeld der Steuerzeile durch das Zeichen für den aktuellen Fenster-Modus ersetzt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

### Beispiel

```
<ENTER> v <LEFTARROW>
```

Die Schreibmarke springt jetzt nicht mehr "überraschenderweise" um, wenn Sie beim vertikalen Weiterbewegen der Schreibmarke den Fensterrand übersehen.

### 4.1.5 Der Such-Modus

Der MAXed kennt fünf Einstellungen des Such-Modus. Die im Abschnitt 3.10 beschriebene Arbeitsweise des MAXed bei Verwendung eines PATTERNS als Kommandoattribut entspricht der Arbeitsweise im Standard-Such-Modus.

#### Anmerkung

Die Einstellung des Such-Modus beeinflusst nicht die *tag*-Suche mit dem Kommando GOTO (siehe Abschnitt 4.2.15 TAGLIST, Kapitel 6:GOTO):  
 <ENTER> 'proc <GOTO>

Für dieses Kommando ist grundsätzlich die intelligente Einstellung des Such-Modus gültig.

### Der Standard-Such-Modus

Diese Einstellung des Such-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Sie stellen den Standard-Such-Modus ein mit

```
<ENTER> * <PLUSFIND>
bzw.
<ENTER> * <MINUSFIND>
```

Dies ist die intelligenteste Variante, die sich in folgenden Eigenschaften von den anderen Einstellungen des Such-Modus unterscheidet:

- Der MAXed sucht im Standard-Such-Modus passende Textstellen prinzipiell im Wortkontext. Sie können bei einem Suchmuster jedoch den linken bzw. rechten Kontext unter Verwendung der spitzen Klammern < bzw. > ausdrücklich angeben oder unberücksichtigt lassen.
- Die nicht-alphanumerischen Zeichen haben in diesem Such-Modus i.a. eine Sonderbedeutung. Wenn Sie Sonderzeichen innerhalb eines PATTERNS als Klartext verwenden wollen, müssen Sie sie mit dem Gegenstrich \ entwerfen.

Im folgenden sind die vier anderen Varianten mit ihren Einschränkungen gegenüber dem Standard-Such-Modus aufgelistet.

## Der \-Such-Modus

In dieser Einstellung des Such-Modus verlieren alle nicht-alphanumerischen Zeichen mit Ausnahme der unten aufgeführten ihre Sonderbedeutung. Eine passende Textstelle wird jedoch weiterhin standardmäßig im Wortkontext gefunden. Die Eigenschaften dieser Einstellung des Such-Modus entsprechen im Standard-Such-Modus dem Fall, in dem Sonderzeichen durch den Gegenschrägstrich \ entwertet werden. Im \-Such-Modus wird ein PATTERN als Suchmuster im Klartext interpretiert. Sie stellen den \-Such-Modus ein mit

```
<ENTER> \ <PLUSFIND>
bzw.
<ENTER> \ <MINUSFIND>
```

Folgende Sonderzeichen behalten ihre Sonderbedeutung:

\ Der Gegenschrägstrich entwertet gültige Sonderzeichen.

< bzw. >

Mit Hilfe der spitzen Klammern geben Sie den linken bzw. den rechten Kontext einer gesuchten Textstelle an.

"mu ster"

Mit Anführungszeichen schließen Sie ein PATTERN ein, das Leerzeichen enthält.

modus Die Kommandoattributierungen zum Einstellen des Such-Modus (\*, \, >, < und <>) sind weiterhin gültig.

## Anmerkung

In dieser Einstellung des Such-Modus wird grundsätzlich zwischen Groß- und Kleinschreibung unterschieden.

### Der <-Such-Modus

Diese Einstellung des Such-Modus stellt eine weitere Einschränkung gegenüber dem \-Such-Modus dar. Der linke Kontext einer passenden Textstelle bleibt grundsätzlich unberücksichtigt. Als rechter Kontext für eine gesuchte Textstelle ist nach wie vor der Wortkontext voreingestellt. Dies entspricht im \-Such-Modus dem Fall, in dem das erste Zeichen eines Suchmusters grundsätzlich eine öffnende spitze Klammer ist, sodaß der linke Kontext beliebig ist. Sie stellen den <-Such-Modus ein mit

```
<ENTER> < <PLUSFIND>  
bzw.  
<ENTER> < <MINUSFIND>
```

Die öffnende spitze Klammer < verliert auch ihre Sonderbedeutung innerhalb eines PATTERNS.

### Der >-Such-Modus

Diese Einstellung des Such-Modus stellt analog zum obigen <-Such-Modus eine weitere Einschränkung gegenüber dem \-Such-Modus dar. Der rechte Kontext einer passenden Textstelle bleibt grundsätzlich unberücksichtigt. Als linker Kontext für eine gesuchte Textstelle ist nach wie vor der Wortkontext voreingestellt. Dies entspricht im \-Such-Modus dem Fall, wo das letzte Zeichen eines Suchmusters grundsätzlich eine schließende spitze Klammer ist, sodaß der rechte Kontext beliebig ist. Sie stellen den >-Such-Modus ein mit

```
<ENTER> > <PLUSFIND>  
bzw.  
<ENTER> > <MINUSFIND>
```

Die schließende spitze Klammer > verliert auch ihre Sonderbedeutung innerhalb eines PATTERNS.

### Der < >-Such-Modus

Diese Einstellung des Such-Modus kombiniert die Eigenschaften der beiden vorstehenden Such-Modi <-Such-Modus und >-Such-Modus. Weder der linke noch der rechte Kontext einer gesuchten Textstelle werden berücksichtigt. Im <>-Modus geben Sie ein Suchmuster stets im Klartext für die gesuchte Textstelle mit beliebigem Kontext ein. Sie stellen den <>-Such-Modus ein mit

```
<ENTER> <> <PLUSFIND>  
bzw.  
<ENTER> <> <MINUSFIND>
```

Die spitzen Klammern < und > verlieren auch ihre Sonderbedeutung innerhalb eines PATTERNS.

### Anmerkung

- Sie können den Such-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter FINDMODE mit dem entsprechenden Kommandoattribut (\*, <, >, <>) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Der Parameter für Textersetzung @\* wird im Eingabefeld der Steuerzeile durch das Zeichen für den aktuellen Such-Modus ersetzt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).
- Die Parameter für Textersetzung gelten bei allen Einstellungen des Such-Modus (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld). In den eingeschränkten Einstellungen des Such-Modus sind lediglich PATTERN-spezifische Sonderzeichen entwertet.

### 4.1.6 Der Makro-Abarbeitungs-Modus

Mit Hilfe des MAXed-Kommandos MACRO können Sie während der MAXed-Sitzung Makros ausführen lassen (siehe Abschnitt 4.7 Arbeiten mit Makros, Kapitel 6: MACRO).

Der MAXed bricht die Abarbeitung eines Makros standardmäßig ab, sobald eines der auszuführenden MAXed-Kommandos eine Fehlermeldung erzeugt. Sie können diese Steuerung durch den MAXed über die Einstellung des Makro-Abarbeitungs-Modus beeinflussen.

Wenn Sie sicher sind, daß ein Makro sinnvoll weiterarbeiten kann, obwohl eine Fehlermeldung aufgetreten ist, können Sie das Makro im Abarbeitungs-Modus CONTINUE ablaufen lassen. Sie stellen diese Einstellung des Makro-Abarbeitungs-Modus ein mit

```
<ENTER> * <MACRO>
```

Die Ausführung eines Makros wird trotz eines zuvor aufgetretenen Fehlers fortgesetzt.

Sie stellen den Makro-Abarbeitungs-Modus TERMINATE wieder ein mit

```
<ENTER> ! <MACRO>
```

Die Ausführung eines Makros wird abgebrochen, sobald ein Fehler auftritt. Die Einstellung TERMINATE des Makro-Abarbeitungs-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

#### Anmerkung

- Sie können den Makro-Abarbeitungs-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter MACCONTINUE mit dem Kommandoattribut \* oder ! belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

### 4.1.7 Der Kommando-Abarbeitungs-Modus

Mit Hilfe des MAXed-Kommandos EXECUTE können Sie während der MAXed-Sitzung SINIX-Kommandos asynchron im Hintergrund ausführen lassen (siehe Abschnitt 4.5.2 Ausführen asynchroner SINIX-Kommandos, Kapitel 6: EXECUTE).

Wenn Sie mehrere SINIX-Kommandos ausführen müssen, dürfen Sie diese dem MAXed in Form eines Pakets übergeben, indem Sie einen Textbereich mit SINIX-Kommandos definieren. Der MAXed bricht die Abarbeitung standardmäßig ab, sobald eines der Kommandos einen Endestatus ungleich Null abliefern. Sie können diese Steuerung durch den MAXed über die Einstellung des Kommando-Abarbeitungs-Modus beeinflussen.

Wenn Sie sicher sind, daß die SINIX-Kommandos voneinander unabhängig sind, können Sie sie mit der im Abarbeitungs-Modus CONTINUE ablaufen lassen. Sie stellen diese Einstellung des Kommando-Abarbeitungs-Modus ein mit

```
<ENTER> * <EXECUTE>
```

Nachfolgende SINIX-Kommandoaufräge werden trotz eines zuvor aufgetretenen Fehlers weiter verfolgt.

Sie stellen den Kommando-Abarbeitungs-Modus TERMINATE wieder ein mit

```
<ENTER> ! <EXECUTE>
```

Die Abarbeitung eines Pakets von SINIX-Kommandos wird abgebrochen, sobald ein Fehler auftritt. Die Einstellung TERMINATE des Kommando-Abarbeitungs-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

**Anmerkung**

- Sie können den Kommando-Abarbeitungs-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie den Parameter EXCONTINUE mit dem Kommandoattribut \* oder ! belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Außerdem ist in der Pseudo-Datei SETUP der Parameter EXECUTE ggf. mit dem aktuellen Zustand eines asynchron laufenden SINIX-Kommandos belegt (siehe Abschnitt 4.5 Die SINIX-Schnittstellen):  
None: es läuft gar nichts;  
Alive: Die Sub-Shell führt gerade ein asynchrones Kommando aus;  
Waiting: Die Sub-Shell, die beim Aufruf eines früheren EXECUTE-Kommandos erzeugt wurde, wartet auf Arbeit.

### 4.1.8 Der Ausgabe-Modus

Die Standard-Ausgabe und die Standard-Fehlerausgabe (Dateikennzahlen 1 und 2) asynchron ablaufender SINIX-Kommandos leitet der MAXed automatisch in die Pseudo-Datei SYSOUTPUT um (siehe Abschnitte 4.5.2 Ausführen asynchroner SINIX-Kommandos, Kapitel 6: EXECUTE, Abschnitt 4.2.11 Die Pseudo-Datei SYSOUTPUT). In dieser Datei können Sie also die Ergebnisse der aufgerufenen Programme begutachten.

Sie können die Form der Ausgabe asynchroner SINIX-Kommandos durch die Einstellung des Ausgabe-Modus beeinflussen.

#### Die Einstellungen OVERWRITE und APPEND

Wenn Sie frühere Ausgaben in der Pseudo-Datei SYSOUTPUT nicht mehr benötigen, können Sie asynchrone SINIX-Kommandos im Ausgabe-Modus OVERWRITE ablaufen lassen, sodaß durch eine neue Programmausführung eine ältere Ausgabe automatisch gelöscht wird. Sie stellen den Ausgabe-Modus OVERWRITE ein mit

```
<ENTER> > <EXECUTE>
```

Beim Aufruf eines asynchron laufenden SINIX-Kommandos wird die Pseudo-Datei SYSOUTPUT zuerst für die zu erwartenden neuen Ausgaben geräumt. Alte Ausgaben gehen Ihnen dadurch verloren, sofern Sie sie nicht in einer anderen Datei abgespeichert haben (siehe Kapitel 6: SAVE).

Sie stellen den Ausgabe-Modus APPEND wieder ein mit

```
<ENTER> > > <EXECUTE>
```

Die Ausgaben mehrerer, in der aktuellen Sitzung hintereinander ausgeführter SINIX-Kommandos werden jeweils am Ende der Pseudo-Datei angehängt. Die Einstellung APPEND des Ausgabe-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

## Die Einstellungen ECHO und NOECHO

Sie können auch festlegen, daß der MAXed automatisch den Aufruf des ausgeführten SINIX-Kommandos vor die Ausgabe in der Pseudo-Datei SYSOUTPUT einfügt. Dieser Eintrag gestattet Ihnen zu einem späteren Zeitpunkt, eine Erklärung dafür zu finden, wie die nachfolgende Ausgabe eigentlich zustande kommen konnte. Sie stellen den Ausgabe-Modus ECHO ein mit

<ENTER> + <EXECUTE>

Der SINIX-Kommandoaufruf wird in der Pseudo-Datei SYSOUTPUT vor der Ausgabe wiedergegeben. Die Kommandoaufrufe werden invers dargestellt, damit Sie sie von den ausgegebenen Daten leichter unterscheiden können! Die Einstellung ECHO des Ausgabe-Modus ist in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

Sie schalten den Ausgabe-Modus NOECHO ein mit

<ENTER> - <EXECUTE>

Es wird lediglich die Ausgabe eines SINIX-Kommandos in die Pseudo-Datei SYSOUTPUT geschrieben.

### Anmerkung

- Sie können den Ausgabe-Modus auch direkt in der Pseudo-Datei SETUP einstellen, indem Sie die Parameter EXAPPEND bzw. EXECHO mit dem entsprechenden Kommandoattribut (> > oder>) bzw. (+ oder-) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

### 4.1.9 Der Zurücksetzen-Modus

Sie können in beliebigem Umfang die Textveränderungen, die Sie während einer laufenden Sitzung vorgenommen haben, wieder mit dem MAXed-Kommando UNDO rückgängig machen (siehe Kapitel 6: UNDO). Dabei ist es häufig von Vorteil, zwischen einer Vorgehensweise, die eine Makro-Ausführung als eine Textveränderungs-Einheit behandelt, und einer anderen, die ein Makro wieder in einzelne Textveränderungs-Schritte aufspaltet, umschalten zu können. Darüberhinaus besteht oft die Notwendigkeit, das Zurücksetzen von Textveränderungen nur für einen Teil Ihrer bearbeiteten Dateien vornehmen zu lassen.

Die Einstellung des Zurücksetzen-Modus ändern Sie, indem Sie beim Aufruf des Kommandos UNDO die Kommandoattribute *modus* (für die Makrobehandlung) oder IDENTIFIER (für die Dateiauswahl) entsprechend angeben. Diese Einstellung bleibt für weitere UNDO-Kommandos erhalten, bis Sie sie wieder explizit durch eine neue Angabe umstellen. Der Aufruf des attributierten Kommandos UNDO hat folgende Syntaxdarstellung:

```
<ENTER> [modus] [richtung] [n] [IDENTIFIER]
          [RANGE] [<TEXTVERÄNDERUNGS-KOMMANDO>] <UNDO>
```

### Die Makrobehandlung

Standardmäßig betrachtet der MAXed beim Zurücksetzen alle Textveränderungen einer Makro-Ausführung als eine Textveränderungseinheit. Sie können jedoch den Zurücksetzen-Modus durch eine entsprechende Angabe für das Kommandoattribut *modus* so einstellen, daß die Ausführung eines Makros in einzelne Textveränderungs-Schritte aufgespalten wird. Sie stellen den Zurücksetzen-Modus auf die Makrobehandlung in Einzelschritten ein mit

```
<ENTER> , [weitere kommandoattribute] <UNDO>
```

Das Zurücksetzen bearbeiteter Dateien auf einen vorausgegangenen Stand erfolgt ab jetzt auch für Textveränderungen, die während einer Makro-Ausführung vorgenommen wurden, in Einzelschritten.

Sie stellen den Zurücksetzen-Modus wieder auf die Makrobehandlung als Textveränderungs-Einheit ein mit

```
<ENTER> ; [weitere kommandoattribute] <UNDO>
```

Sämtliche Änderungen, die durch die Ausführung eines Makros entstanden sind, werden mit einem UNDO-Kommandoaufruf auf einen Schlag rückgängig. Der Zurücksetzen-Modus ist in der Standard-Konfiguration des MAXed mit dieser Makrobehandlung voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

### Die Dateiauswahl

Standardmäßig sind alle bearbeiteten Datei der aktuellen MAXed-Sitzung beim Zurücksetzen von Textveränderungen einbezogen. Sie können jedoch den Zurücksetzen-Modus durch eine entsprechende Angabe für das Kommandoattribut IDENTIFIER so einstellen, daß das Zurücksetzen von Textveränderungen auf die angegebenen Dateien beschränkt bleibt. Sie stellen den Zurücksetzen-Modus auf eine eingeschränkte Dateiauswahl ein mit

```
<ENTER> [weitere kommandoattr.] IDENTIFIER  
[weitere kommandoattr.] <UNDO>
```

Ab jetzt machen Sie nur in den durch IDENTIFIER angegebenen Dateien Textveränderungen rückgängig (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).

Sie stellen den Zurücksetzen-Modus wieder auf die Auswahl aller bearbeiteten Dateien der aktuellen Sitzung ein mit

```
<ENTER> [weitere kommandoattribute] *  
[weitere kommandoattribute] <UNDO>
```

Ab jetzt sind wieder alle bearbeiteten Dateien beim Zurücksetzen einbezogen. Der Zurücksetzen-Modus ist in der Standard-Konfiguration des MAXed mit dieser Dateiauswahl voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

### Anmerkung

- Sie können die Dateiauswahl des Zurücksetzen-Modus auch direkt in der Pseudo-Datei SETUP angeben, indem Sie den Parameter UNDOFILE mit dem gewünschten IDENTIFIER belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

#### 4.1.10 Die Sprach-Modi

Der MAXed kann sich in einem gewissen Rahmen auf Ihre sprachlichen Bedürfnisse einstellen. Er verfügt über zwei Ebenen von Spracheinstellungen. Die eine Ebene betrifft die Umgangssprache, mit der der MAXed Ihnen in Meldungs- und Hilfstexten aufwartet (siehe unten: Der Konversationsmodus). Die andere Ebene bezieht sich lediglich auf den Silbentrenner für Texte in verschiedenen Sprachen und seine Datums- und Zeitausgabe (siehe unten: Der Trennmodus).

##### Der Konversationsmodus

In welcher Sprache Ihnen der MAXed während der MAXed-Sitzung begegnen soll, das müssen Sie schon vor Sitzungsbeginn festlegen. Der Konversationsmodus für die Meldungen des MAXed in der Steuerzeile und die Hilfstexte wird durch die Belegung der Umgebungsvariablen *LANG* bestimmt (siehe Abschnitt 3.5.2 Die Steuerzeile, Kapitel 6: HELP). Der MAXed greift beim Start auf diese Variable zu und stellt für die Dauer der Sitzung die geforderte Spracheinstellung ein.

##### Anmerkung

- Standardmäßig ist die Umgebungsvariable *LANG* vom Betriebssystem so vorgelegt, daß der englische Konversationsmodus eingestellt wird.
- Die korrekten Belegungen der Umgebungsvariablen *LANG* entnehmen Sie bitte der Beschreibung zum [*N*]ative [*L*]anguage [*S*]upport.

**Beispiel**

Sie belegen auf der *Bourne*-Shell die Variable LANG für eine deutschsprachige Konversation mit

```
LANG=De_DE.646  
export LANG
```

Danach können Sie den MAXed aufrufen. Er spricht deutsch mit Ihnen.

Sie ändern die Spracheinstellung für die englische Konversation um mit

```
LANG=En_US.ASCII  
export LANG
```

Nach dem Aufruf meldet sich der MAXed auf Englisch.

## Der Trennmodus

Der MAXed trennt im LEXICAL-Modus beim Zeilenumbruch, wenn Sie beim Schreiben über die rechte Formatgrenze hinauskommen oder Textbereiche mit den Kommandos FILL bzw. FORMAT formatieren (siehe Abschnitte 4.1.1 Die Eingabe-Modi, 4.3.2 Formatränder, 4.3.3 Flatter- und Blocksatz). Dabei geht der MAXed nach sprachabhängigen Trennregeln vor. Außerdem paßt sich die Datums- und Zeit-Ausgabe über die Parameter für Textersetzung *@d* und *@t* der Einstellung des Trennmodus an (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

Sie stellen den Silbentrenner des MAXed auf die deutschen Spracheigenheiten ein mit

```
<ENTER> De <FILL>  
oder  
<ENTER> De <FORMAT>
```

Sie bereiten den Silbentrenner auf englische Texte vor mit

```
<ENTER> En <FILL>  
oder  
<ENTER> En <FORMAT>
```

## Anmerkung

- Die Umstellung des Trennmodus können Sie beliebig während der MAXed-Sitzung vornehmen. Der eingestellte Konversationsmodus ist davon nicht betroffen.
- Sie können den Trennmodus auch direkt in der Pseudo-Datei SETUP verändern, indem Sie den Parameter LANGUAGE mit dem entsprechenden Kommandoattribut (*En*, *De*) belegen (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).
- Der Parameter für Textersetzung *@g* wird im Eingabefeld der Steuerzeile durch das Kommandoattribut für den Trennmodus ersetzt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

#### 4.1.11 Der Lern-Modus

Im Lern-Modus können Sie während der Arbeit in der aktuellen MAXed-Sitzung Makros erstellen (siehe Abschnitt 4.7 Arbeiten mit Makros, Kapitel 6: LEARN). Sie beginnen mit der Definition des Standard-Makros bzw. mit der Definition eines benannten Makros durch

```
<LEARN>
bzw.
<ENTER> makro [makro-parameter] <LEARN>
```

Ab jetzt werden alle Tastenanschläge bis ins Detail aufgezeichnet. Die eventuell angegebenen aktuellen Makro-Parameter eines benannten Makros *makro* können Sie während der Lern-Phase, d.h. die Dauer des Lern-Modus, in der Steuerzeile durch Textersetzung der formalen Makro-Parameter wieder abrufen (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld). Dadurch erstellen Sie ein parametrisiertes Makro, das Sie zur gegebenen Zeit mit anderen aktuellen Parametern wieder aufrufen können.

Während der Lernphase arbeitet MAXed normal weiter, so daß beispielhaft die Funktionsweise des angelegten Makros sichtbar wird. Durch die inverse Darstellung von *Mode* auf der linken Seite der Steuerzeile wird angezeigt, daß gerade eine Lernphase eingeschaltet ist (siehe Abschnitt 3.5.2 Die Steuerzeile).

#### Anmerkung

- Wenn Sie während der Definition eines neuen Makros *makro2* in der Lern-Phase ein bereits definiertes benanntes Makro durch das Kommando  

```
<ENTER> makro1 <MACRO>
```

aufrufen, wird *makro1* ausgeführt und der Aufruf im gerade zu definierenden Makro aufgenommen. Auf diese Weise können Sie Makros ineinander verschachteln. Die von *makro1* ausgelösten Aktionen werden jedoch nicht einzeln im Makro *makro2* abgespeichert, sodaß *makro2* für seine zukünftigen Ausführungen voraussetzt, daß *makro1* definiert ist!
- Das nicht attributierte Kommando <MACRO> beendet jedoch den Lern-Modus, speichert das gerade definierte Makro ab und führt es anschließend das erste Mal aus.

Sie beenden den Lern-Modus und speichern das Standard-Makro bzw. das benannte Makro im zugehörigen Makro-Puffer ab mit  
<LEARN>

Alternativ beenden Sie den Lern-Modus, indem Sie die aktuelle Lern-Phase abrechnen und die Aufzeichnungen nicht in einem Makro-Puffer abspeichern:  
<ENTER> <LEARN>

Sobald Sie die Definition eines Makros abgeschlossen haben, steht es Ihnen für die Dauer der aktuellen Sitzung zur Anwendung zur Verfügung.

Sie können ein nicht mehr benötigtes, definiertes Makro durch die Kommandofolge

<ENTER> [IDENTIFIER] <LEARN> <LEARN>

für die aktuelle MAXed-Sitzung löschen. Dies erreichen Sie ebenso, wenn Sie den Makronamen in der Pseudo-Datei MACROS löschen (siehe Abschnitt 4.2.10 Die Pseudo-Datei MACROS).

### Anmerkung

- Wird eine Makro-Definition erfolgreich abgeschlossen, dann ist das neue Makro als aktuelles Makro im Kommando-Puffer zu MACRO gespeichert (siehe Abschnitte 3.12.2 Kommandos mit Standard-Puffern). Sie können es aufrufen mit  
<MACRO>
- Wenn Sie die Definition des Standard-Makro abgeschlossen haben, ist dieses zusätzlich in einem gesonderten Kommando-Puffer gespeichert, sodaß Sie es auch unabhängig vom aktuellen Makro aufrufen können mit  
<ENTER> <MACRO>
- Makros, die Sie während einer MAXed-Sitzung angelegt haben, werden bei Beendigung der Sitzung gelöscht. Sie haben allerdings die Möglichkeit, die aktuell definierten Makros in einer MACRO-Bibliothek abzuspeichern (siehe Abschnitte 4.7 Arbeiten mit Makros, 7.1 Eine MACRO-Bibliothek erstellen).
- In der Pseudo-Datei SETUP ist der Parameter LEARN im Lern-Modus mit dem Namen des gerade zu erstellenden Makros oder im Falle des Standard-Makros mit der Zeichenkette *on* belegt (siehe Abschnitt 4.2.17 Die Pseudo-Datei SETUP).

## 4.2 Die Pseudo-Dateien

Der MAXed stellt Ihnen für die Bearbeitung Ihrer Dateien zusätzlich sogenannte Pseudo-Dateien zur Verfügung. Diese Dateien geben Ihnen Hilfestellung bei der Arbeit mit dem MAXed, protokollieren einige von Ihnen eingegebene Kommandoattribute und erleichtern Ihnen den Überblick. Die Pseudo-Dateien werden während einer Sitzung automatisch angelegt und existieren temporär bis zum Sitzungsende. Jede der Pseudo-Dateien hat einen eigenen, festgelegten Anwendungszweck. Die Aufgaben der Pseudo-Dateien beinhalten u.a.:

- *On-Line*-Dokumentation über die MAXed-Kommandos
- Sammeln aller Textbereiche, die Sie in den bearbeiteten Dateien verändert haben, sodaß Sie wieder darauf zurückgreifen können
- Auflisten und Einstellen verschiedener aktueller Parameter des MAXed
- Sammeln und Auflisten aller oder einer Auswahl existierender IDENTIFIER (siehe Abschnitt 3.8 Der IDENTIFIER als Kommandoattribut)
- Sammeln und Auflisten eingegebener PATTERN (siehe Abschnitt 3.10 PATTERN als Kommandoattribut) und definierter Umgebungsvariablen
- Ein- oder Ausgabe für SINIX-Kommandos (siehe Abschnitt 4.5 Die SINIX-Schnittstellen)

Die meisten Pseudo-Dateien können Sie nur eingeschränkt oder gar nicht verändern. Sie haben jedoch die Möglichkeit, in ihnen herumzublättern und daraus Textbereiche zu kopieren (siehe Abschnitt 3.3.2 Bewegen der Schreibmarke, Abschnitt 3.3.3 Verschieben des Fensters). Wenn Sie den Inhalt einer Pseudo-Datei über das Sitzungsende hinweg sichern wollen, müssen Sie ihn in einer eigenen Datei abspeichern.

Einige der Pseudo-Dateien können Sie direkt zur Kommandoattributierung eines bestimmten zugehörigen MAXed-Kommandos verwenden: Diese Pseudo-Dateien enthalten eine Auflistung von Kommandoattributen, aus denen Sie eines für den Aufruf des zugehörigen Kommandos auswählen können.

Sie blenden eine mit dem Kommando ANWENDEN-KOMMANDO verbundene Pseudo-Datei im aktuellen Fenster beim Aufruf eines Kommandos nach folgendem Schema ein mit:

```
<ENTER> @ <ANWENDEN-KOMMANDO>  
bzw.  
<ENTER> IDENTIFIER <ANWENDEN-KOMMANDO>
```

Hierbei müssen durch die Angabe IDENTIFIER mehrere passende, existierende IDENTIFIER gleichzeitig angesprochen werden (siehe Abschnitt 3.8 Der IDENTIFIER als Kommandoattribut). Die auswählbaren Kommandoattribute sind untereinander aufgelistet. Im Eingabefeld der Steuerzeile meldet sich der MAXed mit der Aufforderung:

**WAEHLE** kommandoattribut und **gebe** <ANWENDEN-KOMMANDO>

Sie wählen aus, indem Sie die Schreibmarke auf die Zeile mit dem gewünschten Kommandoattribut bewegen, und führen das Kommando durch ein weiteres Betätigen der zugehörigen Kommandotaste aus:  
<ANWENDEN-KOMMANDO>

Sie brechen die Auswahl i.a. ab durch  
<EDIT>

Die Alternativ-Datei, d.h. die Datei, die Sie zuvor im aktuellen Fenster hatten, wird wieder eingeblendet und kein Kommando ausgeführt (siehe Abschnitt 3.4.1 Wechseln zwischen Dateien).

### Anmerkung

Wenn Sie eine Auswahlliste mit PICK- oder ERASE-Puffernamen, Markennamen, Makronamen oder PATTERN aufgerufen haben, wird beim Auslösen der zugehörigen Kommandotaste zuerst die Alternativ-Datei wieder eingeblendet und das Kommando mit dem ausgewählten Kommandoattribut von der aktuellen Schreibmarkenposition aus ausgeführt.

**Beispiel 1**

Sie blenden sich die Pseudo-Datei SYSCOMMAND (siehe unten) im aktuellen Fenster ein mit

```
<ENTER> @ <EXECUTE>
```

Hier haben sich schon mehrere Aufrufe von SINIX-Kommandos angesammelt, die Sie in der aktuellen Sitzung ausführen ließen (siehe Abschnitt 4.5 Die SINIX-Schnittstellen). Da Sie die Kommandoaufrufe noch in einer weiteren MAXed-Sitzung verwenden könnten, vielleicht sogar als Shell-Prozedur, speichern Sie den Inhalt der Pseudo-Datei einfach in einer neuen Datei *my\_special\_commands* ab (siehe Kapitel 6: SAVE):

```
<ENTER> my_special_commands <SAVE>
```

In einer weiteren MAXed-Sitzung erinnern Sie sich an Ihre Kommandosammlung und wollen den Inhalt der Datei einfachheitshalber wieder in die Pseudo-Datei SYSCOMMAND kopieren. Sie blenden sich diese erst mal ein:

```
<ENTER> @ <EXECUTE>
```

und versuchen

```
<ENTER> my_special_commands <PUT>
```

Die Pseudo-Datei SYSCOMMAND können Sie im Gegensatz zu den meisten anderen Pseudo-Dateien nach Belieben verändern; die Kommandosammlung ist an der aktuellen Schreibmarkenposition eingefügt (siehe Kapitel 6: PUT).

**Beispiel 2**

Sie haben sich die Pseudo-Datei EDITFILES erstellt und mit <ERASE> die für Sie momentan uninteressanten Dateinamen gelöscht. Die Auswahlliste brauchen Sie zur Zeit häufiger. Sie speichern also wiederum den Inhalt in einer neuen Datei ab:

```
<ENTER> datei_liste <SAVE>
```

Zu einem späteren Zeitpunkt rufen Sie die Pseudo-Datei EDITFILES mit dieser Liste wieder auf:

```
<ENTER> @ datei_liste <EDIT>
```

### 4.2.1 Die Pseudo-Datei HELP

In dieser Datei können Sie sich die *On-Line*-Dokumentation zum MAXed anzeigen lassen. Sie blenden sich eine alphabetisch geordnete, kommentierte Liste aller zur Verfügung stehenden Hilfsinformationen ein mit

```
<ENTER> @ <HELP>
```

(siehe Kapitel 6: HELP). Sie können sich detailliertere Informationen zu allen MAXed-Kommandos, Hilfsprogrammen und wichtigen Begriffen anzeigen lassen mit

```
<ENTER> begriff <HELP>
```

Informationen zu einem MAXed-Kommando KOMMANDO erhalten Sie aber auch auf ganz einfache Weise durch

```
<ENTER> <HELP> <KOMMANDO>
```

Die HELP-Texte, die Sie sich einblenden lassen, können Sie nicht verändern. In diesen Texten sind alle Begriffe, zu denen es noch weitere Informationen gibt, in Großbuchstaben geschrieben. Zum Aufrufen weiterer HELP-Informationen steht Ihnen in der Pseudo-Datei HELP ein einfaches Verfahren zur Verfügung: Positionieren Sie die Schreibmarke in einem HELP-Text auf den Begriff, zu dem Sie weitere Hilfsinformation wünschen und geben Sie ein:

```
<HELP>
```

Falls die Hilfsinformation existiert, wird sie in der Pseudo-Datei im aktuellen Fenster eingeblendet. Auf diese Weise hangeln Sie sich von einem HELP-Text zum anderen weiter.

Sie blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.2 Die Pseudo-Datei RESTORE

Diese Datei speichert alle Textbereiche, die Sie in ihren bearbeiteten Dateien während der aktuellen MAXed-Sitzung gelöscht, kopiert oder verändert haben (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut). Im einzelnen sind das folgende Aktionen:

- Textbereiche löschen und in ERASE-Puffer kopieren (siehe Abschnitt 3.8.2 Der ERASE-Puffername als IDENTIFIER)
- Textbereiche in PICK-Puffer kopieren (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER)
- Textbereiche im OVERWRITE-Modus überschreiben (siehe Abschnitt 4.1 Die Eingabe-Modi)
- Textbereiche formatieren (siehe Abschnitte 4.3.3 Flatter- und Blocksatz, 4.3.4 Ausrichten und Vereinheitlichen)

Darunter fallen nicht MAXed-Kommandos die nur mit Einzelzeichen arbeiten (siehe Kapitel 6: BACKSPACE, DELCHAR). Die betroffenen Textbereiche werden jeweils ans Ende der Pseudo-Datei RESTORE geschrieben. Falls sie die (beträchtliche) Maximalgröße einer gewöhnlichen Datei erreicht hat, wird sie vom Dateianfang her wieder neu überschrieben (siehe Anhang: Einschränkungen bei der Dateigröße).

Sie blenden die Pseudo-Datei RESTORE im aktuellen Fenster ein mit

<ENTER> <EDIT>

Diese Datei können Sie nicht verändern. Sie haben jedoch die Möglichkeit, aus dieser Datei "verloren gegangene" Textbereiche in einen PICK-Puffer zu kopieren und wieder in eine Ihrer bearbeiteten Dateien zu übertragen.

Sie blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.3 Die Pseudo-Datei OPENFILES

Diese Datei enthält die Liste mit den Namen aller Dateien, die bisher in der aktuellen MAXed-Sitzung geöffnet wurden. Dazu gehören sowohl Ihre bearbeiteten Dateien als auch alle weiteren zur Zeit existierenden Pseudo-Dateien. Sie blenden die Pseudo-Datei OPENFILES im aktuellen Fenster ein mit

```
<ENTER> @ <EDIT>
bzw.
<ENTER> @ <SAVE>
```

Die Namen aller geöffneten Dateien werden zeilenweise in folgender Darstellungsweise aufgelistet:

- Die Namen von Pseudo-Dateien sind zur Unterscheidung in spitzen Klammern < > eingeschlossen.
- Zu Ihrer Übersicht sind die Namen aller Dateien, die Sie seit Sitzungsbeginn verändert haben, invers dargestellt. Die inverse Darstellung eines Dateinamens wird wieder aufgehoben, wenn Sie die entsprechende Datei wieder auf den Stand vor Sitzungsbeginn setzen (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

Sie können diese Datei nicht verändern. Die Schreibmarke ist auf dem Namen der zuletzt bearbeiteten Datei positioniert. Sie wählen sich einen Dateinamen durch Positionieren der Schreibmarke aus und geben ein:

```
<EDIT>
bzw.
<SAVE>
```

Wenn Sie die Kommandotaste <EDIT> betätigen, wird die ausgewählte Datei im aktuellen Fenster eingeblendet. Sie befindet sich noch in genau dem Zustand, in dem Sie sie zuletzt in einem Ihrer Fenster zurückgelassen hatten (siehe Abschnitt 3.4 Arbeiten mit mehreren Dateien, Kapitel 6: EDIT). Die Pseudo-Datei OPENFILES ist nun die Alternativ-Datei dieses Fensters.

Wenn Sie stattdessen die Kommandotaste <SAVE> betätigen, wird der aktuelle Zustand der ausgewählten Datei auf die Festplatte abgespeichert (siehe Abschnitt 3.1 Die Dateihaltung des MAXed, Kapitel 6: SAVE).

#### 4.2.4 Die Pseudo-Datei EDITFILES

Diese Datei enthält eine Auswahlliste mit den Namen existierender Dateien. Sie erstellen eine Auswahlliste, die in der Pseudo-Datei EDITFILES im aktuellen Fenster eingeblendet wird:

```
<ENTER> IDENTIFIER ["PATTERN"] <EDIT>  
bzw.  
<ENTER> @ IDENTIFIER <EDIT>
```

Bei der Eingabe der Kommandoattribute müssen Sie folgendes beachten:

- Wenn Sie den Namen eines Dateiverzeichnisses als IDENTIFIER angeben, wird die Liste der Dateinamen aus dem angegebenen Dateiverzeichnis erstellt, in die Pseudo-Datei EDITFILES eingelesen und diese im aktuellen Fenster eingeblendet. Hierbei werden nur die sichtbaren Dateinamen des Dateiverzeichnisses aufgelistet (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER). Zusätzlich ist auch der Pfadname des übergeordneten Dateiverzeichnisses beigefügt, sodaß Sie sich ggf. wieder im Verzeichnisbaum hocharbeiten können.
- Wenn Sie einen Dateinamen als IDENTIFIER verwenden, der mehrere passende Dateinamen gleichzeitig anspricht, wird eine Auswahlliste der passenden Dateinamen erstellt, in die Pseudo-Datei EDITFILES eingelesen und diese im aktuellen Fenster eingeblendet (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).
- Wenn Sie zusätzlich auch noch ein PATTERN (in Anführungszeichen!) angeben sollten, werden unter den passenden Dateinamen nur jene in die Auswahlliste aufgenommen, die eine zu PATTERN passende Textstelle enthalten (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).
- Wenn Sie als IDENTIFIER den Namen einer Datei angeben, in der Dateinamen untereinander aufgelistet sind, und dem Dateinamen einen durch Leerzeichen getrennten Klammeraffen @ voranstellen, wird der Inhalt dieser Datei in die Pseudo-Datei EDITFILES eingelesen und diese im aktuellen Fenster eingeblendet. Bei relativen Pfadnamen fügt der MAXed automatisch den Pfadnamen des aktuell eingestellten Dateiverzeichnisses vorne an.

Die auswählbaren Dateinamen sind zeilenweise mit vollständigem Pfadnamen aufgelistet. Sie können in dieser Datei einzelne Zeilen oder einen definierten RANGE löschen, um die Einträge, die für Sie momentan nicht interessant sind, zu entfernen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut). Sie blenden die Pseudo-Datei EDITFILES mit der zuletzt erstellten Auswahlliste im aktuellen Fenster ein durch

`<ENTER> * <EDIT>`

Die Datei befindet sich noch in genau dem Zustand, in dem Sie sie zuletzt in einem Ihrer Fenster zurückgelassen hatten. Sie wählen sich einen Dateinamen durch Positionieren der Schreibmarke aus und blenden die Datei ein mit

`<EDIT>`

Wenn Sie den Namen einer gewöhnlichen Datei als Kommandoattribut ausgewählt haben, wird diese Datei im aktuellen Fenster eingeblendet. Die Pseudo-Datei EDITFILES ist nun die Alternativ-Datei dieses Fensters.

Wenn Sie den Namen eines Dateiverzeichnisses ausgewählt haben, werden wiederum die Dateinamen aus diesem Dateiverzeichnis als Auswahlliste in die Pseudo-Datei EDITFILES eingelesen, d.h. der bisherige Inhalt von EDITFILES wird überschrieben und die Auswahlliste im aktuellen Fenster eingeblendet.

### 4.2.5 Die Pseudo-Datei PUTFILES

Diese Datei enthält eine Auswahlliste mit den Namen existierender Dateien. Sie erstellen eine Auswahlliste, die in der Pseudo-Datei PUTFILES im aktuellen Fenster eingeblendet wird, mit

<ENTER> IDENTIFIER <PUT>

Hierbei müssen zwei Voraussetzungen erfüllt sein:

- Es darf kein PICK-Puffernamen zu IDENTIFIER passen (siehe Abschnitt 4.2.7 Pseudo-Datei PICKBUFFERS)
- Durch IDENTIFIER werden mehrere passende Dateinamen gleichzeitig angesprochen (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES).

Die Namen aller passenden Dateinamen werden in die Pseudo-Datei PUTFILES eingelesen und diese im aktuellen Fenster eingeblendet.

Sie blenden die Pseudo-Datei PUTFILES mit der zuletzt erstellten Auswahlliste im aktuellen Fenster ein durch

<ENTER > \* <PUT>

Die Datei befindet sich noch in genau dem Zustand, wie Sie sie zuletzt in einem Ihrer Fenster belassen hatten. Sie wählen sich einen Dateinamen durch Positionieren der Schreibmarke aus und blenden diese Datei ein mit <PUT>

Die Alternativ-Datei wird wieder im aktuellen Fenster eingeblendet und der gesamte Inhalt der ausgewählten Datei an der aktuellen Schreibmarkenposition in die Datei kopiert. Die Pseudo-Datei PUTFILES ist nun die Alternativ-Datei dieses Fensters.

Sie brechen das Kommando ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

#### 4.2.6 Die Pseudo-Datei ERASEBUFFERS

Diese Datei enthält eine Auswahlliste mit den Namen benannter ERASE-Puffer, die Sie bisher in der aktuellen Sitzung erstellt haben (siehe Abschnitt 3.8.2 ERASE-Puffername als IDENTIFIER). Sie blenden die Pseudo-Datei ERASEBUFFERS im aktuellen Fenster ein mit

```
<ENTER> @ <RESTOR>
bzw.
<ENTER> IDENTIFIER <RESTOR>
```

Hierbei müssen durch IDENTIFIER mehrere passende ERASE-Puffernamen gleichzeitig angesprochen werden (siehe Abschnitt 3.8.2 Der ERASE-Puffername als IDENTIFIER). Die Namen aller existierenden ERASE-Puffer inklusive dem \_-ERASE-Puffer bzw. nur die zu IDENTIFIER passenden ERASE-Puffernamen werden in die Pseudo-Datei ERASEBUFFERS eingelesen und diese im aktuellen Fenster eingeblendet. Die ERASE-Puffernamen werden zeilenweise in folgender Darstellungsweise aufgelistet:

```
IDENTIFIER bereich:Text der ersten Zeile des Pufferinhalts
```

An der Position von *bereich* steht der Bereichstyp des mit IDENTIFIER benannten ERASE-Puffers: *RN* (RANGE), *RC* (RECTANGLE) oder *ST* (STREAM). Sie können in dieser Datei einzelne Zeilen oder einen definierten RANGE löschen, um die ERASE-Puffer, zu entfernen, die Sie nicht mehr brauchen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut). Diese ERASE-Puffer sind dann nicht mehr definiert!

Sie wählen sich einen ERASE-Puffer durch Positionieren der Schreibmarke aus und führen das Kommando aus mit

```
<RESTOR>
```

Die Alternativ-Datei wird wieder im aktuellen Fenster eingeblendet und der Inhalt des ausgewählten ERASE-Puffers an der aktuellen Schreibmarkenposition in die Datei kopiert. Die Pseudo-Datei ERASEBUFFERS ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl eines ERASE-Puffers ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.7 Die Pseudo-Datei PICKBUFFERS

Diese Datei enthält eine Auswahlliste mit den Namen benannter PICK-Puffer, die Sie bisher in der aktuellen Sitzung erstellt haben (siehe Abschnitt 3.8.2 PICK-Puffername als IDENTIFIER). Sie blenden die Pseudo-Datei PICKBUFFERS im aktuellen Fenster ein mit

```
<ENTER> @ <PUT>
bzw.
<ENTER> IDENTIFIER <PUT>
```

Hierbei müssen durch IDENTIFIER mehrere passende PICK-Puffernamen gleichzeitig angesprochen werden (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER). Die Namen aller existierenden PICK-Puffer inklusive dem \_-PICK-Puffer bzw. nur die zu IDENTIFIER passenden PICK-Puffernamen werden in die Pseudo-Datei PICKBUFFERS eingelesen und diese im aktuellen Fenster eingeblendet. Die PICK-Puffernamen werden zeilenweise in folgender Darstellungsweise aufgelistet:

IDENTIFIER bereich: Text der ersten Zeile des Pufferinhalts

An der Position von *bereich* steht der Bereichstyp des mit IDENTIFIER benannten PICK-Puffers: *RN* (RANGE), *RC* (RECTANGLE) oder *ST* (STREAM). Sie können in dieser Datei einzelne Zeilen oder einen definierten RANGE löschen, um die PICK-Puffer, zu entfernen, die Sie nicht mehr brauchen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut). Diese PICK-Puffer sind dann nicht mehr definiert. Sie wählen sich einen PICK-Puffer durch Positionieren der Schreibmarke aus und führen das Kommando aus mit

```
<PUT>
```

Die Alternativ-Datei wird wieder im aktuellen Fenster eingeblendet und der Inhalt des ausgewählten PICK-Puffers an der aktuellen Schreibmarkenposition in die Datei kopiert. Die Pseudo-Datei PICKBUFFERS ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl eines PICK-Puffers ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

## 4.2.8 Die Pseudo-Datei LABELS

Diese Datei enthält eine Auswahlliste mit den Namen von Marken, die Sie in der aktuellen Datei gesetzt haben (siehe Abschnitt 3.8.4 Der Markenname als IDENTIFIER). Sie blenden die Pseudo-Datei LABELS im aktuellen Fenster ein mit

```
<ENTER> @ <GOTO>  
bzw.  
<ENTER> IDENTIFIER <GOTO>
```

Hierbei müssen durch IDENTIFIER mehrere passende Markennamen gleichzeitig angesprochen werden (siehe Abschnitt 3.8.4 Der Markenname als IDENTIFIER). Die Namen aller existierenden Marken einschließlich der Standard-Marke bzw. nur die zu IDENTIFIER passenden Markennamen werden in die Pseudo-Datei LABELS eingelesen und diese im aktuellen Fenster eingeblendet. Die Markennamen werden zeilenweise in folgender Darstellungsweise aufgelistet:

```
[IDENTIFIER] zeilennr: Zeilentext mit gekennzeichneteter Marken-  
Pos.
```

Bei der Standard-Marke ist kein IDENTIFIER angegeben. An der Position von *zeilennr* steht die Zeilennummer, in der die mit IDENTIFIER benannte Marke gesetzt ist. Die genaue Position innerhalb des angezeigten Zeilentextes ist durch ein grafisches Zeichen markiert. Sie können in dieser Datei einzelne Zeilen oder einen definierten RANGE (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut) löschen, um die Marken, die Sie nicht mehr brauchen, zu entfernen. Diese Marken sind dann nicht mehr definiert. Sie wählen sich eine Marke durch Positionieren der Schreibmarke aus und führen das Kommando aus mit

```
<GOTO>
```

Die Alternativ-Datei wird wieder im aktuellen Fenster eingeblendet und die Schreibmarke springt an die Position an der die ausgewählte Marke gesetzt ist. Die Pseudo-Datei LABELS ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl einer Marke ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.9 Die Pseudo-Datei FINDHISTORY

Diese Datei enthält die Auswahlliste aller PATTERN, die Sie im Verlauf der MAXed-Sitzung als Kommandoattribute beim Aufruf der Kommandos PLUSFIND und MINUSFIND angegeben haben (siehe Abschnitt 3.10 PATTERN als Kommandoattribut, Kapitel 6: PLUSFIND, MINUSFIND). Sie blenden die Pseudo-Datei FINDHISTORY im aktuellen Fenster ein mit

```
<ENTER> @ <MINUSFIND>  
bzw.  
<ENTER> @ <PLUSFIND>
```

Alle PATTERN, mit denen Sie diese beiden Kommandos bisher attribuiert haben, sind zeilenweise aufgelistet. Die Schreibmarke ist auf dem letzten PATTERN positioniert. Sie können diese Datei nicht verändern. Sie wählen sich ein PATTERN durch Positionieren der Schreibmarke aus und führen ein Kommando aus mit

```
<MINUSFIND>  
bzw.  
<PLUSFIND>
```

Wenn Sie die Kommandotaste <MINUSFIND> betätigen, wird die Alternativ-Datei wieder im aktuellen Fenster eingeblendet. Das Suchkommando wird von der aktuellen Schreibmarkenposition aus in Rückwärtsrichtung ausgeführt.

Wenn Sie stattdessen die Kommandotaste <PLUSFIND> betätigen, wird die Alternativ-Datei wieder im aktuellen Fenster eingeblendet und das Suchkommando von der aktuellen Schreibmarkenposition aus in Vorwärtsrichtung ausgeführt. Die Pseudo-Datei FINDHISTORY ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl eines PATTERNs ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

#### 4.2.10 Die Pseudo-Datei MACROS

Diese Datei enthält eine Auswahlliste mit den Namen benannter Makros, die Sie bisher in der aktuellen Sitzung definiert haben (siehe Abschnitt 3.8.2 Makroname als IDENTIFIER). Sie blenden die Pseudo-Datei MACROS im aktuellen Fenster ein mit

```
<ENTER> @ <MACRO>  
bzw.  
<ENTER> IDENTIFIER <MACRO>
```

Hierbei müssen durch IDENTIFIER mehrere passende Makronamen gleichzeitig angesprochen werden (siehe Abschnitt 3.8.5 Der Makroname als IDENTIFIER). Die Namen aller existierenden Makronamen bzw. nur die zu IDENTIFIER passenden Makronamen werden in die Pseudo-Datei MACROS eingelesen und diese im aktuellen Fenster eingeblendet. Die Makronamen werden zeilenweise in folgender Darstellungsweise aufgelistet:

```
IDENTIFIER [kommentartext zu diesem makro]
```

Einem mit IDENTIFIER benannten Makro wird unter folgender Bedingung ein Kommentartext angefügt: Dieses Makro muß in einer MACRO-Bibliothek, die Sie im Verlauf der Sitzung geladen haben, oder in der Standard-MACRO-Bibliothek definiert sein. Der Quelltext zu dieser MACRO-Bibliothek enthielt in der Kopfzeile der Makro-Definition diesen Kommentartext (siehe Abschnitt 4.7 Arbeiten mit Makros).

Sie können in der Pseudo-Datei MACROS einzelne Zeilen oder einen definierten RANGE löschen, um die Makros, die Sie nicht mehr brauchen, zu entfernen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut). Diese Makros sind dann nicht mehr definiert.

#### Anmerkung

Diejenigen Makros, die in einer MACRO-Bibliothek definiert sind, können Sie selbstverständlich jederzeit wieder für die aktuelle Sitzung definieren, indem Sie diese MACRO-Bibliothek laden.

Sie wählen sich ein Makro durch Positionieren der Schreibmarke aus und führen das Kommando aus mit

<MACRO>

Die Alternativ-Datei wird wieder im aktuellen Fenster eingeblendet und das ausgewählte Makro wird von der aktuellen Schreibmarkenposition aus ausgeführt. Die Pseudo-Datei MACROS ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl eines Makros ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.11 Die Pseudo-Datei SYSOUTPUT

Diese Datei enthält standardmäßig die Ausgaben aller während der MAXed-Sitzung ausgeführten EXECUTE-Kommandos (siehe Abschnitt 4.5 Die SINIX-Schnittstellen, Kapitel 6: EXECUTE). Die Standard-Ausgabe und die Standard-Fehlerausgabe (Dateikennzahlen 1 und 2) eines SINIX-Kommandos, das Sie mit Hilfe des MAXed-Kommandos während einer MAXed-Sitzung starten, werden ans Ende der Pseudo-Datei SYSOUTPUT geschrieben. Sie blenden die Pseudo-Datei SYSOUTPUT im aktuellen Fenster ein mit

```
<ENTER> > <EDIT>
```

Sie können in dieser Datei Textbereiche löschen, jedoch nicht anderweitig bearbeiten. Denjenigen Teil des Ausgabetextes, den Sie benötigen, kopieren Sie sich in eine Ihrer bearbeiteten Dateien oder speichern gleich die ganze Datei unter einem neuen Namen ab (siehe Kapitel 6: SAVE).

Sie blenden die Alternativ-Datei wieder ein mit <EDIT>.

#### Anmerkung

- Sie können den Ausgabe-Modus auf die Pseudo-Datei SYSOUTPUT so einstellen, daß vor einer Ausgabe auch der Kommandoaufruf des SINIX-Kommandos eingefügt wird (siehe Abschnitt 4.1.8 Der Ausgabe-Modus). Er hebt sich durch eine inverse Darstellung von der eigentlichen Ausgabe ab.
- Weiterhin können Sie den Ausgabe-Modus so einstellen, daß die Pseudo-Datei SYSOUTPUT durch jede Ausgabe eines SINIX-Kommandos überschrieben wird.

### 4.2.12 Die Pseudo-Datei SYSHISTORY

Diese Datei enthält alle Aufrufe der SINIX-Kommandos, die Sie bisher mit dem MAXed-Kommando EXECUTE während der MAXed-Sitzung aufgerufen haben (siehe Kapitel 6: EXECUTE). Sie blenden die Pseudo-Datei SYSHISTORY im aktuellen Fenster ein mit

```
<ENTER> < <EDIT>
```

Die bisher aufgerufenen SINIX-Kommandos sind zeilenweise in folgender Darstellung aufgelistet:

- Alle Aufrufe von SINIX-Kommandos, deren Ausführung bereits abgeschlossen ist, sind invers dargestellt.
- Die Aufrufe jener SINIX-Kommandos, bei denen ein Fehler auftrat, die also mit einem Endestatus ungleich Null beendet wurden, sind zusätzlich blinkend dargestellt.

Sie können diese Datei im Gegensatz zur Pseudo-Datei SYSCOMMAND nicht verändern (siehe Abschnitt 4.2.13 Die Pseudo-Datei SYSCOMMAND).

Sie wählen sich ein SINIX-Kommando durch Positionieren der Schreibmarke aus oder definieren einen RANGE mit den Aufrufen mehrerer SINIX-Kommandos (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut). Sie starten die definierten SINIX-Kommandos nochmals mit `<ENTER> [bereich] <EXECUTE>`

Sie blenden die Alternativ-Datei wieder ein mit `<EDIT>`.

#### Anmerkung

Wenn Sie EXECUTE mit mehreren Aufrufen von SINIX-Kommandos attributieren, werden diese wie bei einer Shell-Prozedur in der Reihenfolge des Auftretens abgearbeitet. Beachten Sie dabei, daß die Pseudo-Datei SYSOUTPUT durch jede Ausgabe eines SINIX-Kommandos von neuem überschrieben wird, wenn der Ausgabe-Modus nicht entsprechend eingestellt ist (siehe Abschnitte 4.2.11 Die Pseudo-Datei SYSOUTPUT 4.1.8 Der Ausgabe-Modus).

### 4.2.13 Die Pseudo-Datei SYSCOMMAND

Diese Datei enthält zunächst wie die Pseudo-Datei SYSHISTORY alle Aufrufe der SINIX-Kommandos, die Sie bisher mit dem MAXed-Kommando EXECUTE während der MAXed-Sitzung aufgerufen haben (siehe Abschnitt 4.2.12 Die Pseudo-Datei SYSHISTORY, Kapitel 6: EXECUTE). Sie blenden die Pseudo-Datei SYSCOMMAND im aktuellen Fenster ein mit

```
<ENTER> @ <EXECUTE>
```

Die bisher aufgerufenen SINIX-Kommandos sind zeilenweise aufgelistet. Sie können diese Datei jedoch bearbeiten, um damit beispielsweise Shell-Prozeduren zusammenzustellen und diese teilweise oder im Ganzen auszu-  
testen.

Sie wählen sich ein SINIX-Kommando durch Positionieren der Schreibmarke aus oder definieren einen RANGE mit den Aufrufen mehrerer SINIX-Kommandos (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut). Sie starten die definierten SINIX-Kommandos nochmals mit  
<ENTER> [bereich] <EXECUTE>

Sie blenden die Alternativ-Datei wieder ein mit <EDIT>.

#### Anmerkung

Wenn Sie EXECUTE mit mehreren Aufrufen von SINIX-Kommandos attributieren, werden diese wie bei einer Shell-Prozedur in der Reihenfolge des Auftretens abgearbeitet. Beachten Sie dabei, daß die Pseudo-Datei SYSOUTPUT durch jede Ausgabe eines SINIX-Kommandos von neuem überschrieben wird, wenn der Ausgabe-Modus nicht entsprechend eingestellt ist (siehe Abschnitte 4.2.11 Die Pseudo-Datei SYSOUTPUT 4.1.8 Der Ausgabe-Modus).

#### 4.2.14 Die Pseudo-Datei SCANLIST

Diese Datei enthält eine Auswahlliste mit allen Textstellen, die zu einem bestimmten PATTERN passen und bei einer automatischen Suche in einer oder mehreren Dateien gefunden wurden (siehe Abschnitt 3.10 PATTERN als Kommandoattribut). Aus dieser Auswahlliste können Sie direkt auf jede der gefundenen Textstellen in den jeweiligen Dateien springen. Sie erstellen eine Auswahlliste, die in der Pseudo-Datei SCANLIST im aktuellen Fenster eingeblendet wird, mit

```
<ENTER> "PATTERN" [IDENTIFIER] <GOTO>
bzw.
<ENTER> bereich [IDENTIFIER] <GOTO>
```

Der MAXed durchsucht alle durch IDENTIFIER angegebenen Dateien nach dem Auftreten von Textstellen, die zum angegebenen PATTERN bzw. zu dem angegebenen Bereich passen (siehe Abschnitte 3.8.1 Der Dateiname als IDENTIFIER, 3.10 PATTERN als Kommandoattribut, Kapitel 6: GOTO). Im Eingabefeld der Steuerzeile erhalten Sie die Meldung:

Momentan laeuft: FILE SCAN

Alle Zeilen der Dateien mit einer passenden Textstelle werden in die Pseudo-Datei SCANLIST eingelesen und diese im aktuellen Fenster eingeblendet.

Bei der Eingabe der Kommandoattribute müssen Sie folgendes beachten:

- Bei diesem Kommando müssen Sie ein PATTERN grundsätzlich in Anführungszeichen einschließen. Wenn Sie für PATTERN die leere Zeichenkette "" eingeben, greift der MAXed auf das PATTERN zurück, welches Sie zuletzt bei der Ausführung dieses MAXed-Kommandos verwendet hatten.
- Wenn Sie einen einzeiligen STREAM durch Bewegen der Schreibmarke definieren, wird dieser als Suchmuster verwendet. Bereichsdefinitionen, die über eine Zeile hinausgehen sind bei diesem Kommando nicht erlaubt (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

- Wenn Sie den Namen eines Dateiverzeichnisses als IDENTIFIER verwenden, werden sämtliche Dateien des Dateiverzeichnisses nach dem angegebenen Suchmuster durchsucht, alle gefundenen Textstellen in die Pseudo-Datei SCANLIST eingelesen und diese im aktuellen Fenster eingeblendet. Hierbei werden nur die sichtbaren Dateien des Dateiverzeichnisses in Betracht gezogen (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).
- Wenn Sie einen IDENTIFIER verwenden, der mehrere passende Dateinamen gleichzeitig anspricht, werden alle diese Dateien nach dem angegebenen PATTERN durchsucht, alle gefundenen Textstellen in die Pseudo-Datei SCANLIST eingelesen und diese im aktuellen Fenster eingeblendet (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).
- Wenn Sie gar nichts für IDENTIFIER eingeben, wird ersatzweise die aktuelle Datei nach dem angegebenen Suchmuster durchsucht, alle gefundenen Textstellen in die Pseudo-Datei SCANLIST eingelesen und diese im aktuellen Fenster eingeblendet.

Die Zeilen, in denen eine oder mehrere Textstellen gefunden wurden, sind in folgender Darstellung aufgelistet:

dateiname:zeilenr:zeile mit gefundener textstelle

Jeder gefundenen Textstelle ist der Name der zugehörigen Datei und die Zeilennummer vorangestellt. Sie können in der Pseudo-Datei SCANLIST einzelne Zeilen oder einen definierten RANGE löschen, um die Einträge, die für Sie momentan nicht interessant sind, zu entfernen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut).

Sie blenden die Pseudo-Datei SCANLIST mit der zuletzt erstellten Auswahlliste im aktuellen Fenster ein durch

<ENTER> ” <GOTO>

Die Datei befindet sich noch in genau dem Zustand, in dem Sie sie zuletzt in einem Ihrer Fenster zurückgelassen hatten. Sie wählen sich eine Textstelle durch Positionieren der Schreibmarke aus und führen das Kommando aus mit  
<GOTO>

Der MAXed blendet die angesprochene Datei im aktuellen Fenster ein und springt mit der Schreibmarke auf die ausgewählte Textstelle. Die Pseudo-Datei SCANLIST ist nun die Alternativ-Datei dieses Fensters.

Sie brechen die Auswahl einer Textstelle ab und blenden die Alternativ-Datei wieder ein mit <EDIT>.

### 4.2.15 Die Pseudo-Datei TAGLIST

Diese Datei enthält eine Auswahlliste mit sogenannten *tags*, die zu einem bestimmten PATTERN passen und bei einer automatischen Suche in einer Tag-Datei gefunden wurden (siehe Anhang: Erstellen einer Tag-Datei). Aus dieser Auswahlliste können Sie direkt auf die Definitionsstelle einer Prozedur, die durch eines der Tags aufgeführt ist, in der entsprechenden Quelldatei springen (siehe Kapitel 6: GOTO). Sie erstellen eine Auswahlliste, die in der Pseudo-Datei TAGLIST im aktuellen Fenster eingeblendet wird, mit

```
<ENTER> 'proc [IDENTIFIER] <GOTO>
```

Bei der Eingabe der Kommandoattribute müssen Sie folgendes beachten:

- Es müssen durch *proc* mehrere passende *tags* aus der verwendeten Tag-Datei gleichzeitig angesprochen werden, indem Sie ein entsprechendes PATTERN eingeben (siehe Abschnitt 3.10 PATTERN als Kommandoattribut). Die passenden Tags werden in die Pseudo-Datei TAGLIST eingelesen und diese im aktuellen Fenster eingeblendet. Die *tags* werden zeilenweise aufgelistet.
- Das *tag* für das Hauptprogramm *main* einer Datei *datei.c* ist durch *Mdatei* angeben.
- Falls Sie keinen IDENTIFIER angeben, verwendet der MAXed eine Datei namens *tags* im aktuellen Dateiverzeichnis als Tag-Datei bzw. diejenige Datei, deren Namen Sie in der Umgebungsvariablen *MAXtags* angegeben haben.
- Wenn Sie IDENTIFIER angeben, so erstellt der MAXed automatisch eine Tag-Datei mit dem Namen *tags* bzw. mit dem durch *MAXtags* angegebenen Namen. Dabei verwendet der MAXed alle Dateien, deren Namen zu IDENTIFIER passen (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER). Danach blendet er die Liste aller Tags aus dieser Tag-Datei ein, die zum angegebenen PATTERN *proc* passen.

Sie können in der Pseudo-Datei TAGLIST einzelne Zeilen oder einen definierten RANGE löschen, um die Einträge, die für Sie momentan nicht interessant sind, zu entfernen (siehe Abschnitt 3.9 Bereichsdefinition als Kommandoattribut). Sie wählen sich ein Tag durch Positionieren der Schreibmarke aus und führen das Kommando aus mit <GOTO>

Der MAXed blendet die angesprochene Datei im aktuellen Fenster ein und springt mit der Schreibmarke auf die ausgewählte Definitionsstelle einer Prozedur. Die Pseudo-Datei TAGLIST ist nun die Alternativ-Datei dieses Fensters.

### 4.2.16 Die Pseudo-Datei VARIABLES

Diese Datei enthält die Belegungen aller MAXed-Variablen, die Sie in der aktuellen Sitzung definiert haben (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld, Abschnitt 4.5 Die SINIX-Schnittstellen). Sie blenden die Pseudo-Datei VARIABLES im aktuellen Fenster ein mit

```
<ENTER> = <ENTER>
```

Die bisher definierten MAXed-Variablen sind zeilenweise in folgender Darstellung aufgelistet:

```
variable=wert
```

Die Definition der MAXed-Variablen während der MAXed-Sitzung ist ähnlich der auf der SINIX-Ebene. Sie können diese Datei bearbeiten, um damit neue Variablen zu definieren oder um die Werte bzw. den Namen bereits existierender zu ändern. Sie aktivieren die neu definierten oder geänderten Umgebungsvariablen und blenden gleichzeitig die Alternativ-Datei wieder im aktuellen Fenster ein mit

```
<EDIT>
```

#### Anmerkung

- Wenn Sie eine unzulässige Änderung vorgenommen haben, erhalten Sie eine entsprechende Fehlermeldung im Eingabefeld der Steuerzeile. Die betroffenen Definitionen sind dann nicht gültig.
- MAXed-Variablen werden automatisch exportiert und sind deshalb in der Subshell des MAXed (siehe Abschnitt 4.5 Die SINIX-Schnittstellen Kapitel 6: CLI) gültig.

#### 4.2.17 Die Pseudo-Datei SETUP

Diese Datei enthält eine Reihe von Einträgen mit Parametern, die u.a. die aktuellen Einstellungen der Betriebs-Modi des MAXed und verschiedener aktueller Kommandoattribute anzeigen (Abschnitte 4.1 Die Betriebs-Modi, 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen). Der MAXed übernimmt die Belegungen einiger Parameter aus den Angaben der Konfigurationsdatei (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Sie blenden die Pseudo-Datei SETUP im aktuellen Fenster ein mit

<ENTER> ® <ENTER>

Die Einträge sind zeilenweise in folgender Darstellung aufgelistet:

```
setup-parameter=[belegung] (kommentar zu möglichen belegungswerten)
```

Die aufgelisteten Parameter sind ggf. mit den aktuellen Belegungswerten vorbesetzt. Sie können diese Belegungswerte in der Pseudo-Datei SETUP ändern. Dabei müssen Sie die Regeln für zulässige Werte beachten:

- Hinweise für die jeweils zulässigen Werte entnehmen Sie den Kommentaren in Klammern.
- Die meisten dieser Parameter können Sie während der MAXed-Sitzung auch dadurch einstellen, daß Sie ein zugehöriges MAXed-Kommando mit einem entsprechenden Kommandoattribut aufrufen. Die möglichen Belegungswerte eines solchen Parameters sind identisch mit den möglichen Werten des Kommandoattributs, das bei Aufruf des zugehörigen MAXed-Kommandos diesen Parameter einstellt (siehe unten: Auszüge aus der Pseudo-Datei SETUP).
- Während Sie die Pseudo-Datei SETUP im aktuellen Fenster eingeblendet haben, ist das aktuelle PATTERN temporär so eingestellt, daß Sie mit Hilfe der Kommandos MINUSFIND bzw. PLUSFIND auf die Position des vorausgehenden bzw. nachfolgenden Belegungswertes eines Parameters springen können. Das bisherige aktuelle PATTERN, dessen Belegung durch den Parameter FINDPATTERN angezeigt wird, ist nach Verlassen der Pseudo-Datei SETUP wieder gültig (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).

Sie aktivieren die neuen Parameterbelegungen und blenden gleichzeitig die Alternativ-Datei wieder im aktuellen Fenster ein mit

<EDIT>

### Anmerkung

- Wenn Sie eine unzulässige Änderung vorgenommen haben, erhalten Sie eine entsprechende Fehlermeldung im Eingabefeld der Steuerzeile. Die Alternativ-Datei wird nicht eingeblendet. Die Schreibmarke wird auf den ersten fehlerhaften Belegungswert gesetzt.
- Sie machen alle Ihre Änderungen in dieser Datei wieder rückgängig mit <ENTER> - <EDIT>

Hier sind auszugsweise einige Einträge aus einer Pseudo-Datei SETUP mit Belegungsbeispielen aufgelistet:

```

EDITMODE=I      (Insert, Lexical, Word, Stream)
RETMODE=N      (Normal, Empty, Space, Text)
WINMODE=N      (Normal, Change, Horizontal, Vertical, Move)
FINDMODE=<>    (*, \, <, >, <>)
FINDPATTERN=\alf (Muss zwischen "" stehen, falls Blanks enthalten sind)
REPLACESTRING=Alf (Muss zwischen "" stehen, falls Blanks enthalten sind)
LANGUAGE=E     (En, De, Fr)
LMARGIN=3     (1...9999)
RMARGIN=79    (1...9999)
MACCONTINUE=! (* oder Continue, ! oder Terminate)
DEFDIR=/usr/alf
NEWFILEMODE=600 (owner:rwe, group:rwe, others:rwe)
SCANPATTERN=<ach> (Muss zwischen "" stehen, falls Blanks enthalten sind)
SCANFILES=/usr/alf/a,b (Dateiname darf Wildcards enthalten)
ATTRIBUTES=U:1 (Intensity:*1, Underscore:*1, Blinking:*1, Reversed:*1, ...
UNDOFILE=*    (* (fuer alle) oder offene MAXed-Datei(en))
CLI=N        (None, Alive, #Clilines)
EXECUTE=N    (Active, Waiting, None)
EXCONTINUE=! (* oder Continue, ! oder Terminate)
EXAPPEND=>   (>> oder Append, > oder Overwrite)
EXECHO=-    (+ oder Echo, - oder Noecho)
BAKFILES=B   (Remove, Bak)

```

### 4.3 Das Formatieren

Der MAXed bietet Ihnen eine Reihe von Möglichkeiten, Ihre Texte in Form zu bringen. Sie können ganze Textbereiche formatieren oder während des Schreibens Zeilen geschickt umbrechen und zusammenzufügen. Falls es bei einer dieser Aktionen zu unerwünschten Ergebnissen kommen sollte, stellt der MAXed Mittel bereit, um Schäden zu beheben:

Sie machen die letzten Textveränderungen in den geladenen Dateien schrittweise wieder rückgängig (siehe Kapitel 6: UNDO) mit <UNDO>

Sie blenden sich die Pseudo-Datei RESTORE, in der u.a. alle geänderten Textbereiche der aktuellen Sitzung gespeichert wurden, im aktuellen Fenster ein (siehe Abschnitt 4.2.2 Die Pseudo-Datei RESTORE) mit <ENTER> <EDIT>

### 4.3.1 Tabulatoren

Wenn Sie nicht nur Fließtexte zwischen den beiden Formaträndern schreiben wollen, sondern formal stärker strukturierte Texte, wie z.B. Tabellen, deren Felder linksbündig an bestimmten Spalten beginnen, erweist es sich als vorteilhaft, mit einem oder wenigen Bewegungskommandos die Schreibmarke an die richtigen Positionen bewegen zu können. Zu diesem Zweck führt der MAXed Tabulatoren, die Sie an beliebige Spaltenpositionen setzen dürfen. Sie setzen an der aktuellen Spaltenposition der Schreibmarke einen Tabulator bzw. löschen einen bereits vorhandenen (siehe Kapitel 6: SRTAB) mit

```
<SRTAB>
```

Damit Sie nicht sämtliche Tabulatoren in einer gerade geladenen Datei jedesmal von neuem setzen müssen, definieren Sie sich die öfter verwendeten Tabulatoreinstellungen in speziellen Dateien. In einer solchen Tabulator-Datei beschreiben Sie lediglich die erste Zeile mit den Spaltenangaben der zu setzenden Tabulatoren:

```
spaltenr,spaltennr,spaltennr, ...
```

Alternativ können Sie in jeder Spalte, an der ein Tabulator zu setzen ist, einen Punkt eingeben.

Der MAXed setzt in der aktuellen Datei die Tabulatoren entsprechend den Angaben in der Tabulator-Datei *tab-datei* mit

```
<ENTER> tab-datei <SRTAB>
```

#### Anmerkung

- Die Tabulatoren sind in der Standard-Konfiguration des MAXed an den Spalten 1, 5, 9, ..., 125 gesetzt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Außerdem sind im Bibliotheksverzeichnis des MAXed */usr/lib/MAXed* bereits die Tabulator-Dateien 4 (steht für alle vier Spalten: 1, 5, 9, ..., 125) und 8 (steht für alle acht Spalten: 1, 9, 17, ..., 129) angelegt.
- Tabulatoren sind dateispezifisch. Sie dürfen in einer Datei bis zu 34 Tabulatoren an beliebigen (verschiedenen) Spaltenpositionen setzen. Am unteren Rand des aktuellen Fensters sind mit senkrechten Strichen nach oben die Spaltenpositionen der Tabulatoren gekennzeichnet.

- Beim Beenden des MAXed werden gesetzte Tabulatoren in der benutzer- und terminalspezifischen Status-Datei abgespeichert. Wenn Sie beim nächsten Aufruf des MAXed keine Dateiangaben machen, wird die Status-Datei eingelesen, und die Tabulatoren werden automatisch wieder gesetzt (siehe Kapitel 5: Der Aufruf des MAXed).

### Beispiel

```
<ENTER> $MAXlib/8 <SRTAB>
```

Für die aktuelle Datei setzen Sie die Tabulatoren auf jeder achten Spalte. Voraussetzung: Der Umgebungsvariablen *MAXlib* haben Sie zuvor den Namen des Bibliotheksverzeichnisses des MAXed zugewiesen (siehe Abschnitt 3.4.3 Textersetzung im Eingabefeld):

```
<ENTER> MAXlib=/usr/lib/MAXed <ENTER>
```

Sie bewegen die Schreibmarke in der aktuellen Zeile zur Spalte des nächsten bzw. des vorausgehenden Tabulators (siehe Kapitel 6: BACKTAB, TAB) mit

```
<TAB>
```

bzw.

```
<BACKTAB>
```

### Anmerkung

Je nachdem, wie Sie den Fenster-Modus eingestellt haben, springt die Schreibmarke zum gegenüberliegenden Fensterrand um oder verschiebt gleichzeitig das aktuelle Fenster, wenn Sie die Schreibmarke mit TAB oder BACKTAB über einen Fensterrand hinausbewegen (siehe Abschnitt 4.1.4 Der Fenster-Modus).

### 4.3.2 Formatränder

Wenn Sie während des Schreibens oder hinterher Ihren Text zwischen zwei bestimmten Spaltenpositionen einfassen wollen, setzen Sie zuvor an diesen Spalten die Formatränder.

An den linken Formatrand

der folgenden Zeile wird die Schreibmarke positioniert, wenn Sie in der vorausgehenden Zeile die Kommandotaste <RETURN> betätigen. Wenn Sie den automatischen Zeilenumbruch auslösen (siehe unten), wird der umbrochene Zeilenrest in die nächste Zeile geschoben. Er wird linksbündig am linken Formatrand ausgerichtet und die Schreibmarke dahinter positioniert, sodaß Sie unbesehen weitertippen können. Ein ähnliches Verhalten ergibt sich, wenn Sie den Zeilenumbruch mittels des Kommandos WRAP an beliebiger Stelle per Hand ausführen (siehe Kapitel 6: WRAP, Abschnitt 4.3.5 Umbrechen und Zusammenfügen).

#### *Anmerkung*

Falls in der nächsten Zeile bereits Text links vom linken Formatrand steht, bleibt dieser dort stehen. Der automatisch umbrochene Zeilenrest wird immer in den Text ab der Spalte des linken Formatrandes verschoben. Hierbei wird im Einfüge-Modus der Zeilenrest eingefügt, während er im Überschreibe-Modus OVERWRITE den darunterliegenden Text überschreibt (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Am rechten Formatrand

wird beim Schreiben der automatische Zeilenumbruch ausgelöst (siehe unten). Liegt der rechte Formatrand außerhalb des Bildschirms, so verschiebt sich das Fenster beim Überschreiten der Bildschirmgrenze automatisch nach rechts.

Zwischen den beiden Formaträndern

können Sie bereits bestehenden Text auf verschiedene Weise behandeln:

- Formatieren mit den Kommandos FILL oder FORMAT (siehe Abschnitt 4.3.3 Flatter- und Blocksatz)
- Ausrichten oder zentrieren mit den Kommandos ALIGN, CENTER, COMPRESS und JUSTIFY (siehe Abschnitt 4.3.4 Ausrichten und Vereinheitlichen)

*Anmerkung*

Einige dieser Kommandos funktionieren nicht oder nur eingeschränkt, wenn betroffene Textzeilen links über den linken Formatrand hinausragen.

Sie setzen an der aktuellen Spaltenpositione der Schreibmarke den linken bzw. den rechten Formatrand (siehe Kapitel 6: MARGIN) mit

```
<MARGIN>
bzw.
<ENTER> <MARGIN>
```

Sie setzen beide Formatränder zugleich an den angegebenen Spaltennummern mit

```
<ENTER> linker_rand rechter_rand <MARGIN>
```

**Anmerkung**

- Standardmäßig sind die beiden Formatränder auf Spalte eins gesetzt.
- Formatränder sind dateispezifisch. Sie können also in jeder geöffneten Datei die Formatränder anders setzen. Am oberen Rand des aktuellen Fensters sind die Spalten der Formatränder durch kleine schwarze nach unten zeigende Striche gekennzeichnet (siehe Abschnitt 3.5.1 Das aktuelle Fenster).
- Beim Beenden des MAXed werden gesetzte Formatränder in der benutzer- und terminalspezifischen Status-Datei abgespeichert. Wenn Sie beim nächsten Aufruf des MAXed keine Dateiangaben machen, wird die Status-Datei eingelesen, und die Formatränder werden automatisch wieder gesetzt (siehe Kapitel 5: Der Aufruf des MAXed).

**Beispiel**

```
<ENTER> 3 42 <MARGIN>
```

Für die aktuelle Datei setzen Sie die Formatränder auf den Spalten 3 und 42. Der linke Textrand des anschließend eingegebenen Textes wird sich auf Spalte 3 befinden. Text, der beim Schreiben die 42. Spalte transzendiert, wird an den nächsten Zeilenanfang befördert.

### Der automatische Zeilenumbruch

wird aktiviert, wenn Sie beim Schreiben den rechten Formatrand mit der Schreibmarke überschreiten. Er erfolgt entsprechend dem eingestellten Eingabe-Modus (siehe Abschnitt 4.1.1 Die Eingabe-Modi):

- In allen Einstellungen des Einfüge-Modus außer dem LEXICAL-Modus wird wortweise umbrochen.
- Im LEXICAL-Modus ist der Silbentrenner des MAXed eingeschaltet (siehe Abschnitt 4.1.10 Die Sprach-Modi). Am rechten Formatrand wird getrennt.
- Im OVERWRITE-Modus ist die Umbruchsweise des alternativ verfügbaren Einfüge-Modus gültig.

Sie lösen den automatischen Zeilenumbruch am rechten Formatrand aus, sobald Sie hinter dem rechten Formatrand eine der folgenden Tasten betätigen:

<RETURN>  
bzw.  
<Leerzeichen>

### 4.3.3 Flutter- und Blocksatz

Sie können Textbereiche als Ganzes zwischen den beiden Formaträndern im Flattersatz oder Blocksatz formatieren (siehe Abschnitt 4.3.2 Formatränder. Die einzelnen, betroffenen Textzeilen werden dabei zunächst soweit wie möglich aufgefüllt bzw. umbrochen. Der formatierte Textbereich ersetzt danach den zuvor definierten Textbereich.

#### Im Flattersatz

ist ein linksbündig ausgerichteter Text formatiert, in dem alle überflüssigen Leerzeichen entfernt sind und die einzelnen Zeilen mit ihren (höchstwahrscheinlich) unterschiedlich langen Ausläufern der rechten Formatgrenzung entgegenflattern.

#### Im Blocksatz

ist dagegen ein Text formatiert, dessen einzelne Zeilenbereiche wie beim Flattersatz aufgefüllt wurden und durch Auffüllen mit der richtigen Anzahl Leerzeichen links- und rechtsbündig ausgerichtet wurden.

Sie formatieren den Rest des aktuellen Absatzes, ab der Schreibmarkenposition im Flattersatz bzw. im Blocksatz (siehe Kapitel 6: FILL, FORMAT) mit

```
<FILL>
bzw.
<FORMAT>
```

Sie formatieren den angegebenen Textbereich oder ersatzweise den gesamten aktuellen Absatz im Flutter- bzw. Blocksatz mit

```
<ENTER> [bereich] <FILL>
bzw.
<ENTER> [bereich] <FORMAT>
```

#### Anmerkung

- Ein Absatz ist eine zusammenhängende Texteinheit, die durch den Dateianfang, das Dateieinde oder Leerzeilen begrenzt ist.
- Wenn der rechte Formatrand nicht gesetzt ist, bildet ersatzweise der rechte Rand des aktuellen Fensters die rechte Formatgrenze.
- Als Bereichsangaben sind RANGE, einzeiliger STREAM und sogar RECTANGLE möglich. In den beiden letzteren Fällen nimmt der MAXed komfortablerweise den linken und rechten Bereichsrand als Formatgrenzen her, an denen er den zu formatierenden Text ausrichtet.

Textzeilen, die über den linken Formatrand hinausragen, bleiben grundsätzlich von der Formatierung ausgespart. Der MAXed kann nämlich nur ab dem linken Formatrand formatieren und möchte solche Zeilen dabei nicht zerstören. Damit diese auch in die Formatierung mit einbezogen werden, müssen Sie sie zuvor nach rechts bis an die linke Formatgrenze schieben. Das könnten Sie z.B. so erledigen:

1. Setzen Sie den linken Formatrand zunächst vorübergehend weiter nach links auf eine Spalte vor dem Textbeginn dieser Zeilen.
2. Verschieben Sie den RANGE-Bereich (oder ersatzweise den Rest des aktuellen Absatzes) in einem Zug parallel nach rechts an die vorgehende linke Formatgrenze:  
<ENTER> [RANGE] + <FILL>  
Der angegebene Bereich wird um so viele Spalten nach rechts bewegt, wie die Schreibmarke momentan rechts vom linken Formatrand steht.
3. Setzen Sie die linke Formatgrenze wieder an die frühere Position.

Bei allen Eingabe-Modi

wird durch das Formatieren eines Textbereichs kein anderer Text überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Die Eingabe-Modi unterscheiden sich hier nur in Details:

- Im Überschreibe-Modus formatiert der MAXed alles, was nicht über den linken Formatrand hinausgeht, innerhalb des angegebenen Bereichs (siehe Abschnitt 4.1.1 Die Eingabe-Modi).
- Im Einfüge-Modus nimmt der MAXed jedoch Rücksicht auf führende Leerzeichen: Er formatiert nur über solche Zeilen "hinweg", deren Text von vorneherein an derselben Spalte linksbündig beginnt. Er beginnt also "intelligenterweise" mit einer neuen Formatiereinheit und verschiebt dabei den linken Formatrand temporär, wenn er auf eine Zeile stößt, die weiter oder weniger weit eingerückt ist als die vorhergehende.

#### *Anmerkung*

Eine Sonderbehandlung erfährt beim Formatieren die erste Textzeile des angegebenen Bereichs: Sie darf anders eingerückt sein als der Text der nachfolgenden Zeilen. Der MAXed beläßt dieser Textzeile großzügigerweise die Einrückung und formatiert trotzdem übers Zeilenende hinweg.

Unterschiedliche Behandlung der Leerzeichen:

Die verschiedenen Einstellungen des Einfüge-Modus machen beim Formatieren im Flatter- oder Blocksatz geringe Unterschiede, was das Entfernen überflüssiger Leerzeichen betrifft:

- Beim Formatieren in den Einstellungen INSERT und LEXICAL hält der MAXed zwischen Wörtern prinzipiell ein Leerzeichen als Abstand für ausreichend. Vor Satzzeichen und Klammern, außer den öffnenden Klammern, spart er gleich alle vorhandenen Leerzeichen ein.
- Im Einfüge-Modus WORD reduziert er alle Leerzeichen bis auf eines.
- Im Überschreibe-Modus OVERWRITE gilt die Arbeitsweise des alternativ verfügbaren Einfüge-Modus.

### Der Silbentrenner des MAXed

Der Silbentrenner ist im LEXICAL-Modus bzw. im alternativ verfügbaren Überschreibe-Modus OVERWRITE aktiviert (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Im Gegensatz zu allen anderen Einstellungen der Eingabe-Modi wird in diesen beiden beim Formatieren am rechten Formatrand nicht nur wortweise umbrochen, sondern getrennt!

### Anmerkung

Nach dem Ausführen eines Kommandos FILL oder FORMAT im LEXICAL-Modus ist das aktuelle PATTERN mit dem Silbentrennzeichen – belegt, sodaß Sie durch bloßes Betätigen der Kommandotasten <PLUSFIND> bzw. <MINUSFIND> zu Kontrollzwecken (auch Computer sind nicht allwissend!) sofort an die nächste Trennstelle springen können (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).

Beim Silbentrennen können Sie dem MAXed behilflich sein: Falls Sie einen Trennfehler entdecken, geben Sie dem MAXed einen "Tip", wo sich die richtige Trennstelle befindet, indem Sie die Schreibmarke unter das Anfangszeichen der Silbe setzen, die abgetrennt werden muß. Danach geben Sie eines der folgenden Kommandos ein. Der MAXed wird den aktuellen Absatz nochmal formatieren und dabei die angegebene Trennstelle berücksichtigen. Sie formatieren den ganzen aktuellen Absatz im Flattersatz bzw. Blocksatz mit

```
<ENTER> <FILL>
bzw.
<ENTER> <FORMAT>
```

### Anmerkung

- Selbstverständlich nimmt Ihnen der MAXed alles ab, was Sie ihm als trennbar vorschlagen. Wenn er es an der gewünschten Stelle doch nicht tut, dann liegt das nur daran, daß er dort nicht die Notwendigkeit für eine Trennung sieht!
- Der automatische Silbentrenner arbeitet in der Regel fehlerfrei. Trotzdem können aufgrund der Besonderheiten der deutschen Sprache Fehl- bzw. Nicht-Trennungen auftreten.

### Beispiele

Wort	MAXed-Trennung	korrekte Trennung
Schiffahrt	Schif-fahrt	Schiff-fahrt
Abfalleimer	Abfal-leimer	Abfall-eimer
Wachstum	Wach-stum	Wachs-tum
zwicken	keine Trennung	zwik-ken

In letzterem Falle können Sie den MAXed selbstverständlich nicht durch den Einsatz obiger Kommandos zur korrekten Trennung zwingen!

#### 4.3.4 Ausrichten und Vereinheitlichen

Die folgenden Kommandos zum Arrangieren von Text setzen i.a. voraus, daß die betroffenen Textbereiche nicht über den linken Formatrand hinausreichen. Wenn der rechte Formatrand fehlt, wird ersatzweise der rechte Rand des aktuellen Fensters verwendet.

Sie richten den Textanfang der aktuellen Zeile bzw. des angegebenen Bereichs an der aktuellen Spaltenposition der Schreibmarke aus (siehe Kapitel 6: ALIGN) mit

```
<ALIGN>  
bzw.  
<ENTER> RANGE <ALIGN>
```

Sie zentrieren den Text der aktuellen Zeile bzw. des angegebenen Bereichs (siehe Kapitel 6: CENTER) mit

```
<CENTER>  
bzw.  
<ENTER> [bereich] <CENTER>
```

#### Anmerkung

- Als Bereichsangaben für ALIGN ist nur RANGE möglich.
- Bei CENTER können Sie RANGE, RECTANGLE und einzeliger STREAM angeben. In den beiden letzteren Fällen nimmt der MAXed komfortablerweise den linken und rechten Bereichsrand als Formatgrenzen her. Zwischen diesen sucht er die Mitte für den zu zentrierenden Text.

Sie entfernen im Text der aktuellen Zeile bzw. des angegebenen Bereichs innerhalb der Formatgrenzen überflüssige Leerzeichen (siehe Kapitel 6: COMPRESS) mit

```
<COMPRESS>  
bzw.  
<ENTER> [bereich] <COMPRESS>
```

Sie richten den Text der aktuellen Zeile bzw. des angegebenen Bereichs zwischen den Formatgrenzen aus (siehe Kapitel 6: JUSTIFY) mit

```
<JUSTIFY>  
bzw.  
<ENTER> [bereich] <JUSTIFY>
```

### Anmerkung

- Welche Leerzeichen der MAXed für überflüssig hält, ist abhängig vom der Einstellung des Eingabe-Modus (siehe Abschnitt 4.1.1 Die Eingabe-Modi).
- Wiederum dürfen Sie bei obigen Kommandos als Bereichsangabe RANGE, RECTANGLE und einzeliger STREAM verwenden. Bei den beiden letzteren geht der MAXed nicht nach den gesetzten Formaträndern, sondern nach deren eigenen Bereichsgrenzen!
- Prinzipiell haben die beiden Kommandos COMPRESS und JUSTIFY eine gegensätzliche Arbeitsweise. Dies heißt nicht, daß sie sich gegenseitig aufheben. Es gilt die Faustregel: Wenn zwischen Worten oder Sonderzeichen alle Leerzeichen mit COMPRESS getilgt sind, werden an diesen Stellen durch JUSTIFY keine überflüssigen Leerzeichen mehr eingefügt. JUSTIFY orientiert sich nur an bereits vorhandenen Zwischenräumen.

Die folgenden Kommandos zur Umwandlung von Groß- und Kleinbuchstaben können innerhalb von Makros nützlich sein.

Sie wandeln in der aktuellen Zeile bzw. im angegebenen Bereich alle Großbuchstaben in Kleinbuchstaben um (siehe Kapitel 6: LOWER) mit

<LOWER>

bzw.

<ENTER> [bereich] <LOWER>

Sie wandeln in der aktuellen Zeile bzw. im angegebenen Bereich alle Kleinbuchstaben in Großbuchstaben um (siehe Kapitel 6: LOWER) mit

<UPPER>

bzw.

<ENTER> [bereich] <UPPER>

### **Anmerkung**

- Bei Ziffern und Sonderzeichen unterscheidet der MAXed grundsätzlich nicht zwischen Klein- und Großschreibung

### 4.3.5 Umbrechen und Zusammenfügen

Bei der Texteingabe und bei der Korrektur bestehender Texte kommt es vor, daß Zeilen zu lang geraten und über den rechten Fenster- oder Formatrand hinausragen, weil sie beim Einfügen nicht automatisch umbrochen werden (siehe Abschnitt 4.3.2 Formatränder). Andere wiederum geraten infolge von Löschaktionen zu kurz. Wenn sich die Uneinheitlichkeit über mehrere Zeilen erstreckt, empfiehlt es sich, die Formatränder entsprechend einzustellen und über die Zeilen hinweg im Flatter- oder Blocksatz zu formatieren (siehe Abschnitt 4.3.3 Flatter- und Blocksatz).

Häufig jedoch ist Handarbeit zwischen wenigen Zeilen effizienter, weil dem Formatieren z.B. verschieden große Einrückungen oder momentan falsch postierte Formatgrenzen zur schnellen Durchführung im Wege stehen.

Das MAXed-Kommando WRAP bietet die einfachste Möglichkeit, einen Zeilenrest in die nächste Zeile umzubrechen oder einen Zeilenanfang mit dem Ende der vorausgehenden Zeile zusammenzufügen (siehe Kapitel 6: WRAP): Von Hand lösen Sie einen Umbruch des Zeilenrestes ab der aktuellen Schreibmarkenposition aus mit

<WRAP>

Der Zeilenrest wird an den Anfang des Textes der nächsten Zeile geschoben. Die Schreibmarke wird ebenfalls in der nächsten Zeile am Ende des umbrochenen Zeilenrestes positioniert, sodaß Sie an dieser Stelle sofort Korrekturen oder weitere Eingaben vornehmen können.

#### Anmerkung

- Wenn die folgende Zeile leer sein sollte, wird der umbrochene Zeilenrest an der Spalte des linken Formatrandes eingefügt.
- Wenn in der nächsten Zeile Text steht, fügt das Kommando <WRAP> unabhängig von Formatrand und Eingabe-Modus den Zeilenrest ab der Spalte ein, wo der Text beginnt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Die Einrückung dieses Textes wird also automatisch beibehalten.
- Da es direkt vor den Zeilenende-Zeichen in MAXed-Zeilen keine Leerzeichen gibt, wird beim Einfügen des Zeilenrestes in den Text automatisch und praktischerweise ein trennendes Leerzeichen eingefügt.

Sie trennen den Text des Zeilenanfangs bis vor der Schreibmarkenposition von der aktuellen Zeile ab und fügen ihn am vorausgehenden Zeilenende an mit

<ENTER> <WRAP>

Die Schreibmarke wird in der vorausgehenden Zeile wiederum an der Nahtstelle positioniert.

### **Anmerkung**

Auch diese Kommandovariante funktioniert unabhängig von Formatrand und Eingabe-Modus. Auch hierbei wird ein führendes Leerzeichen als Trennzeichen zwischen ehemaligem Zeilenende und ehemaligem Textanfang eingefügt. In der zuvor aktuellen Zeile beginnt der übriggebliebene Text übrigens wieder an derselben Spalte wie der ehemalige Textanfang. Die Einrückung wurde also beibehalten.

Zuweilen ist es sinnvoll einen Zeilenrest in eine ganz neue Zeile zu schieben. Im Einfüge-Modus trennen Sie den Zeilenrest ab der aktuellen Schreibmarkenposition ab und fügen ihn vor der nächsten Zeile als neue Zeile ein mit

<ENTER> <SPACE>

### **Anmerkung**

- Die Schreibmarke bleibt an der aktuellen Position, dem neuen Zeilenende, stehen.
- Im Einfüge-Modus wird der ehemalige Zeilenrest in der neuen Zeile ab dem linken Formatrand eingeschoben.
- Im Überschreibe-Modus **OVERWRITE** würde stattdessen der aktuelle Zeilenrest mit Leerzeichen überschrieben, was einem Löschen durch folgendes Kommando gleichkäme:  
<ENTER> <ERASE>

Im Einfüge-Modus können Sie mit dem Kommando SPACE geschickt Zeichenketten aus Leerzeichen und Zeilenende-Zeichen einfügen. Sie verschieben dadurch den Zeilenrest in eine vor der nächsten Zeile neu eingefügten Leerzeile an eine gewünschte Spalte. Sie beginnen mit der Definition eines STREAM-Bereichs (siehe Abschnitt 3.9.2 Bereichsdefinition durch Bewegen der Schreibmarke):

<ENTER> <RETURN>

Sie bewegen die Schreibmarke an die gewünschte Spalte und fügen die Leerzeichen ein mit

<SPACE>

Sie fügen zwei Zeilen zusammen, indem Sie das Zeilenende-Zeichen dazwischen löschen. Dazu setzen Sie die Schreibmarke hinter das letzte Zeichen der aktuellen Zeile. Sie fügen die nächste Zeile am Zeilenende an mit

<ENTER> <ERASE>

### Anmerkung

Wenn Sie mit der Schreibmarke innerhalb einer Textzeile sitzen, wird durch dieses Kommando der Zeilenrestes ab der aktuellen Schreibmarkenposition gelöscht; die Schreibmarke ist damit bereits am neuen Zeilenende positioniert. Am Zeilenende aufgerufen, entfernt dieses Kommando jedoch ausdrücklich das Zeilenende-Zeichen.

Sie können den Anfang der aktuellen Zeile und das Zeilenende der folgenden Zeile zusammenfügen. Dazu beginnen Sie mit der Definition eines STREAM-Bereichs (siehe Abschnitt 3.9.2 Bereichsdefinition durch Bewegen der Schreibmarke):

<ENTER> <RETURN>

Sie bewegen die Schreibmarke an die gewünschte Spalte und löschen den definierten Text mit

<ERASE>.

## 4.4 Die Zeichenattribute

Sie können mit dem MAXed Ihren Text durch den Einsatz verschiedener Zeichendarstellungen und Grafik aufbereiten. Hierzu steht Ihnen folgende Auswahl an Zeichenattributen zur Verfügung:

- *intensity* (Halbhelle Darstellung)
- *underscore* (Unterstrichene Darstellung)
- *blinking* (Blinkende Darstellung)
- *reversed* (Inverse Darstellung)
- *graphic* (Grafische Darstellung)

Die verschiedenen Zeichenattribute sind den Darstellungsmöglichkeiten für Einzelzeichen entnommen, die Ihre Datensichtstation bietet. Dazu sind spezielle Codierungen notwendig, die ihre Datensichtstation versteht, mit denen Ihr Drucker jedoch nicht von vorneherein etwas anfangen kann (siehe unten)!

### Die Grafikzeichen

sind einem speziellen grafischen Zeichensatz entnommen. In der Einstellung *graphic* ist Ihre Datensichtstation auf die grafische Darstellung umgestellt, sodaß die eingegebenen alphanumerischen Zeichen und Sonderzeichen auf dem Bildschirm als grafische Zeichen erscheinen. Diese Grafikzeichen speichert der MAXed als MAXed-Grafikzeichen ab. Die Zeichen dieses grafischen Zeichensatzes sind in der Hilfsdatei *GRAPHIC* des MAXed aufgeführt. Sie blenden sich diese Datei im aktuellen Fenster ein (siehe Kapitel 6: HELP) mit

```
<ENTER> graphic <HELP>
```

Aus dieser Datei können Sie sich auch einzelne Zeichen kopieren, die Sie in Ihren Texten einsetzen möchten.

### Anmerkung

Für das Zeichnen von Linien verwenden Sie das Kommando *DRAW* (siehe Abschnitt 4.4.3 Zeichnen von Linien). Dieses Kommando benutzt jedoch auch nur einzelne Zeichen aus dem o.g. grafischen Zeichensatz, um die kontinuierlichen Linien und deren Verzweigungen darzustellen.

## Ausdrucken von Texten:

Texte, in denen Sie Zeichenattribute, also auch Linien, des MAXed verwendet haben, müssen Sie für Ihren Drucker aufbereiten. Der MAXed hat hierfür ein eigenes Hilfsprogramm: *maxlpr* (siehe Abschnitte 4.5.4 Ausdrucken von MAXed-Dateien, 7.4 MAXed-Datei für den Ausdruck aufbereiten, Kapitel 6: PRINT). Für alle unterstützten Druckertypen findet *maxlpr* eine Ersatzlösung zur Darstellung der Zeichenattribute:

<i>intensity</i>	entspricht i.a. der Normalschrift des Druckers
<i>underscore</i>	entspricht i.a. der Unterstreichung des Druckers
<i>blinking</i>	entspricht i.a. der Breitschrift des Druckers
<i>reversed</i>	entspricht i.a. der Fettschrift des Druckers
<i>graphic</i>	wird i.a. durch entsprechende Zeichen dargestellt.

#### 4.4.1 Einstellen und Aktivieren von Zeichenattributen

Um Zeichenattribute in Ihren Texten einsetzen zu können, müssen Sie zuvor die aktuelle Einstellung der Zeichenattribute einstellen, also z.B. "setzen" oder "nicht-setzen", und danach aktivieren.

Die aktuelle Einstellung der Zeichenattribute

wird auf der Steuerzeile im ATTRIBUTE-Feld angezeigt (siehe Abschnitt 3.5.2 Die Steuerzeile). Das ATTRIBUTE-Feld wird in der Steuerzeile nur dann eingeblendet, wenn Sie die Einstellungen der Zeichenattribute verändern und die aktuelle Einstellung aktiviert haben.

Sie springen mit der Schreibmarke zum Ändern der Einstellung der Zeichenattribute ins ATTRIBUTE-Feld der Steuerzeile mit

<ENTER> @ <SETATTRIB>

#### Anmerkung

- Mit den Bewegungskommandos <LEFTARROW> und <RIGHTARROW> können Sie die Schreibmarke innerhalb des ATTRIBUTE-Feldes bewegen (siehe Kapitel 6). Gleichzeitig erhalten Sie am linken Rand der Steuerzeile zur Information den Namen des Zeichenattributs angezeigt, das der aktuellen Schreibmarkenposition im ATTRIBUTE zugeordnet ist.
- Sie ändern die Einstellung, indem Sie an den gewünschten Positionen eines der drei Zeichen 0, 1 oder \* eingeben (siehe unten). Alternativ können Sie diese Zeichen auch durch Betätigen der Kommandotasten <UPARROW> und <DOWNARROW> abrufen.

Sie springen mit der Schreibmarke aus einer der Einstellungspositionen wieder in die aktuelle Datei zurück bei gleichzeitigem Abspeichern und Aktivieren der aktuellen Einstellung mit

<SETATTRIB>

Sie brechen den Vorgang ohne Abspeichern der aktuellen Einstellung ab mit

<ENTER>

**Anmerkung**

- Wenn Sie in dieser Situation mit <SETATTRIB> die Einstellung zwar abspeichern aber nicht aktivieren wollen, so positionieren Sie die Schreibmarke zuvor auf die Lücke zwischen den Vierer-Blöcken.
- Falls Sie mit <ENTER> die Änderungen abbrechen, bleibt die aktuelle Einstellung in ihrem bisherigen Zustand.

Die Einstellungen der Zeichenattribute und ihre Zuordnung im ATTRIBUTE-Feld sind auf folgende Weise dargestellt:

**ATT:**    iubr gxxx

- i        steht für die Einstellung des Zeichenattributes Halbhell ([i]ntensity).
- u        steht für die Einstellung des Zeichenattributes Unterstreichung ([u]nderscore).
- b        steht für die Einstellung des Zeichenattributes Blinken ([b]linking).
- r        steht für die Einstellung des Zeichenattributes Invers ([r]eversesed).
- g        steht für die Einstellung des Zeichenattributes Grafik ([g]raphic).
- x        steht für nicht belegte Einstellungen.

Jede dieser Positionen kann mit einer der folgenden Einstellungen belegt sein:

- 0        bedeutet, das zugeordnete Zeichenattribut ist nicht gesetzt.
- 1        bedeutet, das zugeordnete Zeichenattribut ist gesetzt.
- \*        bedeutet, das zugeordnete Zeichenattribut wird nicht berücksichtigt.

## Ein gesetztes Zeichenattribut (1)

wird nur dann bei jenen Zeichen, die Sie in eine Datei schreiben hinzugefügt, wenn die aktuelle Einstellung der Zeichenattribute aktiviert ist. Weiterhin können Sie gesetzte Zeichenattribute mit dem MAXed-Kommando ATTRIBUTE nachträglich Textbereichen hinzufügen (siehe Abschnitt 4.4.2 Hinzufügen und Entfernen von Zeichenattributen).

## Ein nicht-gesetztes Zeichenattribut (0)

wird bei eingetippten Zeichen grundsätzlich nicht am Bildschirm ausgegeben, ob die aktuelle Einstellung nun aktiviert ist oder nicht. Sie können aber nicht-gesetzte Zeichenattribute mit dem MAXed-Kommando ATTRIBUTE nachträglich wieder aus Textbereichen entfernen.

## Ein nicht-berücksichtigtes Zeichenattribut (\*)

wird auch nicht am Bildschirm ausgegeben, wenn Sie Zeichen eintippen, selbst wenn die aktuelle Einstellung aktiviert ist. Beim nachträglichen Hinzufügen oder Entfernen von Zeichenattributen in Textbereichen mit dem Kommando ATTRIBUTE werden nicht-berücksichtigte Zeichenattribute weder hinzugefügt noch entfernt.

**Anmerkung**

Wenn die aktuelle Einstellung nicht aktiviert ist, werden beim Schreiben keine Zeichenattribute hinzugefügt. Wenn Sie jedoch in einem Text schreiben, dem bereits Zeichenattribute hinzugefügt sind, übernehmen die eingegebenen Zeichen die unterliegenden Zeichenattribute. Sie aktivieren oder deaktivieren die aktuelle Einstellung der Zeichenattribute mit

<SETATTRIB>

## Setzen und Aktivieren in einem Kommando

Sie ändern die aktuelle Einstellung der Zeichenattribute und setzen Sie in einem Kommando mit

```
<ENTER> [setze-attribute] [, [lösche-attribute] <SETATTRIB>
```

Bei der Eingabe der Kommandoattributierung müssen Sie folgendes beachten:

- Für *setze-attribute* und *lösche-attribute* können Sie Namen der gewünschten Zeichenattribute eingeben wie sie in der Konfigurationsdatei definiert sind (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Laut Standard-Konfigurationsdatei kennt der MAXed Zeichenattribute mit den folgenden Namen:

```
r[eversed] b[linking] u[nderscore] i[ntensity] g[raphic]
```

Die Namen trennen Sie durch je ein Leerzeichen. Sie können diese Namen auch bis zum ersten Zeichen abkürzen und bei ausschließlicher Angabe der Anfangszeichen die trennenden Leerzeichen weglassen, solange die Eingabe eindeutig bleibt (siehe unten: Beispiel).

- Für *setze-attribute* geben Sie die Zeichenattribute an, die Sie mit der Einstellung "gesetzt" (1) aktivieren wollen.
- Für *lösche-attribute* geben Sie die Zeichenattribute an, die Sie mit der Einstellung "nicht-gesetzt" (0) aktivieren wollen. I.a. werden Sie nur solche Zeichenattribute angeben, die bereits gesetzt sind, da alle anderen ihre bisherige Einstellung (0 bzw. \*) behalten.
- Geben Sie zwar das Komma, jedoch nichts für *lösche-attribute* an, versieht der MAXed alle Zeichenattribute außer jenen, die Sie bei *setze-attribute* angeben, mit der Einstellung "nicht-gesetzt" (0).

**Anmerkung**

- Diejenigen Zeichenattribute, die in der aktuellen Einstellung gesetzt (1) sind, die Sie jedoch weder bei *setze-attribute* noch bei *lösche-attribute* angeben, werden nach der Aktivierung auf "nicht-berücksichtigen" (\*) eingestellt sein.
- Die Belegung der Zeichenattribute ist in der Standard-Konfiguration des MAXed mit 1 für *reversed* (invers) und 0 für alle anderen Positionen des ATTRIBUTE-Feldes voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Es bleibt Ihnen unbenommen, beim Start des MAXed eine eigene Konfigurationsdatei mit Ihren bevorzugten Einstellungen anzugeben (siehe Kapitel 5: Der Aufruf des MAXed).
- Beim Beenden des MAXed wird die aktuelle Einstellung der Zeichenattribute in der benutzer- und terminalspezifischen Status-Datei abgespeichert. Wenn Sie beim nächsten Aufruf des MAXed keine Dateiangaben machen, wird die Status-Datei eingelesen, und die Einstellung der Zeichenattribute automatisch übernommen (siehe Kapitel 5: Der Aufruf des MAXed).

**Beispiel 1**

```
<ENTER> underscore reversed <SETATTRIB>
```

Damit belegen Sie die Einstellungen "Unterstreichen" und "Invers" mit "gesetzt" (1), die restlichen Zeichenattribute behalten ihre bisherige Einstellung, und die Einstellung der Zeichenattribute wird aktiviert.

**Beispiel 2**

```
<ENTER> rg,i <SETATTRIB>
```

Sie setzen *reversed* und *graphic* auf 1, *intensity* auf 0, belassen die restlichen Zeichenattribute bei ihrer bisherigen Einstellung und aktivieren das Ganze auch noch. Vorsicht: Das Kommando

```
<ENTER> gr,i <SETATTRIB>
```

würde nicht dasselbe bewirken: Der MAXed würde nämlich die Zeichenkette *gr* als Abkürzung des Namens *graphic* erkennen!

**Beispiel 3**

Sie haben momentan von den Zeichenattributen genug und wollen einfach alle Einstellungen wieder mit "nicht-gesetzt" (0) belegen. Nichts einfacher als das:

<ENTER> , <SETATTRIB>

Damit die Anzeige des ATTRIBUTE-Feldes auch noch verschwindet, drücken Sie nochmal <SETATTRIB>.

#### 4.4.2 Hinzufügen und Entfernen von Zeichenattributen

Sie müssen nicht gleich während des Schreibens die Zeichen mit den gewünschten Zeichenattributen versehen. Sie können Texte nachträglich mit den richtigen Zeichenattribute versehen.

Einige Varianten des Kommandos ATTRIBUTE greifen auf die aktuelle Einstellung der Zeichenattribute zurück. Jene mit der Einstellung

0	werden im Text entfernt
1	werden im Text hinzugefügt
*	werden nicht berücksichtigt

Gemäß der aktivierten, aktuellen Einstellung der Zeichenattribute fügen Sie in der aktuellen Zeile bzw. im angegebenen Textbereich Zeichenattribute hinzu und entfernen ggf. einige der bereits vorhandenen Zeichenattribute:

```
<ATTRIBUTE>
bzw.
<ENTER> bereich <ATTRIBUTE>
```

#### Anmerkung

Diese beiden Kommandoformen greifen auf die aktuelle Einstellung der Zeichenattribute zurück, wenn sie momentan aktiviert ist! Falls nicht, geht der MAXed davon aus, daß sämtliche im angegebenen Textbereich vorhandenen Zeichenattribute zu löschen sind!

Mit einer anderen Kommandoform ändern Sie die aktuelle Einstellung der Zeichenattribute, aktivieren sie und setzen Sie im angegebenen Textbereich. Sie fügen Zeichenattribute im angegebenen Textbereich hinzu bzw. entfernen sie entsprechend den Angaben *setze-attribute* bzw. *lösche-attribute*:

```
<ENTER> bereich [setze-attribute] [, [lösche-attribute] <ATTRIBUTE>
```

Die Angaben für *setze-attribute* und *lösche-attribute* entsprechen denen beim Kommando SETATTRIB (siehe Abschnitt 4.4.1).

### 4.4.3 Zeichnen von Linien

Wenn Sie Linien, Rahmen usw. zeichnen wollen, steht Ihnen dazu das Kommando DRAW zur Verfügung (siehe Kapitel 6: DRAW). Es arbeitet ebenfalls mit den Zeichen des grafischen Zeichensatzes (siehe Abschnitt 4.4). Seine Anwendung ist jedoch recht unkompliziert. Der zugrundeliegende Text wird unabhängig vom aktuellen Eingabe-Modus von den Linien überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Sie schalten den Zeichnen-Modus ein mit

<DRAW>

Sie zeichnen ab jetzt Linien, wenn Sie die ARROW-Kommandotasten betätigen (siehe Abschnitt 4.1.2 Der Zeichnen-Modus). Mit jeder beliebigen anderen Taste beenden Sie den Zeichnen-Modus.

#### Anmerkung

Sie brauchen für diese Kommandovariante die aktuelle Einstellung der Zeichenattribute nicht zu aktivieren (siehe Abschnitt 4.4.1 Einstellen und Aktivieren von Zeichenattributen). Wenn die aktuelle Einstellung nicht aktiviert ist, verwendet der MAXed im Zeichnen-Modus automatisch Zeichen aus dem grafischen Zeichensatz. Wenn Sie die aktuelle Einstellung aktiviert haben sollten, versucht der MAXed damit zu arbeiten.

Sie zeichnen im angegebenen Textbereich in einem Zug eine Linie bzw. einen Rahmen mit

<ENTER> [bereich] <DRAW>

**Anmerkung**

- Der MAXed greift hierbei explizit auf die aktuelle Einstellung der Zeichenattribute zu, wenn sie aktiviert ist. Wenn Sie die aktuelle Einstellung nicht aktiviert haben, erreichen Sie mit diesem Kommando nichts.
- Zum Zeichnen von Linien erscheint das Setzen des Zeichenattributes *graphic* sinnvoll. Wenn Sie andere Zeichenattribute einstellen, wird der MAXed aber auch diese einsetzen.
- Als Bereichsangaben sind RANGE, einzeiliger STREAM und RECTANGLE erlaubt. In den beiden ersten Fällen wird eine einzige Linie in der aktuellen Spalte bzw. Zeile über den gesamten definierten Bereich gezogen. Im letzteren Fall zeichnet der MAXed einen Rahmen entlang den Begrenzungslinien des Rechtecks.

## 4.5 Die SINIX-Schnittstellen

Der MAXed ist ein integrierter Bestandteil der SINIX-Umgebung: Er läuft als Sohn-Prozeß der Shell, aus der Sie ihn aufgerufen haben. Entsprechend sind für den MAXed alle Variablen, die aus der Aufruf-Shell exportiert wurden, als Umgebungsvariablen gültig (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld: Ersetzen von MAXed-Variablen).

Der MAXed arbeitet auf dem Dateisystem seiner Aufrufumgebung. Als aktuelles Dateiverzeichnis ist jenes, von dem aus Sie den MAXed aufgerufen haben, eingestellt. Sie können das aktuelle Dateiverzeichnis während der aktuellen MAXed-Sitzung jedoch auch mit dem MAXed Kommando `DEFDIR` umstellen (siehe Kapitel 6: `DEFDIR`). Der MAXed verwendet Ihre Dateiangaben für den Zugriff auf das umgebende Dateisystem, wobei sich relative Pfadnamen auf das aktuelle Dateiverzeichnis beziehen (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).

### Aufrufen von SINIX-Kommandos

Sie haben die Möglichkeit, SINIX-Kommandos oder auch selbsterstellte Programme während der MAXed-Sitzung direkt aufzurufen. Dazu wechseln Sie explizit in die CLI-Sub-Shell des MAXed (siehe Abschnitt 4.5.1 Wechseln in die CLI-Sub-Shell). Oder Sie attributieren die MAXed-Kommandos `EXECUTE` oder `FILTER` mit einem gewünschten SINIX-Kommandoaufruf. Der MAXed reicht diese SINIX-Kommandoaufrufe an eine Sub-Shell weiter, welche die Ausführung übernimmt (siehe Abschnitte 4.5.2 Ausführen asynchroner SINIX-Kommandos, 4.5.3 Ausführen von Filter-Programmen). Mit dem Kommando `PRINT` können Sie ein vordefiniertes Druckkommando anstoßen (siehe Abschnitt 4.5.4 Ausdrucken von MAXed-Dateien).

### Die Sub-Shells des MAXed

Der MAXed richtet sich automatisch sogenannte Sub-Shells ein, wenn Sie mit einem der Kommandos EXECUTE, FILTER oder PRINT ein SINIX-Kommando aus der MAXed Sitzung heraus anstoßen oder explizit mit dem Kommando CLI in die Sub-Shell wechseln. Jede Sub-Shell übernimmt jenes aktuelle Dateiverzeichnis, das zum Zeitpunkt ihrer Erstellung für den MAXed eingestellt ist. Eine Sub-Shell existiert von da an parallel zum MAXed-Prozeß und steht weiterhin für die MAXed-Kommandos zur Verfügung, bei deren Aufruf sie angelegt wurde: Sie führt entweder weitere SINIX-Kommandos aus, die Sie während der Sitzung einem der Kommandos EXECUTE, FILTER oder PRINT übergeben. Oder Sie wird durch das Kommando CLI wieder eingeblendet. Dadurch bleiben auch alle in einer Sub-Shell definierten Variablen bis zum Beenden der Sub-Shell gültig. Die Sub-Shells des MAXed existieren höchstens bis zum Sitzungsende.

Alle MAXed-Variablen werden automatisch in MAXed-eigene Sohn-Prozesse, also z.B. die Sub-Shells (siehe unten) exportiert, sodaß Sie dort in jedem Falle Zugriff auf diese Variablen als auch auf die Umgebungsvariablen des MAXed haben (siehe Abschnitt 4.2.16 Die Pseudo-Datei VARIABLES).

### 4.5.1 Wechseln in die CLI-Sub-Shell

Sie können aus der MAXed-Sitzung heraus direkt in eine Sub-Shell des MAXed gelangen, ohne die Sitzung beenden zu müssen: Sie rufen das MAXed-Kommando *C[ommand] L[ine] I[nterpreter]* auf (siehe Kapitel 6: CLI). Auf dieser Shell-Ebene stehen Ihnen prinzipiell alle SINIX-Möglichkeiten offen. Der MAXed "schläft" solange, bis Sie wieder in die Sitzung zurückkehren. Der aktuelle Zustand der Sitzung ist völlig identisch geblieben. Auf diese Weise haben Sie die Möglichkeit, den MAXed jederzeit, befristet, unbürokratisch (kein automatisches Abspeichern der Dateien auf Platte) und in aller Schnelle zu verlassen, um andere Arbeiten von der Shell-Ebene aus zu verrichten, während der MAXed im Hintergrund seiner Wiedererweckung harrt.

Sie blenden die CLI-Sub-Shell des MAXed ein mit  
<CLI>

Sie blenden die MAXed-Sitzung wieder ein, die Sub-Shell bleibt erhalten und geht in den Hintergrund, indem Sie den MAXed wieder aufrufen mit  
m [IDENTIFIER]

Durch den Aufruf des MAXed von der CLI-Sub-Shell aus wird er nicht nochmals neu gestartet, sondern die aktuelle Sitzung wieder eingeblendet. Die Angabe eines IDENTIFIER hat in diesem Falle die gleiche Wirkung wie die Kommandofolge:

```
m  
<ENTER> IDENTIFIER <EDIT>
```

Alternativ zu dieser Form der Rückkehr in die aktuelle MAXed-Sitzung können Sie die Sub-Shell auch explizit beenden, indem Sie *eof* eingeben, z.B. mit  
CTRL-D

Sobald Sie wieder das Kommando CLI aufrufen, legt der MAXed automatisch eine neue CLI-Sub-Shell an.

Sie können den MAXed-Bildschirm auch in zwei Zeilenbereiche unterteilen: der obere Bereich zeigt die Fenster des MAXed, der untere zeigt die aktuelle CLI-Sub-Shell (siehe Abschnitt 3.5 Der Bildschirm des MAXed).

Sie reservieren die unteren  $n$  Zeilen für die CLI-Sub-Shell mit

<ENTER> n <CLI>

Beide Bereiche bleiben von jetzt an sichtbar, auch wenn Sie zwischen dem MAXed und der CLI-Sub-Shell hin- und herwechseln. Aus dem MAXed-Bereich heraus löschen Sie wieder den CLI-Bereich mit

<ENTER> <CLI>

Die CLI-Sub-Shell wird dadurch nicht beendet.

### 4.5.2 Ausführen asynchroner SINIX-Kommandos

Mit Hilfe des MAXed-Kommandos EXECUTE führen Sie ein SINIX-Kommando asynchron aus (siehe Kapitel 6: EXECUTE): Sie rufen es auf, müssen aber nicht auf seine Beendigung zu warten, sondern können wie gewohnt mit dem MAXed weiterarbeiten. Zum Zeichen, daß momentan im Hintergrund ein SINIX-Kommando ausgeführt wird, stellt der MAXed die Anzeige des aktuellen Eingabe-Modus am linken Rand der Steuerzeile invers dar (siehe Abschnitte 4.1.1 Die Eingabe-Modi, 3.5.2 Die Steuerzeile).

Sie starten ein SINIX-Kommando *SINIX-kommando*, das asynchron aber kontrolliert im Hintergrund abläuft mit

```
<ENTER> SINIX-kommando [kommando-parameter] <EXECUTE>
```

Sie wiederholen das zuletzt ausgeführte SINIX-Kommando mit  
<EXECUTE>

#### Anmerkung

Sie können EXECUTE jedes beliebige Programm, das auf der SINIX-Ebene greifbar ist, ausführen lassen. Die zugehörigen Kommando-Parameter geben Sie wie auf der Shell-Ebene an: Falls ein Parameter ein oder mehrere Leerzeichen enthält, müssen Sie ihn in Hochkommata ('pa ra') oder Anführungszeichen ("me ter") setzen.

Der MAXed bietet Ihnen jedoch noch mehr Komfort für den Aufruf von SINIX-Kommandos: Sie können sich in einer Datei einen Textbereich mit Aufrufen von einem oder mehreren SINIX-Kommandos auswählen und dem MAXed-Kommando EXECUTE als Kommandoattribut mitgeben. Dazu bieten sich z.B. die Pseudo-Dateien SYSHISTORY und SYSCOMMAND an (siehe unten: Protokollieren der SINIX-Kommandoaufrufe). Sie lassen das SINIX-Kommando in der aktuellen Zeile bzw. den angegebenen Textbereich mit SINIX-Kommandos asynchron im Hintergrund ausführen mit

```
<ENTER> <EXECUTE>  
bzw.  
<ENTER> bereich <EXECUTE>
```

**Anmerkung**

- Wenn Sie mit der Bereichsangabe mehrere SINIX-Kommandos aufrufen, führt sie der MAXed hintereinander in der Reihenfolge ihres Auftretens aus.
- Es gibt jedoch gewisse Einschränkungen gegenüber Aufrufmöglichkeiten auf der *Bourne*-Shell-Ebene: Ein einzelner SINIX-Kommandoaufruf darf sich für den Aufruf mit EXECUTE nur über eine Zeile erstrecken!

Sobald ein asynchron ausgeführtes SINIX-Kommando beendet ist, meldet Ihnen dies der MAXed in der Steuerzeile

**\*\*\*\* EXECUTE: Normales Programmende** oder

**\*\*\*\* EXECUTE: Exit-Status nicht null**

Im letzteren Fall wurde ein SINIX-Kommando mit einem Ende-Status ungleich Null beendet. Damit Sie diese Mitteilung nicht übersehen, versetzt der MAXed die Anzeige des aktuellen Eingabe-Modus danach auch noch in blinkenden Zustand.

Sie brechen ein durch EXECUTE gestartetes SINIX-Kommando ab mit  
<ENTER> <CANCEL>

Die Standard-Eingabe eines asynchron laufenden SINIX-Kommandos leitet der MAXed automatisch nach */dev/null* um, sodaß es immer das Signal *EOF* (leere Eingabe) erhält. Vor diesem Hintergrund erscheint es nicht sinnvoll, ein Programm mit EXECUTE zu starten, das von Standard-Eingabe (Dateikennzahl 0) liest, ohne diese gleichzeitig umzulenken.

Die Standard-Ausgabe und die Standard-Fehlerausgabe (Dateikennzahlen 1 und 2) leitet der MAXed automatisch in die Pseudo-Datei *SYSOUTPUT* um (siehe Abschnitt 4.2.11 Die Pseudo-Datei *SYSOUTPUT*). In dieser Datei können Sie also die Ergebnisse der aufgerufenen Programme begutachten. Sie blenden sich die Pseudo-Datei *SYSOUTPUT* ins aktuelle Fenster mit  
<ENTER> > <EDIT>

### Der Ausgabe-Modus

ist in der Standard-Konfiguration des MAXed auf APPEND und ECHO voreingestellt. Dadurch werden die Ausgaben mehrerer, nacheinander ausgeführter Programme zusammen mit dem Kommandoaufruf jeweils am Ende der Pseudo-Datei SYSOUTPUT angehängt werden (siehe Abschnitte 4.1.8 Der Ausgabe-Modus, 7.3 Eine neue Konfigurationsdatei erstellen).

### Der Kommando-Abarbeitungs-Modus

ist in der Standard-Konfiguration des MAXed auf TERMINATE voreingestellt. Dadurch wird die Ausführung mehrerer SINIX-Kommandos abgebrochen, sobald eines der Kommandos einen Ende-Status ungleich Null abliefern (siehe Abschnitt 4.1.7 Der Kommando-Abarbeitungs-Modus).

### Protokollieren der SINIX-Kommandoaufrufe:

Der MAXed sammelt alle von Ihnen erhaltenen SINIX-Kommandoaufrufe in zwei Pseudo-Dateien: In der Pseudo-Datei SYSHISTORY werden alle SINIX-Kommandoaufrufe der aktuellen Sitzung chronologisch aufgelistet (siehe Abschnitt 4.2.12 Die Pseudo-Datei SYSHISTORY). Sie blenden sich die Pseudo-Datei SYSHISTORY ins aktuelle Fenster mit  
<ENTER> < <EDIT>

In der Pseudo-Datei SYSCOMMAND werden die SINIX-Kommandoaufrufe ebenfalls eingetragen. Sie können diese Datei jedoch beliebig bearbeiten, um Kommandoaufrufe zusammenzustellen und auszutesten (siehe Abschnitt 4.2.13 Die Pseudo-Datei SYSCOMMAND). Sie blenden sich die Pseudo-Datei SYSCOMMAND ins aktuelle Fenster mit  
<ENTER> @ <EXECUTE>

**Beispiel 1**

Sie möchten eine Datei löschen, die Sie im Verlauf der Sitzung geöffnet oder neu angelegt haben. Sie blenden sich diese Datei im aktuellen Fenster ein mit

```
<ENTER> datei <EDIT>
```

Sie führen das SINIX-Kommando zum Löschen der Datei asynchron im Hintergrund aus mit

```
<ENTER> rm @f <EXECUTE>
```

Der Inhalt der aktuellen Datei verschwindet aus dem aktuellen Fenster. Die Datei wird auf der Festplatte gelöscht.

**Beispiel 2**

Sie wollen einige Dateien aus dem Dateiverzeichnis *verz1* ins Verzeichnis *verz2* kopieren. Zunächst lassen Sie sich alle Dateinamen von *verz1* in der Pseudo-Datei EDITFILES (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES) auflisten mit

```
<ENTER> verz1 <EDIT>
```

Sie wollen den *\_PICK*-Puffer verwenden und löschen ihn deshalb zu Beginn mit *<ERASE>* in der Pseudo-Datei PICKBUFFERS (siehe Abschnitt 3.8.3 Der *PICK*-Puffername als IDENTIFIER). Dann blenden Sie wieder EDITFILES ein:

```
<ENTER> * <EDIT>
```

Mit *<PICK>* kopieren Sie jeden gewünschten Dateinamen in den *\_PICK*-Puffer. Dann rufen Sie ein (!) SINIX-Kommando zum Kopieren auf:

```
<ENTER> cp %_ verz2 <EXECUTE>
```

Der Inhalt des *\_PICK*-Puffers, alle kopierten Dateinamen, wird dabei im Eingabefeld als einzeliger STREAM ausgegeben und das SINIX-Kommando *cp* mit den Dateinamen als Argumente ausgeführt (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

**Beispiel 3**

Sie lassen sich Ihre momentan laufenden Prozesse auflisten mit

<ENTER> ps -l <EXECUTE>

Sie blenden sich die Ausgabe in der Pseudo-Datei SYSOUTPUT ein mit

<ENTER> > <EDIT>

**Beispiel 4**

Sie haben gerade ein C-Quellprogramm korrigiert. Sie speichern die Datei ab mit

<SAVE>

Sie lassen das Programm asynchron im Hintergrund übersetzen mit

<ENTER> cc @f <EXECUTE>

Während der C-Compiler beschäftigt ist, arbeiten Sie an anderem weiter. Sobald die Übersetzung beendet ist, gibt der MAXed eine entsprechende Meldung in der Steuerzeile aus. In diesem Fall blenden Sie die Pseudo-Datei SYSOUTPUT ein mit

<ENTER> > <EDIT>

um die Fehlermeldungen zu begutachten.

### 4.5.3 Ausführen von Filter-Programmen

Mit Hilfe des MAXed-Kommandos FILTER sind Sie in der Lage, sogenannte Filter-Programme mit definierten Textbereichen als Eingabe aufzurufen (siehe Kapitel 6: FILTER).

#### Filter-Programme

sind Programme, die ihre Eingabedaten von Standard-Eingabe (Dateikennzahl 0) lesen, diese weiterverarbeiten und ihre Ergebnisdaten auf Standard-Ausgabe (Dateikennzahl 1) wieder ausgeben.

Sie rufen ein Filter-Programm *filter-programm*, das den angegebenen Bereich als Eingabedaten erhält und durch seine Ausgabedaten ersetzt, auf mit

```
<ENTER> [RANGE] filter-programmm [programm-parameter] <FILTER>
```

Sie wiederholen das zuletzt ausgeführte Filter-Programm, das den Rest des aktuellen Absatzes ab der aktuellen Zeile als Eingabe- und Ausgabebereich zugewiesen bekommt mit  
<FILTER>

#### Anmerkung

- Sie können FILTER jedes beliebige Filter-Programm, das auf der SINIX-Ebene greifbar ist, ausführen lassen, sofern es mit Ihren Eingabedaten zurecht kommt. Einen zugehörigen Parameter setzen Sie wie auf der Shell-Ebene in in Hochkommata (*'pa ra'*) oder Anführungszeichen (*"me ter"*), falls sie ein oder mehrere Leerzeichen enthalten.
- Als Bereichsangaben sind nur RANGE-Bereiche erlaubt. Falls Sie hierfür nichts angeben, wird ersatzweise der aktuelle Absatz angenommen (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).
- Der MAXed definiert den gesamten Textbereich, der durch die Bereichsdefinition festgelegt ist, als Bereich für die Standard-Eingabe und Standard-Ausgabe des aufgerufenen Filter-Programms. Nach Ablauf des Filter-Programms wird der angegebene Textbereich unabhängig von der aktuellen Einstellung des Eingabe-Modus durch die Standard-Ausgabe des Filter-Programms ersetzt (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

- Wenn Sie explizit verhindern wollen, daß ein "Filter"-Programm einen Textbereich für die Ein- und Ausgabe zugeteilt bekommt, so lassen Sie dem Programm *EOF* (leere Eingabe) signalisieren:

<ENTER> 0 filter-programm <FILTER>

Die Ausgabe wird in diesem Fall vor der aktuellen Zeile eingefügt.

#### 4.5.4 Ausdrucken von MAXed-Dateien

Wenn Sie in dem Text einer MAXed-Datei, die Sie ausdrucken wollen, auf MAXed-Zeichenattribute und nationale Sonderzeichen wie z.B. Umlaute verzichtet haben, können Sie die Datei ohne Probleme direkt der Standard-Druckroutine des Betriebssystems übergeben (siehe Abschnitte 4.4 Die Zeichenattribute, Kapitel 6: FONT). Sie drucken eine Datei z.B. während der MAXed-Sitzung aus mit

```
<ENTER> lpr schalter dateinamen <EXECUTE>
```

Falls Ihr Text auch Zeichenattribute oder nationale Sonderzeichen enthält, müssen Sie diese für den verwendeten Drucker erst übersetzen lassen. Ein geeignetes Filter-Programm ist das MAXed-Hilfsprogramm *maxlpr* (siehe Abschnitt 7.4 Eine MAXed-Datei für den Ausdruck aufbereiten). Es bereitet MAXed-Dateien für spezielle Druckertypen auf und ruft standardmäßig automatisch die Standard-Druckroutine *lpr* auf. Sie drucken eine solche Datei z.B. während der MAXed-Sitzung aus mit

```
<ENTER> maxlpr schalter dateinamen <EXECUTE>
```

#### Anmerkung

Berücksichtigen Sie bei diesen Kommandoaufrufen, daß die Druckroutinen stets nur jene Dateiversionen zur Kenntnis nehmen, die Sie zuvor mit SAVE auf Platte abgespeichert haben (siehe Kapitel 6: SAVE)!

Um diesen "Bürokratismus" zu umgehen, hat der MAXed zum Ausdrucken aber auch ein spezielles Kommando parat: PRINT (siehe Kapitel 6: PRINT). Damit können Sie die Dateiversion im aktuellen Fenster als Ganzes oder teilweise ausdrucken lassen.

Beim Aufruf des Kommandos `PRINT` stößt der MAXed jenen Druckauftrag an, mit dessen Kommandoaufruf Sie die Umgebungsvariable `PRINT` belegt haben. Standardmäßig ist diese Variable nicht definiert. Im Standard-Fall verwendet der MAXed die Druckroutine `lpr`. Das funktioniert ganz gut, solange Sie keine Zeichenattribute oder nationale Sonderzeichen in Ihren Texten verwendet haben. Vorausgesetzt wird natürlich, daß Ihr Betriebssystem die Druckroutine `lpr` zur Verfügung stellt. Um solchen Schwierigkeiten von vorneherein zu vermeiden, belegen Sie die Variable `PRINT` mit einem Kommandoaufruf vor, der alle Ihre druckerbetriebstechnischen Besonderheiten berücksichtigt, z.B. in der *Bourne-Shell*:

```
PRINT="maxlpr -druckertyp -pbx"  
export PRINT  
oder  
PRINT="maxlpr -pbx -dru=druckergruppe"  
export PRINT
```

Die benötigten Schalter entnehmen Sie bitte dem Abschnitt 7.4 "Eine MAXed-Datei für den Ausdruck aufbereiten" in Verbindung mit einem geübten Blick auf den angeschlossenen Drucker. Am besten definieren Sie die Variable in der benutzerspezifischen Datei `.profile`.

Sie lassen den Text, der gerade im aktuellen Fenster sichtbar ist, bzw. den angegebenen Textbereich von der durch die Umgebungsvariable `PRINT` angegebenen Druckroutine ausdrucken mit

```
<PRINT>  
bzw.  
<ENTER> [RANGE] <PRINT>
```

### Anmerkung

- MAXed-Zeichenattribute und die in der Konfigurationsdatei definierten nationale Sonderzeichen kann nur das MAXed-Hilfsprogramm `maxlpr` für die Standard-Druckroutine aufbereiten.

## 4.6 Arbeiten mit Textbausteinen

Textbausteine sind im wesentlichen PICK-Puffer, die Sie in einer oder mehreren MACRO-Bibliotheken abgespeichert haben (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER).

Während einer MAXed-Sitzung legen Sie PICK-Puffer an, indem Sie Textbereiche mit Hilfe des Kommandos PICK kopieren (siehe Kapitel 6: PICK) mit

```
<ENTER> [bereich] [IDENTIFIER] <PICK>
```

Sie verwenden PICK-Puffer, indem Sie ihren Inhalt mit Hilfe des Kommandos PUT wieder in eine Ihrer Dateien kopieren (siehe Kapitel 6: PUT) mit

```
<ENTER> [IDENTIFIER] <PUT>
```

PICK-Puffer, die Sie während einer MAXed-Sitzung anlegen, existieren höchstens bis zum Ende der Sitzung. Der MAXed speichert sie nur temporär. Damit Ihnen bestimmte PICK-Puffer für weitere MAXed-Sitzungen zur Verfügung stehen, speichern Sie sie in MACRO-Bibliotheken ab, die Sie bei einer Sitzung erneut laden können, um die darin enthaltenen Textbausteine für die aktuelle Sitzung zu aktivieren.

Prinzipiell gibt es zwei Vorgehensweisen, um Textbausteine in einer MACRO-Bibliothek abzuspeichern:

- Sie können zunächst ganz einfach alle während einer MAXed-Sitzung definierten, benannten PICK-Puffer und Makros in der Standard-MACRO-Bibliothek bzw. in der angegebenen MACRO-Bibliothek abspeichern (siehe Kapitel 6: SAVE) mit

```
<ENTER> # <SAVE>
```

bzw.

```
<ENTER> #IDENTIFIER <SAVE>
```

- Sie können aber auch eine Quelldatei erstellen und diese in eine für den MAXed lesbare MACRO-Bibliothek übersetzen.

Mit beiden Verfahrensweisen können Sie gleiche Resultate erzielen. Die Standard-MACRO-Bibliothek wird beim Start des MAXed normalerweise automatisch geladen (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen). Sie laden während einer Sitzung eine MACRO-Bibliothek mit:

```
<ENTER> #IDENTIFIER <EDIT>
```

Die darin definierten Textbausteine bzw. Makros stehen Ihnen für den weiteren Verlauf zur Verfügung, und zwar in Form von benannten PICK-Puffern bzw. benannten Makros.

Für die Erstellung und Veränderung von MACRO-Bibliotheken stellt Ihnen der MAXed verschiedene Dienstprogramme zur Verfügung (siehe Kapitel 7: MAXed-Dienstprogramme).

## 4.7 Arbeiten mit Makros

Beim Arbeiten mit dem MAXed werden bestimmte Arbeitsschritte immer wieder auf gleiche oder ähnliche Weise ablaufen und die Kommandofolgen sich höchstens durch einige Parameter (Kommandoattribute) unterscheiden. Um sich die "Handarbeit" zu ersparen, können Sie sich für den MAXed solche Befehlssequenzen einmal erstellen und in sogenannten Makros abspeichern (siehe Abschnitte 3.8.5 Der Makroname als IDENTIFIER, 4.1.11 Der Lern-Modus).

Während einer MAXed-Sitzung legen Sie Makros an, indem Sie sie in der Lernphase mit Hilfe des Kommandos LEARN erstellen (siehe Kapitel 6: LEARN) mit

```
<ENTER> IDENTIFIER [makro-parameter] <LEARN>
...
<LEARN>
```

Sie führen Makros aus, indem Sie sie mit dem Kommando MACRO aufrufen (siehe Kapitel 6: MACRO) mit

```
<ENTER> IDENTIFIER [makro-parameter] <MACRO>
```

Makros, die Sie während einer MAXed-Sitzung anlegen, existieren höchstens bis zum Ende der Sitzung. Der MAXed speichert sie nur temporär. Damit Ihnen bestimmte Makros für weitere MAXed-Sitzungen zur Verfügung stehen, speichern Sie sie in MACRO-Bibliotheken ab, die Sie bei einer Sitzung erneut laden können, um dadurch die darin enthaltenen Makros für die aktuelle Sitzung zu aktivieren.

Prinzipiell gibt es zwei Vorgehensweisen, um Makros in einer MACRO-Bibliothek abzuspeichern:

- Sie können zunächst ganz einfach alle während einer MAXed-Sitzung definierten, benannten Makros und PICK-Puffer in der Standard-MACRO-Bibliothek bzw. in der angegebenen MACRO-Bibliothek abspeichern (siehe Kapitel 6: SAVE) mit

```
<ENTER> # <SAVE>
bzw.
<ENTER> #IDENTIFIER <SAVE>
```

- Sie können aber auch eine Quelldatei erstellen und diese in eine für den MAXed lesbare MACRO-Bibliothek übersetzen.

Mit beiden Verfahrensweisen können Sie gleiche Resultate erzielen. Die Standard-MACRO-Bibliothek wird beim Start des MAXed normalerweise automatisch geladen (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen). Sie laden während einer Sitzung eine MACRO-Bibliothek mit:

```
<ENTER> #IDENTIFIER <EDIT>
```

Die darin definierten Makros bzw. Textbausteine stehen Ihnen für den weiteren Verlauf zur Verfügung, und zwar in Form von benannten Makros bzw. benannten PICK-Puffern.

Für die Erstellung und Veränderung von MACRO-Bibliotheken stellt Ihnen der MAXed verschiedene Dienstprogramme zur Verfügung (siehe Kapitel 7: MAXed-Dienstprogramme).

## 4.8 Arbeiten mit MAXed-Steuerprogrammen

MAXed-Steuerprogramme sind C-Programme, die sich während einer MAXed-Sitzung als "Filter" zwischen die Eingaben des Benutzers von der Tastatur und dem MAXed-Prozeß schieben. Durch MAXed-Steuerprogramme können Sie wie mit Makros eine Reihe von MAXed-Kommandos absetzen lassen. Jedoch steht Ihnen hierbei im Gegensatz zu Makros der volle Sprachumfang der Programmiersprache C zur Verfügung. Dies verleiht MAXed-Steuerprogrammen größere Flexibilität. Solche Programme übernehmen die volle Kontrolle über den MAXed, indem sie die Eingaben von der Tastatur in modifizierter Form an den MAXed weiterreichen bzw. von sich aus Texteingaben oder MAXed-Kommandos auslösen.

MAXed-Steuerprogrammen ist auch der direkte Zugriff auf Dateizeilen und Zeichenketten-Argumente des MAXed möglich (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld). Damit haben sie also (prinzipiell!) die gleichen Möglichkeiten, wie ein Benutzer des MAXed!

### C-Schnittstelle:

Bei der Erstellung von MAXed-Steuerprogrammen verwenden Sie Schnittstellen-Funktionen, mit deren Hilfe Sie mit dem gerade laufenden MAXed-Prozeß kommunizieren (siehe Abschnitt 7.5 MAXed-Steuerprogramme).

### Aufruf eines MAXed-Steuerprogramms:

Ein MAXed-Steuerprogramm starten Sie beim Aufruf des MAXed durch Angabe eines entsprechenden Schalters (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen) mit

**m -pr = steuerprogramm**

Oder Sie starten es während der MAXed-Sitzung mit dem Kommandos MACRO (siehe Abschnitt 3.8.6 Der Name eines Steuerprogramms als IDENTIFIER, Kapitel 6: MACRO):

<ENTER> steuerprogramm <MACRO>

Das Steuerprogramm läuft dann im Hintergrund und steuert den MAXed-Prozeß mit Hilfe von Prozeßkommunikationsmitteln wie *Pipes*. Dabei läßt es sich alle Benutzereingaben von der Tastatur geben. Der MAXed erhält nur das, was ihm das Steuerprogramm an Eingaben und Kommandos schickt!



## 5 Der Aufruf des MAXed

Im Abschnitt 5.1 finden Sie die Beschreibung des SINIX-Kommandos für den Aufruf des MAXed.

Der MAXed stellt Hilfsmittel zur Verfügung, die nach einer nicht ordnungsgemäßen Beendigung einer MAXed-Sitzung eine Rekonstruktion der verlorengegangenen aktuellen Dateiversion(en) ermöglichen. Abschnitt 5.2 skizziert den automatischen Ablauf, den Sie starten können, nachdem der MAXed beim Aufruf von sich aus eine Rekonstruktion angeboten hat.

## 5.1 maxed - Den MAXed aufrufen

Die Syntax des SINIX-Kommandos für den Aufruf des MAXed lautet:

```
maxed [-schalter] [IDENTIFIER [zeile [spalte [PATTERN]]]]
oder
m [-schalter] [IDENTIFIER [zeile [spalte [PATTERN]]]]
```

Sie rufen den MAXed von der Shell aus auf. Als Angabe für die optionalen Argumente steht Ihnen folgendes zur Auswahl:

### IDENTIFIER

für den Namen der zu bearbeitenden Datei. Wenn Sie keinen Namen angeben, wird die Status-Datei eingelesen (siehe unten).

Sie können mehrere, bereits existierende Dateien auf einmal ansprechen, wenn Sie als IDENTIFIER folgendes angeben (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER):

- einen regulären Ausdruck für Dateinamen entsprechend den SINIX-Konventionen
- mehrere, durch jeweils ein Komma (ohne Leerzeichen!) getrennte IDENTIFIER
- ein Dateiverzeichnis

Wenn genau ein existierender Dateiname zu IDENTIFIER paßt, holt der MAXed diese Datei ins aktuelle Fenster. Wenn mehrere Dateinamen passen, erstellt der MAXed eine Liste dieser Dateien und schreibt diese Liste in die Pseudo-Datei EDITFILES (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES). Diese Pseudo-Datei wird im aktuellen Fenster eingeblendet und Sie können sich aus der Liste eine der Dateien auswählen.

### zeile

für die Nummer der Zeile, auf die die Schreibmarke beim Einblenden der angegebenen Datei gesetzt wird. Das Fenster wird so positioniert, daß die Schreibmarke darin sichtbar ist.

**spalte**

für die Nummer der Spalte in der angegebenen Zeile, auf die die Schreibmarke beim Einblenden der angegebenen Datei gesetzt wird. Das Fenster wird so positioniert, daß die Schreibmarke darin sichtbar ist.

**PATTERN**

Beim Einblenden der Datei wird die Schreibmarke zunächst auf die angegebene Zeilen- und Spaltenposition gesetzt (siehe oben) und danach ein Suchkommando mit dem angegebenen PATTERN in Vorwärtsrichtung ausgeführt (siehe Kapitel 6: PLUSFIND).

**Anmerkung**

Die optionalen Argumente können Sie nur in der oben angegebenen Reihenfolge eingeben.

Sie können einen oder mehrere der folgenden Schalter angeben:

**-a[llfiles]**

Alle geöffneten Fenster mit den dazugehörigen aktuellen und alternativen Dateien der vorangegangenen MAXed-Sitzung werden zu Sitzungsbeginn automatisch eingerichtet. Pseudo-Dateien aus der früheren Sitzung können nicht mehr rekonstruiert werden.

**-f[ile] = alternativdatei**

Die angegebene Datei wird als Alternativ-Datei geladen.

**-c[onfiguration] = konfigurationsdatei**

Der MAXed wird mit der angegebenen Konfigurationsdatei gestartet (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Wenn Sie diesen Schalter nicht angeben, wird die voreingestellte Konfigurationsdatei eingelesen (siehe unten).

**-d[efinition] = makro-datei**

Der MAXed lädt beim Aufruf die angegebene MACRO-Bibliotheksdater (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen). Wenn Sie diesen Schalter nicht angeben, wird die Standard-MACRO-Bibliothek eingelesen (siehe unten).

-pr[ogram] = programm

Der MAXed startet nach dem Aufruf automatisch das angegebene MAXed-Steuerprogramm (siehe Abschnitt 7.5 MAXed-Steuerprogramme).

-pa[ge]length = n

Beim MAXed-Aufruf wird die Seitenlänge *n* eingestellt. Dadurch erhalten Sie auf dem Bildschirm eine Orientierungshilfe für die Paginierung Ihrer Dateien. Dies beeinflusst jedoch nicht den Ausdruck der Dateien! In der MAXed-Steuerzeile wird dann die Angabe *Ln* durch *Pl* infolgender Darstellungsweise ersetzt:

P1 2,3 bedeutet z.B.: Seite 2, Zeile 3

-t[ag] = proc

Beim Aufruf wird die eingestellte Tag-Datei nach einem zum PATTERN *proc* passenden *tag* durchsucht und bei Erfolg die Definitionsstelle der angesprochenen Prozedur eingeblendet bzw. die Pseudo-Datei TAGLIST erstellt (siehe Abschnitt 4.2.15). Dies entspricht der Funktionsweise des MAXed-Kommandos

<ENTER> 'proc <GOTO>

-te[xt] = sprache

Beim Aufruf wird der angegebene Sprach-Modus für den Silbentrenner sowie die Datums- und Zeitausgabe eingestellt: *En* oder *De*

-i[gnore]

Der MAXed greift beim Start nach einem System-Absturz nicht auf die bei der letzten Sitzung erstellte Protokolldatei zurück (siehe Abschnitt 5.2 Die letzte MAXed-Sitzung rekonstruieren).

-w[riteprotect]

Lädt für diese Sitzung Dateien nur zum Lesen. Dadurch werden sie vor unbeabsichtigtem Überschreiben geschützt.

### Konfigurationsdatei:

Wenn Sie beim Aufruf keine Konfigurationsdatei angeben (siehe oben: Schalter *-c*), so wird die voreingestellte Konfigurationsdatei eingelesen und der MAXed entsprechend den Angaben in dieser Datei eingestellt. Dabei wird in folgender Reihenfolge nach der Existenz einer Konfigurationsdatei gesucht (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen):

1. Die benutzer- und terminalspezifische Konfigurationsdatei *X\$USERttyf* (*tty* steht für die Nummer der Datensichtstation). Wenn Sie die Umgebungsvariable *MAXtemp* mit dem Namen eines Dateiverzeichnisses belegt haben, sucht der MAXed in diesem nach der Konfigurationsdatei. Ansonsten sucht er sie im Dateiverzeichnis */usr/lib/MAXed/tmp*.
2. Die terminalspezifische Konfigurationsdatei *mcfg.\$TERM*. Wenn Sie die Umgebungsvariable *MAXlib* mit dem Namen eines Dateiverzeichnisses belegt haben, sucht der MAXed in diesem nach der Konfigurationsdatei. Ansonsten sucht er sie im Dateiverzeichnis */usr/lib/MAXed*.
3. Die Standard-Konfigurationsdatei */usr/lib/MAXed/maxconfig.std*

### MACRO-Bibliothek:

Wenn Sie beim Aufruf keine MACRO-Bibliothek angeben (siehe oben: Schalter *-d*), so wird die benutzerspezifische Standard-MACRO-Bibliothek *X\$USER.mcd* geladen. Wenn Sie die Umgebungsvariable *MAXtemp* mit dem Namen eines Dateiverzeichnisses belegt haben, sucht der MAXed in diesem nach der Konfigurationsdatei. Ansonsten sucht er sie im Dateiverzeichnis */usr/lib/MAXed/tmp*. Alle darin definierten Makros und Textbausteine stehen Ihnen für die MAXed-Sitzung zur Verfügung (siehe Abschnitt 7.1 Eine neue MACRO-Bibliothek erstellen).

### Status-Datei:

Wenn Sie beim Aufruf keinen IDENTIFIER angeben, so liest der MAXed – wenn möglich – die Informationen aus der benutzer- und terminalspezifischen Status-Datei */usr/lib/MAXed/tmp/X\$USER\$tyv* ein. Das bedeutet u.a., daß die zuletzt bearbeitete Datei auf den Bildschirm gebracht wird. Schreibmarke und Dateiausschnitt werden so positioniert, wie sie bei der Beendigung der letzten Sitzung standen, und die am Ende der letzten Sitzung gültigen Formatgrenzen, Tabulatoren, Marken oder Voreinstellungen der Zeichenattribute sind wieder gesetzt.

**Beispiel 1**

```
m -p=60 max2.2
```

Damit wird beim Laden des MAXed die Datei *max2.2* aufgerufen. Die Seitenlänge wird auf 60 Zeilen pro Seite festgelegt. Danach können Sie sich dann orientieren, wenn der angeschlossene Drucker, auf dem Sie die Datei *max2.2* ausdrucken wollen, genau 60 Zeilen pro Seite beschreibt.

**Beispiel 2**

```
m -f=macbib.src -d=MACBIB3 max2.2
```

Damit wird beim Laden des MAXed die Datei *max2.2* aufgerufen. Als Alternativ-Datei wird die Datei *macbib.src* bereitgestellt. Sofern vorhanden wird an Stelle der Standard-MACRO-Bibliothek die Bibliotheksdatei *MACBIB3* geöffnet, deren Makro-Definitionen und Textbausteine während der MAXed-Sitzung zur Verfügung stehen.

## 5.2 Die letzte MAXed-Sitzung rekonstruieren

Während der Sitzung zeichnet der MAXed alle Änderungen an den geöffneten Dateien in einer speziellen Datei auf, der sogenannten Protokolldatei. Wird eine MAXed-Sitzung nicht ordnungsgemäß beendet, kann der MAXed die vorgenommenen Änderungen unter Umständen nicht auf der Festplatte abspeichern. Zur Rekonstruktion dieser Sitzung ruft der MAXed automatisch das Hilfsprogramm `mrecover` auf, das wiederum auf die Protokolldatei zurückgreift.

Bei einem Neustart entdeckt der MAXed eine nicht ordnungsgemäße Beendigung der letzten Sitzung normalerweise selbst. Auf dem Bildschirm erscheint folgende Meldung:

**WARNUNG:** Frühere MAXed-Sitzung nicht ordnungsgemäß abgeschlossen !  
Soll sie wiederhergestellt werden ? [j/n]:

Geben Sie *j* ein für "ja", stellt der MAXed die vergangene Sitzung automatisch wieder her. Mit Hilfe der Protokolldatei werden die Änderungen der letzten Sitzung übernommen.

Geben Sie *n* ein für "nein", wird der MAXed ganz normal gestartet. Änderungen, die Sie nach dem letzten SAVE-Kommando in der Sitzung vorgenommen hatten, gehen verloren (siehe Kapitel 6: SAVE).

Die Rekonstruktion einer MAXed-Sitzung läuft folgendermaßen ab:

Haben Sie während der MAXed-Sitzung geöffnete Dateien mit den Kommandos

<SAVE>

oder

<ENTER> <SAVE>

gesichert, so existieren für jede dieser Dateien ab diesem Zeitpunkt folgende Versionen (siehe Abschnitt 3.1 Die Dateihaltung des MAXed).

- die aktuelle Version einer Datei *datei* in einer temporären Datei
- die Sicherung dieser aktuellen Version auf der Festplatte unter dem angegebenen Dateinamen *datei*
- die frühere (*Backup*-)Version unter dem Dateinamen *datei.bak*

Sobald Sie die Kopie einer Datei *datei* im Arbeitsspeicher durch Bearbeiten mit dem MAXed verändern, die eingeblendete Dateiversion ist also noch nicht gesichert, richtet MAXed für die Datei *datei* auf Platte einen Schreibschutz ein, um diese bisher aktuellste gesicherte Version vor Veränderungen anderer Benutzer zu schützen (siehe Abschnitt 3.1 Die Dateihaltung des MAXed, Kapitel 6: SAVE). Nach der "nicht ordnungsgemäßen Beendigung" des MAXed geht die temporäre Version mit eventuell weiteren nicht gesicherten Änderungen verloren. Jedoch hat MAXed den gesamten Sitzungsverlauf in der Protokolldatei aufgezeichnet.

Beim nächsten Aufruf trifft der MAXed Vorbereitungen für eine Rekonstruktion des letzten Zustandes der temporären Dateien. Dazu kopiert er zunächst jede durch das Kommando SAVE abgesicherte, aktuellste Version auf der Festplatte namens *datei* in eine neue Datei *datei.sav*. Die früheren Versionen in den *.bak*-Dateien werden nun wieder unter ihrem alten Namen *datei* eingespielt. Das Hilfsprogramm *mrecover* erzeugt aus der Protokolldatei eine *keystroke*-Datei, welche die betroffenen Dateien unter dem alten Namen ohne das Suffix *.bak* als Ausgangsmaterial für die Rekonstruktion verwendet, bei welcher der festgehaltene Sitzungsverlauf nochmals durchgespielt wird.

### Beispiel

Sie editieren und verändern während der MAXed-Sitzung die Dateien *MAX* und *Moritz*. *Moritz* speichern Sie mit dem Kommando <SAVE> ab. Beim Bearbeiten der Datei *MAX* hängt sich Ihre Datensichtstation auf. Sie versuchen vergeblich, dem MAXed-Prozeß von einer anderen Datensichtstation aus das *Signal 1* zu schicken, damit er sich selbst beenden kann. Sie können Ihren Rechner nur noch abschalten und neu starten. Auf die Meldung

WARNUNG: Frühere MAXed-Sitzung nicht ordnungsgemaess abgeschlossen !  
Soll sie wiederhergestellt werden ? [j/n]:

geben Sie *j* ein. Daraufhin wird zunächst *Moritz* in die Datei *Moritz.sav* kopiert. Dann wird die Datei *Moritz.bak* in die Datei *Moritz* kopiert. Von der Datei *MAX* gibt es nach wie vor nur diese eine Version auf der Festplatte. Der Stand der beiden betroffenen Dateien vor der letzten MAXed-Sitzung ist jetzt wieder in den Dateien *MAX* und *Moritz* gespeichert. Der letzte Sitzungsverlauf wird auf diesen beiden Dateien entsprechend der in der Protokolldatei aufgezeichneten Änderungen nochmals durchgespielt.

Dateien vor der letzten MAXed-Sitzung:

Moritz, MAX

Dateien nach dem Öffnen in der letzten MAXed-Sitzung:

Moritz, MAX

Dateien während der letzten MAXed-Sitzung nach dem Kommando  
<SAVE>:

Moritz, Moritz.bak, MAX

Dateien nach dem Neustart:

Moritz, Moritz.bak, Moritz.sav, MAX

—

—

—

—



## ALIGN - Text an der Position der Schreibmarke ausrichten

<ALIGN>  
<ENTER> RANGE <ALIGN>

Das Kommando ALIGN richtet den definierten Textbereich so aus, daß der Text an der aktuellen Position der Schreibmarke beginnt. (siehe Abschnitt 4.3.4 Ausrichten und Vereinheitlichen). Geht der Text über den linken Formatrand hinaus, bewirkt das Kommando keine Textverschiebung (siehe Kapitel 6: MARGIN).

ALIGN ist ein textveränderndes Kommando. Der auszurichtende Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<ALIGN>

Verschiebt den Textanfang der aktuellen Zeile an die aktuelle Spaltenposition der Schreibmarke.

<ENTER> RANGE <ALIGN>

Richtet den Text des angegebenen RANGE-Bereichs an der aktuellen Spaltenposition der Schreibmarke aus.

**ATTRIBUTE - Zeichenattribute setzen**

**<ATTRIBUTE>**  
**<ENTER>** [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM]  
[setze-attribute] [, [lösche-attribute]] **<ATTRIBUTE>**

Das Kommando ATTRIBUTE dient zum nachträglichen Setzen und Zurücksetzen von Zeichenattributen im Text. Damit können Sie Text z.B. invers oder unterstrichen oder als Grafikzeichen darstellen (siehe Abschnitt 4.4.2 Hinzufügen und Entfernen von Zeichenattributen). Als Zeichenattribute stehen standardmäßig zur Verfügung:

- r[eversed] (invers)
- b[linking] (blinkend)
- u[nderscore] (unterstrichen)
- i[ntensity] (halbhell) und
- g[raphic] (Grafik)

Für einige der Kommandovarianten von ATTRIBUTE ist es erforderlich, daß die aktuelle Einstellung der Zeichenattribute aktiviert ist (siehe Abschnitt 4.4 Die Zeichenattribute, Kapitel 6: SETATTRIB):

- 0 Zeichenattribute mit dieser Einstellung werden im Text entfernt.
- 1 Zeichenattribute mit dieser Einstellung werden im Text hinzugefügt.
- \* Zeichenattribute mit dieser Einstellung werden nicht berücksichtigt.

Mit Hilfe des Kommandos ATTRIBUTE haben Sie auch die Möglichkeit, die aktuelle Einstellung der Zeichenattribute in einem Zug zu verändern, zu aktivieren und die entsprechenden Zeichenattribute dem Text hinzuzufügen.

ATTRIBUTE ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

**<ATTRIBUTE>**

Setzt die voreingestellten Zeichenattribute für die aktuelle Zeile und löscht hier bereits gesetzte Zeichenattribute, wenn die Voreinstellung für die Zeichenattribute nicht aktiviert ist (siehe Kapitel 6: SETATTRIB).

<ENTER> <ATTRIBUTE>

Setzt im Zeilenrest ab der aktuellen Schreibmarkenposition die Zeichenattribute gemäß der aktuellen Einstellung bzw. entfernt hier bereits vorhandene Zeichenattribute, wenn die aktuelle Einstellung der Zeichenattribute nicht aktiviert ist (siehe Kapitel 6: SETATTRIB).

<ENTER> *bereich* <ATTRIBUTE>

Setzt im angegebenen Bereich die Zeichenattribute gemäß der aktuellen Einstellung bzw. entfernt hier bereits vorhandene Zeichenattribute, wenn die aktuelle Einstellung der Zeichenattribute nicht aktiviert ist (siehe Kapitel 6: SETATTRIB). Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> *bereich* [*setze-attribute*] [, [*lösche-attribute*]] <ATTRIBUTE>

bestimmten Text an. Die aktuelle Einstellung der Zeichenattribute wird verändert und aktiviert: Alle links vom Komma stehenden Zeichenattribute werden auf "gesetzt" eingestellt, alle rechts vom Komma stehenden auf "nicht-gesetzt" eingestellt (siehe Kapitel 6: SETATTRIB). Danach werden im angegebenen Textbereich die Zeichenattribute gemäß der aktuellen Einstellung hinzugefügt oder entfernt. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

### Anmerkung

Texte, in denen Sie Zeichenattribute gesetzt haben, können Sie nur mit Hilfe des Dienstprogramms *maxlpr* ausdrucken (siehe Abschnitte 4.5.4 Ausdrucken von MAXed-Dateien, 7.4 MAXed-Dateien für den Ausdruck aufbereiten).

**Beispiel**

<ENTER> / reversed blinking,underscore intensity graphic <ATTRIBUTE>

In der abgekürzten Schreibweise sieht das Kommando dann so aus:

<ENTER> / rb,uiq <ATTRIBUTE>

Sofern der LEXICAL-Modus eingeschaltet ist, wird durch dieses Kommando, unabhängig von der bisherigen Voreinstellung für Zeichenattribute, der durch die aktuelle Position der Schreibmarke gekennzeichnete Satz invers und blinkend dargestellt.

## BACKSPACE - Einzelne Zeichen löschen

<BACKSPACE>

Zurückholen der letzten Kommandoattributierung:  
<ENTER> <BACKSPACE>

Das Kommando BACKSPACE verschiebt die Schreibmarke um eine Spalte nach links und löscht das Zeichen auf der neuen Position. Das wird jedoch entsprechend der Einstellung des Eingabe-Modus unterschiedlich gehandhabt (siehe Abschnitt 4.1.1 Die Eingabe-Modi, Kapitel 6: INSERTMODE):

- Ist der MAXed im Überschreibe-Modus, wird das Zeichen auf dieser neuen Position durch ein Leerzeichen ersetzt.
- Ist der MAXed im Einfüge-Modus, verschiebt sich der nachfolgende Text der Zeile um eine Spalte nach links. Das Zeichen auf der neuen Position der Schreibmarke wird also ersatzlos gelöscht bzw. durch das folgende, nach links verschobene Zeichen überschrieben.

### *Ausnahme*

Im LEXICAL-Modus verschiebt sich lediglich der nachfolgende Rest des aktuellen Wortes um eine Spalte nach links. Der übrige Zeilenrest rechts des aktuellen Wortes bleibt unverändert stehen.

<BACKSPACE>

Löscht das Zeichen links von der aktuellen Position der Schreibmarke.

<ENTER> <BACKSPACE>

Setzt die Schreibmarke ins Eingabefeld der Steuerzeile und holt die letzte Kommandoattributierung aus dem Eingabepuffer zurück (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer).

## BACKTAB - Tabulatorsprung nach links

<BACKTAB>

Kommandoverstärkung:

<ENTER> n <BACKTAB>

Start einer Bereichsdefinition für STREAM:

<ENTER> <BACKTAB>

Das Kommando BACKTAB verschiebt die Schreibmarke nach links zur nächsten Tabulatorposition. Die Tabulatorpositionen können Sie mit dem Kommando SRTAB verändern (siehe Abschnitt 4.3.1 Tabulatoren, Kapitel 6: SRTAB).

<BACKTAB>

Bewegt die Schreibmarke nach links zur nächsten Tabulatorposition.

<ENTER> <BACKTAB>

Start der Definition eines STREAM-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <BACKTAB>

Bewegt die Schreibmarke um  $n$  Tabulatorpositionen nach links.

### Anmerkung

- Das Verhalten der Schreibmarke beim Bewegen über den Rand des aktuellen Fensters ist abhängig vom Fenster-Modus (siehe Abschnitt 4.1.4 Der Fenster-Modus). Standardmäßig springt sie dabei zum gegenüberliegenden Fensterrand um und bewegt sich von dort aus weiter. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster.

- Mit <BACKTAB> können Sie die Schreibmarke auch im Eingabefeld der Steuerzeile nach links verschieben (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Im Eingabefeld sind grundsätzlich an jeder achten Spalte (1, 9, 17, ...) Tabulatoren gesetzt. Dabei ist ein Umspringen oder Wechseln des Fensters nicht möglich. Der sichtbare Ausschnitt des Eingabefeldes wird jedoch um eine Ausschnittslänge mitverschoben, wenn die Schreibmarke den rechten Rand des Eingabefeldes erreicht und das Ende der eingegebenen Zeichenkette nicht sichtbar ist.

## CANCEL - Abbruch eines Kommandos

<CANCEL>  
<ENTER> <CANCEL>

Das Kommando CANCEL bricht die Ausführung von Kommandos ab. Nach <CANCEL> kann die Datei sofort weiter bearbeitet werden. Die Schreibmarke steht wieder an der alten Position vor dem Aufruf des abgebrochenen Kommandos.

Die Kommandos werden so abgebrochen, daß auch bereits erfolgte Änderungen in der bearbeiteten Datei rückgängig gemacht werden.

Folgende Kommandos können immer abgebrochen werden:

ALIGN, CENTER, COMPRESS, EXECUTE, FILL, FORMAT, GOTO, JUSTIFY, LOWER, MACRO, MINUSFIND, PLUSFIND, REPLACE, UPPER.

Folgende Funktionen können nur dann abgebrochen werden, wenn sie auf RECTANGLE angewendet werden:

ERASE, PICK, PUT, SPACE, RESTOR.

<CANCEL>

Bricht die Ausführung eines der oben genannten MAXed-Kommandos ab (Ausnahme: EXECUTE).

<ENTER> <CANCEL>

Bricht die Ausführung von SINIX-Kommandos oder Programmen ab, die Sie asynchron im Hintergrund ausführen lassen (siehe Abschnitt 4.5.2 Ausführen asynchroner SINIX-Kommandos, Kapitel 6: EXECUTE).

## CENTER - Text zentrieren

```
<CENTER>  
<ENTER> [RANGE/RECTANGLE/einzeiliger [BLOCK/RECORD]STREAM] <CENTER>
```

Das Kommando CENTER zentriert den Text im angegebenen Textbereich zwischen linkem und rechtem Formatrand, wobei der MAXed den rechten Fensterrand als Ersatz für einen fehlenden rechten Formatrand akzeptiert (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN).

CENTER ist ein textveränderndes Kommando. Der auszurichtende Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<CENTER>

Zentriert den Text in der aktuellen Zeile.

<ENTER> bereich <CENTER>

Wendet das CENTER-Kommando auf den definierten Bereich an. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM-Argument nur einzeilig sein. Der linke und rechte Rand von STREAM oder RECTANGLE werden als temporäre Formatränder betrachtet. Zwischen diesen wird der Text zentriert.

**CHLINELEN - Zwischen verschiedenen Zeilenlängen umschalten**

<CHLINELEN>

Das Kommando CHLINELEN ist nur an bestimmten Terminals verfügbar (siehe Referenzkarte). Damit schalten Sie zwischen der Standard-Zeilenlänge und überlanger Zeilenlänge um. In der Regel entspricht dies 80 Zeichen bzw. 132 Zeichen pro Zeile. Die Zeilenlängen, die Sie mit CHLINELEN auswählen können, sind in der Konfigurationsdatei festgelegt.

<CHLINELEN>

Schaltet zwischen der Standard-Zeilenlänge und überlanger Zeilenlänge um.

## CHWINDOW - Zwischen Fenstern wechseln

```
<CHWINDOW>  
<ENTER> n <CHWINDOW>
```

Das Kommando CHWINDOW aktiviert ein anderes Fenster des MAXed-Bildschirms. (siehe Kapitel 6: HWINDOW, VWINDOW). Beim Wechsel des Fensters ohne Angabe der Fensternummer wird die Reihenfolge eingehalten, in der die Fenster definiert worden sind. Die Schreibmarke befindet sich danach innerhalb des neuen aktiven Fensters. Die anderen, nicht aktiven Fenster werden durch punktierte Begrenzungslinien markiert (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten).

```
<CHWINDOW>
```

Aktiviert das Fenster mit der nächsthöheren Fensternummer. Falls das aktuelle Fenster jenes mit der höchsten Nummer ist, wird das Fenster mit der Nummer 0 aktiviert.

```
<ENTER> n <CHWINDOW>
```

Aktiviert das Fenster *n*. Für *n* kann eine sedezimale Ziffer, also eine einstellige Zahl (0 bis 9) oder ein einzelner Buchstabe (*a* bis *f*) eingesetzt werden. Die sedezimale Ziffer wird in der rechten unteren Ecke jedes nicht aktiven Fensters angezeigt. Dadurch können also alle möglichen 16 Fenster innerhalb des Bildschirms direkt angesprochen werden.

### Beispiel

```
<ENTER> 3 <CHWINDOW>
```

Das Kommando macht das Fenster mit der Fensternummer 3 zum aktuellen Fenster, sofern momentan wenigstens vier Fenster existieren.

## CLI - Den MAXed vorübergehend verlassen

<CLI>  
<ENTER> [ n ] <CLI>

Das Kommando CLI bewirkt einen vorübergehenden Wechsel in die CLI-Sub-Shell (siehe Abschnitt 4.5.1 Die CLI-Sub-Shell). Der Bildschirm wird (teilweise) gelöscht. Das Betriebssystem wartet dann auf Ihre Eingabe. Der MAXed ist momentan in den Hintergrund geschoben und wird dadurch reaktiviert, daß Sie die Sub-Shell wieder verlassen, indem Sie den MAXed aufrufen, oder die Sub-Shell mit *eof* beenden (siehe Anmerkung).

<CLI>

Schaltet vorübergehend zum Betriebssystem um. Dabei wird zuvor der gesamte Bildschirm gelöscht.

<ENTER> n <CLI>

Bewirkt im Prinzip das gleiche wie <CLI>. Jedoch wird der Bildschirm in zwei Bereiche eingeteilt, wodurch eine gleichzeitige Anzeige von MAXed-Fenstern und Sub-Shell-Bereich möglich ist: Die unteren *n* Zeilen können benutzt werden, um Befehle des Betriebssystems auszuführen; auf dem oberen Teil des Bildschirms sind weiterhin die bearbeiteten Fenster und die Steuerzeile des MAXed zu sehen. Die Größe der einzelnen MAXed-Fenster wird vor der Positionierung der Schreibmarke auf der ersten Kommandozeile des CLI-Bereichs neu eingestellt und die Steuerung dann an das Betriebssystem übergeben. Diese Einteilung des Bildschirms bleibt erhalten, wenn Sie durch einen MAXed-Aufruf den MAXed-Bereich reaktivieren oder durch ein einfaches Kommando <CLI> später wieder in die Sub-Shell wechseln.

<ENTER> <CLI>

Stellt dem MAXed, während er gerade aktiviert ist, wieder den gesamten Bildschirm zur Verfügung, nachdem Sie ihn zuvor mit dem o.g. Kommando geteilt hatten. Diese Kommandovariante löscht lediglich den CLI-Bereich des Fensters und bewirkt keinen Übergang ins Betriebssystem.

### Anmerkung

- Ein neuerlicher Aufruf des MAXed aus der CLI-Sub-Shell bewirkt, daß der aktuelle Sitzungsstand wieder eingeblendet wird (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen). Wenn Sie beim Aufruf einen IDENTIFIER angeben, wird das Kommando EDIT mit diesem Kommandoattribut ausgeführt.
- Die CLI-Sub-Shell können Sie wie jede andere Sub-Shell mit der Eingabe *eof* beenden. Dies erreichen Sie z.B. durch CTRL-D

In der SINIX-Shell liefert auch die Taste *END* (entspricht der Taste für das MAXed-Kommando <EXIT>) diesen Code.

- Jedes MAXed-Fenster benötigt wenigstens eine zu bearbeitende Zeile. Wird der Wert für *n* zu hoch angesetzt, könnten eventuell nicht genügend Zeilen für MAXed-Fenster übrigbleiben. In diesem Fall überläßt der MAXed wie bei der Funktion <CLI> den gesamten Bildschirm ohne Argument dem Betriebssystem.

### Beispiel

```
<ENTER> 5 <CLI>
```

Das Kommando reserviert die unteren 5 Bildschirmzeilen für die Ein- und Ausgabe von (SINIX-)Kommandos bzw. Daten auf Betriebssystemebene. Die aktuelle Bildschirmseite der Maxed-Sitzung bleibt noch teilweise eingeblendet. Mit Eingabe des Kommandos *m* schalten Sie wieder zur Arbeit in die zuvor aktuelle Bildschirmseite des MAXed um. Die unteren fünf Zeilen, die für die Subshell reserviert waren, bleiben erhalten. Wenn Sie diese löschen wollen, um wieder den gesamten Bildschirm zur Arbeit mit MAXed ausnützen zu können, geben Sie folgendes Kommando ein:

```
<ENTER> <CLI>
```

## CODE - Steuerzeichen eingeben

```
<CODE> x
<CODE> <KOMMANDO>
<ENTER> <CODE> <KOMMANDO>
```

Mit Hilfe des Kommandos CODE setzen Sie einen "Code" an die aktuelle Schreibmarkenposition. Der Begriff "Code" hat hierbei zwei unterschiedliche Bedeutungen:

- ein nicht-druckbares Steuerzeichen, ein sogenanntes CNTRL-Zeichen (ASCII-Code 0 bis 31):

Zunächst wird ein "Bullet", ein grafisches Zeichen wie Bereichsmarkierungen bei einer Bereichsdefinition, an der aktuellen Schreibmarkenposition gesetzt. Wenn Sie anschließend eines der unten aufgeführten Zeichen eingeben, stellt die Kombination dieser beiden Zeichen die "Codierung" des entsprechenden CNTRL-Zeichens dar: Der MAXed speichert das "Bullet" als "Maskierung" des beliebigen Folgezeichens mit dem Dezimalwert 31 (oktal 37) ab. Mit einer solchen "Maskierung" werden jedoch nur die unten aufgeführten Zeichen als CNTRL-Zeichen anerkannt. Ein CTRL-A geben Sie demnach ein durch:

```
<CODE> A
```

- die Bezeichnung eines MAXed-Kommandos, also sein ausgeschriebener "Name", wie Sie ihn in Quelldateien von MACRO-Bibliotheken verwenden (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen):

Hierbei betätigen Sie im Anschluß die Kommandotaste(n) eines MAXed-Kommandos (siehe Referenzkarte). Dadurch wird lediglich der Kommandoname als Zeichenkette an die aktuelle Schreibmarkenposition geschrieben. Das "Bullet" verschwindet wieder. Damit ersparen Sie sich die Tipparbeit, wenn Sie z.B. Makros oder Textbausteine im Quelltext erstellen.

Das Einfügen eines "Codes" vollzieht sich wie das Einfügen einer Zeichenkette durch das Kommando PUT (siehe Kapitel 6: PUT).

&lt;CODE&gt;

Setzt ein Bullet an die aktuelle Position der Schreibmarke.

Für die erste der o.g. Codierungsformen sind im folgenden die Zeichen mit einem dezimalen ASCII-Code zwischen 64 und 95 aufgeführt, mit denen Sie ein Steuerzeichen, genannt CNTRL-Zeichen, in den Text setzen, wenn Sie eines der Zeichen unmittelbar danach eingeben:

Zeichen → erzeugtes Steuerzeichen mit ASCII-Code (dezimal)

@	→	CNTRL-@	0
A	→	CNTRL-A	1
B	→	CNTRL-B	2
...	→	CNTRL-...	...
Z	→	CNTRL-Z	26
[	→	CNTRL-[	27
\	→	CNTRL-\	28
]	→	CNTRL-]	29
^	→	CNTRL-^	30
-	→	CNTRL--	31

&lt;CODE&gt; &lt;KOMMANDO&gt;

Zweite Codierungsformen". Schreibt die Bezeichnung des MAXed-Kommandos KOMMANDO in den Text. Wenn Sie hierbei die Kommandotaste eines Tasten-Makros verwenden, wird der gesamte "Makro-Text" ausgeschrieben (siehe Abschnitt 6.2 Makro-Definitionen zur Tastenbelegung).

&lt;ENTER&gt; &lt;CODE&gt; &lt;KOMMANDO&gt;

Variante des obigen Kommandos. Gibt die Bezeichnung des MAXed-Kommandos KOMMANDO auf der Steuerzeile aus:

**CODE:** <KOMMANDO>

Damit können Sie z.B. vor Ausführen eines Tasten-Makros nochmal überprüfen, welche Kommandofolge Sie auf die entsprechende Taste bzw. Tastenkombination gelegt haben (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen).

**Beispiel 1**

<CODE> L

Damit setzen Sie die "Codierung" des Steuerzeichens CNTRL-L an die aktuelle Schreibmarkenposition, was Ihren Drucker beim Ausdrucken dieser Textstelle zu einem Seitenvorschub veranlaßt.

**Beispiel 2**

<ENTER> <CODE> <ERASEWORD>

Auf der Steuerzeile erscheint folgende Ausgabe:

CODE: <ENTER>, <ERASE>

Sie hätten die Bezeichnung dieser Kommandofolge aber auch gleich in den aktuellen Text einfügen können:

<CODE> <ERASEWORD>

## COMPARE - Textstellen vergleichen

```
<COMPARE>
<ENTER> IDENTIFIER / [RANGE/RECTANGLE/[BLOCK/RECORD] STREAM] <COMPARE>
```

Das Kommando COMPARE vergleicht den Textbereich eines PICK-Puffers mit dem Text im aktuellen Fenster ab der aktuellen Schreibmarken-Position. Mit einer der beiden folgenden Meldungen in der Steuerzeile informiert Sie der MAXed, ob die beiden Texte übereinstimmen (siehe Abschnitte 3.8.3 Der PICK-Puffername als IDENTIFIER, 3.5.2 Die Steuerzeile):

```
Text stimmt ueberein.
oder
**** Keine Textuebereinstimmung.
```

Stimmen die beiden Texte nicht überein, positioniert der MAXed die Schreibmarke an die Stelle, an der sich der nachfolgende Text das erste Mal vom Inhalt des Standard-PICK-Puffers unterscheidet.

Die Zeichenattribute vorhandenen berücksichtigt der MAXed beim Textvergleich nicht grundsätzlich (siehe Abschnitt 4.4 Die Zeichenattribute). Die Art und Weise, wie sie beim Vergleich berücksichtigt werden, hängt von ihrer aktuellen Einstellung ab:

- Ist die aktuelle Einstellung der Zeichenattribute nicht aktiviert, dann berücksichtigt der MAXed Zeichenattribute in den zu vergleichenden Textbereichen grundsätzlich nicht (siehe Kapitel 6: SETATTRIB).
- Ist die aktuelle Einstellung dagegen aktiviert, dann berücksichtigt der MAXed alle Zeichenattribute bis auf jene, die Sie auf "nicht-berücksichtigen" (\*) gesetzt haben.

### *Anmerkung*

Auf die Eigenheiten des Einfüge-Modus STREAM sei lediglich hingewiesen (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

```
<COMPARE>
Vergleicht den Inhalt des Standard-PICK-Puffers mit dem Text ab der
aktuellen Schreibmarken-Position wie oben beschrieben (siehe
3.12.2 Kommandos mit Standard-Puffern).
```

<ENTER> IDENTIFIER <COMPARE>

Vergleicht den Inhalt des mit IDENTIFIER benannten PICK-Puffers mit dem Text ab der aktuellen Schreibmarken-Position wie oben beschrieben.

<ENTER> bereich <COMPARE>

Vergleicht den durch das Bereich-Argument definierten Textauschnitt mit dem Text ab der aktuellen Position der Schreibmarke wie oben beschrieben. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

### Beispiel

Sie kopieren einen Zeilenbereich (Zeile 10 bis 25), welcher bereits an einigen Stellen mit dem Zeichenattribut *underscore* (unterstrichen) durchsetzt ist, mit Hilfe des PICK-Puffers *puf* aus der aktuellen Datei in die Datei *dat2* vor der Zeile 45 ein:

```
<ENTER> 10-25 puf <PICK>
<ENTER> dat2 <EDIT>
<ENTER> 45 <GOTO>
<ENTER> i <INSERTMODE>
<ENTER> puf <PUT>
```

Nun bereiten Sie diesen Textbereich in *dat2* mit Invertierungen grafisch auf (siehe Kapitel 6: ATTRIBUTE, SETATTRIB). Um sicher zu sein, daß Sie den Text im Vergleich zum Original ansonsten nicht verändert haben, führen Sie das Vergleichskommando aus. Damit das neu hinzugefügte Zeichenattribut nicht berücksichtigt wird, müssen Sie dies zuvor in der Voreinstellung mit "\*" kennzeichnen (ATTRIBUTE-Feld: 010\* 0000) und <SETATTRIB> aktivieren.

```
<ENTER> 45 <GOTO>
<ENTER> puf <COMPARE>
```

Die 16 Zeilen ab Zeile 45 werden mit den 16 im PICK-Puffer *puf* verglichen, wobei die Zeichenattributierung auf Übereinstimmung bei Unterstreichungen abgeprüft wird, nicht jedoch bei Invertierungen. Falls sich die Texte jedoch in anderem als den gesetzten Attributen *reversed* (invers) unterscheiden, erscheint die Meldung  
 \*\*\*\* Keine Textübereinstimmung  
 in der Steuerzeile und die Schreibmarke springt zur ersten Stelle, die nicht mit dem Pufferoriginal übereinstimmt.

## COMPRESS - Überflüssige Leerzeichen entfernen

```
<COMPRESS>
<ENTER> [RANGE/RECTANGLE/einzeiliger [BLOCK/RECORD]STREAM] <COMPRESS>
```

Durch das Kommando COMPRESS werden in einem definierten Textbereich innerhalb der Formatgrenzen überflüssige Leerzeichen entfernt. Welche Leerzeichen der MAXed für überflüssig hält, ist abhängig vom aktuellen Einfüge-Modus (siehe Abschnitte 4.3.4 Ausrichten und Vereinheitlichen 4.1.1 Die Eingabe-Modi).

COMPRESS ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

```
<COMPRESS>
```

Komprimiert den Text der aktuellen Zeile durch Löschen der überflüssigen Leerzeichen.

```
<ENTER> <COMPRESS>
```

Komprimiert den Zeilenrest der aktuellen Zeile ab der aktuellen Spalte.

```
<ENTER> bereich <COMPRESS>
```

Komprimiert den angegebenen Bereich. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM nur einzeilig sein. Der linke und rechte Rand von STREAM oder RECTANGLE werden als temporäre Formatränder betrachtet. Der Text wird zwischen diesen komprimiert.

### Beispiel

```
<ENTER> w <INSERTMODE>
<ENTER> 42 <COMPRESS>
```

Nach dem Umschalten auf den WORD-Modus werden durch Aufruf des Kommandos COMPRESS alle mehrfach auftretenden Leerzeichen in den 42 Zeilen ab der Position der Schreibmarke entfernt, die sich rechts des linken Formatrandes befinden.

## DEFDIR - Dateiverzeichnis anzeigen/wechseln

```
<DEFDIR>
<ENTER> [IDENTIFIER/STREAM] <DEFDIR>
```

Das Kommando DEFDIR gibt den absoluten Pfadnamen des aktuellen Dateiverzeichnisses in der MAXed-Steuerzeile aus oder stellt es neu ein (siehe Abschnitt 4.5 Die SINIX-Schnittstellen).

```
<DEFDIR>
```

Zeigt den absoluten Pfadnamen des aktuellen Dateiverzeichnisses in der MAXed-Steuerzeile an. Zu Beginn der MAXed-Sitzung ist dasjenige Dateiverzeichnis aktuell, von dem aus der MAXed aufgerufen wurde.

```
<ENTER> IDENTIFIER <DEFDIR>
```

Stellt das mit IDENTIFIER angegebene Dateiverzeichnis als aktuelles Dateiverzeichnis ein. Es kann ein absoluter oder ein relativer Pfadname benutzt werden (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER).

```
<ENTER> <DEFDIR>
```

Stellt das aktuelle Dateiverzeichnis neu ein. Dabei wird als Name die erste zusammenhängende Zeichenkette in der aktuellen Zeile verwendet.

```
<ENTER> STREAM <DEFDIR>
```

Stellt das aktuelle Dateiverzeichnis neu ein. Dabei wird als Name das Textstück verwendet, das im einzeiligen STREAM-Bereich angegeben ist.

### Anmerkung

Sie können das aktuelle Dateiverzeichnis nur auf den Namen eines anderen existierenden Dateinamen umstellen. Falls Sie einen falschen Namen angeben, erhalten Sie die Meldung:

```
**** Dateiliste nicht gefunden.
```

### Beispiel

```
<ENTER> MAXLIB=/usr/lib/MAXed <ENTER>
<ENTER> $MAXLIB <DEFDIR>
```

Das Kommando stellt den Wert der MAXed-Variablen MAXLIB als aktuelles Dateiverzeichnis ein (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

## DELCHAR - Ein Zeichen löschen

<DELCHAR>

Das Kommando DELCHAR löscht in jedem Eingabe-Modus das Zeichen, auf dem die Schreibmarke steht (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Der Text rechts der Position der Schreibmarke wird dabei um eine Spalte nach links verschoben.

### Ausnahme

Im LEXICAL-Modus verschiebt sich lediglich der nachfolgende Rest des aktuellen Wortes um eine Spalte nach links. Der übrige Zeilenrest rechts des aktuellen Wortes bleibt unverändert stehen.

<DELCHAR>

Löscht das Zeichen "unter" der Schreibmarke.

### Anmerkung

Mit <DELCHAR> können Sie auch Korrekturen bei der Eingabe im Eingabefeld der Steuerzeile vornehmen (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer).

**DOWNARROW - Die Schreibmarke nach unten bewegen**`<DOWNARROW>``<ENTER> n / h / v / m / c <DOWNARROW>`

Kommandoverstärkung:

`<ENTER> n <DOWNARROW>`

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

`<ENTER> <DOWNARROW>`

Das Kommando DOWNARROW bewegt die Schreibmarke eine Zeile nach unten. Im standardmäßig eingestellten Fenster-Modus springt die Schreibmarke auf den gegenüberliegenden Fensterrand um, wenn Sie sie auf den Rand des aktuellen Fensters bewegen. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Sie können den Fenster-Modus jedoch auch so einstellen, daß die Schreibmarke stattdessen den Dateiausschnitt verschiebt oder in ein angrenzendes Fenster überwechselt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

`<DOWNARROW>`

Bewegt die Schreibmarke um eine Zeile nach unten.

`<ENTER> <DOWNARROW>`

Start der Definition eines RANGE- oder RECTANGLE-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich um eine Zeile nach unten (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

`<ENTER> n <DOWNARROW>`Bewegt die Schreibmarke um *n* Zeilen nach unten.`<ENTER> modus <DOWNARROW>`

Stellt den aktuellen Fenster-Modus ein (siehe Abschnitt 4.1.4 Der Fenster-Modus). Davon betroffen sind alle ARROW- und TAB-Kommandos beim Bewegen der Schreibmarke auf bzw. über den Rand des aktuellen Fensters. Für *modus* können Sie folgendes eingeben:

n[ormal] Die Schreibmarke erhält für alle vier Richtungen die Eigenschaft, zum gegenüberliegenden Fensterrand umzuspringen. Dieser Fenster-Modus ist standardmäßig eingestellt.

h[orizontal]

Für horizontale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando oder durch Texteingabe den rechten bzw. linken Fensterrand überschreitet. Am Zeilenanfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.

v[ertical] Für vertikale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando den oberen bzw. unteren Fensterrand überschreitet. Am Dateianfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.

m[ove] Kombiniert die Einstellungen *horizontal* und *vertikal*. Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt.

c[hangewindow]

Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt. Stattdessen wechselt die Schreibmarke (falls möglich) über den Rand des aktuellen Fensters in das benachbarte Fenster, das zum aktuellen Fenster wird (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten). Beim Überschreiten des äußersten Fensterrandes springt die Schreibmarke allerdings nach wie vor um.

### Anmerkung

<DOWNARROW> besitzt im Eingabefeld der Steuerzeile eine spezielle Bedeutung: Mit <DOWNARROW> bewegen Sie die Schreibmarke an das Ende des bisher eingegebenen Textes im Eingabefeld (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer).

## DRAW - Linien und Umrahmungen zeichnen

```
<DRAW>
<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM]
         [setze-attribut] [, [lösche-attribut]] <DRAW>
<ENTER> [ a / b ] <DRAW>
```

Mit dem Kommando DRAW überschreiben Sie unabhängig vom eingestellten Eingabe-Modus in einem Textbereich eine Spalte bzw. Zeile mit einer Linie oder Sie zeichnen einen Rahmen entlang den Begrenzungslinien eines Rechtecks (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Beim Zeichnen von rechtwinklig aufeinanderstoßenden Linien setzt der MAXed automatisch passende Eckteile oder Verzweigungen ein.

Einige der Kommandovarianten von DRAW verwenden die aktuelle Einstellung der Zeichenattribute, wenn diese aktiviert ist (siehe Abschnitt 4.4 Die Zeichenattribute, Kapitel 6: SETATTRIB).

DRAW ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<DRAW>

Schaltet den Zeichnen-Modus ein (siehe Abschnitte 4.1.2 Der Zeichnen-Modus, 4.4.3 Zeichnen von Linien). Sie zeichnen ab jetzt Linien, wenn Sie anschließend die ARROW-Kommandotasten verwenden:

```
<LEFTARROW>   zeichnet eine horizontale Linie nach links weiter.
<RIGHTARROW>  zeichnet eine horizontale Linie nach rechts weiter.
<DOWNARROW>   zeichnet eine vertikale Linie nach unten weiter.
<UPARROW>     zeichnet eine vertikale Linie nach oben weiter.
```

### Anmerkung

- Durch Betätigen jeder anderen Kommandotaste beenden Sie den Zeichnen-Modus.
- Unter der Voraussetzung, daß die aktuelle Einstellung nicht aktiviert ist, werden im Zeichnen-Modus automatisch Zeichen aus dem grafischen Zeichensatz verwendet (siehe Abschnitt 4.4.1 Einstellen und Aktivieren von Zeichenattributen).

- Wenn Sie beim Zeichnen mit den ARROW-Kommandos an den Rand des aktuellen Fensters stoßen, wird das Fenster unabhängig vom aktuellen Fenster-Modus in der entsprechenden Richtung mitbewegt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

<ENTER> <DRAW>

Überschreibt unter der Voraussetzung, daß die aktuelle Einstellung nicht aktiviert ist, den Zeilenrest der aktuellen Zeile ab der aktuellen Position der Schreibmarke mit einer durchgezogenen Linie.

<ENTER> *bereich* <DRAW>

Überschreibt unter der Voraussetzung, daß die aktuelle Einstellung nicht aktiviert ist, den in dem angegebenen Bereich eine durchgezogenen Linie.

Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM nur einzeilig sein. Im Falle eines RANGE wird die aktuelle Spalte mit einer senkrechten Linie überschrieben, während bei einem STREAM eine waagrechte Linie gezogen wird. Bei einem RECTANGLE-Bereich wird ein Rechteck dieser Größe gezeichnet. Dieses Kommando eignet sich gut, um Texte einzurahmen.

<ENTER> *bereich* [*setze-attribut*] [, [*lösche-attribut*]] <DRAW>

Zeichnet in dem angegebenen Bereich eine durchgezogene Linie entsprechend der Neueinstellung der Zeichenattribute: Bei dieser Kommandovariante wird – wie beim Kommando ATTRIBUTE – die aktuelle Einstellung der Zeichenattribute zusätzlich neu eingestellt und aktiviert. Alle links vom Komma stehenden Attribute werden gesetzt, alle rechts vom Komma stehenden rückgesetzt (siehe Kapitel 6: SETATTRIB). Für *bereich* gilt gleiches wie beim obigen Kommando.

<ENTER> *linientyp* <DRAW>

Stellt den Linientyp für das Zeichnen im Zeichnen-Modus ein (siehe oben). Folgende Linientypen sind in der Standard-Konfiguration des MAXed schon voreingestellt (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen):

*a* stellt eine größere Linienstärke ein.

*b* stellt eine kleinere Linienstärke ein.

**Anmerkung**

Texte, in denen Sie Zeichenattribute gesetzt haben, können Sie nur mit Hilfe des Dienstprogramms maxlpr ausdrucken (siehe Abschnitte 4.5.4 Ausdrucken von MAXed-Dateien, 7.4 MAXed-Dateien zum Ausdruck aufbereiten).

**Beispiel**

<ENTER> ; <DRAW>

Überschreibt den Text der aktuellen Zeile bis zum Zeilenende mit einer durchgezogenen Linie.

## EDIT - Eine Datei anzeigen/erstellen

```

<EDIT>
<ENTER> [ [ = / + ] IDENTIFIER ]
          /[ IDENTIFIER [ zeile[ spalte[PATTERN]]] ]
          /[ IDENTIFIER "PATTERN" ]
          /[ @ IDENTIFIER ]
          /[ @ / * / > / < ]
          /[ - / ^ ]
          /[ #[IDENTIFIER] ] <EDIT>

```

Das Kommando EDIT bringt i.a. eine andere Datei zur Bearbeitung in das aktuelle Fenster (siehe Abschnitte 3.8.1 Der Dateiname als IDENTIFIER). Die bisher aktuelle Datei wird dabei zur alternativen Datei. Wenn Sie die neue Datei während der momentanen MAXed-Sitzung bereits im Fenster hatten, entspricht die Position von Fenster und Schreibmarke der Position zum Zeitpunkt des Ausblendens der Datei. Der aktuelle Dateiname wird in der Steuerzeile angezeigt (siehe Abschnitt 3.5 Der Bildschirm des MAXed).

<EDIT>

Bringt die Alternativ-Datei in das aktuelle Fenster. Das ist die Datei, die unmittelbar vor der jetzt aktuellen im Fenster war. Die Position von Fenster und Schreibmarke entspricht der Position zum Zeitpunkt des Ausblendens der Datei.

### *Anmerkung*

- Wenn Sie sich nacheinander verschiedene Pseudo-Dateien einblenden, hält der MAXed immer die zuletzt eingeblendete Benutzerdatei als Alternativ-Datei.
- Sie können beim Aufruf des MAXed auch gleich festlegen, welche Datei zu Beginn als Alternativ-Datei bereit gehalten werden soll (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

<ENTER> IDENTIFIER <EDIT>

Holt die durch IDENTIFIER angegebene Datei bzw. die Pseudo-Datei EDITFILES mit der Liste aller passenden Dateinamen ins aktuelle Fenster (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES).

*Anmerkung*

Sie können auch dieselbe Datei, die Sie bereits im aktuellen Fenster eingeblendet haben, nochmals einblenden lassen, z.B. mit

<ENTER> @f <EDIT>

Mit <EDIT> schalten Sie nun zwischen zwei alternativen Schreibmarkenpositionen hin und her (siehe Abschnitte 3.8.1 Der Dateiname als IDENTIFIER, 3.6.4 Textersetzung im Eingabefeld).

<ENTER> @ IDENTIFIER <EDIT>

Liest den Inhalt der durch IDENTIFIER angegebenen Datei in die Pseudo-Datei EDITFILES ein und blendet diese im aktuellen Fenster ein. Die angegebene Datei sollte eine (u.U. selbst erstellte) Liste mit Dateinamen enthalten. IDENTIFIER darf hierbei keine Sonderzeichen enthalten, die den Dateinamen abkürzen (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES).

<ENTER> = IDENTIFIER <EDIT>

Holt die durch IDENTIFIER angegebene Datei bzw. die Pseudo-Datei EDITFILES mit der Liste aller passenden Dateinamen ins aktuelle Fenster, falls es eine Datei dieses Namens bereits gibt (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES). Existiert sie nicht, wird eine entsprechende Meldung ausgegeben und keine neue Datei erstellt.

<ENTER> + IDENTIFIER <EDIT>

Holt die durch IDENTIFIER angegebene Datei bzw. die Pseudo-Datei EDITFILES mit der Liste aller passenden Dateinamen ins aktuelle Fenster, falls es eine Datei dieses Namens noch nicht gibt (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES). Existiert sie bereits, wird eine entsprechende Meldung ausgegeben und die Datei wird nicht geladen.

<ENTER> IDENTIFIER [zeile [spalte [PATTERN]]] <EDIT>

Holt die durch IDENTIFIER angegebene Datei bzw. die Pseudo-Datei EDITFILES mit der Liste aller passenden Dateinamen ins aktuelle Fenster (siehe Abschnitt 4.2.4). Durch die Angabe von Zeilen- und Spaltennummer sowie PATTERN können Sie sich gleich den gewünschten Dateiausschnitt einblenden lassen: Beim Einblenden der Datei wird die Schreibmarke zunächst auf die angegebene Zeilen- und Spaltenposition gesetzt und danach ein Suchkommando mit dem angegebenen PATTERN in Vorwärtsrichtung ausgeführt (siehe Kapitel 6: PLUSFIND).

Die Angaben sind optional, die oben genannte Reihenfolge muß eingehalten werden. In der Pseudo-Datei EDITFILES erhalten Sie ggf. die Liste aller Dateinamen, die die angegebenen "Forderungen" erfüllen.

<ENTER> IDENTIFIER "PATTERN" <EDIT>

Holt die durch IDENTIFIER angegebene Datei bzw. die Pseudo-Datei EDITFILES mit der Liste aller passenden Dateinamen ins aktuelle Fenster (siehe Abschnitt 4.2.2). Durch die Angabe von PATTERN "fordern" Sie vom MAXed, nur Dateien auszuwählen, die eine passende Textstelle zu diesem PATTERN enthalten. Sie müssen also mit einer Fehlermeldung rechnen, wenn keine passende Datei vorhanden ist.

<ENTER> <EDIT>

Holt die Pseudo-Datei RESTORE in das aktuelle Fenster (siehe Abschnitt 4.2.2). Hier werden sämtliche Textbereiche, die Sie in der aktuellen Sitzung gelöscht, verändert oder kopiert haben, für Sie zur Wiederverwendung gespeichert.

<ENTER> @ <EDIT>

Blendet die Pseudo-Datei OPENFILES mit der Liste aller in der aktuellen MAXed-Sitzung geöffneten Dateien ein (siehe Abschnitt 4.2.3 Die Pseudo-Datei OPENFILES).

<ENTER> \* <EDIT>

Blendet die Pseudo-Datei EDITFILES wieder mit der zuletzt erstellten Liste von Dateinamen ein (siehe Abschnitt 4.2.2). Die Position des Fensters und der Schreibmarke entspricht der zum Zeitpunkt des Verlassens der Datei. Wenn EDITFILES noch nicht existiert, wird sie neu angelegt und der Stern \* als IDENTIFIER verwendet, sodaß es folgendem Kommando entspricht:

<ENTER> . <EDIT>

<ENTER> > <EDIT>

Blendet die Pseudo-Datei SYSOUTPUT ein, in der die Ausgaben zuvor ausgeführter SINIX-Kommandos aufgelistet sind (siehe Abschnitte 4.2.11 Die Pseudo-Datei SYSOUTPUT, 4.5.2 Ausführen asynchroner SINIX-Kommandos).

<ENTER> < <EDIT>

Blendet die Pseudo-Datei SYSHISTORY ein, in der alle Aufrufe von SINIX-Kommandos aufgelistet sind, die Sie in der aktuellen Sitzung asynchron im Hintergrund ausführen ließen (siehe Abschnitte 4.2.12 Die Pseudo-Datei SYSHISTORY, 4.5.2 Ausführen asynchroner SINIX-Kommandos).

<ENTER> - <EDIT>

Versetzt die gerade bearbeitete Datei in den ursprünglichen Zustand vor Sitzungsbeginn. Alle Änderungen, die Sie während der Sitzung vorgenommen haben, auch solche, die Sie bereits durch das Kommando SAVE gesichert haben, werden verworfen (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

<ENTER> ^ <EDIT>

Macht alle Änderungen in der gerade bearbeiteten Datei seit dem letzten Abspeichern mit dem Kommando SAVE rückgängig. Haben Sie noch kein SAVE-Kommando eingegeben, wirkt das Kommando wie das vorstehende Kommando (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei, Kapitel 6: SAVE).

<ENTER> # <EDIT>

Interpretiert die aktuelle Textdatei als Quelldatei einer MACRO-Bibliothek, übersetzt sie mit Hilfe des Dienstprogramms mdefine zur benutzerspezifischen Standard-MACRO-Bibliothek und öffnet diese anschließend. Die bisherige Standard-MACRO-Bibliothek wird dadurch überschrieben (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen).

<ENTER> # IDENTIFIER <EDIT>

Öffnet die durch IDENTIFIER angegebene MACRO-Bibliothek und lädt die darin definierten PICK-Puffer und Makros (siehe Abschnitte 4.6 Arbeiten mit Textbausteinen, 4.7 Arbeiten mit Makros).

**Beispiel 1**

Die Eingabe

```
<ENTER> Datei 34 Haus <EDIT>
```

führt zu der Fehlermeldung

```
**** Unzulaessiges Argument.
```

Eine richtige Eingabe für den Aufruf der Datei *Datei* lautet:

```
<ENTER> Datei 34 8 Haus <EDIT>
```

Das Wort *Haus* wird ab Zeile 34 und Spalte 8 gesucht.

**Beispiel 2**

```
<ENTER> MAX* "BLOCK" <EDIT>
```

Dieses Kommando gibt eine Liste aller Dateien des aktuellen Dateiverzeichnisses aus, deren Name mit *KMAX* beginnt und deren Text das Wort *BLOCK* enthält.

**ENDLINE - Sprung an das Zeilenende**

<ENDLINE>

Mit dem Kommando ENDLINE setzen Sie die Schreibmarke an das Ende der aktuellen Zeile, falls diese beschrieben ist. Ist die aktuelle Zeile leer oder befindet sich die Schreibmarke schon am Zeilenende, bleibt sie unverändert stehen.

<ENDLINE>

Bewegt die Schreibmarke an das Ende einer beschriebenen Zeile.

**Anmerkung**

Sie können die Schreibmarke mit

<ENTER> ; <PLUSFIND>

an das nächste Zeilenende einer beschriebenen Zeile setzen. Dabei wird jedoch gleichzeitig das aktuelle PATTERN überschrieben (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).

**ENTER - Kommandoattribute eingeben**

```

<ENTER>
<ENTER> [ [kommandoattribute] kommando[fragment] ]
          /[ variable=wert ]
          /[ = beliebige zeichenketten ]
          /[ = ]
          /[ zeichenkette ? PATTERN [nachricht] ] <ENTER>
          /[ zeichenkette ! PATTERN [nachricht] ] <ENTER>
<ENTER> <BACKSPACE>

```

Das Kommando ENTER schaltet den MAXed aus der aktuellen Datei für die Eingabe von Kommandoattributen um. Dabei springt die Schreibmarke an den Anfang der MAXed-Steuerzeile (siehe Abschnitt 3.6 Prinzipielles zur Kommandoattributierung).

Korrektur im Eingabefeld:

Zur Korrektur bisher eingegebener Zeichenketten-Attribute und zum Bewegen der Schreibmarke im Eingabefeld stehen Ihnen einige Kommandos zur Verfügung (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer):

```

<BACKSPACE>, <DELCHAR>, <ARROW-KOMMANDO>, <TAB-KOMMANDO>,
<WORD-KOMMANDO>

```

Textersetzung im Eingabefeld:

Nach der Eingabe von Zeichenketten-Attributen und anschließendem Betätigen einer beliebigen Kommandotaste (siehe unten) expandiert der MAXed das Eingabefeld. Dabei ersetzt er alle vorhandenen Parameter für Textersetzung, Namen von MAXed-Variablen, PICK-Puffernamen und SINIX-Kommandoaufrufe durch ihre Werte, Inhalte bzw. Ausgaben (siehe Abschnitt 3.6.3 Sonderzeichen und Textersetzung im Eingabefeld). Danach wird das aufgerufene Kommando mit der expandierten Kommandoattributierung ausgeführt.

<ENTER>

Bewegt die Schreibmarke aus dem aktuellen Fenster in das Eingabefeld der Steuerzeile (siehe Abschnitte 3.3 Arbeiten mit der aktuellen Datei, 3.5.2 Die Steuerzeile).

Sie können im Eingabefeld Kommandoattribute eingeben (siehe Abschnitte 3.6.2 Das Eingabefeld und der Eingabepuffer, 3.6.1 Das Format der Kommandoattributierung):

- Bereichsdefinition durch Bewegen der Schreibmarke (siehe Abschnitt 3.9.2)
- Druckbare Zeichen als Zeichenketten-Attribute (siehe Abschnitte 3.8 Der IDENTIFIER als Kommandoattribut, 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut, 3.10 PATTERN als Kommandoattribut, 3.11 Ersetzungszeichenkette als Kommandoattribut).

Durch Betätigen der entsprechenden Kommandotaste <KOMMANDO> (siehe Referenzkarte) lösen Sie ein attributiertes MAXed-Kommando KOMMANDO aus.

<ENTER> <BACKSPACE>

Gibt die zuletzt im Anschluß an das einleitende <ENTER> eingegebene Kommandoattributierung aus dem Eingabepuffer in der MAXed-Steuerzeile aus. Sie können diese Attributierung korrigieren und direkt wiederverwenden für das nächste Kommando (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer, Kapitel 6: GOT0).

<ENTER> kommandoattribute <ENTER>

Ein abschließendes <ENTER> bricht i.a. die Kommandoattributierung ab (siehe unten).

<ENTER> kommando[-fragment] <ENTER>

Führt ein Kommando aus, falls der MAXed die eingegebene Zeichenkette als Bezeichnung eines MAXed-Kommandos oder zumindest als eindeutige Abkürzung einer solchen Kommando- bezeichnung erkennt (siehe Abschnitt 3.6.1 Das Format der Kommandoattributierung: Eine Alternative zum Abschicken von MAXed-Kommandos). Dieses Kommando entspricht <KOMMANDO>.

*Anmerkung*

Mit der Technik dieses und der folgenden Kommandos steht Ihnen eine Alternative zum Kommandoaufruf mittels Kommandotaste zur Verfügung.

<ENTER> kommando[-fragment] <ENTER>

Entspricht mit den führenden Leerzeichen folgendem Kommando:  
<ENTER> <KOMMANDO>

<ENTER> kommandoattribute kommando[-fragment] <ENTER>

Entspricht mit den vorausgestellten Kommandoattributen folgendem Kommando:  
<ENTER> *kommandoattribute* <KOMMANDO>

<ENTER> variable=zeichenkette <ENTER>

Weist der MAXed-Variablen *variable* eine Zeichenkette als Wert zu (siehe Abschnitte 3.6.4 Textersetzung im Eingabefeld, 4.5 Die SINIX-Schnittstellen).

<ENTER> = beliebige zeichenketten <ENTER>

Führt eine Textersetzung im Eingabefeld durch (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld: Entwertung und Auswertung auf der Steuerzeile). Die expandierten Zeichenketten werden in der Steuerzeile angezeigt. Es wird kein weiteres MAXed-Kommando ausgelöst.

<ENTER> = <ENTER>

Ruft die Pseudo-Datei VARIABLES auf, in der alle während der MAXed-Sitzung definierten Variablen aufgelistet sind (siehe Abschnitt 4.2.16).

<ENTER> zeichenkette ? PATTERN [nachricht] <ENTER>

Vergleicht PATTERN mit *zeichenkette*. Falls die Zeichenkette nicht zum PATTERN paßt, wird *nachricht* in der MAXed-Steuerzeile ausgegeben. Falls das Kommando innerhalb eines Makros verwendet wird, und der Makro-Abarbeitungs-Modus auf TERMINATE gesetzt ist, so wird die Makro-Ausführung abgebrochen, wenn die Zeichenkette nicht zum PATTERN paßt (siehe Abschnitt 4.1.6 Der Makro-Abarbeitungs-Modus). Die Zeichenkette oder das PATTERN müssen in Anführungszeichen gesetzt werden, wenn sie Leerzeichen enthalten.

<ENTER> zeichenkette ! PATTERN [nachricht] <ENTER>

Funktioniert ähnlich wie oben, nur wird ein evtl. laufendes Makro mit der Ausgabe von *nachricht* genau dann abgebrochen, wenn die Zeichenkette zu PATTERN paßt.

### Beispiel 1

<ENTER> datum @d . <REPLACE>

Das Kommando ersetzt in der gesamten Datei (angegeben durch den Punkt) das Wort *datum* durch die aktuelle Datumsangabe (angegeben durch *@d*).

### Beispiel 2

Sie haben beispielsweise folgendes Makro in einer MACRO-Bibliothek definiert:

```
<MACRO=ha5>
<ENTER>@2<PLUSFIND>
<ENTER>"Softwarearbeitsplatz"<REPLACE>
<ENTER>@1<PLUSFIND>
<ENTER>"SIEMENS AG"<REPLACE>
<ENDMACRO>
```

so wird beim Aufruf des Makros durch das Kommando

<ENTER> ha5 S\ . SAST <MACRO>

@1 mit S\ . und @2 mit SAST identifiziert. Da der Punkt ein Sonderzeichen innerhalb eines PATTERN ist, muß er durch den Gegenschrägstrich entwertet werden. Im Text wird dann S. durch SIEMENS AG und SAST durch Softwarearbeitsplatz ersetzt.

### Beispiel 3

<ENTER> pat he <ENTER>

Die Eingabe des Fragments *pat he* wird als pattern HELP identifiziert, d.h. diese Kommandoingabe bringt die Hilfsinformation zu PATTERN in das aktuelle Fenster.

## ERASE - Textbereiche löschen

```
<ERASE>
<ENTER> [ [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] [ > / >> ] [IDENTIFIER] ]
        / [ [IDENTIFIER] [ > / >> ] [IDENTIFIER] ] <ERASE>
```

Das Kommando ERASE löscht den angegebenen Textbereich und speichert ihn in einem ERASE-Puffer. Durch ein RESTOR-Kommando können Sie gelöschte Textstellen wieder in den Text kopieren (siehe Abschnitte 3.8.2 Der ERASE-Puffernamen als IDENTIFIER, 3.12.2 Kommandos mit Standard-Puffern, Kapitel 6: RESTOR).

ERASE ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<ERASE>

Löscht die aktuelle Zeile vollständig und speichert sie im Standard-ERASE-Puffer. Den Inhalt dieses Puffers können Sie mit <RESTOR> wieder in den Text kopieren.

<ENTER> IDENTIFIER <ERASE>

Löscht die aktuelle Zeile vollständig und speichert sie in den ERASE-Puffer mit dem Namen IDENTIFIER. Auf einen benannten ERASE-Puffer können Sie mit

<ENTER> IDENTIFIER <RESTOR>  
wieder zugreifen.

<ENTER> <ERASE>

Löscht den Zeilenrest der aktuellen Zeile ab der aktuellen Spalte und speichert ihn im Standard-ERASE-Puffer.

Befindet sich die Schreibmarke am Zeilenende, wird der Zeilenvorschub gelöscht, sodaß die nachfolgende Zeile an die aktuelle Zeile angehängt wird (siehe Abschnitt 4.3.5 Umbrechen und Zusammenfügen). In diesem Fall wird nichts in den Standard-ERASE-Puffer geschrieben. Die Positionen von Schreibmarke und Fenster bleiben unverändert.

<ENTER> *bereich* <ERASE>

Löscht den angegebenen Bereich und speichert ihn im Standard-ERASE-Puffer. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> *bereich* [ > ] IDENTIFIER <ERASE>

Löscht den angegebenen Bereich und speichert ihn in den mit IDENTIFIER bezeichneten ERASE-Puffer. Die schließende spitze Klammer > ist ein Hinweis auf die Kopierrichtung. Sie können sie auch weglassen. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> *bereich* > > [IDENTIFIER] <ERASE>

Löscht den angegebenen Bereich und hängt ihn ans Ende des mit IDENTIFIER bezeichneten ERASE-Puffers an. Falls IDENTIFIER weggelassen wird, wird der Standard-ERASE-Puffer als Ziel verwendet. Für *bereich* sind RANGE, STREAM und RECTANGLE möglich. Auch RECORD und BLOCK als Spezialfall von STREAM sind erlaubt.

<ENTER> [ID1] [ > ] [ID2] <ERASE>

Kopiert den Text aus dem ERASE-Puffer mit dem Namen ID1 in den ERASE-Puffer mit dem Namen ID2. Die schließende spitze Klammer > ist ein Hinweis auf die Kopierrichtung. Sie können sie auch weglassen. Falls ID1 oder ID2 weggelassen wird, wird der Standard-ERASE-Puffer als Quelle bzw. Ziel benutzt. In diesem Fall müssen Sie die schließende spitze Klammer > angeben, damit sich diese Kommandoattributierung nicht mit einer der oben beschriebenen überschneidet!.

#### *Anmerkung*

Mit diesem Kommando wird nur ein Pufferinhalt kopiert, jedoch kein Text gelöscht.

<ENTER> [ID1] > > [ID2] <ERASE>

Hängt den Text aus dem ERASE-Puffer mit dem Namen ID1 ans Ende des ERASE-Puffers mit dem Namen ID2. Falls ID1 oder ID2 weggelassen wird, so wird Standard-ERASE-Puffer als Quelle bzw. Ziel benutzt.

#### *Anmerkung*

Mit diesem Kommando wird nur ein Pufferinhalt kopiert, jedoch kein Text gelöscht.

**Anmerkung**

- Bei einigen der vorstehenden Kommandos haben Sie die Möglichkeit, das Zeichen > mit anzugeben. Bis auf einen Fall wird die Wirkung der Kommandos dadurch nicht beeinflusst. Das Zeichen ist als Hinweis auf die Kopierrichtung gedacht. Wollen Sie jedoch Text an einen Puffer anhängen, müssen Sie die Zeichen > > unbedingt mit angeben!
- Wenn Sie durch ein ERASE-Kommando nur den Inhalt eines benannten ERASE-Puffers zusätzlich in einen anderen benannten ERASE-Puffer abspeichern, so wird dieser zuvor gelöschte Text nicht nochmals in die Pseudo-Datei RESTORE aufgenommen.

**Beispiel 1**

<ENTER> , ~ , <ERASE>

Das Wort an der Position der Schreibmarke wird ohne die folgenden Leerzeichen gelöscht.

<ENTER> , <ERASE>

entspricht <ERASEWORD> und löscht mit dem aktuellen Wort auch ggf. folgende Leerzeichen (siehe Abschnitt 6.2: MACRO-Definitionen zur Tastenbelegung).

**Beispiel 2**

<ENTER> ' ~ <ERASE>

Der durch die Position der Schreibmarke gekennzeichnete BLOCK wird ab BLOCK-Anfang bis zur Position der Schreibmarke gelöscht. Die öffnende Klammer wird nicht gelöscht.

**Beispiel 3**

<ENTER> 4 id2 <ERASE>

Löscht die nächsten vier Zeilen ab der aktuellen Zeile (inklusive aktuelle Zeile) und schreibt sie in den ERASE-Puffer *id2*.

**Beispiel 4**

<ENTER> / id3 <ERASE>

Löscht den durch die aktuelle Position der Schreibmarke markierten RECORD und schreibt diesen in den ERASE-Puffer *id3*.

## EXCHANGE - Textbereiche vertauschen

`<EXCHANGE>`  
`<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] <EXCHANGE>`

Mit Hilfe des Kommandos EXCHANGE können Sie zwei Textbereiche (unter Umständen in verschiedenen Dateien) miteinander vertauschen: Dem EXCHANGE-Kommando muß immer ein ERASE-Kommando vorausgehen. Sie löschen damit einen der beiden Textbereiche. Danach kennzeichnen Sie den anderen Textbereich und führen das Kommando EXCHANGE aus. Der gekennzeichnete Textbereich wird gegen den Inhalt des Standard-ERASE-Puffers ausgetauscht und an der früheren Position des vorher gelöschten Textes wieder eingefügt.

Die beiden Textbereiche werden unabhängig vom eingestellten Eingabemodus immer eingefügt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Der Inhalt des Standard-ERASE-Puffers wird beim Austausch nicht nochmal verändert.

EXCHANGE ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

`<EXCHANGE>`

Ersetzt die aktuelle Zeile durch den Inhalt des Standard-ERASE-Puffers und schreibt die aktuelle Zeile an die Position des zuletzt gelöschten Textes.

`<ENTER> <EXCHANGE>`

Ersetzt den Zeilenrest ab der aktuellen Position der Schreibmarke durch den Inhalt des Standard-ERASE-Puffers und schreibt den Zeilenrest an die Position des zuletzt gelöschten Textes.

`<ENTER> bereich <EXCHANGE>`

Ersetzt den angegebenen Bereich durch den Inhalt des Standard-ERASE-Puffers und schreibt ihn an die frühere Position dieses gelöschten Textes. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

**Anmerkung**

Die Position, ab der die zu vertauschenden Textstellen eingesetzt werden, entspricht der ehemaligen Anfangsposition der jeweils anderen Textstelle. Eine Mischung verschiedener Bereichstypen kann somit ganz originelle Ergebnisse haben! (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut).

**Beispiel 1**

```
<ERASE>  
<EXCHANGE>
```

Zwei aufeinanderfolgende Zeilen werden ausgetauscht.

**Beispiel 2**

```
<ENTER> 15 <GOTO>  
<ERASE>  
<ENTER> 120 <GOTO>  
<ENTER> ' <EXCHANGE>
```

Der durch die Position der Schreibmarke (um die Zeile 120 herum) gekennzeichnete BLOCK (') wird einschließlich der öffnenden und schließenden Klammern durch den Inhalt des Standard-ERASE-Puffers, die gelöschte Zeile 15, ersetzt. Der gekennzeichnete BLOCK wird am Anfang der Zeile 15 eingefügt.

**Beispiel 3**

```
<ERASE>  
<ENTER> Moritz <EDIT>  
<ENTER> 15 <GOTO>  
<EXCHANGE>
```

Mit diesen Kommandos wird die aktuelle Zeile der Alternativ-Datei mit der Zeile 15 der Datei *Moritz* vertauscht.

**Beispiel 4**

```
<ENTER> MAX <PLUSFIND>  
<ENTER> A <LABEL>  
<ENTER> MORITZ <PLUSFIND>  
<ENTER> B <LABEL>  
<ENTER> A-B <EXCHANGE>
```

Mit dieser Kommandofolge wird der Bereich von der Zeile des ersten Vorkommens von *MAX* bis zu der Zeile des darauffolgenden ersten Vorkommens von *MORITZ* durch den Text im Standard-ERASE-Puffer ersetzt. Der gekennzeichnete Zeilenbereich wird an der früheren Position des gelöschten Textes aus dem Standard-ERASE-Puffer eingefügt.

## EXECUTE - Shell-Kommandos aufrufen

```
<EXECUTE>
<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD] STREAM]
          / [SINIXkommando [kommandoparameter]]
          / [ + / - / > / >> / * / ! ] <EXECUTE>
```

Das Kommando EXECUTE ermöglicht es, die Ausführung von SINIX-Kommandos, die keine Eingabe von der Standard-Eingabe benötigen, aus der aktuellen MAXed-Sitzung heraus aufzurufen und im Hintergrund ablaufen zu lassen, während Sie weiterhin Ihrer Dateiarbeit nachgehen (siehe Abschnitt 4.5.2 Ausführen asynchroner SINIX-Kommandos).

Auf Standard-Eingabe (Dateikennzahl 0) erhält das aufgerufene Programm immer das Signal für "leere Eingabe". Die Standard-Ausgabe und die Standard-Fehlerausgabe (Dateikennzahlen 1 und 2) sind auf die Pseudo-Datei SYSOUTPUT umgeleitet. Die dem Kommando EXECUTE übergebenen Kommandoaufrufe werden am Ende der Pseudo-Dateien SYS-HISTORY und SYSCOMMAND angefügt (siehe Abschnitte 4.2.12, 4.2.13).

### Achtung

- Für den Notfall: Sie brechen das aktuell asynchron laufende SINIX-Kommando bzw. die Ausführung des gesamten Pakets von SINIX-Kommandos, das Sie dem MAXed-Kommando EXECUTE übergeben haben, ab mit  
<ENTER> <CANCEL>
- Sie können den MAXed mit dem Kommando EXIT erst beenden, wenn alle EXECUTE-Operationen abgeschlossen sind.

<EXECUTE>

Führt das SINIX-Kommando aus dem Kommando-Puffer zu EXECUTE asynchron aus, das bereits bei einem vorhergehenden EXECUTE-Kommando festgelegt und im Puffer gespeichert worden ist (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen). Diese Eingabe ist nur als Wiederholungskommando geeignet.

<ENTER> SINIX-kommando [kommando-parameter] <EXECUTE>

Führt das angegebene SINIX-Kommando mit den evtl. dazugehörigen Parametern asynchron aus.

<ENTER> *bereich* <EXECUTE>

Interpretiert den durch einen Bereich festgelegten Text als ein Paket von mehreren, nacheinander abzuarbeitenden SINIX-Kommandos. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> @ <EXECUTE>

Blendet die Pseudo-Datei SYSCOMMAND im aktuellen Fenster ein, in der vorangegangene SINIX-Kommandoaufrufe und ggf. Ihre zusätzlichen Eintragungen aufgelistet sind (siehe Abschnitt 4.2.13 Die Pseudo-Datei SYSCOMMAND).

<ENTER> + <EXECUTE>

Stellt für den Ausgabe-Modus ECHO ein (siehe Abschnitt 4.1.8 Der Ausgabe-Modus). Dadurch wird vor jeder Ausgabe in der Pseudo-Datei SYSOUTPUT der zugehörige SINIX-Kommandoaufruf angefügt (siehe Abschnitt 4.2.11 Die Pseudo-Datei SYSOUTPUT).

<ENTER> - <EXECUTE>

Stellt für den Ausgabe-Modus NOECHO wieder ein (siehe oben). Dadurch werden die Kommandoaufrufe in der Pseudo-Datei SYSOUTPUT nicht angezeigt. Dies ist die Standard-Einstellung.

<ENTER> > <EXECUTE>

Stellt für den Ausgabe-Modus OVERWRITE ein (siehe Abschnitt 4.1.8 Der Ausgabe-Modus). Dadurch wird der bisherige Inhalt der Pseudo-Datei SYSOUTPUT vor jedem Aufruf eines asynchron laufenden SINIX-Kommandos gelöscht (siehe Abschnitt 4.2.11).

<ENTER> > > <EXECUTE>

Stellt für den Ausgabe-Modus wieder APPEND ein (siehe oben). Dadurch werden die einzelnen Ausgaben am Ende der Pseudo-Datei SYSOUTPUT nacheinander angehängt. Dies ist die Standard-Einstellung.

<ENTER> \* <EXECUTE>

Stellt für den Kommando-Abarbeitungs-Modus CONTINUE ein (siehe Abschnitt 4.1.7 Der Kommando-Abarbeitungs-Modus). Dadurch wird die Ausführung weiterer SINIX-Kommandos, die Sie dem MAXed-Kommando EXECUTE in einem Paket übergeben haben, auch dann fortgesetzt, wenn zwischendurch bei der Abarbeitung ein Ende-Status ungleich Null auftritt. Standardmäßig wird die Befehlsausführung abgebrochen, wenn ein Fehler während der Ausführung eines Befehls auftritt.

<ENTER> ! <EXECUTE>

Stellt für den Kommando-Abarbeitungs-Modus wieder TERMINATE ein (siehe oben). Die Ausführung mehrerer SINIX-Kommandos in einem Paket wird abgebrochen, wenn bei der Abarbeitung ein Ende-Status ungleich Null auftritt. Dies ist die Standard-Einstellung.

### Anmerkung

Ein einzelner SINIX-Kommandoaufruf, den Sie innerhalb eines Pakets von mehreren Kommandoaufrufen als Bereichsattribut dem MAXed-Kommando EXECUTE mitgeben, darf sich höchstens über eine Zeile erstrecken.

### Beispiel 1

Die Schreibmarke stehe in der ersten Zeile des folgenden Textabschnitts:

```
fgrep -n LABEL /usr/lib/nls/msg/{De,En}/maxedh/*.hlp >labli  
lpr labli
```

Die Eingabe der beiden Kommandos

<ENTER> + <EXECUTE>

<ENTER> 2 <EXECUTE>

bewirkt das Einschalten von ECHO für die nachfolgend ausgeführten SINIX-Kommandos und danach die Ausführung der durch den Zeilenbereich (RANGE) markierten Kommandos. Als Ergebnis wird die Datei *labli* im aktuellen Dateiverzeichnis angelegt. Sie enthält die Dateinamen und die Zeilennummern der MAXed-Informationsdateien, die das Wort *LABEL* enthalten. Diese Datei wird anschließend ausgedruckt. Während der Ausführung kann mit dem MAXed ganz normal weitergearbeitet werden.

**Beispiel 2**

Sie wollen mehrere Dateien aus einem anderen Dateiverzeichnis mit dem Pfadnamen */user/hans* in Ihr eigenes kopieren. Sie lassen sich das Verzeichnis auflisten mit

```
<ENTER> /user/hans <EDIT>
```

Sie löschen mit dem Kommando *<ERASE>* die Namen uninteressanter Dateien. Nun kopieren Sie den Bereich mit den Dateinamen in einen PICK-Puffer *puffer* mit

```
<ENTER> . puffer <PICK>
```

Sie brauchen jetzt nur noch folgende Kommandozeile abzuschicken:

```
<ENTER> cp %puffer $HOME <EXECUTE>
```

## EXIT - Den MAXed verlassen

<EXIT>  
<ENTER> [ r / a / c ] <EXIT>

Das Kommando EXIT beendet den MAXed. Dateien, die während der Sitzung verändert wurden, werden gesichert, indem die neuen Versionen dieser Dateien auf Platte geschrieben werden. Gleichzeitig wird eine benutzer- und terminalspezifische Status-Datei im Verzeichnis */usr/lib/MAXed/tmp* erstellt. Diese enthält für die zuletzt bearbeiteten Dateien Informationen zu den geöffneten Dateien, den Positionen von Schreibmarke und Fenster, linken und rechten Formatgrenzen, gesetzten Tabulatoren, Marken und Voreinstellungen der Zeichenattribute. Beim nächsten Aufruf von MAXed ohne Angabe eines Dateinamens wird diese Datei zum Herstellen des entsprechenden Zustandes für die MAXed-Sitzung verwendet (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

<EXIT>

Beendet den MAXed. Alle geänderten Dateien werden gesichert. Die zugehörigen Dateiversionen bei Sitzungsbeginn werden in *.bak*-Dateien gesichert.

<ENTER> <EXIT>

Beendet den MAXed. Alle geänderten Dateien werden gesichert. Dabei werden aber die zugehörigen *.bak*-Dateien gelöscht.

<ENTER> r <EXIT>

Beendet den MAXed ohne Sichern der geänderten Dateien. Alle Änderungen seit dem letzten SAVE-Kommando gehen verloren (siehe Kapitel 6: QUIT).

<ENTER> a <EXIT>

Entspricht vorstehendem Kommando. Jedoch wird die Status-Datei nicht aktualisiert, sondern so beibehalten wie vor der MAXed-Sitzung (siehe Kapitel 6: QUIT).

*Anmerkung*

Die Änderungen, die Sie während der aktuellen MAXed-Sitzung mit Hilfe des Kommandos SAVE abgespeichert haben, bleiben von den beiden vorstehenden Kommandos verschont! Eine Datei setzen Sie wieder auf den Stand vor Sitzungsbeginn mit dem Kommando:

<ENTER> - <EDIT>

(siehe Kapitel 6: EDIT, SAVE).

<ENTER> c <EXIT>

Erleichtert das Erstellen bzw. schrittweise Verändern einer Konfigurationsdatei. In der Wirkung entspricht es zunächst dem einfachen <EXIT>. Dabei wird jedoch die zuletzt im aktuellen Fenster dargestellte Datei als Quell-Datei für eine MAXed-Konfiguration betrachtet und mit Hilfe des Dienstprogramms *mconfig* automatisch als neue benutzer- und terminalspezifische Konfigurationsdatei eingerichtet (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Danach wird ohne weiteres Zutun MAXed mit den neuen Konfigurationsdaten wieder gestartet.

**Anmerkung**

- Das Kommando EXIT wird nicht ausgeführt, wenn noch Befehle durch das Kommando EXECUTE ausgeführt werden.
- Sie können in der Pseudo-Datei SETUP auch direkt einstellen, ob *.bak*-Dateien bei Beendigung der MAXed-Sitzung erstellt oder ggf. gelöscht werden sollen, indem Sie den Parameter BAKFILES mit dem entsprechende Kommandoattribut (*b* oder *r*) belegen (siehe Abschnitt 4.217 Die Pseudo-Datei SETUP).

## FILL - Im Flattersatz formatieren

```
<FILL>
<ENTER> [RANGE/RECTANGLE/einzeiliger [BLOCK/RECORD]STREAM]
        / [ [RANGE] [ De / En / + / - ] <FILL>
```

Das Kommando FILL formatiert den gekennzeichneten Textbereich im Flattersatz. Textzeilen werden aufgefüllt und der Text wird linksbündig ausgerichtet. Dabei werden gemäß dem eingestellten Eingabe-Modus überflüssige Leerzeichen zwischen Wörtern entfernt (siehe Abschnitte 4.3.3 Flatter- und Blocksatz, 4.1.1 Die Eingabe-Modi).

Der Textbereich, innerhalb dessen Sie formatieren können, wird durch den linken Formatrand begrenzt. Beim Formatieren wird der Text an beiden Formaträndern ausgerichtet, wobei der MAXed den rechten Fensterrand als Ersatz für einen fehlenden rechten Formatrand akzeptiert (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN).

Haben Sie als Eingabe-Modus den LEXICAL-Modus eingestellt, wird automatisch am rechten Formatrand getrennt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Das aktuelle PATTERN wird außerdem im Anschluß mit dem Zeichen für das Zeilenende-Zeichen, dem Bindestrich -, belegt, sodaß Sie sofort mit <PLUSFIND> oder <MINUSFIND> die Trennstellen nachkontrollieren können (siehe Abschnitt 3.10 PATTERN als Kommandoattribut, Kapitel 6: MINUSFIND, PLUSFIND).

FILL ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<FILL>

Formatiert den Rest des durch die aktuelle Position der Schreibmarke gekennzeichneten Textabsatzes ab der aktuellen Zeile im Flattersatz.

<ENTER> <FILL>

Formatiert den durch die aktuelle Position der Schreibmarke gekennzeichneten Textabsatz ab der aktuellen Zeile im Flattersatz. Im LEXICAL-Modus wird zusätzlich an der Position der Schreibmarke ein Trennvorschlag vermerkt den der MAXed bei Bedarf berücksichtigt (siehe Abschnitt 4.3.3 Flatter- und Blocksatz).

<ENTER> *bereich* <FILL>

Formatiert den angegebenen Bereich im Flattersatz. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM nur einzeilig sein. Der linke und rechte Rand von STREAM oder RECTANGLE werden als temporäre Formatränder betrachtet. Zwischen diesen wird der Text formatiert.

<ENTER> *sprache* <FILL>

Sie stellen die Datumsausgabe und den Silbentrenner des MAXed für die Trennung von Texten der angegebenen Sprache ein (siehe Abschnitte 3.6.4 Textersetzung im Eingabefeld: Parameter für Textersetzung, 4.1.10 Die Sprach-Modi: Trennmodus). Als *sprache* stellen Sie z.B. ein:

**De**            für den deutschen Silbentrenner  
**En**            für den englischen Silbentrenner

<ENTER> [RANGE] + <FILL>

Bewirkt für den angegebenen Bereich eine Parallelverschiebung des Textes nach rechts, ohne eine Formatierung auszulösen. Der Text wird um so viele Spalten nach rechts verschoben, wie die Schreibmarke rechts vom linken Formatrandes steht. Auch wenn der Text dabei über den rechten Formatrand hinausweist, wird er nicht neu umbrochen. Mit diesem Kommando können Sie einen Text unabhängig vom eingestellten Eingabe-Modus verschieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Wird der RANGE nicht explizit angegeben, erfolgt die Verschiebung des Textes ab der aktuellen Zeile bis zum Absatzende.

<ENTER> [RANGE] - <FILL>

Bewirkt für den angegebenen Bereich eine Parallelverschiebung des Textes nach links, ohne eine Formatierung auszulösen. Der Text wird um so viele Spalten nach links verschoben, wie die Schreibmarke rechts vom linken Formatrandes steht. Mit diesem Kommando können Sie einen Text unabhängig vom eingestellten Eingabe-Modus bis zur linken Formatgrenze verschieben.

Wird der RANGE nicht explizit angegeben, erfolgt die Verschiebung des Textes ab der aktuellen Zeile bis zum Absatzende.

*Achtung*

Beachten Sie jedoch, daß jene Spalten des nach links verschobenen Textes, die über den linken Formatrand hinausragen, gelöscht werden!

**Anmerkung**

Beim Formatieren kann es passieren, daß Leerzeilen entstehen, weil der ursprüngliche Text gerafft wird. Nur wenn als linker Formatrand Spalte 1 eingestellt ist, werden diese Leerzeilen automatisch gelöscht.



## FILTER - Ein FILTER-Programm aufrufen

```
<FILTER>
<ENTER> [RANGE] filterprogramm [programmparameter] <FILTER>
```

Das Kommando FILTER ermöglicht die Ausführung von Filter-Programmen, deren Standard-Eingabe (Dateikennzahl 0) wie auch die Standard-Ausgabe (Dateikennzahl 1) auf einen definierten Textbereich in Ihrer aktuellen Datei umgelenkt sind. Dadurch können Sie ein Filter-Programm veranlassen, seine Eingabedaten aus einem angegebenen Textbereich zu lesen, diese zu bearbeiten und durch die Ausgabe zu ersetzen (siehe Abschnitt 4.5.3 Ausführen von Filter-Programmen).

FILTER ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<FILTER>

Wiederholt das zuletzt eingegebene FILTER-Kommando. Jeder Aufruf von FILTER wird solange im Kommando-Puffer zu FILTER gespeichert, bis ein neuer, geänderter Aufruf den Pufferinhalt überschreibt. Als Eingabebereich für das FILTER-Kommando gilt der durch die Position der Schreibmarke gekennzeichnete Text ab der aktuellen Zeilenposition der Schreibmarke bis zum Absatzende.

### *Achtung*

In den Kommando-Puffer zu FILTER wird der zuletzt definierte RANGE-Bereich nicht mit aufgenommen! Das bedeutet, daß bei der Wiederholung eines Filter-Programms der durch die Position der Schreibmarke gekennzeichnete Textabsatz ab der Zeilenposition der Schreibmarke bis zum Absatzende als Eingabebereich definiert ist!

<ENTER> <FILTER>

Entspricht <FILTER>. Als Eingabebereich gilt hier der ganze durch die Schreibmarke markierte Textabsatz.

```
<ENTER> [RANGE] filter-programm [programm-parameter] <FILTER>
```

Führt das angegebene Filter-Programm mit den evtl. dazugehörigen Parametern aus.

Die durch die Angabe RANGE definierten Textzeilen sind die Eingabedaten für das Filter-Programm. Nach Beendigung des Filter-Programms wird der RANGE-Bereich durch die Ausgabe des Filter-Programms ersetzt. Bei fehlendem RANGE-Argument werden die Zeilen ab der aktuellen Zeile bis zum Ende des Textabsatzes als Eingabedaten genommen.

#### *Anmerkung*

Wenn Sie explizit verhindern wollen, daß ein Filter-Programm einen bestimmten Zeilenbereich Ihrer Datei als "seinen" Eingabebereich betrachtet, den es bei der Ausgabe löscht, rufen Sie es mit folgender Angabe für RANGE auf:

```
<ENTER> 0 filter-programm <FILTER>
```

Die ausgegebenen Text-Zeilen des Filter-Programms werden vor der aktuellen Zeile eingeschoben.

#### **Beispiel 1**

```
<ENTER> 0 mail <FILTER>
```

Das Kommando schiebt die Post-Mitteilung vor den Text der aktuellen Zeile.

#### **Beispiel 2**

Steht die Schreibmarke in der ersten Zeile des folgenden Textabschnitts, der durch eine Leerzeile abgeschlossen ist:

```
Weigand, Max 7000 Stuttgart, Wilhelm-Busch-Str. 13
Artmann, Wilhelmine 8000 München, Hansastr. 6
Gefrörer, Moritz 6800 Mannheim, Sast-Allee 1
König, Carola 1000 Berlin, Wattstr. 41
Hitchcock, Alfred J. 7570 Baden-Baden, Jägersteig 11
Lorenz, Konrad G. 6600 Saarbrücken, Grauganspfad 7
```

so wird dieser nach Eingabe des Kommandos

```
<ENTER> sort <FILTER>
```

durch den nachstehenden Absatz ersetzt:

```
Artmann, Wilhelmine 8000 München, Hansastr. 6
Gefrörer, Moritz 6800 Mannheim, Sast-Allee 1
Hitchcock, Alfred J. 7570 Baden-Baden, Jägersteig 11
König, Carola 1000 Berlin, Wattstr. 41
Lorenz, Konrad G. 6600 Saarbrücken, Grauganspfad 7
Weigand, Max 7000 Stuttgart, Wilhelm-Busch-Str. 13
```

**FONT - Den Zeichensatz umschalten**

<FONT>

Das Kommando FONT wird an *7-Bit*-Terminals für die Eingabe nationaler Sonderzeichen zur Verfügung gestellt. Bei *8-Bit*-Terminals stellt das Kommando eine Alternative zur üblichen Eingabe von Zeichen aus der "oberen Hälfte" des *ISO-8859/1*-Zeichensatzes dar. Mit FONT wird hier ggf. die Eingabe vereinfacht.

Das Kommando FONT schaltet für das anschließend eingegebene Zeichen den alternativen Zeichensatz ein. Für die nächsten Zeichen, die Sie eingeben, gilt wieder der Standard-Zeichensatz. Der alternative Zeichensatz ist in der Konfigurationsdatei eingetragen und wird bei Sitzungsbeginn zur Verfügung gestellt (siehe Abschnitt 7.2 Eine neue Konfigurationsdatei erstellen).

<FONT>

Schaltet auf den alternativen Zeichensatz um. Dieser ist allerdings nur für das unmittelbar nach der <FONT>-Taste eingegebene Zeichen wirksam.

In der Standard-Konfiguration sind u.a. folgende Zeichentasten belegt:

a	mit	ä,	A	mit	Ä
o	mit	ö,	O	mit	Ö
u	mit	ü,	U	mit	Ü
s	mit	ß,			
S	mit	§,			

**Anmerkung**

- Es ist durchaus auch eine andere Tastenbelegung denkbar. Sie finden in der *On-Line*-Dokumentation eine vollständige Liste derjenigen Zeichen aus dem *ISO-8859/1*-Zeichensatz, die in der Standard-Konfiguration auf bestimmte Tasten gelegt sind (siehe Kapitel 6: HELP). Dazu geben Sie folgendes Kommando:

```
<HELP> <FONT>
```

- Texte mit Zeichen aus dem alternativen Zeichensatz werden mit Hilfe des Dienstprogramms *maxlpr* auf den unterstützten Druckern ausgegeben (siehe Abschnitt 7.4 Eine MAXed-Datei für den Ausdruck aufbereiten).

**Beispiel**

```
h<FONT>a<FONT>slich
```

bewirkt die Abbildung von häßlich auf dem Bildschirm.

## FORMAT - Im Blocksatz formatieren

<FORMAT>  
 <ENTER> [RANGE/RECTANGLE/einzeiliger [BLOCK/RECORD]STREAM]  
 / [ [RANGE] [ De / En / + / - ] <FORMAT>

Das Kommando **FORMAT** formatiert den gekennzeichneten Textbereich im Blocksatz. Textzeilen werden aufgefüllt, und der Text wird links- und rechtsbündig ausgerichtet und gemäß dem eingestellten Eingabe-Modus von überflüssige Leerzeichen zwischen den Wörtern befreit (siehe Abschnitte 4.3.3 Flatter- und Blocksatz, 4.1.1 Die Eingabe-Modi). Dazu wird die erforderliche Anzahl an Leerzeichen eingefügt.

Der Textbereich, innerhalb dessen Sie formatieren können, wird durch den linken Formatrand begrenzt. Beim Formatieren wird der Text an beiden Formaträndern ausgerichtet, wobei der MAXed den rechten Fensterrand als Ersatz für einen fehlenden rechten Formatrand akzeptiert (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN).

Haben Sie als Eingabe-Modus den LEXICAL-Modus eingestellt, wird automatisch am rechten Formatrand getrennt (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Das aktuelle PATTERN wird außerdem im Anschluß mit dem Zeichen für das Zeilenende-Zeichen, dem Bindestrich -, belegt, sodaß Sie sofort mit <PLUSFIND> oder <MINUSFIND> die Trennstellen nachkontrollieren können (siehe Abschnitt 3.10 PATTERN als Kommandoattribut, Kapitel 6: MINUSFIND, PLUSFIND).

**FORMAT** ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei **RESTORE** gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos **UNDO** können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<FORMAT>

Formatiert den Rest des durch die aktuelle Position der Schreibmarke gekennzeichneten Textabsatzes ab der aktuellen Zeile im Blocksatz.

<ENTER> <FORMAT>

Formatiert den durch die aktuelle Position der Schreibmarke gekennzeichneten Textabsatz ab der aktuellen Zeile im Blocksatz. Im LEXICAL-Modus wird zusätzlich an der Position der Schreibmarke ein Trennvorschlag vermerkt, den der MAXed bei Bedarf berücksichtigt (siehe Abschnitt 4.3.3 Flatter- und Blocksatz).

<ENTER> *bereich* <FORMAT>

Formatiert den angegebenen Bereich im Blocksatz. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM nur einzeilig sein. Der linke und rechte Rand von STREAM oder RECTANGLE werden als temporäre Formatränder betrachtet. Zwischen diesen wird der Text formatiert.

<ENTER> *sprache* <FORMAT>

Sie stellen die Datumsausgabe und den Silbentrenner des MAXed für die Trennung von Texten der angegebenen Sprache ein (siehe Abschnitte 3.6.4 Textersetzung im Eingabefeld: Parameter für Textersetzung, 4.1.10 Die Sprach-Modi: Sprach-Modus für den Silbentrenner). Als *sprache* stellen Sie z.B. ein:

**De** für den deutschen Silbentrenner  
**En** für den englischen Silbentrenner

<ENTER> [RANGE] + <FORMAT>

Bewirkt für den angegebenen Bereich eine Parallelverschiebung des Textes nach rechts, ohne eine Formatierung auszulösen. Der Text wird um so viele Spalten nach rechts verschoben, wie die Schreibmarke rechts vom linken Formatrandes steht. Auch wenn der Text dabei über den rechten Formatrand hinausweist, wird er nicht neu umbrochen. Mit diesem Kommando können Sie einen Text unabhängig vom eingestellten Eingabe-Modus verschieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Wird der RANGE nicht explizit angegeben, erfolgt die Verschiebung des Textes ab der aktuellen Zeile bis zum Absatzende.

<ENTER> [RANGE] - <FORMAT>

Bewirkt für den angegebenen Bereich eine Parallelverschiebung des Textes nach links, ohne eine Formatierung auszulösen. Der Text wird um so viele Spalten nach links verschoben, wie die Schreibmarke rechts vom linken Formatrandes steht. Mit diesem Kommando können Sie einen Text unabhängig vom eingestellten Eingabe-Modus bis zur linken Formatgrenze verschieben.

Wird der RANGE nicht explizit angegeben, erfolgt die Verschiebung des Textes ab der aktuellen Zeile bis zum Absatzende.

*Achtung*

Beachten Sie jedoch, daß jene Spalten des nach links verschobenen Textes, die über den linken Formatrand hinausragen, gelöscht werden!

**Anmerkung**

Beim Formatieren kann es passieren, daß Leerzeilen entstehen, weil der ursprüngliche Text gerafft wird. Nur wenn als linker Formatrand Spalte 1 eingestellt ist, werden diese Leerzeilen automatisch gelöscht.

## GOTO - Positionieren auf ausgewählten Textstellen

```

<GOTO>
<ENTER> [ IDENTIFIER ]
          / [ [zeile] [,spalte] ]
          / [ "PATTERN" [IDENTIFIER] ]
          / [ ' [proc] [IDENTIFIER] ]
          / [ RANGE/RECTANGLE/[BLOCK/RECORD]STREAM ]
          / [ . / @ / " ] <GOTO>

```

Das Kommando GOTO setzt die Schreibmarke an die angegebene Position im Text. Liegt das Sprungziel außerhalb des aktuellen Fensters wird dieses dorthin verschoben. Mit GOTO springen Sie absolute Positionen an, die Sie z.B. durch Zeilen- und Spaltennummer oder durch Namen gesetzter Marken angeben (siehe Abschnitt 3.8.4 Der Markenname als IDENTIFIER, Kapitel 6: LABEL).

<GOTO>  
Sprung an den Dateianfang. Die Schreibmarke bleibt in der gleichen Spalte.

<ENTER> . <GOTO>  
Sprung an das Dateieinde. Die Schreibmarke bleibt in der gleichen Spalte.

<ENTER> [zeile] [,spalte] <GOTO>  
Sprung zu der Dateiposition mit der angegebenen Zeilen- und Spaltennummer. Geben Sie nur die Zeilennummer an, bleibt die Schreibmarke in der aktuellen Spalte.

<ENTER> <GOTO>  
Sprung an die Position der Standard-Marke.

<ENTER> IDENTIFIER <GOTO>  
Setzt die Schreibmarke an die Position der mit IDENTIFIER benannten Marke bzw. blendet die Pseudo-Datei LABELS mit der Liste aller passenden Markennamen im aktuelle Fenster ein (siehe Abschnitt 4.2.8 Die Pseudo-Datei LABELS).

<ENTER> @ <GOTO>  
Blendet die Pseudo-Datei LABELS mit der Liste aller momentan in der aktuellen Datei gesetzten Marken ein (siehe Abschnitt 4.2.8 Die Pseudo-Datei LABELS).

<ENTER> "PATTERN" [IDENTIFIER] <GOTO>

Startet die Funktion FILE-SCAN. Danach wird die Pseudo-Datei SCANLIST mit der Liste aller zum angegebenen PATTERN passenden Textstellen aus allen Dateien, deren Namen zu IDENTIFIER passen, eingeblendet (siehe Abschnitte 4.2.14 Die Pseudo-Datei SCANLIST, 3.10 PATTERN als Kommandoattribut).

Bei Eingabe der leeren Zeichenkette "" für PATTERN wird ersatzweise auf das zuletzt beim Aufruf dieses Kommandos verwendete PATTERN zurückgegriffen (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen). Bei fehlender Angabe für IDENTIFIER wird die aktuelle Datei durchsucht.

#### *Anmerkung*

Das aktuelle PATTERN wird erst dann mit dieser PATTERN-Angabe belegt, wenn Sie aus der Pseudo-Datei SCANLIST heraus mit <GOTO> auf eine der gefundenen Textstellen aus den angegebenen Dateien springen.

<ENTER> " <GOTO>

Blendet die Pseudo-Datei SCANLIST wieder mit der zuletzt erstellten Liste von gefundenen Textstellen ein (siehe Abschnitt 4.2.14 Die Pseudo-Datei SCANLIST). Die Position des Fensters und der Schreibmarke entspricht der zum Zeitpunkt des Verlassens der Datei.

<ENTER> bereich <GOTO>

Sprung an das andere Ende des angegebenen Bereichs. Für *bereich* sind RANGE, RECTANGLE und mehrzeiliger STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

#### *Anmerkung*

- Dieses Kommando ist dann sinnvoll, wenn der angegebene Bereich zuvor durch Bewegen der Schreibmarke definiert wurde und noch im Eingabepuffer gespeichert ist, sodaß Sie ihn wieder ins Eingabefeld zurückholen können (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Die tatsächliche Kommandofolge lautet demnach:

<ENTER> <BACKSPACE> <GOTO>

- Handelt es sich um einen einzeiligen STREAM, so verwendet der MAXed die definierte Textstelle als Suchmuster und blendet die die Pseudo-Datei SCANLIST mit der Liste aller damit identischen Textstellen aus der aktuellen Datei ein.

<ENTER> '[proc] [IDENTIFIER] <GOTO>

Blendet die Datei mit der Definitionsstelle der Prozedur ein, die Sie durch *proc* angeben bzw. die Pseudo-Datei TAGLIST mit der Liste aller zum PATTERN *proc* passenden Prozedurnamen ein (siehe Abschnitte 4.2.15 Die Pseudo-Datei TAGLIST, 3.10 PATTERN als Kommandoattribut). Wenn Sie nur das Hochkomma ', jedoch nichts für *proc* eingeben, wird ersatzweise das aktuelle Wort als *tag* verwendet.

Wenn Sie IDENTIFIER angeben, wird automatisch eine neue *tag*-Datei für alle Dateien erstellt, deren Namen zu IDENTIFIER passen, und diese *tag*-Datei für die oben beschriebene Ausführung verwendet.

#### Anmerkung

Standardmäßig verwendet der MAXed eine Datei namens *tags* bzw. die durch die Umgebungsvariable MAXtags festgelegte *tag*-Datei (siehe Abschnitt 4.2.15 Die Pseudo-Dateien: TAGLIST, Anhang: Erstellen einer Tag-Datei).

#### Beispiel

Das vorausgehende Kommando lautete:

<ENTER> <PLUSPAGE> <UPEND> <FORMAT>

Dadurch hatten Sie ein RECTANGLE im Blocksatz formatiert. Die Schreibmarke befindet sich nun wieder am Anfang des formatierten Bereichs. Sie wollen aber am anderen Ende weiterarbeiten. Sie gelangen dorthin mit

<ENTER> <BACKSPACE> <GOTO>

## HELP - Hilfsinformationen abrufen

<HELP>  
<ENTER> begriff / @ <HELP>  
<ENTER> <HELP> <KOMMANDO>

Mit dem Kommandos HELP blenden Sie sich die *On-Line*-Dokumentation des MAXed in der Pseudo-Datei HELP ein (siehe Abschnitt 4.2.1). In den eingeblendeten Hilfstexten sind die Begriffe, für die Sie weitere Hilfsinformationen aufrufen können, in Großbuchstaben geschrieben. Sie können sich einen der Begriffe durch Positionieren der Schreibmarke auswählen. Durch nochmaliges Betätigen von <HELP> wird dann die entsprechende Information auf den Bildschirm gebracht.

<HELP>  
Zeigt die Übersicht aller zur Verfügung stehenden Hilfsinformationen.

<ENTER> begriff <HELP>  
Blendet zu dem genannten Begriff die Hilfsinformation ein. Alle Begriffe, zu denen es Hilfstexte gibt, sind in der Übersichtsdatei (siehe oben) aufgelistet. Dazu zählen z.B. alle MAXed-Kommandonamen und Bereichstypen. Sie können *begriff* abkürzen.

<ENTER> <HELP> <KOMMANDO>  
Blendet die Hilfsinformation zum MAXed-Kommando KOMMANDO ein.

<ENTER> @ <HELP>  
Blendet eine kommentierte Liste aller verfügbaren Hilfsinformationen ein.

## HOME - Schnelle Bewegungen der Schreibmarke

```
<HOME>
<ENTER> [ <RIGHTARROW>/<LEFTARROW>/<UPARROW>/<DOWNARROW> ]
          / [ [zeile] [,spalte] ]
          / [ vbewegung hbewegung / hbewegung vbewegung ] <HOME>
```

Das Kommando dient zum schnellen Bewegen der Schreibmarke an die Ränder des aktuellen Fensters und zum diagonalen Verschieben des Dateiausschnitts (siehe Abschnitte 3.3.2 Bewegen der Schreibmarke, 3.3.3 Verschieben des Dateiausschnitts).

<HOME>

Bewegt die Schreibmarke zur linken oberen Ecke des Fensters. Das Fenster wird dabei nicht verschoben.

<ENTER> <HOME>

Verschiebt das aktuelle Fenster so, daß die aktuelle Position der Schreibmarke in der linken oberen Ecke des Fensters steht.

<ENTER> [zeile] [,spalte] <HOME>

Bewegt das aktuelle Fenster so, daß die durch *zeile* und *spalte* festgelegte Position in der linken oberen Ecke des Fensters steht.

<ENTER> <RIGHTARROW> <HOME>

Bewegt die Schreibmarke zum rechten Rand des aktuellen Fensters.

<ENTER> <LEFTARROW> <HOME>

Bewegt die Schreibmarke zum linken Rand des aktuellen Fensters.

<ENTER> <UPARROW> <HOME>

Bewegt die Schreibmarke in die oberste Zeile des aktuellen Fensters.

<ENTER> <DOWNARROW> <HOME>

Bewegt die Schreibmarke in die unterste Zeile des aktuellen Fensters.

### *Anmerkung*

Bei den vorstehenden vier Kommandovarianten springt die Schreibmarke an den Fensterrand in der aktuellen Zeile bzw. Spalte.

<ENTER> v-bewegung h-bewegung <HOME>

<ENTER> h-bewegung v-bewegung <HOME>

Bewegt die Schreibmarke entsprechend der Bewegungskombination zu einer Ecke des aktuellen Fensters. *h-bewegung1* und *v-bewegung2* bezeichnen eine vertikale und eine horizontale Bewegung der Schreibmarke mit ARROW-, TAB- oder WORD-Kommandos (siehe Abschnitt 3.3.2 Bewegen der Schreibmarke).

**Beispiel**

Die Schreibmarke bewegt sich nach dem Kommando

<HOME> <UPARROW> in die linke untere Ecke

<ENTER> <DOWNARROW> <RIGHTARROW> <HOME> in die rechte untere Ecke

<HOME> <UPARROW> <LEFTARROW> in die rechte untere Ecke

<HOME> <LEFTARROW> in die rechte obere Ecke.

Bei obigen Kommandos wird zum Teil die Umspring-Eigenschaft am Fensterrand ausgenutzt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

## HWINDOW - Den Bildschirm horizontal teilen

```
<HWINDOW>
<ENTER> [ [ = / + ] IDENTIFIER ]
          /[ IDENTIFIER [ zeile[ spalte[PATTERN]]] ]
          /[ IDENTIFIER "PATTERN" ]
          /[ @ IDENTIFIER ]
          /[ @ / * / > / < ] <HWINDOW>
```

Das Kommando HWINDOW erzeugt ein neues Fenster durch horizontale Teilung des aktuellen Fensters. Der untere Teil davon ist nun das aktuelle Fenster. Dieses wird, wenn nicht anders angegeben, an den Anfang der neuen aktuellen Datei gesetzt. Das verkleinerte alte Fenster bleibt in der zuvor aktuellen Datei an der ursprünglichen Position. Sie können das Kommando HWINDOW in einigen Fällen mit denselben Kommandoattributen versehen wie das Kommando EDIT (siehe Kapitel 6: EDIT). Mit Hilfe des Kommandos CHWINDOW wechseln Sie zwischen den Fenstern (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten).

<HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Dabei wird die zuvor aktuelle Datei nochmals im (neuen) aktuellen Fenster eingeblendet.

<ENTER> <HWINDOW>

Entfernt das zuletzt mit einem HWINDOW- oder VWINDOW-Kommando eingerichtete Fenster, das ist jenes mit der höchsten Fensternummer. Das Fenster, von dem das gelöschte Fenster abgeteilt worden war, erhält wieder seine ursprüngliche Größe und wird zum aktuellen Fenster.

<ENTER> IDENTIFIER <HWINDOW>

Erzeugt ein neues Fenster durch horizontale Teilung des aktuellen Fensters an der aktuellen Zeile. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> IDENTIFIER <EDIT>  
ausgeführt.

<ENTER> = IDENTIFIER <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> = IDENTIFIER <EDIT> ausgeführt.

<ENTER> + IDENTIFIER <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> + IDENTIFIER <EDIT> ausgeführt.

<ENTER> IDENTIFIER [zeile [spalte [PATTERN]]] <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> IDENTIFIER [zeile [spalte [PATTERN]]] <EDIT> ausgeführt.

<ENTER> IDENTIFIER "PATTERN" <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> IDENTIFIER "PATTERN" <EDIT> ausgeführt.

<ENTER> @ <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> @ <EDIT> ausgeführt.

<ENTER> \* <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> \* <EDIT> ausgeführt.

<ENTER> > <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> > <EDIT>  
ausgeführt.

<ENTER> < <HWINDOW>

Erzeugt durch horizontale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der untere Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> < <EDIT>  
ausgeführt.

## INSERTMODE - Den Eingabemodus wählen

<INSERTMODE>  
<ENTER> [ o / i / w / l / s ] <INSERTMODE>

Mit dem Kommando INSERTMODE stellen Sie den Eingabe-Modus ein (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Sie können zum einen zwischen den beiden grundsätzlichen, alternativ verfügbaren Eingabe-Modi zum Einfügen und Überschreiben umschalten. Daneben ist eine Feineinstellung der Einfüge-Modi möglich:

- INSERT-Modus (Standard-Einfüge-Modus)
- WORD-Modus
- LEXICAL-Modus
- STREAM-Modus

<INSERTMODE>

Schaltet zwischen den alternativ verfügbaren Eingabe-Modi zum Einfügen und Überschreiben (OVERWRITE) um.

### *Anmerkung*

Im OVERWRITE-Modus sind die Umbrucheigenschaften und die Art und Weise der Behandlung von Leerzeichen entsprechend des alternativ verfügbaren Einfüge-Modus gültig.

<ENTER> <INSERTMODE>

bzw.

<ENTER> o <INSERTMODE>

Stellt den Überschreibe-Modus OVERWRITE ein. Beim Schreiben und Kopieren wird der unterliegende Text überschrieben.

<ENTER> i <INSERTMODE>

Stellt den Standard-Einfüge-Modus INSERT ein. Beim Einfügen wird rechts von der Schreibmarke stehender Text unverändert nach rechts verschoben, Leerzeichen werden nicht entfernt. An der rechten Formatgrenze wird wortorientiert umbrochen.

<ENTER> w <INSERTMODE>

Stellt den Einfüge-Modus WORD ein. Wird der Text durch weitere Texteingabe oder Einfügen von Textbereichen nach rechts verschoben, werden mehrfach auftretende Leerzeichen zwischen den Wörtern sukzessive zu einem Leerzeichen komprimiert. An der rechten Formatgrenze wird wortorientiert umbrochen.

<ENTER> l <INSERTMODE>

Stellt den Einfüge-Modus LEXICAL ein. Wird der Text durch weitere Texteingabe oder Einfügen von Textbereichen nach rechts verschoben, werden mehrfach auftretende Leerzeichen zwischen den Wörtern, vor Satzzeichen und nach öffnenden bzw. vor schließenden Klammern sukzessive zu einem Leerzeichen komprimiert.

*Anmerkung*

- Im LEXICAL-Modus ist der automatische Silbentrenner eingeschaltet, d.h. an einer rechten Formatgrenze wird automatisch getrennt. Diese Eigenschaft bleibt auch beim Umschalten in den OVERWRITE-Modus wirksam (siehe Abschnitt 4.1.10 Die Sprach-Modi).
- Beim Löschen von Einzelzeichen mit den Kommandos BACKSPACE und DELCHAR verschiebt sich lediglich der nachfolgende Rest des aktuellen Wortes um eine Spalte nach links. Der übrige Zeilenrest rechts des aktuellen Wortes bleibt unverändert stehen.

<ENTER> s <INSERTMODE>

Stellt den Einfüge-Modus STREAM ein. Wird der Text durch weitere Texteingabe oder Einfügen von Textbereichen nach rechts verschoben, werden alle auftretenden Leerzeichen sukzessive entfernt.

**JUSTIFY - Text links- und rechtsbündig ausrichten**

```
<JUSTIFY>
<ENTER> [RANGE/RECTANGLE/einzeiliger [BLOCK/RECORD]STREAM] <JUSTIFY>
```

Das Kommando JUSTIFY richtet den Text in dem angegebenen Textbereich links- und rechtsbündig an den Formaträndern aus, wobei der MAXed den rechten Fensterrand als Ersatz für einen fehlenden rechten Formatrand akzeptiert (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN). Dazu werden zwischen den Wörtern Leerzeichen eingefügt.

JUSTIFY ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

```
<JUSTIFY>
```

Richtet den Text der aktuellen Zeile links- und rechtsbündig aus.

```
<ENTER> <JUSTIFY>
```

Richtet den Text des Zeilenrestes ab der aktuellen Position der Schreibmarke links- und rechtsbündig aus. Die Schreibmarke wird als temporärer linker Formatrand betrachtet.

```
<ENTER> bereich <JUSTIFY>
```

Richtet den Text im angegebenen Bereich links- und rechtsbündig aus. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK. Allerdings darf ein STREAM nur einzeilig sein. Der linke und rechte Rand von STREAM oder RECTANGLE werden als temporäre Formatränder betrachtet, zwischen denen ausgerichtet wird.

**Beispiel**

```
<ENTER> 5 42 <INSERTMODE>
<ENTER> 42 <COMPRESS>
```

In den folgenden 42 Zeilen wird der Text der einzelnen Zeilen an den Formaträndern (Spalten 5 und 42) links- und rechtsbündig ausgerichtet. Wörter, die über die Formatränder hinausragen bleiben unverändert stehen.

## LABEL - Textstellen markieren

```
<LABEL>
<ENTER> [ IDENTIFIER / , ] <LABEL>
```

Mit dem Kommando LABEL wird die aktuelle Position der Schreibmarke in der Datei mit einer nicht sichtbaren Marke gekennzeichnet. Marken sind dateispezifisch, d.h. der MAXed hält für jede geöffnete Datei einen eigenen Satz an Marken. Alle momentan gültigen Marken, also diejenigen, die Sie in der aktuellen Datei gesetzt haben, können Sie für das Kommando GOTO zum Anspringen ihrer jeweiligen Dateiposition verwenden (siehe Abschnitt 3.8.4 Der Markenname als IDENTIFIER, Kapitel 6: GOTO).

Sie können Marken auch zur Definition von Bereichen verwenden (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut).

```
<LABEL>
```

Setzt die Standard-Marke an der Position der Schreibmarke.

```
<ENTER> IDENTIFIER <LABEL>
```

Markiert die aktuelle Position in der Datei durch eine nicht sichtbare Marke mit dem angegebenen Namen.

```
<ENTER> <LABEL>
```

Löscht die Marke an der aktuellen Position.

```
<ENTER> , <LABEL>
```

Markiert die aktuelle Position der Schreibmarke in der Datei durch eine nicht sichtbare Marke, die als IDENTIFIER das Wort, unter dem die Schreibmarke steht, zugeteilt bekommt.

### Anmerkung

Gesetzte Marken werden bei Verlassen des MAXed mit EXIT in der benutzer- und terminalspezifischen Status-Datei gespeichert. Wenn Sie den MAXed beim nächsten Mal ohne Angabe eines Dateinamens aufrufen, werden diese wieder gesetzt (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

**Beispiel 1**

<ENTER> marke3 <LABEL>

setzt an der aktuellen Position der Schreibmarke die Marke *marke3*.

**Beispiel 2**

Mit

<ENTER> ® <GOTO>

werden in der Pseudo-Datei LABELS die in der bearbeiteten Datei gesetzten Marken angezeigt. In diesem Fall zumindest *marke3*.

## LEARN - Ein Makro anlegen

```
<LEARN>
<ENTER> [ IDENTIFIER [p1 [p2 ... [p8 ]]] ] <LEARN>
<ENTER> [IDENTIFIER] <LEARN> <LEARN>
```

Das Kommando LEARN ermöglicht die Erstellung von Makros während der Bearbeitung einer Datei, also in der Lernphase. Alle Kommandos, die Sie im Lern-Modus eingeben, werden bis zur Beendigung der Lernphase gespeichert. Den Lernmodus starten und beenden Sie mit dem Kommando LEARN (siehe Abschnitte 3.8.5 Der Makroname als IDENTIFIER, 4.1.11 Der Lern-Modus).

<LEARN>

Start oder Ende einer Lernphase. Mit einem ersten Drücken der Taste <LEARN> schalten Sie den Lern-Modus für die Definition des Standard-Makro ein. Mit einem zweiten <LEARN> schließen Sie diese ab (siehe Abschnitt 3.12.2 Kommandos mit Standard-Puffern).

### *Anmerkung*

Das (unbenannte) Standard-Makro existiert für die Dauer der aktuellen MAXed-Sitzung unabhängig von den anderen benannten Makros bzw. dem aktuellen Makro. Es wird nur dann überschrieben, wenn Sie es neu definieren.

<ENTER> IDENTIFIER <LEARN>

Leitet die Definition eines benannten Makros ein. Mit dem Drücken der Taste <LEARN> schließen Sie diese ab. Geben Sie während der Lernphase eines anderen Makros das Kommando

<ENTER> IDENTIFIER <LEARN>

ein, so beenden Sie damit dessen Lernphase. Dieses Makro gilt dann als vollständig definiert. Gleichzeitig beginnen Sie die Lernphase für das neue, durch IDENTIFIER angegebene Makro.

<ENTER> IDENTIFIER [p1 [p2 ... [p8 ]]] <LEARN>

Beim Aufruf des Kommandos LEARN können Sie bis zu acht Zeichenketten als aktuelle Makro-Parameter *p1*, *p2*,... *p8* angeben. Diese aktuellen Makro-Parameter verwenden Sie während der Lernphase, indem Sie sie innerhalb von Kommandoattributierungen im Eingabefeld durch Angabe der zugeordneten formalen Makro-Parameter *@1*, *@2*,... *@8* einsetzen lassen (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld: Parameter für Textersetzung). Damit erstellen Sie parametrisierte Makros, denen Sie beim Aufruf in gleicher Weise wieder Zeichenketten als aktuelle Parameter mitgeben (siehe auch unten: Beispiel 2).

#### *Achtung*

Eine Zeichenkette, die Leerzeichen enthält, müssen Sie – wie dies bei der Eingabe von Kommandoattributen üblich ist – in Anführungszeichen setzen, da sie sonst mehrere aktuelle Makro-Parameter darstellen würde. Das müssen Sie insbesondere dann beachten, wenn innerhalb von Zeichenketten durch Textersetzung Leerzeichen entstehen können!

<ENTER> <LEARN>

Schaltet den Lern-Modus aus, bricht jedoch die aktuelle Lernphase ab. Alles, was Sie in dieser Lernphase eingegeben haben, wird vergessen. Kein neues Makro wird definiert. Alte Makro-Definitionen, auch das aktuelle Makro, werden nicht überschrieben. Die während der Lernphase ausgeführten MAXed-Kommandos werden selbstverständlich nicht zurückgenommen.

<ENTER> IDENTIFIER <LEARN> <LEARN>

bzw.

<LEARN> <LEARN>

Löscht das durch IDENTIFIER angegebene Makro bzw. das Standard-Makro, das in der aktuellen Sitzung definiert ist.

#### **Anmerkung**

Wollen Sie komplexe Makros definieren, empfiehlt es sich, eine Quelldatei für die Makro-Definition(en) zu erstellen und diese mit dem Dienstprogramm *mdefine* in eine MACRO-Bibliothek zu überführen (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen).

**Beispiel 1**

```
<ENTER> makro42 <LEARN>
```

schaltet die Lern-Phase für das Makro *makro42* ein. Die folgenden Eingaben werden sowohl in der bearbeiteten Datei ausgeführt als auch in dem Makro *makro42* gespeichert.

```
<GOTO>           gehe an den Dateianfang;
<ENTER> ; <PLUSFIND> finde das Ende der nächsten nicht leeren
                  Zeile;
<ENTER> 1 65 <MARGIN> setze den linken und rechten Formatrand;
<FILL>           formatiere den nachfolgenden Absatz im
                  Flattersatz.
```

Damit soll der Inhalt des Makros *makro42* definiert sein. Um die Lernphase zu beenden, geben Sie ein:

```
<LEARN>
```

**Beispiel 2**

Sie haben im PICK-Puffer *abs* ein RECTANGLE mit Ihrer Adresse und im PICK-Puffer *gru* den STREAM

"Mit freundlichen Grüßen!" inklusive der Anführungszeichen gespeichert (siehe Kommando PICK). Die Variable *ORT* haben Sie mit Ihrem Wohnort belegt durch

```
<ENTER> ORT=München <ENTER>
```

Nun wollen Sie zum soundsovielten Mal (es geht nämlich auf -Weihnachten zu) eine Ladung Briefpost zum Ausdruck fertig machen. Diese Briefe haben Sie im Dateiverzeichnis *post* mit der Endung *.aus* versehen. Damit Sie nicht in allen Dateien Briefkopf und Schluß einzeln eintragen müssen, führen Sie eine Makro-Definition in der Lernphase durch.

```
<ENTER> post/* .aus <EDIT>
<DOWNARROW>
```

Die Pseudo-Datei EDITFILES mit der Liste der Dateien, welche auf *.aus* enden, ist im aktuellen Fenster eingeblendet und die Schreibmarke steht auf dem Namen der ersten zu bearbeitenden Datei.

```

<ENTER> brief $ORT gru <LEARN>
                                die Lernphase für das Makro "brief"
                                mit zwei Parametern beginnt;
                                Datei aus der Liste auswählen;
<EDIT>
<GOTO>
<HOME>                          nach Hause gehen;
<ENTER> 4 <RIGHTARROW>          Position für linken Rand;
<MARGIN>                          linken Formatrand setzen;
<ENTER> i <INSERTMODE>          gezielt INSERT-Modus setzen;
<SPACE>                          Leerzeile einfügen;
<ENTER> t <RETURN>              RETURN-Taste in Text-Modus schalten;
<ENTER> abs + <PUT>             Absender als RECTANGLE einfügen, vorher
                                Platz schaffen;
<ENTER> "sehr gee" <PLUSFIND>    Anrede suchen (Suchmuster
                                enthält Leerzeichen!);
<ENTER> fraulherr <PLUSFIND>     entweder ist's eine "sie" oder ein "er";
<ENTER> <PICK>                   Zeilenrest in Standard-PICK-Puffer;
<ENTER> . <GOTO>                 das ist das Ende!
<ENTER> 3 <RETURN>              dreimal zurückkehren;
Ich wünsche Ihnen,              normale Texteingabe;
<ENTER> ' % ein frohes Fest!' <PUT>
                                Standard-PICK-Puffer für "%"
                                einsetzen;
<RETURN>
<RETURN>                          zwei Leerzeilen;
<ENTER> '@1, den @d' <PUT>       Ort und Datum;
<RETURN>
<RETURN>                          noch zwei Leerzeilen;
<ENTER> 40 <RIGHTARROW>         auf Spalte 41;
<ENTER> @2 <PUT>                 und ganz liebe / freundliche Grüße;
<EDIT>                          die Datei EDITFILES wieder einblenden;
<LINEFEED>                       auf nächste Datei positionieren;

```

Danach haben Sie die Schreibmarke auf den Namen der nächsten Briefdatei aus der Liste positioniert. Sie schließen die Makro-Definition ab mit

```
<LEARN>
```

Um die restlichen 30 Briefe bearbeiten zu lassen, müssen Sie lediglich das folgende Kommando starten:

```
<ENTER> 30 brief $ORT gru <MACRO>
```

Es wird erwartet, daß die Dateien die Anrede "Sehr geehrte(r) ..." enthalten. Ist das nicht der Fall, wird die Makro-Ausführung abgebrochen. Wenn Sie allerdings zuvor das Kommando

<ENTER> \* <MACRO>

eingeben, erzwingen Sie damit die Fortführung.

Bei der Parameterübergabe in diesem Beispiel ist folgendes zu beachten: dem Makro *brief* wird beim Aufruf die Variable *ORT* übergeben, die durch die Textersetzung vor dem Start des Makros durch ihren Wert ersetzt wird (siehe Kapitel 6: ENTER). Als zweiter aktueller Makro-Parameter ist der IDENTIFIER des PICK-Puffers *gru* angegeben. Diese beiden Zeichenketten sind danach den Makro-Parametern *@1* und *@2* zugeordnet.

**LEFTARROW - Die Schreibmarke nach links bewegen**

<LEFTARROW>  
 <ENTER> n / h / v / m / c <LEFTARROW>

Kommandoverstärkung:  
 <ENTER> n <LEFTARROW>

Start einer Bereichsdefinition für STREAM:  
 <ENTER> <LEFTARROW>

Das Kommando LEFTARROW bewegt die Schreibmarke eine Spalte nach links. Im standardmäßig eingestellten Fenster-Modus springt die Schreibmarke auf den gegenüberliegenden Fensterrand um, wenn Sie sie auf den Rand des aktuellen Fensters bewegen. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Sie können den Fenster-Modus jedoch auch so einstellen, daß die Schreibmarke stattdessen den Dateiausschnitt verschiebt oder in ein angrenzendes Fenster überwechselt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

<LEFTARROW>  
 Bewegt die Schreibmarke um eine Spalte nach links.

<ENTER> <LEFTARROW>  
 Start der Definition eines STREAM-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <LEFTARROW>  
 Bewegt die Schreibmarke um *n* Spalten nach links.

<ENTER> modus <LEFTARROW>  
 Stellt den aktuellen Fenster-Modus ein (siehe Abschnitt 4.1.4 Der Fenster-Modus). Davon betroffen sind alle ARROW- und TAB-Kommandos beim Bewegen der Schreibmarke auf bzw. über den Rand des aktuellen Fensters. Für *modus* können Sie folgendes eingeben:

- n[ormal] Die Schreibmarke erhält für alle vier Richtungen die Eigenschaft, zum gegenüberliegenden Fensterrand umzuspringen. Dieser Fenster-Modus ist standardmäßig eingestellt.
- h[orizontal] Für horizontale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando oder durch Texteingabe den rechten bzw. linken Fensterrand überschreitet. Am Zeilenanfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.
- v[ertical] Für vertikale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando den oberen bzw. unteren Fensterrand überschreitet. Am Dateianfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.
- m[ove] Kombiniert die Einstellungen *horizontal* und *vertikal*. Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt.
- c[hangewindow] Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt. Stattdessen wechselt die Schreibmarke (falls möglich) über den Rand des aktuellen Fensters in das benachbarte Fenster, das zum aktuellen Fenster wird (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten). Beim Überschreiten des äußersten Fensterrandes springt die Schreibmarke allerdings nach wie vor um.

### Anmerkung

Mit <LEFTARROW> können Sie die Schreibmarke auch im Eingabefeld der Steuerzeile nach links verschieben (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Dabei ist ein Umspringen oder Wechseln des Fensters nicht möglich. Der sichtbare Ausschnitt des Eingabefeldes wird jedoch um eine Ausschnittslänge mitverschoben, wenn die Schreibmarke den linken Rand des Eingabefeldes erreicht und der Anfang des Eingabefeldes nicht sichtbar ist.

## LINEFEED - Zeilenweises Verschieben des Dateiausschnitts

<LINEFEED>

Kommandoverstärkung:

<ENTER> n <LINEFEED>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <LINEFEED>

Das Kommando LINEFEED verschiebt den Dateiausschnitt im aktuellen Fenster und bewegt gleichzeitig die Schreibmarke um eine Zeile nach unten. Die aktuelle Spalte bleibt dieselbe. Die Schreibmarke verändert also ihre relative Position innerhalb des Fensters nicht (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<LINEFEED>

Verschiebt den Dateiausschnitt und die Schreibmarke um eine Zeile nach unten.

<ENTER> <LINEFEED>

Verschiebt den Dateiausschnitt um ein Drittel seiner Höhe nach unten. Dies entspricht dem Kommando <PLUSLINE>. Allerdings wird die Schreibmarke ebenso weit nach unten bewegt. Sie bleibt also im Fenster auf gleicher Höhe stehen.

<ENTER> n <LINEFEED>

Verschiebt den Dateiausschnitt und die Schreibmarke um  $n$  Zeilen nach unten.

### Anmerkung

Dieses Kommando ist besonders in Makros nützlich. Sie entgehen der Gefahr des Umspringens der Schreibmarke am unteren Fensterrand, indem Sie das Fenster mitbewegen (siehe Abschnitt 4.1.4 Der Fenster-Modus).

## LOWER - Kleinschreibung

<LOWER>  
<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] <LOWER>

Das Kommando LOWER wandelt Großbuchstaben in Kleinbuchstaben um. Alle anderen Zeichen bleiben unverändert (siehe Abschnitt 4.3.4 Ausrichten und Vereinheitlichen).

LOWER ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<LOWER>  
Wandelt alle Großbuchstaben der aktuellen Zeile in Kleinbuchstaben um.

<ENTER> <LOWER>  
Wandelt alle Großbuchstaben von der aktuellen Spalte bis zum Ende der aktuellen Zeile in Kleinbuchstaben um.

<ENTER> *bereich* <LOWER>  
Führt das Kommando LOWER in dem angegebenen Bereich aus. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

## MACRO - Ein Makro ausführen

```

<MACRO>
<ENTER> [ IDENTIFIER [p1 [p2 ... [p8]]] ]
          / [ steuerprogramm [programmparameter] ]
          / [ @ / * / ! ] <MACRO>
          <ENTER> n [ IDENTIFIER [p1 [p2 ... [p8]]] ] <MACRO>

```

Mit dem Kommando `MACRO` werden definierte Makros oder `MAXed`-Steuerprogramme zur Ausführung aufgerufen (siehe Abschnitte 3.8.5 Der Makroname als `IDENTIFIER`, 3.8.6 Der Name eines Steuerprogramms als `IDENTIFIER`). Makros können Sie in der Lern-Phase mit Hilfe des Kommandos `LEARN` erstellen (siehe Kapitel 6: `LEARN`). Außerdem stehen Ihnen die Makros zur Verfügung, die in einer geladenen `MACRO`-Bibliothek definiert sind (siehe Abschnitte 4.7 Arbeiten mit Makros, 7.1 Eine `MACRO`-Bibliothek erstellen, Kapitel 6: `LEARN`). Steuerprogramme erstellen Sie als `C`-Quellprogramme (siehe Abschnitt 4.8 Arbeiten mit `MAXed`-Steuerprogrammen).

Der Kommando-Puffer zu `MACRO` enthält immer das aktuelle Makro, d.h. jenes benannte oder unbenannte Makro, das Sie zuletzt mit dem Kommando `LEARN` definiert oder mit dem Kommando `MACRO` ausgeführt haben (siehe Abschnitte 3.12.2 Kommandos mit Standard-Puffern, 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen).

<MACRO>

Führt das aktuelle, d.h. das im Kommando-Puffer zu `MACRO` gespeicherte Makro einmal aus. Dieses ist durch die zuletzt ausgeführte Lernphase mittels des Kommandos `LEARN` oder durch den vorangegangenen Makro-Aufruf mittels des Kommandos `MACRO` überschrieben worden.

<ENTER> <MACRO>

Führt das Standard-Makro einmal aus. Dieses haben Sie zuvor bei einer Lernphase mittels des Kommandos `LEARN` als unbenanntes Makro definiert.

<ENTER> IDENTIFIER <MACRO>

Führt das mit `IDENTIFIER` benannte Makro aus.

<ENTER> n [IDENTIFIER] <MACRO>

Führt das mit `IDENTIFIER` benannte Makro bzw. das aktuelle Makro  $n$ -mal aus. Der Bildschirm wird erst nach Beendigung der  $n$ -maligen Ausführung aktualisiert.

<ENTER> [n] IDENTIFIER [p1 [p2 ... [p8]]] <MACRO>

Sie können beim Aufruf eines parametrisierten Makro bis zu 8 aktuelle Makro-Parameter angeben, die während der Makro-Ausführung durch die zugeordneten formalen Makro-Parameter @1, @2,..., @8 auf der Steuerzeile abgerufen werden (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

<ENTER> steuerprogramm [programm-parameter] <MACRO>

Startet das MAXed-Steuerprogramm *steuerprogramm*, falls kein Makro mit dem angegebenen Namen *steuerprogramm* definiert ist (siehe Abschnitte 3.8.6 Der Name eines Steuerprogramms als IDENTIFIER, 4.8 Arbeiten mit Steuerprogrammen, 7.5 MAXed-Steuerprogramme).

<ENTER> @ <MACRO>

Blendet die Pseudo-Datei MACROS mit der Liste aller aktuell definierten Makros ein (siehe Abschnitt 4.2.10 Die Pseudo-Datei MACROS).

<ENTER> \* <MACRO>

Stellt den Makro-Abarbeitungs-Modus auf CONTINUE ein (siehe Abschnitt 4.1.6 Der Makro-Abarbeitungs-Modus). Dies bewirkt, daß die Ausführung von Makros fortgesetzt wird, auch wenn bei der Abarbeitung Fehler auftreten. Standardmäßig wird die Kommandoausführung abgebrochen, wenn ein Fehler während der Ausführung eines Makros auftritt.

<ENTER> ! <MACRO>

Stellt den Makro-Abarbeitungs-Modus wieder auf TERMINATE ein (siehe oben). Die Ausführung eines Makros wird abgebrochen, wenn bei der Abarbeitung Fehler auftreten. Dies ist die Standard-Einstellung.

### Beispiel

<ENTER> 3 makro2 MAXed <MACRO>

Das Kommando ersetzt im Makro *makro2* den Parameter @1 durch *MAXed* und bewirkt, daß das Makro dreimal ausgeführt wird.

## MARGIN - Linken und rechten Formatrand setzen

```
<MARGIN>
<ENTER> [ linker_rand rechter_rand ]
          / [ bewegung_der_schreibmarke ] <MARGIN>
```

Mit dem Kommando MARGIN werden der linke und rechte Rand des Textformats gesetzt, an dem sich Zeilenumbruch und Zeilenschaltung orientieren (siehe Abschnitte 4.3.2 Formatränder, 4.3 Das Formatieren, 4.1.3 Der Zeilenschaltungs-Modus).

Standardmäßig sind beide Formatränder auf der ersten Spalte des aktuellen Fensters gesetzt. In dieser Einstellung werden sie nicht bei der Textbearbeitung berücksichtigt.

Wenn Sie den linken Formatrand auf eine beliebige andere Spaltenposition setzen, wird der Textanfang einer Zeile beim Zeilenumbruch an dieser Spalte ausgerichtet. Außerdem wird die Schreibmarke bei der Zeilenschaltung mit Hilfe des Kommandos RETURN immer an den linken Formatrand gesetzt.

Wenn Sie den rechten Formatrand auf eine Spalte rechts des linken Formatrandes gesetzt haben, wird das Textende einer Zeile beim Zeilenumbruch an dieser Spalte vorgenommen. Außerdem erfolgt während der Texteingabe beim Überschreiben der rechten Formatgrenze ein automatischer Zeilenumbruch.

```
<MARGIN>
  Setzt die linke Formatgrenze an der aktuellen Spaltenposition der
  Schreibmarke.
```

```
<ENTER> <MARGIN>
  Setzt die rechte Formatgrenze an der aktuellen Spalte.
```

```
<ENTER> linker_rand rechter_rand <MARGIN>
  Setzt die linke Formatgrenze an der Spaltennummer linker_rand, die
  rechte an der Spaltennummer rechter_rand.
```

```
<ENTER> bewegung_der_schreibmarke <MARGIN>
  Setzt die linke bzw. die rechte Formatgrenze an die Spaltenpositionen
  an denen die Bereichsgrenzen dieser Bereichsdefinition liegen (siehe
  Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).
```

**Anmerkung**

- Abhängig vom eingestellten Eingabe-Modus wird wortweise umbrochen oder getrennt (siehe Abschnitt 4.1.1 Die Eingabe-Modi).
- Gesetzte Formatränder werden bei Verlassen des MAXed mit EXIT in der benutzer- und terminalspezifischen Status-Datei gespeichert. Wenn Sie den MAXed das nächsten Mal ohne Angabe eines Dateinamens aufrufen, werden sie wieder gesetzt (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

**Beispiel**

```
<ENTER> 15 95 <MARGIN>
```

legt den linken Formatrand bei Spalte 15, den rechten bei Spalte 95 fest.

## MINUSFIND - Textstelle suchen

```

<MINUSFIND>
<ENTER> [ PATTERN ]
          /[ [ % ] [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] ]
          /[ RANGE/RECTANGLE/[BLOCK/RECORD]STREAM PATTERN ]
          /[%[IDENTIFIER] ]
          /[@ / / / ' ] <MINUSFIND>

```

Das Kommando MINUSFIND sucht in der aktuellen Datei rückwärts nach der ersten mit dem definierten Suchmuster übereinstimmenden Textstelle. Suchmuster geben Sie in Form eines PATTERNS oder einer Bereichsdefinition an (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut, 3.10 PATTERN als Kommandoattribut). Durch ein PATTERN können Sie mehrere alternative Suchmuster gleichzeitig definieren.

Wenn Sie kein Suchmuster angeben, greift das Kommando PLUSFIND auf das aktuelle PATTERN zurück. Das aktuelle PATTERN wird immer dann überschrieben, wenn Sie ein Suchkommando aufrufen, bei dem Sie ein PATTERN als Kommandoattribut angeben (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen).

Wenn eine zum definierten Suchmuster passende Textstelle gefunden ist, wird die Schreibmarke auf das erste Zeichen der gesuchten Textstelle positioniert. Gegebenenfalls verschiebt sich das Fenster, sodaß die Schreibmarke im aktuellen Fenster sichtbar bleibt. Wird keine übereinstimmende Textstelle bis zum Anfang der Datei gefunden, erscheint in der Steuerzeile eine der folgenden Meldungen:

```

**** PATTERN nicht gefunden.
**** RANGE nicht gefunden.
**** RECTANGLE nicht gefunden.
**** STREAM nicht gefunden.

```

Schreibmarke und Fenster bleiben dann in ihrer alten Position.

<MINUSFIND>

Sucht ab der aktuellen Position der Schreibmarke rückwärts nach der ersten Zeichenkette, die mit dem aktuellen PATTERN übereinstimmt. Diese Eingabe ist nur als Wiederholungskommando geeignet.

<ENTER> <MINUSFIND>

Sucht rückwärts nach einer übereinstimmenden Zeichenkette. Das Suchmuster ist dabei das aktuelle Wort, unter dem die Schreibmarke positioniert ist.

<ENTER> PATTERN <MINUSFIND>

Sucht rückwärts nach einer zu PATTERN passenden Zeichenfolge.

<ENTER> / <MINUSFIND>

Findet den Anfang des durch die Position der Schreibmarke gekennzeichneten RECORD. Die Schreibmarke wird auf den Anfang des RECORD positioniert. (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut: RECORD als Spezialfall eines STREAM-Bereichs).

<ENTER> ' <MINUSFIND>

Findet den Anfang eines BLOCKs in einer verschachtelten Klammerstruktur. Die Schreibmarke wird auf den Anfang des kleinsten umgebenden BLOCKs positioniert. Wenn notwendig, wird das Fenster verschoben (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut: BLOCK als Spezialfall eines STREAM-Bereichs).

#### *Anmerkung*

Dieses Kommando steht Ihnen standardmäßig als Tasten-Makro <MINUSBLOCK> zur Verfügung (siehe Abschnitt 6.2 Die Tasten-Makros).

<ENTER> bereich <MINUSFIND>

Sucht von der aktuellen Position der Schreibmarke aus rückwärts nach demjenigen Text, der identisch ist mit dem durch die Bereichs-angabe gekennzeichneten Text. Der Bereich muß durch Bewegen der Schreibmarke definiert sein (siehe Abschnitt 3.9.1). Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> % bereich <MINUSFIND>

Sucht von der aktuellen Position der Schreibmarke aus rückwärts nach demjenigen Text, der identisch ist mit dem durch die Bereichsdefinition gekennzeichneten Text. Der Bereich muß durch ein Zeichenketten-Attribut angegeben sein (siehe Abschnitt 3.9.2). Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

*Anmerkung*

Das Prozentzeichen % vor der Bereichsangabe ist deshalb notwendig, weil der MAXed diese sonst als PATTERN interpretieren würde (siehe oben).

<ENTER> bereich PATTERN <MINUSFIND>

Sucht innerhalb des angegebenen Bereichs rückwärts nach PATTERN. Wenn Sie für PATTERN die leere Zeichenkette "" angeben, wird das aktuelle PATTERN verwendet. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

*Anmerkung*

Das Kommando unterscheidet sich von den vorangegangenen Kommandos dadurch, daß es innerhalb des Bereichs nach einem angegebenen PATTERN sucht!

<ENTER> %[IDENTIFIER] <MINUSFIND>

Sucht rückwärts nach einem Textbereich, der mit dem Inhalt des angegebenen PICK-Puffers identisch ist. Falls IDENTIFIER nicht angegeben ist, wird der Standard-PICK-Puffer verwendet (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER).

*Anmerkung*

Bei dieser Form der Kommandoattributierung wird keine Textersetzung im Eingabefeld durchgeführt (siehe Abschnitt 3.6.4: Ersetzen von PICK-Puffernamen). Der Vorteil ist, daß der Textbereich des angegebenen PICK-Puffers nicht durch eine Textersetzung in einen einzeiligen STREAM umgewandelt und als PATTERN verwendet wird! Im Gegensatz dazu findet bei folgendem Kommando eine Textersetzung statt:

<ENTER> "%[IDENTIFIER]" <MINUSFIND>

<ENTER> @ <MINUSFIND>

Blendet die Pseudo-Datei FINDHISTORY mit der Liste aller PATTERN ein, die Sie im bisherigen Verlauf der Sitzung beim Aufruf der Kommandos MINUSFIND und PLUSFIND als Kommandoattribut angegeben haben. (siehe Abschnitt 4.2.9 Die Pseudo-Datei FINDHISTORY).

**Anmerkung**

- Bei der Suche mit PATTERN werden Zeichenattribute nicht berücksichtigt.
- Bei der Suche mit einer Bereichsdefinition als Suchmuster werden gesetzte Zeichenattribute nur dann berücksichtigt, wenn die aktuelle Einstellung der Zeichenattribute aktiviert ist: Alle gesetzten Zeichenattribute in zwei zu vergleichenden Textstellen, die in der aktuellen Einstellung nicht mit Stern \* für "nicht berücksichtigen" angezeigt werden, müssen übereinstimmen (siehe Abschnitt 4.4 Die Zeichenattribute).

**Beispiel**

<ENTER> , <MINUSFIND>

Der MAXed verwendet das aktuelle Wort als Suchmuster und sucht rückwärts nach einer identischen Textstelle.

**MINUSLINE - In der Datei zurückblättern**

<MINUSLINE>

Kommandoverstärkung:

<ENTER> n <MINUSLINE>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <MINUSLINE>

Das Kommando MINUSLINE schiebt den Dateiausschnitt im aktuellen Fenster um etwa ein Drittel seiner Höhe nach oben. Die aktuelle Spalte wird beibehalten. Befindet sich die ursprüngliche, aktuelle Zeile noch innerhalb des Fensters, bleibt die Schreibmarke in dieser Zeile stehen. Anderenfalls bewegt sich die Schreibmarke um die gleiche Anzahl Zeilen nach oben, bleibt also im Fenster auf gleicher Höhe stehen. Falls das Fenster den Anfang der Datei zeigt, wird der Dateiausschnitt nicht verschoben (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<MINUSLINE>

Verschiebt den Dateiausschnitt um einige Zeilen nach oben.

<ENTER> <MINUSLINE>

Verschiebt den Dateiausschnitt so, daß die aktuelle Zeile zur letzten Zeile des Fensters wird.

<ENTER> n <MINUSLINE>

Verschiebt den Dateiausschnitt um  $n$  Zeilen nach oben.

## MINUSPAGE - Seitenweise zurückblättern

<MINUSPAGE>

Kommandoverstärkung:

<ENTER> n <MINUSPAGE>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <MINUSPAGE>

Das Kommando MINUSPAGE schiebt den Dateiausschnitt seitenweise nach oben. Die Größe einer Seite ist durch die Zahl der Zeilen im aktuellen Fensters bestimmt. Die Schreibmarke verändert ihre relative Position innerhalb des Fensters nicht. Die aktuelle Spalte wird beibehalten. Falls das Fenster den Anfang der Datei zeigt, wird der Dateiausschnitt nicht verschoben (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<MINUSPAGE>

Verschiebt den Dateiausschnitt und die Schreibmarke um eine Seite nach oben.

<ENTER> <MINUSPAGE>

Start der Definition eines RANGE- oder RECTANGLE-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Dateiausschnitt und Schreibmarke bewegen sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <MINUSPAGE>

Verschiebt den Dateiausschnitt und die Schreibmarke um *n* Seiten nach oben.

## PICK - Textbereiche zwischenspeichern

```
<PICK>
<ENTER> [ [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] [ > / >> ] [IDENTIFIER] ]
        / [ IDENTIFIER ] [ > / >> ] [ IDENTIFIER ] ] <PICK>
```

Das Kommando PICK wird zum Kopieren von Textstellen verwendet. Dabei wird zunächst der Text in einen Speicher, den sogenannten PICK-Puffer, kopiert. Von dort kann er dann durch das Kommando PUT an die gewünschte Position kopiert werden (siehe Abschnitte 3.8.3 Der PICK-Puffername als IDENTIFIER, 3.12.2 Kommandos mit Standard-Puffern, Kapitel 6: PUT).

<PICK>

Kopiert die aktuelle Zeile in den Standard-PICK-Puffer. Den Inhalt dieses Puffers können Sie mit <PUT> wieder in den Text kopieren.

<ENTER> IDENTIFIER <PICK>

Kopiert die aktuelle Zeile in den PICK-Puffer mit dem Namen IDENTIFIER. Auf einen benannten PICK-Puffer können Sie mit <ENTER> IDENTIFIER <PUT> wieder zugreifen.

<ENTER> <PICK>

Kopiert den Zeilenrest der aktuellen Zeile ab der aktuellen Position der Schreibmarke in den Standard-PICK-Puffer.

<ENTER> *bereich* <PICK>

Kopiert den angegebenen Bereich in den Standard-PICK-Puffer. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> *bereich* [ > ] IDENTIFIER <PICK>

Kopiert den angegebenen Bereich in den mit IDENTIFIER bezeichneten PICK-Puffer. Die schließende spitze Klammer > ist ein Hinweis auf die Kopierrichtung. Sie können sie auch weglassen. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> *bereich* >> [IDENTIFIER] <PICK>

Kopiert den angegebenen Bereich und hängt ihn ans Ende des mit IDENTIFIER bezeichneten PICK-Puffers an. Falls IDENTIFIER weggelassen wird, wird der Standard-PICK-Puffer als Ziel verwendet. Für *bereich* sind RANGE, STREAM und RECTANGLE möglich. Auch RECORD und BLOCK als Spezialfall von STREAM sind erlaubt.

<ENTER> [ID1] [ > ] [ID2] <PICK>

Kopiert den Text aus dem PICK-Puffer mit dem Namen ID1 wird in den PICK-Puffer mit dem Namen ID2 kopiert. Die schließende spitze Klammer > ist ein Hinweis auf die Kopierrichtung. Sie können sie auch weglassen. Falls ID1 oder ID2 weggelassen wird, wird der Standard-PICK-Puffer als Quelle bzw. Ziel benutzt. In diesem Fall müssen Sie die schließende spitze Klammer > angeben, damit sich diese Kommandoattributierung nicht mit einer der oben beschriebenen überschneidet!.

<ENTER> [ID1] >> [ID2] <PICK>

Hängt den Text aus dem PICK-Puffer mit dem Namen ID1 ans Ende des PICK-Puffers mit dem Namen ID2. Falls ID1 oder ID2 weggelassen wird, wird der Standard-PICK-Puffer als Quelle bzw. Ziel benutzt.

### Anmerkung

- Bei einigen der vorstehenden Kommandos haben Sie die Möglichkeit, das Zeichen > mit anzugeben. Bis auf einen Fall wird die Wirkung der Kommandos dadurch nicht beeinflusst. Das Zeichen ist als Hinweis auf die Kopierrichtung gedacht. Wollen Sie jedoch Text an einen Puffer anhängen, müssen Sie die Zeichen >> unbedingt mitangeben!
- Wenn Sie durch ein PICK-Kommando nur den Inhalt eines benannten PICK-Puffers zusätzlich in einen anderen benannten PICK-Puffer abspeichern, so wird dieser zuvor kopierte Text nicht nochmals in die Pseudo-Datei RESTORE aufgenommen.

**Beispiel 1**

<ENTER> ' pick2 <PICK>

Das Kommando kopiert den durch die Position der Schreibmarke gekennzeichneten BLOCK einschließlich der öffnenden und schließenden Klammern in den PICK-Puffer *pick2*.

**Beispiel 2**

<ENTER> 7 pick2 <PICK>

Das Kommando kopiert 7 Zeilen ab der aktuellen Zeile inklusive aktueller Zeile in den PICK-Puffer *pick2*. Der frühere Puffer-Inhalt wird überschrieben.

## PLUSFIND - Textstelle suchen

```

<PLUSFIND>
<ENTER> [ PATTERN ]
          / [ [ % ] [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] ]
          / [ RANGE/RECTANGLE/[BLOCK/RECORD]STREAM PATTERN ]
          / [ %[IDENTIFIER] ]
          / [ @ / / / ' ] <PLUSFIND>

```

Das Kommando PLUSFIND sucht in der aktuellen Datei vorwärts nach der ersten mit dem definierten Suchmuster übereinstimmenden Textstelle. Suchmuster geben Sie in Form eines PATTERNS oder einer Bereichsdefinition an (siehe Abschnitt 3.9 Die Bereichsdefinition als Kommandoattribut, 3.10 PATTERN als Kommandoattribut). Durch ein PATTERN können Sie mehrere alternative Suchmuster gleichzeitig definieren.

Wenn Sie kein Suchmuster angeben, greift das Kommando PLUSFIND auf das aktuelle PATTERN zurück. Das aktuelle PATTERN wird immer dann überschrieben, wenn Sie ein Suchkommando aufrufen, bei dem Sie ein PATTERN als Kommandoattribut angeben (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen).

Wenn eine zum definierten Suchmuster passende Textstelle gefunden ist, wird die Schreibmarke auf das erste Zeichen der gesuchten Textstelle positioniert. Gegebenenfalls verschiebt sich das Fenster, sodaß die Schreibmarke im aktuellen Fenster sichtbar bleibt. Wird keine übereinstimmende Textstelle bis zum Anfang der Datei gefunden, erscheint in der Steuerzeile eine der folgenden Meldungen:

```

**** PATTERN nicht gefunden.
**** RANGE nicht gefunden.
**** RECTANGLE nicht gefunden.
**** STREAM nicht gefunden.

```

Schreibmarke und Fenster bleiben dann in ihrer alten Position.

<PLUSFIND>

Sucht ab der aktuellen Position der Schreibmarke vorwärts nach der ersten Zeichenfolge, die mit dem aktuellen PATTERN übereinstimmt. Diese Eingabe ist nur als Wiederholungskommando geeignet.

<ENTER> <PLUSFIND>

Sucht vorwärts nach einer übereinstimmenden Zeichenfolge. Das Suchmuster ist dabei das aktuelle Wort, unter dem die Schreibmarke positioniert ist.

<ENTER> PATTERN <PLUSFIND>

Sucht vorwärts nach einer zu PATTERN passenden Zeichenfolge.

<ENTER> / <PLUSFIND>

Findet das Ende des durch die Position der Schreibmarke gekennzeichneten RECORD. Die Schreibmarke wird auf das Ende des RECORD positioniert. (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut: RECORD als Spezialfall eines STREAM-Bereichs).

<ENTER> ' <PLUSFIND>

Findet das Ende eines BLOCKs in einer verschachtelten Klammerstruktur. Die Schreibmarke wird auf das Ende des kleinsten umgebenden BLOCKs positioniert. Wenn notwendig, wird das Fenster verschoben (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut: BLOCK als Spezialfall eines STREAM-Bereichs).

#### *Anmerkung*

Dieses Kommando steht Ihnen standardmäßig als Tasten-Makro <PLUSBLOCK> zur Verfügung (siehe Abschnitt 6.2 Die Tasten-Makros).

<ENTER> bereich <PLUSFIND>

Sucht von der aktuellen Position der Schreibmarke aus vorwärts nach demjenigen Text, der identisch ist mit dem durch die Bereichsangabe gekennzeichneten Text. Der Bereich muß durch Bewegen der Schreibmarke definiert sein (siehe Abschnitt 3.9.1). Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

<ENTER> % bereich <PLUSFIND>

Sucht von der aktuellen Position der Schreibmarke aus vorwärts nach demjenigen Text, der identisch ist mit dem durch die Bereichsdefinition gekennzeichneten Text. Der Bereich muß durch ein Zeichenketten-Attribut angegeben sein (siehe Abschnitt 3.9.2). Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

*Anmerkung*

Das Prozentzeichen % vor der Bereichsangabe ist deshalb notwendig, weil der MAXed diese sonst als PATTERN interpretieren würde (siehe oben).

<ENTER> bereich PATTERN <PLUSFIND>

Sucht innerhalb des angegebenen Bereichs vorwärts nach PATTERN. Wenn Sie für PATTERN die leere Zeichenkette "" angeben, wird das aktuelle PATTERN verwendet. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

*Anmerkung*

Das Kommando unterscheidet sich von den vorangegangenen Kommandos dadurch, daß es innerhalb des Bereichs nach einem angegebenen PATTERN sucht!

<ENTER> %[IDENTIFIER] <PLUSFIND>

Sucht vorwärts nach einem Textbereich, der mit dem Inhalt des angegebenen PICK-Puffers identisch ist. Falls IDENTIFIER nicht angegeben ist, wird der Standard-PICK-Puffer verwendet (siehe Abschnitt 3.8.3 Der PICK-Puffername als IDENTIFIER).

*Anmerkung*

Bei dieser Form der Kommandoattributierung wird keine Textersetzung im Eingabefeld durchgeführt (siehe Abschnitt 3.6.4: Ersetzen von PICK-Puffernamen). Der Vorteil ist, daß der Textbereich des angegebenen PICK-Puffers nicht durch eine Textersetzung in einen einzeiligen STREAM umgewandelt und als PATTERN verwendet wird! Im Gegensatz dazu findet bei folgendem Kommando eine Textersetzung statt:

<ENTER> "%[IDENTIFIER]" <PLUSFIND>

<ENTER> @ <PLUSFIND>

Blendet die Pseudo-Datei FINDHISTORY mit der Liste aller PATTERN ein, die Sie im bisherigen Verlauf der Sitzung beim Aufruf der Kommandos MINUSFIND und PLUSFIND als Kommandoattribut angegeben haben. (siehe Abschnitt 4.2.9 Die Pseudo-Datei FINDHISTORY).

### Anmerkung

- Bei der Suche mit PATTERN werden Zeichenattribute nicht berücksichtigt.
- Bei der Suche mit einer Bereichsdefinition als Suchmuster werden gesetzte Zeichenattribute nur dann berücksichtigt, wenn die aktuelle Einstellung der Zeichenattribute aktiviert ist: Alle gesetzten Zeichenattribute in zwei zu vergleichenden Textstellen, die in der aktuellen Einstellung nicht mit Stern \* für "nicht berücksichtigen" angezeigt werden, müssen übereinstimmen (siehe Abschnitt 4.4 Die Zeichenattribute).

### Beispiel

<ENTER> , <PLUSFIND>

Der MAXed verwendet das aktuelle Wort als Suchmuster und sucht vorwärts nach einer identischen Textstelle.

## PLUSLINE - In der Datei vorblättern

<PLUSLINE>

Kommandoverstärkung:

<ENTER> *n* <MINUSLINE>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <PLUSLINE>

Das Kommando PLUSLINE schiebt den Dateiausschnitt im aktuellen Fenster um etwa ein Drittel seiner Höhe nach unten. Die aktuelle Spalte wird beibehalten. Befindet sich die ursprüngliche aktuelle Zeile noch innerhalb des Fensters, bleibt die Schreibmarke in dieser Zeile stehen. Anderenfalls bewegt sich die Schreibmarke um die gleiche Anzahl Zeilen nach unten, bleibt also im Fenster auf gleicher Höhe stehen (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<PLUSLINE>

Verschiebt den Dateiausschnitt um einige Zeilen nach unten.

<ENTER> <PLUSLINE>

Verschiebt den Dateiausschnitt so, daß die aktuelle Zeile zur ersten Zeile des Fensters wird.

<ENTER> *n* <PLUSLINE>

Verschiebt den Dateiausschnitt um *n* Zeilen nach unten.

**PLUSPAGE - Seitenweise vorblättern**

<PLUSPAGE>

Kommandoverstärkung:

<ENTER> n <PLUSPAGE>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <PLUSPAGE>

Das Kommando PLUSPAGE schiebt den Dateiausschnitt seitenweise nach unten. Die Größe einer Seite ist durch die Zahl der Zeilen im aktuellen Fensters bestimmt. Die Schreibmarke verändert ihre relative Position innerhalb des Fensters nicht. Die aktuelle Spalte wird beibehalten (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<PLUSPAGE>

Verschiebt den Dateiausschnitt und die Schreibmarke um eine Seite nach unten.

<ENTER> <PLUSPAGE>

Start der Definition eines RANGE- oder RECTANGLE-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Dateiausschnitt und Schreibmarke bewegen sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <PLUSPAGE>

Verschiebt den Dateiausschnitt und die Schreibmarke um  $n$  Seiten nach unten.

## PRINT - Text ausdrucken

```
<PRINT>  
<ENTER> RANGE <PRINT>
```

Das Kommando PRINT druckt die aktuell bearbeitete Datei ganz oder teilweise aus. Dabei verwendet es standardmäßig das Betriebssystem-Kommando *lpr*. Die Einstellung ändern Sie, wenn Sie der Umgebungsvariablen PRINT ein anderes Druckkommando als Wert zuweisen (siehe Abschnitt 4.5.4 Ausdrucken von MAXed-Dateien).

```
<PRINT>  
Druckt den Text des aktuellen Fensters aus.
```

```
<ENTER> <PRINT>  
Druckt den Text ab der aktuellen Position der Schreibmarke bis zum Ende des aktuellen Fensters.
```

```
<ENTER> RANGE <PRINT>  
Führt das PRINT-Kommando für die durch das RANGE-Argument definierten Zeilen aus.
```

### Anmerkung

- Wenn Sie auch Texte mit MAXed-Zeichenattributen oder nationalen Sonderzeichen ausdrucken wollen, müssen Sie diese für den Ausdruck vom Dienstprogramm *maxlpr* aufbereiten lassen (siehe Abschnitte 4.5.4 Ausdrucken von MAXed-Dateien, 7.4 MAXed-Dateien für den Ausdruck aufbereiten, 4.4 Die Zeichenattribute, Kapitel 6: FONT).

### Beispiel

```
<ENTER> . <PRINT>  
Druckt die gesamte aktuelle Datei.
```

## PUT - Zwischengespeicherte Textbereiche einsetzen

```
<PUT>
<ENTER> [ [IDENTIFIER] [ + / - / / / \ / | / ; ]
          / [ @ ]
          / [ "PATTERN" / 'zeichenkette' ] <PUT>
```

Kommandoverstärkung:

```
<ENTER> n [IDENTIFIER] [ + / - / / / \ / | / ; ] <PUT>
```

Das Kommando PUT kopiert den Inhalt eines PICK-Puffers in den Text des aktuellen Fensters. PICK-Puffer erstellen Sie beim Kopieren von Textbereichen mit dem Kommando PICK (siehe Abschnitte 3.8.3 Der PICK-Puffername als IDENTIFIER, 3.12.2 Kommandos mit Standard-Puffern, Kapitel 6: PICK).

Sie können Textbereiche kopieren, indem Sie diese mit Hilfe des Kommandos PICK abspeichern und mit PUT wieder in den Text einsetzen. Dabei wird der Textbereich abhängig vom aktuell eingestellten Einfüge-Modus an der aktuellen Schreibmarkenposition eingefügt, oder ein eventuell an dieser Stelle bereits vorhandener Text wird von dem Textbereich überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

PUT ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

```
<PUT>
```

Kopiert den Inhalt des Standard-PICK-Puffers in den Text ab der aktuellen Position der Schreibmarke in die Datei. Der Pufferinhalt wird im Einfüge-Modus vor der aktuellen Schreibmarkenposition eingefügt oder überschreibt im Überschreibe-Modus den unterliegenden Text ab der aktuellen Schreibmarkenposition.

```
<ENTER> n <PUT>
```

Kopiert den Inhalt des PICK-Puffers *n*-mal in den Text ab der aktuellen Position der Schreibmarke.

<ENTER> IDENTIFIER <PUT>

Kopiert den Inhalt des mit IDENTIFIER benannten PICK-Puffers in den Text ab der aktuellen Position der Schreibmarke bzw. blendet die Pseudo-Datei PICKBUFFERS mit der Liste aller passenden PICK-Puffernamen im aktuellen Fenster ein (siehe Abschnitt 4.2.7 Die Pseudo-Datei PICKBUFFERS).

Falls der MAXed keinen passenden PICK-Puffernamen findet, sucht er nach einem passenden Dateinamen! In diesem Fall kopiert er den Inhalt der durch IDENTIFIER angegebenen Datei an die aktuelle Position der Schreibmarke bzw. blendet die Pseudo-Datei PUTFILES mit der Liste aller passenden Dateinamen im aktuellen Fenster ein (siehe Abschnitt 4.2.5 Die Pseudo-Datei PUTFILES).

<ENTER> n IDENTIFIER <PUT>

Kopiert den Inhalt des mit IDENTIFIER benannten PICK-Puffers *n*-mal in Text ab der aktuellen Position der Schreibmarke.

<ENTER> [n] [IDENTIFIER] + <PUT>

Fügt unabhängig vom eingestellten Eingabe-Modus und vom Bereichstyp des PICK-Puffers den Zeilenbereich, über den sich der Pufferinhalt erstreckt, vor der aktuellen Zeile ein. Füllt danach den Zeilenbereich *n*-mal mit dem Pufferinhalt auf.

<ENTER> [n] [IDENTIFIER] - <PUT>

Überschreibt unabhängig vom eingestellten Eingabe-Modus und vom Bereichstyp des PICK-Puffers den Zeilenbereich, über den sich der Pufferinhalt erstreckt, ab der aktuellen Zeile mit Leerzeilen. Füllt danach den Zeilenbereich *n*-mal mit dem Pufferinhalt auf.

<ENTER> [n] [IDENTIFIER] / <PUT>

Überlagert den vorhandenen Text mit dem Inhalt des PICK-Puffers. Dabei wird der bisherige Text überschrieben. Leerzeichen im PICK-Puffer lassen vorhandenen Text jedoch unverändert. Das Kommando dient der Erzeugung von Mustern und mehrspaltigem Text. Diese Funktion ist nur auf Puffer, in denen ein RANGE oder RECTANGLE steht, anwendbar.

<ENTER> [n] [IDENTIFIER] \ <PUT>

Unterlegt den vorhandenen Text mit dem Inhalt des PICK-Puffers. Dabei werden nur die Leerzeichen des vorhandenen Textes durch den Inhalt des PICK-Puffers überschrieben. Die Zeichen des bisherigen Textes bleiben erhalten. Das Kommando dient der Erzeugung von Mustern und mehrspaltigen Texten. Diese Funktion ist nur auf Puffer, in denen ein RANGE oder RECTANGLE steht, anwendbar.

<ENTER> [n] [IDENTIFIER] <PUT>

Entspricht dem Kommando

<ENTER> [n] [IDENTIFIER] <PUT>.

Allerdings ist der Text des Pufferinhalts unabhängig davon, ob im PICK-Puffer ein STREAM, RANGE oder RECTANGLE steht, an der aktuellen Spaltenposition der Schreibmarke ausgerichtet. Die mehrfach kopierten Textbereiche beginnen immer in einer neuen Zeile.

<ENTER> [n] [IDENTIFIER] ; <PUT>

Die Arbeitsweise dieses Kommandos ist je nach Bereichstyp des PICK-Puffers unterschiedlich:

- Enthält der PICK-Puffer ein RECTANGLE, dann wird dieses rechts im Anschluß an den vorhandenen Text angehängt. Ist  $n$  größer als 1, so werden die RECTANGLES rechts nebeneinander kopiert.
- Enthält der PICK-Puffer ein RANGE, dann wird dieses wie bei einem RECTANGLE rechts im Anschluß an den vorhandenen Text angehängt. Ist  $n$  größer als 1, so werden die RANGES rechts nebeneinander kopiert.
- Es sind nur einzeilige STREAMs zulässig. Enthält der PICK-Puffer einen einzeiligen STREAM, dann wird dieser wie bei einem RECTANGLE rechts im Anschluß an den vorhandenen Text angehängt. Ist  $n$  größer als 1, so werden die STREAMs rechts nebeneinander kopiert.

<ENTER> @ <PUT>

Blendet die Pseudo-Datei PICKBUFFERS mit der Liste aller momentan definierten benannten PICK-Puffer ein (siehe Abschnitt 4.2.7 Die Pseudo-Datei PICKBUFFERS).

<ENTER> "PATTERN" <PUT>

Kopiert ein zu PATTERN passendes Wort aus der aktuellen Datei an die aktuelle Position der Schreibmarke. Dabei muß das PATTERN Sonderzeichen enthalten, die nicht entwertet sind. Der MAXed sucht zunächst in der aktuellen Datei rückwärts nach der nächsten zu PATTERN passenden Textstelle. Bei erfolgloser Suche probiert er es nochmals in Vorwärtsrichtung. Die übereinstimmende Zeichenkette wird dann bis zu ihren Wortgrenzen erweitert und an die aktuellen Position der Schreibmarke in den Text kopiert.

*Anmerkung*

- Für den weiteren Gebrauch durch PUT-Kommandos wird das Wort in den Standard-PICK-Puffer aufgenommen. Außerdem wird das aktuelle PATTERN mit dem angegebenen PATTERN belegt, sodaß es für weitere Such- und Ersetzungskommandos verfügbar ist (siehe Abschnitt 3.10 PATTERN als Kommandoattribut).
- Das PATTERN müssen Sie immer in Anführungszeichen setzen. Es darf jedoch keine Leerzeichen enthalten, da ja nur im Wortkontext gesucht wird! Dieses Kommando ist insbesondere beim Schreiben von Quell-Programmen nützlich, um Schreibfehler beim Einfügen eines vorher definierten IDENTIFIER zu vermeiden.
- Wenn das PATTERN nur eine Zeichenkette im Klartext darstellt, entspricht dieses Kommando dem nachfolgend beschriebenen.

<ENTER> 'zeichenkette' <PUT>

Kopiert die angegebene Zeichenkette an die aktuelle Position der Schreibmarke. Die Zeichenkette müssen Sie hierbei immer in Hochkommata setzen. Sie kann beliebig sein und darf auch Leerzeichen enthalten (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

**Beispiel 1**

In der Datei wurde ein IDENTIFIER *xxTextbaustein* definiert. Mit dem Kommando

```
<ENTER> "x{2}>" <PUT>
```

wird dann ab der aktuellen Schreibmarken-Position (zuerst rückwärts, dann vorwärts) die Datei nach dem nächsten Vorkommen von *xx* durchsucht. Da die nächste (und vermutlich einzige) passende Zeichenkette das Wort *xxTextbaustein* ist, wird das PATTERN *x{2}>* zu diesem Wort erweitert und dieses an der aktuellen Position der Schreibmarke eingesetzt.

**Beispiel 2**

Mit folgendem Kommando weisen Sie der Variablen *ORT* die Zeichenkette *München* zu:

```
<ENTER> ORT=München <ENTER>
```

Mit dem Kommando

```
<ENTER> '$ORT, den @d' <PUT>
```

könnten Sie nun stets in Briefen Ort und Datum einfügen, ohne sich um das aktuelle Tagesdatum kümmern zu müssen.

## QUIT - Beenden des MAXed ohne Sichern

<QUIT>

Mit dem Kommando QUIT beenden Sie die aktuelle MAXed-Sitzung. Der MAXed sichert die geöffneten und geänderten Dateien nicht auf der Festplatte ab. Alle Änderungen der aktuellen Sitzung gehen verloren, wenn Sie sie nicht zuvor mit Hilfe des Kommandos SAVE abgespeichert haben (siehe Abschnitt 3.2 Starten und Beenden des MAXed, 3.3.1 Einblenden und Sichern einer Datei, Kapitel 6: EXIT, SAVE). <QUIT> ist nicht auf der Tastatur der Datensichtstation konfiguriert.

<ENTER> quit <ENTER>

Beendet die aktuelle MAXed-Sitzung ohne Abspeichern der geöffneten Dateien.

### Anmerkung

Das Kommando QUIT ist identisch mit dem Kommando  
<ENTER> a <EXIT>

## **REFRESH - Den Bildschirm neu schreiben**

**<REFRESH>**  
**<ENTER> <REFRESH>**

Das Kommando REFRESH schreibt den gesamten MAXed-Bildschirm neu. Fehlerhafte Zeichendarstellungen durch Übertragungsfehler oder Systemmeldungen werden durch das Kommando gelöscht. Sie können das Kommando auch verwenden, um die Bildschirmausgabe bei Makros oder MAXed-Steuerprogrammen ein- bzw. auszuschalten (siehe Abschnitte 7.1 Eine MACRO-Bibliothek erstellen, 7.5 MAXed-Steuerprogramme).

**<REFRESH>**

Schreibt den gesamten MAXed-Bildschirm neu. Bei Übertragungs- oder Bildschirmfehlern werden die fehlerhaften Zeichendarstellungen beseitigt. Innerhalb von Makros oder MAXed-Steuerprogrammen hebt dieses Kommando das unten beschriebene Kommando auf.

**<ENTER> <REFRESH>**

Schaltet die Ausgabe auf den Bildschirm aus. Dies ist jedoch nur als Kommando innerhalb eines Makros oder eines MAXed-Steuerprogramms anwendbar. Die Ausgabe auf den Bildschirm kann durch ein <REFRESH> wieder eingeschaltet werden. Bei Beendigung des Makros oder des Steuerprogramms wird die Ausgabe auf den Bildschirm automatisch wieder eingeschaltet und der Bildschirm aktualisiert.

## REPLACE - Ersetzen von Textstellen

```
<REPLACE>
<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM]
          [[PATTERN] ersetzungszeichenfolge] <REPLACE>

<ENTER> [PATTERN] ersetzungszeichenfolge
          [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] <REPLACE>
```

Das Kommando REPLACE dient zum Suchen und Ersetzen von Zeichenketten. Eine zum verwendeten PATTERN passende Zeichenkette wird durch die Ersetzungszeichenkette ersetzt (siehe Abschnitte 3.10 PATTERN als Kommandoattribut, 3.11 Die Ersetzungszeichenkette als Kommandoattribut).

Wenn Sie kein PATTERN angeben, greift das Kommando REPLACE auf das aktuelle PATTERN zurück. Das aktuelle PATTERN wird jedesmal überschrieben, wenn Sie ein Suchkommando aufrufen, bei dem Sie ein PATTERN als Kommandoattribut angeben (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen).

Wird die zu ersetzende Zeichenkette nicht gefunden, erscheint die Meldung

```
**** Nichts passt.
```

REPLACE ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

```
<REPLACE>
  Ersetzt die nächste passende Textstelle im aktuellen Zeilenrest. Dazu werden das aktuelle PATTERN und die zuletzt eingegebene Ersetzungszeichenkette verwendet (siehe Abschnitte 3.12.1 Kommandos mit konstanter Standard-Ausführung, 3.12.3 Kommandos mit aktuell gespeicherten Attributen). Diese Eingabe ist nur als Wiederholungskommando geeignet.
```

```
<ENTER> [bereich] [[PATTERN] ersetzungszeichenfolge] <REPLACE>
bzw.
```

```
<ENTER> [PATTERN] ersetzungszeichenfolge [bereich] <REPLACE>
  Ersetzt in dem angegebenen Bereich jede Zeichenkette, die zum verwendeten PATTERN paßt, durch die ersetzungszeichenfolge. Für
```

*bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

Bei der Eingabe der Kommandoattribute ist folgendes zu beachten:

- Wenn Sie den Bereich durch Bewegen der Schreibmarke definieren, legen Sie ihn wie üblich zu Beginn der Kommandoattributierung fest (siehe Abschnitt 3.9.1).
- Wenn Sie den Bereich durch ein Zeichenketten-Attribut festlegen, so müssen Sie ihn bei diesem Kommando als letztes von drei Kommandoattributen eingeben (siehe Abschnitt 3.9.2)!
- Fehlt die Angabe *bereich*, wird die aktuelle Zeile ab der Position der Schreibmarke angenommen (siehe Abschnitt 3.12.1 Kommandos mit konstanter Standard-Ausführung).
- Ein einzelnes Zeichenkettenattribut wird prinzipiell als Ersetzungszeichenkette betrachtet!
- Für PATTERN bzw. *ersetzungszeichenfolge* wird das aktuelle PATTERN bzw. die zuletzt eingegebene Ersetzungszeichenkette verwendet (siehe Abschnitt 3.12.3 Kommandos mit aktuell gespeicherten Kommandoattributen).

### Anmerkung

- Wollen Sie die einzelnen Ersetzungsoperationen vor der tatsächlichen Ersetzung jeweils überprüfen, verwenden Sie am besten das Kommandopaar  
`<ENTER> PATTERN <FIND-KOMMANDO>`  
`<ENTER> ersetzungszeichenkette <REPLACE>`  
 Die nachfolgenden Ersetzungen erreichen Sie durch Wiederholen von  
`<FIND-KOMMANDO> <REPLACE>`  
 (siehe Kapitel 6: MINUSFIND, PLUSFIND).
- Wollen Sie nur das PATTERN ändern, den Ersetzungsbegriff aber beibehalten, dann geben Sie am besten folgendes Kommando:  
`<ENTER> [bereich] PATTERN @r <REPLACE>`  
 Der Parameter für Textersetzung wird zur zuletzt verwendeten Ersetzungszeichenkette expandiert (siehe Abschnitt 3.6.4 Textersetzung im Eingabefeld).

**Beispiel 1**

```
<ENTER> maxed MAXed . <REPLACE>
```

Das Kommando ersetzt in der gesamten aktuellen Datei alle zum PATTERN *maxed* passenden Zeichenketten, also z.B. *maxed*, *axed*, *mAxed* oder *MAXed* durch die Ersetzungszeichenfolge *MAXed*.

**Beispiel 2**

```
<ENTER> ^; | ;-. <REPLACE>
```

Alle Leerzeilen zwischen aktueller Zeile und Dateiende werden gelöscht.

**Beispiel 3**

```
<ENTER> ^.{2}Kap\.<.* A.{3d,0,10,}\neu . <REPLACE>
```

In der aktuellen Datei werden die Kapitelüberschriften neu numeriert: Alle Zeilen, die ab der 3. Spalte einer Zeile mit *Kap.* beginnen, werden gefunden. Dann wird der Rest dieser Zeilen (der beliebig aussehen kann) ersetzt durch *A.zähler\neu*, wobei zähler einen drei Zeichen großen Platz darstellt, der für die Numerierung freigehalten wird. Diese beginnt bei 0, wird immer um 10 erhöht und rechtsbündig eingetragen. Folgende gefundene Textzeilen

```
Kap. 3 (Stückwerk aus Version 1.0)
```

```
...
```

```
kap. ?? (alles neu)
```

```
...
```

```
KAP. 586\alte Version
```

```
...
```

werden durch obiges REPLACE-Kommando verändert zu

```
Kap.A 0\neu
```

```
...
```

```
kap.A 10\neu
```

```
...
```

```
KAP.A 20\neu
```

```
...
```

**Beispiel 4**

Der Inhalt der aktuellen Datei ist

```
test
test
tist
tost
test
tust
tust
```

Die Eingaben

```
<ENTER> t.*t {d,1,1,!} . <REPLACE>
<UNDO>
<ENTER> t.*t {d,1,1,=} . <REPLACE>
<UNDO>
<ENTER> t.*t {d,1,1} . <REPLACE>
```

liefern nacheinander

1	1	1
1	2	2
2	1	3
3	1	4
1	3	5
4	1	6
4	2	7

## RESTOR - Gelöschte Textbereiche wieder einsetzen

```
<RESTOR>
<ENTER> [ [IDENTIFIER] [ + / - / / \ ]
          / [ @ ] <RESTOR>
```

Kommandoverstärkung:

```
<ENTER> n [IDENTIFIER] [ + / - / / \ ] <RESTOR>
```

Das Kommando RESTOR kopiert den Inhalt eines ERASE-Puffers in den Text des aktuellen Fensters. ERASE-Puffer erstellen Sie beim Löschen von Textbereichen mit dem Kommando ERASE (siehe Abschnitte 3.8.2 Der ERASE-Puffername als IDENTIFIER, 3.12.2 Kommandos mit Standard-Puffern, Kapitel 6: ERASE).

Sie können Textbereiche verschieben, indem Sie diese mit Hilfe des Kommandos ERASE ausschneiden (löschen) und mit RESTOR wieder in den Text einsetzen (kopieren). Dabei wird der Textbereich abhängig vom aktuell eingestellten Einfüge-Modus an der aktuellen Schreibmarkenposition eingefügt, oder ein eventuell an dieser Stelle bereits vorhandener Text wird von dem Textbereich überschrieben (siehe Abschnitt 4.1.1 Die Einfüge-Modi).

RESTOR ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

```
<RESTOR>
```

Kopiert den Inhalt des Standard-ERASE-Puffers in den Text ab der aktuellen Position der Schreibmarke in die Datei. Der Pufferinhalt wird im Einfüge-Modus vor der aktuellen Schreibmarkenposition eingefügt oder überschreibt im Überschreibe-Modus den unterliegenden Text ab der aktuellen Schreibmarkenposition.

```
<ENTER> n <RESTOR>
```

Kopiert den Inhalt des Standard-ERASE-Puffers *n*-mal in den Text ab der aktuellen Position der Schreibmarke.

<ENTER> IDENTIFIER <RESTOR>

Kopiert den Inhalt des mit IDENTIFIER benannten ERASE-Puffers in den Text ab der aktuellen Position der Schreibmarke bzw. blendet die Pseudo-Datei ERASEBUFFERS mit der Liste aller passenden ERASE-Puffernamen im aktuellen Fenster ein (siehe Abschnitt 4.2.6 Die Pseudo-Datei ERASEBUFFERS).

<ENTER> n IDENTIFIER <RESTOR>

Kopiert den Inhalt des mit IDENTIFIER benannten ERASE-Puffers *n*-mal in den Text ab der aktuellen Position der Schreibmarke.

<ENTER> [n] [IDENTIFIER] + <RESTOR>

Fügt unabhängig vom eingestellten Eingabe-Modus und vom Bereichstyp des ERASE-Puffers den Zeilenbereich, über den sich der Pufferinhalt erstreckt, vor der aktuellen Zeile ein. Füllt danach den Zeilenbereich *n*-mal mit dem Pufferinhalt auf.

<ENTER> [n] [IDENTIFIER] - <RESTOR>

Überschreibt unabhängig vom eingestellten Eingabe-Modus und vom Bereichstyp des ERASE-Puffers den Zeilenbereich, über den sich der Pufferinhalt erstreckt, ab der aktuellen Zeile mit Leerzeilen. Füllt danach den Zeilenbereich *n*-mal mit dem Pufferinhalt auf.

<ENTER> [n] [IDENTIFIER] / <RESTOR>

Überlagert den vorhandenen Text mit dem Inhalt des ERASE-Puffers. Dabei wird der bisherige Text überschrieben. Leerzeichen im ERASE-Puffer lassen vorhandenen Text jedoch unverändert. Das Kommando dient der Erzeugung von Mustern und mehrspaltigem Text. Diese Funktion ist nur auf Puffer, in denen ein RANGE oder RECTANGLE steht, anwendbar.

<ENTER> [n] [IDENTIFIER] \ <RESTOR>

Unterlegt den vorhandenen Text mit dem Inhalt des ERASE-Puffers. Dabei werden nur die Leerzeichen des vorhandenen Textes durch den Inhalt des ERASE-Puffers überschrieben. Die Zeichen des bisherigen Textes bleiben erhalten. Das Kommando dient der Erzeugung von Mustern und mehrspaltigen Texten. Diese Funktion ist nur auf Puffer, in denen ein RANGE oder RECTANGLE steht, anwendbar.

<ENTER> @ <RESTOR>

Blendet die Pseudo-Datei ERASEBUFFERS mit der Liste aller momentan definierten benannten ERASE-Puffer ein (siehe Abschnitt 4.2.6 Die Pseudo-Datei ERASEBUFFERS).

**RETURN - Zeilenschaltung**

```
<RETURN>
<ENTER> [ n / s / e / t ]
          / [ IDENTIFIER ] <RETURN>
```

Kommandoverstärkung:

```
<ENTER> n <RETURN>
```

Start einer Bereichsdefinition für STREAM:

```
<ENTER> <RETURN>
```

Das Kommando RETURN bewegt die Schreibmarke in die nächste Zeile. Im standardmäßig eingestellten Zeilenschaltungs-Modus wird die Schreibmarke dabei an die Spalte des linken Formatrandes gesetzt (siehe Abschnitt 4.3.2 Formatränder, Kapitel 6: MARGIN). Sie können den Zeilenschaltungs-Modus jedoch auch so einstellen, daß die Schreibmarke stattdessen an die Spalte springt, an der der Text der voranstehenden Zeile beginnt (siehe Abschnitt 4.1.3 Der Zeilenschaltungs-Modus).

Außerdem bestimmt der Zeilenschaltungs-Modus, ob die Schreibmarke beim Sprung auf eine zusätzlich eingesetzte Leerzeile nach der aktuellen Zeile gesetzt wird; das bedeutet im Detail das Einfügen einer zusätzlichen Leerzeile oder das Überschreiben der nächsten Zeile mit Leerzeichen (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Befindet sich die Schreibmarke vor Ausführung von RETURN in der letzten Zeile des aktuellen Fensters, wird zunächst ein PLUSLINE-Kommando ausgeführt (siehe Kapitel 6: PLUSLINE).

Ist ein rechter Formatrand festgelegt und befindet sich die Schreibmarke vor Ausführung von RETURN rechts davon, wird ein automatischer Zeilenumbruch durchgeführt (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

Eine geeignete Einstellung erlaubt Ihnen also eine kontinuierliche Texteingabe, ohne das Fenster, die Formatgrenzen oder den nachfolgenden Text beachten zu müssen.

```
<RETURN>
```

Bewegt die Schreibmarke entsprechend dem eingestellten Modus. Standard ist der NORMAL-Modus (siehe unten).

<ENTER> <RETURN>

Start der Definition eines STREAM-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <RETURN>

Führt das Kommando <RETURN> entsprechend dem eingestellten Zeilenschaltungs-Modus *n*-mal aus.

<ENTER> s[pace] <RETURN>

Schaltet in den Zeilenschaltungs-Modus SPACE um. Vor dem Sprung der Schreibmarke in die neue Zeile wird nach der aktuellen Zeile eine zusätzliche Leerzeile eingesetzt. Dadurch können Sie auf unkomplizierte Weise innerhalb von Texten etwas einfügen oder ändern.

<ENTER> e[mpy] <RETURN>

Schaltet in den Zeilenschaltungs-Modus EMPTY um. Dieser Modus ähnelt dem SPACE-Modus; es wird allerdings nur dann eine Leerzeile eingefügt, wenn der aktuellen Zeile keine Leerzeile folgt.

<ENTER> t[ext] <RETURN>

Schaltet in den Zeilenschaltungs-Modus TEXT um. Die Schreibmarke wird in die nächste Zeile zur ersten nicht leeren Spalte bewegt. Wenn die nächste Zeile leer ist, springt die Schreibmarke dort in die Spalte, in der das erste nicht leere Zeichen der aktuellen Zeile steht, höchstens aber an den linken Formatrand.

<ENTER> n[ormal] <RETURN>

Schaltet in den Zeilenschaltungs-Modus NORMAL um. Die Schreibmarke wird in der nächsten Zeile an den linken Formatrand bewegt. Dies ist die Standard-Einstellung.

<ENTER> IDENTIFIER <RETURN>

Entspricht dem Kommando

<ENTER> IDENTIFIER <EDIT>

(siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER, Kapitel 6: EDIT). IDENTIFIER kann jede zulässige Eingabe für einen Dateinamen sein mit Ausnahme der oben genannten Zeichenketten zum Einstellen des Zeilenschaltungs-Modus.

**Beispiel**

<ENTER> e t <RETURN>

Das Kommando bewegt die Schreibmarke in der nächsten, ggf. neu einzufügenden Leerzeile auf die Spaltenposition, ab der in der aktuellen Zeile der Text beginnt.

**RIGHTARROW - Die Schreibmarke nach rechts bewegen**

```
<RIGHTARROW>
<ENTER> n / h / v / m / c <RIGHTARROW>
```

Kommandoverstärkung:  
 <ENTER> n <RIGHTARROW>

Start einer Bereichsdefinition für STREAM:  
 <ENTER> <RIGHTARROW>

Das Kommando RIGHTARROW bewegt die Schreibmarke um eine Spalte nach rechts. Im standardmäßig eingestellten Fenster-Modus springt die Schreibmarke auf den gegenüberliegenden Fensterrand um, wenn Sie sie auf den Rand des aktuellen Fensters bewegen. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Sie können den Fenster-Modus jedoch auch so einstellen, daß die Schreibmarke stattdessen den Dateiausschnitt verschiebt oder in ein angrenzendes Fenster überwechselt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

```
<RIGHTARROW>
  Bewegt die Schreibmarke um eine Spalte nach rechts.
```

```
<ENTER> <RIGHTARROW>
  Start der Definition eines STREAM-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).
```

```
<ENTER> n <RIGHTARROW>
  Bewegt die Schreibmarke um n Spalten nach rechts.
```

```
<ENTER> modus <RIGHTARROW>
  Damit stellen Sie den aktuellen Fenster-Modus ein (siehe Abschnitt 4.1.4 Der Fenster-Modus). Davon betroffen sind alle ARROW- und TAB-Kommandos beim Bewegen der Schreibmarke auf bzw. über den Rand des aktuellen Fensters. Für modus können Sie folgendes eingeben:
```

n[ormal] Die Schreibmarke erhält für alle vier Richtungen die Eigenschaft, zum gegenüberliegenden Fensterrand umzuspringen. Dieser Fenster-Modus ist standardmäßig eingestellt.

h[orizontal] Für horizontale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando oder durch Texteingabe den rechten bzw. linken Fensterrand überschreitet. Am Zeilenanfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.

v[ertical] Für vertikale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando den oberen bzw. unteren Fensterrand überschreitet. Am Dateianfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.

m[ove] Kombiniert die Einstellungen *horizontal* und *vertikal*. Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt.

c[hangewindow] Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt. Stattdessen wechselt die Schreibmarke (falls möglich) über den Rand des aktuellen Fensters in das benachbarte Fenster, das zum aktuellen Fenster wird (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten). Beim Überschreiten des äußersten Fensterrandes springt die Schreibmarke allerdings nach wie vor um.

### Anmerkung

Mit <RIGHTARROW> können Sie die Schreibmarke auch im Eingabefeld der Steuerzeile nach rechts verschieben (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Dabei ist ein Umspringen oder Wechseln des Fensters nicht möglich. Der sichtbare Ausschnitt des Eingabefeldes wird jedoch um eine Ausschnittslänge mitverschoben, wenn die Schreibmarke den rechten Rand des Eingabefeldes erreicht und das Ende der eingegebenen Zeichenkette nicht sichtbar ist.

## RVLINEFEED - Zeilenweises Verschieben des Dateiausschnitts

<RVLINEFEED>

Kommandoverstärkung:

<ENTER> n <RVLINEFEED>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:

<ENTER> <RVLINEFEED>

Das Kommando RVLINEFEED verschiebt den Dateiausschnitt im aktuellen Fenster und bewegt gleichzeitig die Schreibmarke um eine Zeile nach oben. Die aktuelle Spalte bleibt dieselbe. Die Schreibmarke verändert also ihre relative Position innerhalb des Fensters nicht. Falls das Fenster den Anfang der Datei zeigt, wird der Dateiausschnitt nicht verschoben, sondern lediglich die Schreibmarke um eine Zeile nach oben bis zur ersten Zeile bewegt (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<RVLINEFEED>

Verschiebt den Dateiausschnitt und die Schreibmarke um eine Zeile nach oben.

<ENTER> <RVLINEFEED>

Verschiebt den Dateiausschnitt um ein Drittel seiner Höhe nach oben. Dies entspricht dem Kommando <MINUSLINE>. Allerdings wird die Schreibmarke ebenso weit nach oben bewegt. Sie bleibt also im Fenster auf gleicher Höhe stehen.

<ENTER> n <RVLINEFEED>

Verschiebt den Dateiausschnitt und die Schreibmarke um *n* Zeilen nach oben.

### *Anmerkung*

Dieses Kommando ist besonders in Makros nützlich. Sie entgehen der Gefahr des Umspringens der Schreibmarke am oberen Fensterrand, indem Sie das Fenster mitbewegen (siehe Abschnitt 4.1.4 Der Fenster-Modus).

## SAVE - Eine Datei sichern

```
<SAVE>
<ENTER> [ [ = / + ] [mode] IDENTIFIER ]
          / [ [ = / + ] [mode] #IDENTIFIER ]
          / [ # / @ ] <SAVE>
```

Das Kommando SAVE wird in der Regel benutzt, um während der laufenden MAXed-Sitzung bearbeitete Dateien zu sichern. Falls Sie eine Datei bearbeiten, die bereits vor der aktuellen MAXed-Sitzung existierte, wird diese ursprüngliche Version beim Absichern der aktuellen Version mit dem Kommando SAVE gleichzeitig in eine sogenannte *.bak*-Datei geschrieben (siehe Abschnitt 3.3.1 Einblenden und Sichern einer Datei).

<SAVE>

Sichert die Datei des aktuellen Fensters mit allen Veränderungen unter ihrem Namen ab, indem diese neue Version der Datei auf die Festplatte geschrieben wird. Die alte Dateiversion *datei* wird in einer Datei *datei.bak* gesichert.

<ENTER> <SAVE>

Sichert alle Dateien, die seit Anfang der Sitzung oder seit dem letzten Sichern verändert wurden wie oben beschrieben.

<ENTER> IDENTIFIER <SAVE>

Sichert die aktuelle Datei unter durch IDENTIFIER angegebenen Namen ab. Existiert bereits eine Datei mit diesem Namen, erscheint zuerst eine entsprechende Warnung. Nochmaliges Eingeben von <SAVE> überschreibt diese Datei.

<ENTER> = IDENTIFIER <SAVE>

Sichert die aktuelle Datei unter durch IDENTIFIER angegebenen Namen nur dann ab, falls es eine Datei dieses Namens bereits gibt. Existiert sie nicht, wird eine entsprechende Meldung ausgegeben und es erfolgt keine Sicherung.

<ENTER> + IDENTIFIER <SAVE>

Sichert die aktuelle Datei unter durch IDENTIFIER angegebenen Namen nur dann ab, falls es eine Datei dieses Namens noch nicht gibt. Existiert sie bereits, wird eine entsprechende Meldung ausgegeben und es erfolgt keine Sicherung.

<ENTER> mode IDENTIFIER <SAVE>

Sichert die aktuelle Datei mit den in *mode* (oktal) angegebenen Zugriffsrechten unter dem durch IDENTIFIER angegebenen Namen ab (siehe SINIX-Manual 1).

<ENTER> # <SAVE>

Sichert alle aktuell definierten und benannten Makros und alle benannten PICK-Puffer in der Standard-MACRO-Bibliothek. Diese wird dabei neu angelegt (siehe Abschnitte 4.6 Arbeiten mit Textbausteinen, 4.7 Arbeiten mit Makros, 5.1 maxed – Den MAXed aufrufen).

<ENTER> # IDENTIFIER <SAVE>

Speichert alle aktuell definierten und benannten Makros und alle benannten PICK-Puffer (Textbausteine), in der mit IDENTIFIER angegebenen MACRO-Bibliothek ab. Existiert bereits eine Datei gleichen Namens, erscheint zuerst eine entsprechende Warnung. Nochmaliges Eingeben von <SAVE> überschreibt diese Datei (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen). Für MACRO-Bibliotheksnamen sind keine Sonderzeichen zulässig.

<ENTER> = # IDENTIFIER <SAVE>

Entspricht dem Kommando

<ENTER> # IDENTIFIER <SAVE>

Allerdings funktioniert das Abspeicherungsverfahren für Makros und Textbausteine analog zu

<ENTER> = IDENTIFIER <SAVE>

<ENTER> + # IDENTIFIER <SAVE>

Entspricht dem Kommando

<ENTER> # IDENTIFIER <SAVE>

Allerdings funktioniert das Abspeicherungsverfahren für Makros und Textbausteine analog zu

<ENTER> + IDENTIFIER <SAVE>

<ENTER> mode # IDENTIFIER <SAVE>

Entspricht dem Kommando

<ENTER> # IDENTIFIER <SAVE>

Allerdings funktioniert das Abspeicherungsverfahren für Makros und Textbausteine analog zu

<ENTER> mode IDENTIFIER <SAVE>

<ENTER> @ <SAVE>

Blendet die Pseudo-Datei `OPENFILES` mit einer Liste der von MAXed während der aktuellen Sitzung eröffneten Dateien ein, aus der Sie Dateien zum Abspeichern auswählen können (siehe Abschnitt 4.2.3 Die Pseudo-Datei `OPENFILES`).

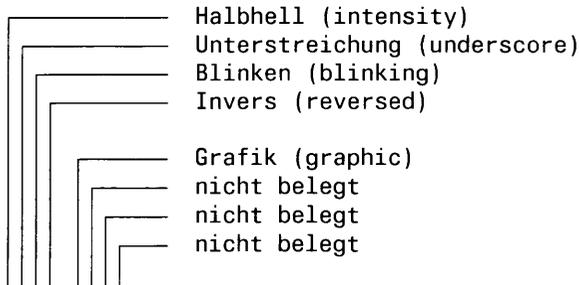
## SETATTRIB - Zeichenattribute setzen/löschen

```

<SETATTRIB>
<ENTER> [ @ ]
          / [ [setze-attribut] [ , [lösche-attribut] ] ]
          <SETATTRIB>

```

Mit Hilfe des Kommandos SETATTRIB ändern Sie die Einstellung der Zeichenattribute und aktivieren bzw. deaktivieren die aktuelle Einstellung. Damit können Sie Text z.B. invers oder unterstrichen oder als Grafikzeichen darstellen (siehe Abschnitt 4.4 Die Zeichenattribute). Die verfügbaren Zeichenattribute sind jeweils einer Position des ATTRIBUTE-Feldes zugeordnet (siehe Abschnitt 3.5.2 Die Steuerzeile)::



ATT: iubr gxxx

Jede dieser Positionen des ATTRIBUTE-Feldes kann mit einer der folgenden Einstellungen belegt sein:

- 0 bedeutet, das zugeordnete Zeichenattribut ist "nicht-gesetzt".
- 1 bedeutet, das zugeordnete Zeichenattribut ist "gesetzt".
- \* bedeutet, das zugeordnete Zeichenattribut wird "nicht-berücksichtigt".

Wenn Sie die aktuelle Einstellung der Zeichenattribute aktiviert haben, zeigt Ihnen der MAXed die Belegung der einzelnen Zeichenattribute im ATTRIBUTE-Feld in der Steuerzeile an.

Jedes Zeichen, das Sie bei aktivierter Einstellung eingeben, erscheint mit den gesetzten Zeichenattributen. Außerdem werden die Kommandos ATTRIBUTE, COMPARE und DRAW beeinflusst (siehe Kapitel 6: ATTRIBUTE, COMPARE, DRAW).

<SETATTRIB>

Aktiviert bzw. deaktiviert die aktuelle Einstellung der Zeichenattribute.

<ENTER> <SETATTRIB>

Übernimmt die Zeichenattribute, die an der aktuellen Schreibmarkenposition vorhanden sind für die aktuelle Einstellung der Zeichenattribute und aktiviert diese gleichzeitig:

- vorhandene Zeichenattribute werden in der aktuellen Einstellung auf "gesetzt" (1) gesetzt.
- alle anderen Zeichenattribute werden in der aktuellen Einstellung auf "nicht-gesetzt" (0) gesetzt.

<ENTER> @ <SETATTRIB>

Setzt die Schreibmarke auf das ATTRIBUTE-Feld in der MAXed-Steuerezeile mit den aktuellen Zeichenattribut-Einstellungen. Sie können hier die Voreinstellung der Zeichenattribute ändern (siehe Abschnitt 4.4.1 Einstellen und Aktivieren von Zeichenattributen). Danach können Sie die aktuelle Einstellung abspeichern:

- Mit Betätigen von <SETATTRIB> speichern Sie die Änderungen ab. Wenn Sie die Schreibmarke dabei nicht zwischen den beiden Vierer-Blöcken des ATTRIBUTE-Feldes positioniert haben, wird diese aktuelle Einstellung gleichzeitig aktiviert.
- Mit <ENTER> brechen Sie die Änderung ab. Die Einstellung bleibt in ihrem vorigen Zustand.

<ENTER> [setze-attribute] [, [lösche-attribute]] <SETATTRIB>

Verändert und aktiviert die aktuelle Einstellung der Zeichenattribute in einem Kommando:

Sie geben für *setze-attribute* bzw. *lösche-attribute* die Namen der Zeichenattribute an – durch Leerzeichen getrennt – wie sie in der Konfigurationsdatei definiert sind (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Laut Standard-Konfigurationsdatei kennt der MAXed die o.g. Zeichenattribute unter den folgenden Namen:

r[eversted] b[linking] u[nderscore] i[ntensity] g[raphic]

Sie können diese Namen auch bis zum ersten Zeichen abkürzen und bei ausschließlicher Angabe der Anfangszeichen die trennenden Leerzeichen weglassen, solange die Eingabe eindeutig bleibt.

- Für *setattributes* geben Sie die Zeichenattribute an, die Sie mit der Einstellung "gesetzt" (1) aktivieren wollen.
- Für *resetattributes* geben Sie die Zeichenattribute an, die Sie mit der Einstellung "nicht-gesetzt" (0) aktivieren wollen.
- Diejenigen Attribute, die weder bei *setattributes* noch bei *resetattributes* aufgeführt werden, erhalten automatisch die Einstellung "nicht-berücksichtigen" (\*).
- Alle Zeichenattribute, die Sie nicht ausdrücklich bei *setze-attribute* aufführen, erhalten automatisch die Einstellung "nicht-gesetzt" (0), wenn Sie außer *setze-attribute* nur das Komma angeben.

### Anmerkung

- Texte, in denen Sie Zeichenattribute gesetzt haben, können Sie nur mit Hilfe des Dienstprogramms *maxlpr* ausdrucken (siehe Abschnitte 4.5.4 Ausdrucken von MAXed-Dateien, 7.4 MAXed-Dateien für den Ausdruck aufbereiten).
- Beim Suchen mit Hilfe von Bereichsdefinitionen und beim Vergleichen von Textstellen werden die Zeichenattribute berücksichtigt, die aktiviert sind und nicht die Einstellung "nicht-berücksichtigen" haben (siehe Kapitel 6: COMPARE, MINUSFIND, PLUSFIND).

### Beispiel 1

<ENTER> , <SETATTRIB>

Damit setzen Sie alle Zeichenattribute mit dem Kommando zurück auf "nicht-gesetzt" (0).

**Beispiel 2**

Die Schreibmarke steht am Anfang folgender Textzeile:  
Nur MAXed ist blinkend und invers dargestellt, der Rest unterstrichen.

Geben Sie das Kommando

<ENTER> <SETATTRIBUT>

und setzen Sie die Werte für die Voreinstellung der Zeichenattribute auf 0011 0000

Drücken Sie noch einmal <SETATTRIBUT>, um die neue Voreinstellung zu übernehmen. Markieren Sie das Wort *MAXed* als STREAM.

Geben Sie das Kommando <ATTRIBUTE>. *MAXed* wird invers und blinkend dargestellt.

Ändern Sie jetzt die Voreinstellung auf: 01\*\* 0000

Geben Sie das Kommando <SETATTRIBUT>, damit die neue Voreinstellung übernommen wird und anschließend das Kommando <ATTRIBUTE>. Die gesamte Zeile wird unterstrichen, *MAXed* invers, blinkend und unterstrichen dargestellt.

**Beispiel 3**

<ENTER> unders blink,int rever graphic <SETATTRIB>

<ENTER> ub,irg <SETATTRIB>

Beide Kommandos bewirken, daß bei der aktuellen Einstellung der Zeichenattribute *underscore* und *blinking* auf "gesetzt" (1), *reversed*, *intensity* und *graphic* auf "nicht-gesetzt" (0) eingestellt werden.

## SPACE - Textbereiche mit Leerzeichen beschreiben

`<SPACE>`  
`<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD]STREAM] <SPACE>`

Das Kommando SPACE setzt Leerzeichen in die bearbeitete Datei. Dabei werden Leerzeichen bzw. Leerzeilen in identischem Umfang wie der definierte Textbereich abhängig vom aktuell eingestellten Einfüge-Modus an der aktuellen Schreibmarkenposition eingefügt, oder der definierte Textbereich wird mit Leerzeichen überschrieben (siehe Abschnitt 4.1.1 Die Eingabe-Modi).

SPACE ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

`<SPACE>`

Schreibt an die aktuellen Schreibmarken-Position eine Leerzeile. Diese wird im Einfüge-Modus vor der aktuellen Zeile eingefügt oder überschreibt im Überschreibe-Modus die aktuelle Zeile.

`<ENTER> <SPACE>`

Schreibt in die aktuelle Zeile ab der Position der Schreibmarke Leerzeichen und das Zeilenende-Zeichen:

- Im Überschreibe-Modus OVERWRITE wird folglich der Zeilenrest gelöscht. Dies entspricht in der Wirkung dem Kommando `<ENTER> <ERASE>`
- Im Einfüge-Modus wird ein zusätzliches Zeilenende-Zeichen eingefügt, sodaß der Text ab der Position der Schreibmarke in eine neue, nachfolgende Zeile an den linken Formatrand geschoben wird.

`<ENTER> bereich <SPACE>`

Schreibt einen Bereich aus Leerzeichen in identischem Umfang wie der durch die Bereichsangabe festgelegte Textbereich an die aktuelle Schreibmarken-Position.

Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

**Anmerkung**

Bei Bereichen, die Zeilenenden einschließen, werden die Zeilenende-Zeichen beim Einfügen oder Überschreiben mit SPACE vom definierten Textbereich übernommen, sodaß Bereichstyp und -umfang identisch bleiben.

**Beispiel 1**

<ENTER> 5 <SPACE>

Im Einfüge-Modus werden vor der aktuellen Zeile 5 Leerzeilen eingefügt. Im OVERWRITE-Modus werden ab der aktuellen Zeile die nächsten 5 Zeilen mit Leerzeichen überschrieben.

**Beispiel 2**

<ENTER> .-! <SPACE>

Im Einfüge-Modus werden vor den Dateianfang so viele Leerzeilen eingefügt, wie zwischen dem Dateianfang und der Standard-Marke liegen. Dies ist unabhängig von der aktuellen Position der Schreibmarke. Im OVERWRITE-Modus wird der Text vom Dateianfang bis zur Standard-Marke mit Leerzeichen überschrieben.

**Beispiel 3**

<ENTER> 5-10 <SPACE>

Im Einfüge-Modus wird vor der Zeile 5 ein RANGE von 6 Leerzeilen eingeschoben. Im OVERWRITE-Modus werden die Zeilen 5 bis 10 durch Leerzeichen überschrieben. Dies ist unabhängig von der aktuellen Position der Schreibmarke.

## SRTAB - Tabulatoren setzen/löschen

`<SRTAB>`  
`<ENTER> IDENTIFIER <SRTAB>`

Das Kommando SRTAB setzt Tabulatoren bzw. löscht bereits gesetzte Tabulatoren. Diese Tabulatoren gehören zu der Datei im aktuellen Fenster und werden am unteren Rand des aktuellen Fensters durch eine Markierung angezeigt (siehe Abschnitt 4.3.1 Tabulatoren).

`<SRTAB>`  
Setzt einen neuen bzw. löscht einen bereits gesetzten Tabulator an der aktuellen Spaltenposition der Schreibmarke.

`<ENTER> IDENTIFIER <SRTAB>`  
Setzt die Tabulatoren in der aktuellen Datei entsprechend den in der durch IDENTIFIER angegebenen Datei festgelegten Tabulatorpositionen.

In einer solchen Tabulator-Datei legen Sie die gewünschten Tabulatorpositionen folgendermaßen fest:

- durch Eingabe der Spaltenzahlen (getrennt durch Kommata) in der ersten Zeile
- durch Setzen von Punkten in den Spalten der ersten Zeile, in denen Sie Tabulatoren haben wollen. Zwischen den Punkten können beliebige andere druckbare Zeichen angeben sein.

### Anmerkung

- Im Verzeichnis `/usr/lib/MAXed`, stehen die Standard-Tabulator-Dateien 4 und 8 zu Ihrer Verfügung.
- Gesetzte Tabulatoren werden bei Verlassen des MAXed mit EXIT in der benutzer- und terminalspezifischen Status-Datei gespeichert. Wenn Sie den MAXed das nächsten Mal ohne Angabe eines Dateinamens aufrufen, werden diese wieder gesetzt (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen).

**Beispiel**

Sie legen die Tabulator-Datei *tab1* an mit  
<ENTER> *tab1* <EDIT> <EDIT>

Sie geben die gewünschten Tabulatorpositionen ein mit  
3, 10, 17, 24, 38, 45, 59

Sie sichern die Datei mit  
<SAVE>

Sie rufen die Tabulator-Datei *tab1* zum Laden der Tabulatoren auf mit  
<ENTER> *tab1* <SRTAB>

## STARTLINE - Sprung an den Zeilenanfang

`<STARTLINE>`

Mit dem Kommando STARTLINE setzen Sie die Schreibmarke an den Anfang der aktuellen Zeile, falls diese beschrieben ist. Ist die aktuelle Zeile leer oder befindet sich die Schreibmarke schon am Zeilenanfang, bleibt sie unverändert stehen.

`<STARTLINE>`

Bewegt die Schreibmarke an den Anfang einer beschriebenen Zeile.

### Anmerkung

Sie können die Schreibmarke mit  
`<ENTER> ^ <MINUSFIND>`  
an den vorausgehenden Zeilenanfang einer beschriebenen Zeile setzen.  
Dabei wird jedoch gleichzeitig das aktuelle PATTERN überschrieben  
(siehe Abschnitt 3.10 PATTERN als Kommandoattribut).

## TAB - Tabulatorsprung nach rechts

<TAB>

Kommandoverstärkung:

<ENTER> n <TAB>

Start einer Bereichsdefinition für STREAM:

<ENTER> <TAB>

Das Kommando TAB verschiebt die Schreibmarke nach links zur nächsten Tabulatorposition. Die Tabulatorpositionen können Sie mit dem Kommando SRTAB verändern (siehe Abschnitt 4.3.1 Tabulatoren, Kapitel 6: SRTAB).

<TAB>

Bewegt die Schreibmarke nach rechts zur nächsten Tabulatorposition.

<ENTER> <TAB>

Start der Definition eines STREAM-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <TAB>

Bewegt die Schreibmarke um  $n$  Tabulatorpositionen nach rechts.

**Anmerkung**

- Das Verhalten der Schreibmarke beim Bewegen über den Rand des aktuellen Fensters ist abhängig vom Fenster-Modus (siehe Abschnitt 4.1.4 Der Fenster-Modus). Standardmäßig springt sie dabei zum gegenüberliegenden Fensterrand um und bewegt sich von dort aus weiter. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster.
- Mit <TAB> können Sie die Schreibmarke auch im Eingabefeld der Steuerzeile nach rechts verschieben (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer). Im Eingabefeld sind grundsätzlich an jeder achten Spalte (1, 9, 17,...) Tabulatoren gesetzt. Dabei ist ein Umspringen oder Wechseln des Fensters nicht möglich. Der sichtbare Ausschnitt des Eingabefeldes wird jedoch um eine Ausschnittslänge mitverschoben, wenn die Schreibmarke den rechten Rand des Eingabefeldes erreicht und das Ende der eingegebenen Zeichenkette nicht sichtbar ist.

## UNDO - Textveränderungen rückgängig machen

```
<UNDO>
<ENTER> [ ; / , ] [ ! ]
           [IDENTIFIER] [RANGE] [<TEXTVERÄNDERUNGS-KOMMANDO>] <UNDO>

Kommandoverstärkung:
<ENTER> [ ; / , ] [ ! ] n
           [IDENTIFIER] [RANGE] [<TEXTVERÄNDERUNGS-KOMMANDO>] <UNDO>
```

Mit dem Kommando UNDO machen Sie schrittweise oder in einem festgelegten Umfang Textveränderungen in den bearbeiteten Dateien der aktuellen Sitzung wieder rückgängig. Unter die textverändernden Kommandos fallen alle Kommandos, die

- vorhandenen Text löschen (siehe Kapitel 6: ERASE, ERASEWORD)
- vorhandenen Text überschreiben (siehe Kapitel 6: PUT, RESTOR, SPACE)
- vorhandenen Text formatieren oder mit Zeichenattributen versehen (siehe Kapitel 6: ALIGN, ATTRIBUTE, CENTER, COMPRESS, DRAW, FILL, FORMAT, JUSTIFY, LOWER, UPPER, WRAP)
- vorhandene Textstellen ersetzen oder austauschen (siehe Kapitel 6: EXCHANGE, FILTER, REPLACE)
- Textbereiche einfügen (siehe Kapitel 6: FILTER, PUT, RESTOR, SPACE)
- Textveränderungen durch Makros und Tasten-Makros (siehe Kapitel 6: MACRO, Abschnitt 6.2 Die Tasten-Makros)

Das Eingeben von Text durch Tippen der Einzelzeichen und Löschen von Einzelzeichen durch BACKSPACE oder DELCHAR innerhalb einer Zeile wird als eine Textveränderungs-Einheit betrachtet, solange Sie die Zeile nicht verlassen. Das bedeutet, daß Sie alle Eingaben und Änderungen von Einzelzeichen, die Sie in einer einzigen Zeile vornehmen, ohne diese zu verlassen und ohne eines der o.g. textverändernden Kommandos auszuführen, in der Standard-Ausführung des Kommandos UNDO auf einmal zurücknehmen.

Abhängig vom eingestellten Zurücksetzen-Modus werden Makro-Ausführungen als Textveränderungs-Einheiten betrachtet oder wieder in einzelne Textveränderungs-Schritte aufgespalten (siehe Abschnitt 4.1.9 Der Zurücksetzen-Modus).



Diese Einstellung des Zurücksetzen-Modus bleibt auch für weitere UNDO-Kommandos gültig, sofern sie dabei nicht explizit umgestellt wird (siehe Abschnitt 4.1.9 Der Zurücksetzen-Modus).

#### richtungsumkehr

Keht die Wirkungsrichtung für dieses UNDO-Kommando um:

! Bei Angabe des Ausrufungszeichens werden bereits zurückgenommene Textveränderungen wie beim Kommando <ENTER> <UNDO> (siehe oben) in Vorwärtsrichtung wieder durchgespielt.

n gibt die Anzahl der Textveränderungen an, die durch dieses Kommando auf einmal zurückgenommen werden sollen.

#### IDENTIFIER

Legt die Auswahl der Dateien fest, die von der Rücknahme der Textveränderungen betroffen sein sollen (siehe Abschnitt 3.8.1 Der Dateiname als IDENTIFIER). Diese Einstellung des Zurücksetzen-Modus bleibt auch für weitere UNDO-Kommandos gültig, sofern sie dabei nicht explizit umgestellt wird (siehe Abschnitt 4.1.9 Der Zurücksetzen-Modus).

#### *Anmerkung*

Sie sprechen alle geöffneten Dateien durch den IDENTIFIER \* an. Dies ist die Standard-Einstellung.

#### RANGE

Stellt ein Abbruchkriterium dar. Dieses UNDO-Kommando wird abgebrochen, sobald im angegebenen RANGE-Bereich eine Textveränderung zurückgenommen werden müßte (siehe Abschnitt 3.9.2 Bereichsdefinition durch Zeichenketten-Attribut).

#### <TEXTVERÄNDERUNGS-KOMMANDO>

Stellt bei Angabe eines der oben aufgeführten textverändernden Kommandos in spitzen Klammern ein Abbruchkriterium dar. Dieses UNDO-Kommando wird abgebrochen, sobald das angegebene Kommando zurückgenommen werden müßte.

## UPARROW - Die Schreibmarke nach oben bewegen

<UPARROW>  
 <ENTER> n / h / v / m / c <UPARROW>

Kommandoverstärkung:  
 <ENTER> n <UPARROW>

Start einer Bereichsdefinition für RANGE oder RECTANGLE:  
 <ENTER> <UPARROW>

Das Kommando UPARROW bewegt die Schreibmarke um eine Zeile nach oben. Im standardmäßig eingestellten Fenster-Modus springt die Schreibmarke auf den gegenüberliegenden Fensterrand um, wenn Sie sie auf den Rand des aktuellen Fensters bewegen. Sie bleibt also innerhalb des sichtbaren Dateiausschnitts im aktuellen Fenster (siehe Abschnitt 3.5 Der Bildschirm des MAXed). Sie können den Fenster-Modus jedoch auch so einstellen, daß die Schreibmarke stattdessen den Dateiausschnitt verschiebt oder in ein angrenzendes Fenster überwechselt (siehe Abschnitt 4.1.4 Der Fenster-Modus).

<UPARROW>  
 Bewegt die Schreibmarke um eine Zeile nach oben.

<ENTER> <UPARROW>  
 Start der Definition eines RANGE- oder RECTANGLE-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich wie oben beschrieben (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <UPARROW>  
 Bewegt die Schreibmarke um *n* Zeilen nach oben.

<ENTER> *modus* <UPARROW>  
 Damit stellen Sie den aktuellen Fenster-Modus ein (siehe Abschnitt 4.1.4 Die Fenster-Modus). Davon betroffen sind alle ARROW- und TAB-Kommandos beim Bewegen der Schreibmarke auf bzw. über den Rand des aktuellen Fensters. Für *modus* können Sie folgendes eingeben:

- n[ormal] Die Schreibmarke erhält für alle vier Richtungen die Eigenschaft, zum gegenüberliegenden Fensterrand umzuspringen. Dieser Fenster-Modus ist standardmäßig eingestellt.
- h[orizontal] Für horizontale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando oder durch Texteingabe den rechten bzw. linken Fensterrand überschreitet. Am Zeilenanfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.
- v[ertical] Für vertikale Bewegungen ist die Umspring-Eigenschaft abgestellt. Das Fenster wird nun grundsätzlich mitbewegt, wenn die Schreibmarke durch ein Bewegungs-Kommando den oberen bzw. unteren Fensterrand überschreitet. Am Dateianfang bewegt sich die Schreibmarke allerdings nicht mehr weiter.
- m[ove] Kombiniert die Einstellungen *horizontal* und *vertikal*. Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt.
- c[hangewindow] Die Umspring-Eigenschaft ist in allen vier Richtungen abgestellt. Stattdessen wechselt die Schreibmarke (falls möglich) über den Rand des aktuellen Fensters in das benachbarte Fenster, das zum aktuellen Fenster wird (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten). Beim Überschreiten des äußersten Fensterrandes springt die Schreibmarke allerdings nach wie vor um.

### Anmerkung

<UPARROW> besitzt im Eingabefeld der Steuerzeile eine spezielle Bedeutung: Mit <UPARROW> bewegen Sie die Schreibmarke an den Anfang der bisher eingegebenen Zeichenkette (siehe Abschnitt 3.6.2 Das Eingabefeld und der Eingabepuffer).

## UPEND - Die Schreibmarke ans vorige Zeilenende bewegen

<UPEND>

Start einer Bereichsdefinition für STREAM:

<ENTER> <UPEND>

Das Kommando UPEND setzt die Schreibmarke ans Ende der vorhergehenden Zeile. Falls notwendig, wird auch das aktuelle Fenster bewegt.

<UPEND>

Bewegt die Schreibmarke ans Ende der vorigen Zeile.

<ENTER> <UPEND>

Start der Definition eines STREAM-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich ans Ende der vorigen Zeile (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

### Beispiel 1

Angenommen, Sie befinden sich mit Ihrer Schreibmarke mitten in einer vollgeschriebenen Textzeile. Nun wollen Sie den Anfang der Zeile bis zur aktuellen Position der Schreibmarke löschen und den Rest ans Ende der vorherigen Zeile anhängen. Sie erreichen dies mit

<ENTER> <UPEND> <ERASE>

### Beispiel 2

Die Schreibmarke steht wiederum mitten in einer Textzeile. Der RETURN-Modus ist weder EMPTY noch SPACE (siehe Kapitel 6:RETURN). Sie wollen jetzt genau den Rest der aktuellen Zeile und die nächste Zeile in den Standard-PICK-Puffer kopieren:

<ENTER> <RETURN> <RETURN> <UPEND> <PICK>

## UPPER - Großschreibung

<UPPER>  
<ENTER> [RANGE/RECTANGLE/[BLOCK/RECORD] STREAM] <UPPER>

Das Kommando UPPER wandelt Kleinbuchstaben in Großbuchstaben um. Alle anderen Zeichen bleiben unverändert (siehe Abschnitt 4.3.4 Ausrichten und Vereinheitlichen).

UPPER ist ein textveränderndes Kommando. Der ursprüngliche Textbereich wird bei der Kommandoausführung in die Pseudo-Datei RESTORE gerettet (siehe Abschnitt 4.2.2). Mit Hilfe des Kommandos UNDO können Sie Textveränderungen wieder rückgängig machen (siehe Kapitel 6: UNDO).

<UPPER>  
Wandelt die Kleinbuchstaben der aktuellen Zeile in Großbuchstaben um.

<ENTER> <UPPER>  
Wandelt die Kleinbuchstaben der aktuellen Zeile ab der Schreibmarken-Position bis zum Zeilenende in Großbuchstaben um.

<ENTER> *bereich* <UPPER>  
Führt das Kommando UPPER in dem angegebenen Bereich aus. Für *bereich* sind RANGE, RECTANGLE und STREAM möglich. Spezialfälle eines STREAM sind RECORD und BLOCK.

## VWINDOW - Den Bildschirm vertikal teilen

```

<VWINDOW>
<ENTER> [ [ = / + ] IDENTIFIER ]
          /[ IDENTIFIER [ zeile[ spalte[PATTERN]]] ]
          /[ IDENTIFIER "PATTERN" ]
          /[ @ IDENTIFIER ]
          /[ @ / * / > / < ] <VWINDOW>

```

Das Kommando VWINDOW erzeugt ein neues Fenster durch vertikale Teilung des aktuellen Fensters. Das rechte Fenster ist nun das aktuelle Fenster. Dieses wird, wenn nicht anders angegeben, an den Anfang der neuen aktuellen Datei gesetzt. Das verkleinerte alte Fenster bleibt in der zuvor aktuellen Datei an der ursprünglichen Position. Sie können das Kommando VWINDOW in einigen Fällen mit denselben Kommandoattributen versehen wie das Kommando EDIT (siehe Kapitel 6: EDIT). Mit Hilfe des Kommandos CHWINDOW wechseln Sie zwischen den Fenstern (siehe Abschnitt 3.4.2 Dateiausschnitte in mehreren Fenstern bearbeiten).

<VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Dabei wird die zuvor aktuelle Datei nochmals im (neuen) aktuellen Fenster eingeblendet.

<ENTER> <VWINDOW>

Entfernt das zuletzt mit einem HWINDOW- oder VWINDOW-Kommando eingerichtete Fenster, das ist jenes mit der höchsten Fensternummer. Das Fenster, von dem das gelöschte Fenster abgeteilt worden war, erhält wieder seine ursprüngliche Größe und wird zum aktuellen Fenster.

<ENTER> IDENTIFIER <VWINDOW>

Erzeugt ein neues Fenster durch vertikale Teilung des aktuellen Fensters an der aktuellen Spalte. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> IDENTIFIER <EDIT>  
ausgeführt.

<ENTER> = IDENTIFIER <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> = IDENTIFIER <EDIT>  
ausgeführt.

<ENTER> + IDENTIFIER <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> + IDENTIFIER <EDIT>  
ausgeführt.

<ENTER> IDENTIFIER [zeile [spalte [PATTERN]]] <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> IDENTIFIER [zeile [spalte [PATTERN]]] <EDIT>  
ausgeführt.

<ENTER> IDENTIFIER "PATTERN" <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> IDENTIFIER "PATTERN" <EDIT>  
ausgeführt.

<ENTER> @ <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> @ <EDIT>  
ausgeführt.

<ENTER> \* <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando

<ENTER> \* <EDIT>  
ausgeführt.

<ENTER> > <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> > <EDIT> ausgeführt.

<ENTER> < <VWINDOW>

Erzeugt durch vertikale Unterteilung des aktuellen Fensters an der Position der Schreibmarke ein neues Fenster. Der rechte Teil ist nun das aktuelle Fenster. Im übrigen wird das Kommando <ENTER> < <EDIT> ausgeführt.

### Beispiel

Sie wollen die Datei im aktuellen Fenster mit anderen Dateien vergleichen, die nicht im aktuellen Dateiverzeichnis stehen. Sie geben z.B. ein:

<ENTER> .. <VWINDOW>

Das übergeordnete Dateiverzeichnis wird im neuen vertikalen Fenster aufgelistet. Durch Positionieren der Schreibmarke und Betätigen von <EDIT> können Sie sich eine Datei einblenden lassen bzw., wenn Sie ein Dateiverzeichnis auswählen, sich dieses auflisten lassen und so zu der gesuchten Datei durchwählen.

## WAIT - Warten auf das Ende eines asynchronen Programms

<WAIT>

Das Kommando WAIT bewirkt, daß der MAXed auf die Beendigung aller mit dem Kommando EXECUTE angestoßenen, asynchron laufenden SINIX-Programme wartet (siehe Abschnitt 4.5.2 Ausführen asynchroner SINIX-Kommandos, Kapitel 6: EXECUTE). Dieses Kommando ist nur beim Ablauf eines Makros bzw. eines MAXed-Steuerprogramms sinnvoll. Sie können es durch die Angabe seiner Kommandozeichnung in Quelldateien von MACRO-Bibliotheken oder C-Programmen zur Steuerung des MAXed verwenden (siehe Abschnitte 7.1 Eine MACRO-Bibliothek erstellen, 7.5 MAXed-Steuerprogramme). <WAIT> ist nicht auf der Tastatur der Datensichtstation konfiguriert.

<WAIT>

Wartet auf die Beendigung sämtlicher asynchron im Hintergrund laufender SINIX-Kommandos, die durch das Kommando EXECUTE gestartet wurden.

### Anmerkung

WAIT kann eingesetzt werden, um abzuwarten, bis ein asynchron laufendes SINIX-Programm seine Ausgaben in der Pseudo-Datei SYSOUTPUT abgelegt hat. Diese Ausgaben können anschließend weiter verarbeitet werden (siehe Abschnitt 4.2.11 Die Pseudo-Datei SYSOUTPUT).

## WINL - Den Dateiausschnitt nach links verschieben

<WINL>  
<ENTER> n / ^ <WINL>

Start einer Bereichsdefinition für STREAM:  
<ENTER> <WINL>

Das Kommando WINL verschiebt den Dateiausschnitt im aktuellen Fenster um einige Spalten nach links oder zur ersten Spalte der Datei, wenn diese näher ist. Die Schreibmarke bleibt in der gleichen Zeile. Ist die aktuelle Spalte noch innerhalb des Fensters, bleibt die Schreibmarke in dieser Spalte stehen. Anderenfalls bewegt sie sich um die gleiche Anzahl Spalten nach links, sodaß sie im aktuellen Fenster sichtbar bleibt (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

<WINL>  
Verschiebt den Dateiausschnitt um einige Spalten nach links.

<ENTER> <WINL>  
Start der Definition eines STREAM-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Fenster und Schreibmarke bewegen sich um einige Spalten nach links (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

<ENTER> n <WINL>  
Verschiebt den Dateiausschnitt um  $n$  Spalten nach links.

<ENTER> ^ <WINL>  
Verschiebt den Dateiausschnitt nach links zur ersten Spalte der Datei.

## WINR - Den Dateiausschnitt nach rechts verschieben

```
<WINR>  
<ENTER> n <WINR>
```

Start einer Bereichsdefinition für STREAM:  
<ENTER> <WINR>

Das Kommando WINR verschiebt den Dateiausschnitt im aktuellen Fenster um einige Spalten nach rechts. Die Schreibmarke bleibt in der gleichen Zeile. Ist die aktuelle Spalte noch innerhalb des Fensters, bleibt die Schreibmarke in dieser Spalte stehen. Anderenfalls bewegt sie sich um die gleiche Anzahl Spalten nach rechts, sodaß sie im aktuellen Fenster sichtbar bleibt (siehe Abschnitt 3.3.3 Verschieben des Dateiausschnitts).

```
<WINR>
```

Verschiebt den Dateiausschnitt um einige Spalten nach rechts.

```
<ENTER> <WINR>
```

Start der Definition eines STREAM-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Fenster und Schreibmarke bewegen sich um einige Spalten nach rechts (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

```
<ENTER> n <WINR>
```

Verschiebt den Dateiausschnitt um  $n$  Spalten nach rechts.

## WORDLEFT - Die Schreibmarke wortweise nach links bewegen

<WORDLEFT>

Start einer Bereichsdefinition für STREAM:  
<ENTER> <WORDLEFT>

Das Kommando WORDLEFT bewegt die Schreibmarke nach links zum nächstliegenden Wortanfang. Gegebenenfalls verschiebt sich der Dateiauschnitt. Steht die Schreibmarke auf dem ersten Wort der aktuellen Zeile, werden die Schreibmarke und eventuell auch das Fenster an den Zeilenanfang positioniert. Ein weiteres Kommando WORDLEFT bewirkt einen Sprung der Schreibmarke ans Ende der vorhergehenden Zeile (siehe 3.3.2 Bewegen der Schreibmarke).

<WORDLEFT>

Bewegt die Schreibmarke nach links zum nächstliegenden Wortanfang.

<ENTER> <WORDLEFT>

Start der Definition eines STREAM-Bereichs. Das Ende (bzw. der Anfang) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich nach links zum nächstliegenden Wortanfang (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

## WORDRIGHT - Die Schreibmarke wortweise nach rechts bewegen

<WORDRIGHT>

Start einer Bereichsdefinition für STREAM:

<ENTER> <WORDRIGHT>

Das Kommando WORDRIGHT bewegt die Schreibmarke nach rechts zum Anfang des nächsten Wortes. Gegebenenfalls verschiebt sich der Dateiauschnitt. Steht die Schreibmarke auf dem letzten Wort der aktuellen Zeile, springt sie zur ersten Spalte hinter diesem Wort. Ein weiteres Kommando WORDRIGHT bewirkt einen Sprung der Schreibmarke nach links in die erste Spalte der nächsten Zeile (siehe Abschnitt 3.3.2 Bewegen der Schreibmarke).

<WORDRIGHT>

Bewegt die Schreibmarke nach rechts zum Anfang des nächsten Wortes.

<ENTER> <WORDRIGHT>

Start der Definition eines STREAM-Bereichs. Der Anfang (bzw. das Ende) des Bereichs wird an der aktuellen Schreibmarkenposition festgelegt. Die Schreibmarke bewegt sich nach rechts zum Anfang des nächsten Wortes (siehe Abschnitt 3.9.1 Bereichsdefinition durch Bewegen der Schreibmarke).

## WRAP - Text in die nächste Zeile bringen

`<WRAP>`  
`<ENTER> <WRAP>`

Das Kommando `WRAP` umbricht den Text der aktuellen Zeile ab der Position der Schreibmarke bis zum Zeilenende in die nächste Zeile (siehe Abschnitt 4.3.4 Umbrechen und Zusammenfügen):

- Ist die nächste Zeile eine Leerzeile, wird der umbrochene Text an den linken Formatrand gestellt.
- Befindet sich schon Text in der nächsten Zeile, wird der Text aus der vorausgehenden Zeile am Textanfang eingefügt. Dadurch wird der bisherige Text nach verschoben.

`<WRAP>`

Umbricht den Text zwischen der Position der Schreibmarke und dem Zeilenende der aktuellen Zeile in die nächste Zeile.

`<ENTER> <WRAP>`

Verschiebt (in umgekehrter Richtung als das voranstehende Kommando) den Text zwischen dem Textanfang der aktuellen Zeile und dem Zeichen vor der Position der Schreibmarke mit einem führenden Leerzeichen an das Ende der vorhergehenden Zeile. Der restliche Text der aktuellen Zeile wird nach links an den ursprünglichen Textanfang gesetzt.

### Anmerkung

- Das Kommando `WRAP` arbeitet unabhängig vom eingestellten Eingabemodus (siehe Abschnitt 4.1.1 Die Eingabe-Modi). Vorhandener Text wird grundsätzlich nicht überschrieben, sondern nach rechts verschoben.
- Zwischen dem umbrochenen und dem nach rechts verschobenen Text wird ggf. ein trennendes Leerzeichen gesetzt.

### 6.2 MACRO-Definitionen zur Tastenbelegung

Einige Kommandos, die standardmäßig bestimmten Kommandotasten zugeordnet sind, sind keine Einzelkommandos sondern Tasten-Makros aus der Standard-Konfigurationsdatei (siehe Abschnitt 7.3 Eine neue Konfigurationsdatei erstellen). Bei der Anwendung ist dies zu beachten. Da diese Kommandobezeichnungen keine Schlüsselbegriffe darstellen, können sie auch nicht als solche in Makro-Definitionen innerhalb der Quell-Dateien von MACRO-Bibliotheken verwendet werden (siehe Abschnitt 7.1 Eine MACRO-Bibliothek erstellen)!

Liste der definierten Tasten-Makros:

#### <ERASEWORD>

Löscht das Wort, auf dem die Schreibmarke steht und speichert es im Standard-ERASE-Puffer. Dabei wird der nachfolgende Text nach links zurückgezogen. Außerdem werden alle folgenden Leerzeichen bis zum nächsten Wort der aktuellen Zeile mit gelöscht. Das Wort ist (zusammen mit den Leerzeichen) im Standard-ERASE-Puffer gespeichert, sodaß Sie es an anderer Stelle mit <RESTOR> wieder einsetzen können (siehe Kapitel 6: ERASE, RESTOR).

#### <ERASEWORD>

Entspricht der Kommandofolge  
<ENTER> , <ERASE>

#### <PICKWORD>

Kopiert das Wort, auf dem die Schreibmarke steht, in den Standard-PICK-Puffer. Dazu gehören auch alle folgenden Leerzeichen bis zum nächsten Wort der aktuellen Zeile.

#### <PICKWORD>

Entspricht der Kommandofolge  
<ENTER> , <PICK>

### <MINUSBLOCK>

Bewegt die Schreibmarke zum BLOCK-Anfang des kleinsten umgebenden BLOCKs.

### <MINUSBLOCK>

Entspricht der Kommandofolge  
<ENTER> ' <MINUSFIND>

### <PLUSBLOCK>

Bewegt die Schreibmarke zum BLOCK-Ende des kleinsten umgebenden BLOCKs.

### <PLUSBLOCK>

Entspricht der Kommandofolge  
<ENTER> ' <PLUSFIND>

—

—

—

—

### 7 Dienstprogramme und Steuerprogramme des MAXed

Der MAXed stellt Ihnen mehrere Dienstprogramme zur Verfügung, die Ihnen das Erstellen und Bearbeiten von MACRO-Bibliotheken sowie das Ausdrucken von MAXed-Dateien ermöglichen.

Im einzelnen sind dies folgende Programme:

**mdefine**

übersetzt Quelldateien mit Definitionen von Makros und Textbausteinen in MACRO-Bibliotheken (siehe Abschnitt 7.1).

**mlist**

erstellt aus MACRO-Bibliotheksdateien wieder lesbaren Quellcode (siehe Abschnitt 7.2).

**mconfig**

ermöglicht das Anlegen einer neuen Konfigurationsdatei (siehe Abschnitt 7.3).

**maxlpr**

ermöglicht Ihnen die Aufbereitung von MAXed-Dateien, in denen Sie Zeichenattribute gesetzt oder mit Hilfe des Kommandos FONT spezielle nationale Sonderzeichen geschrieben haben (siehe Abschnitt 7.4).

Sie können diese Dienstprogramme auf der SINIX-Shell-Ebene aufrufen.

Das Erstellen von MAXed-Steuerprogrammen ist in Abschnitt 7.5 beschrieben.

## 7.1 Eine MACRO-Bibliothek erstellen

Es gibt grundsätzlich zwei Möglichkeiten, eine MACRO-Bibliothek zu erstellen:

- Sie können alle benannten Makros und Textbausteine (PICK-Puffer), die Sie in der aktuellen MAXed-Sitzung neu erstellt haben oder die durch Öffnen einer schon existierenden MACRO-Bibliothek aktiviert wurden, vor Beendigung der Sitzung mit Hilfe von  
`<ENTER> # [datei] <SAVE>`  
 in die Standard- bzw. in die angegebene MACRO-Bibliothek *datei* abspeichern. Die angesprochene MACRO-Bibliothek wird dabei neu angelegt (siehe Kapitel 6: SAVE).
- Für kompliziertere Makros oder auch, um eine bestimmte Zusammengehörigkeit von Textbausteinen zu erreichen, empfiehlt es sich, Makros und Textbausteine in einer Quelldatei zu erstellen, und diese danach mit dem Dienstprogramm *mdefine* in eine MACRO-Bibliothek zu übersetzen.

MACRO-Bibliotheken öffnen Sie beim Aufruf des MAXed, aber auch während der Sitzung mit

```
<ENTER> #datei <EDIT>
```

(siehe Abschnitt 5.1 maxed – Den MAXed aufrufen, Kapitel 6: EDIT). Damit stehen Ihnen die darin definierten Makros und Textbausteine zusammen mit den aktuell definierten zur Verfügung. Sie greifen auf sie zu mit

```
<ENTER> IDENTIFIER <MACRO>
```

bzw.

```
<ENTER> IDENTIFIER <PUT>
```

Der Kommandoaufruf für *mdefine* hat folgendes Format:

**mdefine** *quelldatei* [MACRO-bibliothek]

*quelldatei*

Steht für eine oder mehrere Quelldateien. Diese müssen durch Kommata voneinander getrennt sein. Es können auch Sonderzeichen für Dateinamen benutzt werden.

MACRO-bibliothek

Steht für den Dateinamen der zu erstellenden MACRO-Bibliothek. Eine bereits bestehende Datei gleichen Namens wird überschrieben. Für MACRO-Bibliotheksnamen sind keine Sonderzeichen zulässig. Wenn Sie keinen Dateinamen einer MACRO-Bibliothek angeben, schreibt MAXed die übersetzte Quelldatei in die Datei */usr/lib/MAXed/tmp/X\$USER.mcd*. Diese Datei wird im Normalfall beim MAXed-Aufruf als benutzerspezifische Standard-MACRO-Bibliothek geöffnet.

### 7.1.1 Makros in einer MACRO-Bibliothek anlegen

Quelldateien für Makro-Definitionen müssen wie folgt aussehen:

```
<MACRO=IDENTIFIER1>
text<schlüsselwort>text<schlüsselwort>...
.
.
<schlüsselwort>...text...
<ENDMACRO>

<MACRO=IDENTIFIER2 kommentar>
<schlüsselwort>...text...
.
.
text / kommentar
text<schlüsselwort>text<schlüsselwort>...
<ENDMACRO>
```

Beim Erstellen einer Makro-Definition in einer Quelldatei ist folgendes zu beachten:

- Eine Makro-Definition muß mit der Festlegung des Makronamens (IDENTIFIER) beginnen und mit dem Schlüsselwort <ENDMACRO> enden.
- Lautet der Makroname *X\_INIT*, wird das Makro automatisch beim Öffnen der MACRO-Bibliothek einmal ausgeführt.
- Leerzeichen und Tabulatoren innerhalb einer Makro-Definition werden wie Text behandelt. Kommandoattributierungen werden wie üblich durch <ENTER> eingeleitet und durch ein beliebiges anderes Kommando abgeschlossen. Hier dienen Leerzeichen und Tabulatoren als Trennzeichen zwischen den einzelnen Kommandoattributen.
- Ein Zeilenvorschub innerhalb einer Makro-Definition wird immer ignoriert. Um bei der Ausführung des Makros einen Zeilenvorschub auszulösen, müssen Sie explizit das Schlüsselwort <RETURN> angeben.
- Schlüsselwörter bestehen immer aus einem Bezeichner in spitzen Klammern. Sie können die Bezeichner in Groß- oder Kleinbuchstaben schreiben. Wollen Sie eine spitze Klammern als Text erscheinen lassen, müssen Sie in der Quelldatei zwei Klammern in direkter Folge << bzw. >> angeben.

- Kommentare stehen entweder in der ersten Zeile der Makro-Definition zwischen IDENTIFIER und der schließenden spitzen Klammer >. Diese Kommentare werden in der Pseudo-Datei MACROS angezeigt (siehe Abschnitt 4.2.10). Oder sie beginnen in einer beliebigen Zeile mit einem Schrägstrich /. Der Kommentartext wird ab dem Schrägstrich bis zum Ende der Zeile ignoriert. Wollen Sie einen Schrägstrich als Text erscheinen lassen, so müssen Sie in der Quelldatei zwei Schrägstriche in direkter Folge // angeben.
- Quelldateien dürfen keine Zeichenattribute enthalten. Nationale Sonderzeichen geben Sie wie üblich mit Hilfe des Kommandos FONT ein (siehe Abschnitt 4.4 Die Zeichenattribute, Kapitel 6: FONT). Das Kommando <FONT> ist aber für Makro-Definitionen nicht als Schlüsselwort verfügbar!

Folgende Schlüsselwörter können Sie innerhalb einer Makro-Definition - auch eindeutig abgekürzt - benutzen:

<ALIGN>	<DRAW>	<INSERTMODE>	<PLUSPAGE>	<TAB>
<ATTRIBUTE>	<EDIT>	<JUSTIFY>	<PRINT>	<UPARROW>
<BACKSPACE>	<ENDLINE>	<LABEL>	<PUT>	<UNDO>
<BACKTAB>	<ENTER>	<LEARN>	<QUIT>	<UPEND>
<CANCEL>	<ERASE>	<LEFTARROW>	<REFRESH>	<UPPER>
<CENTER>	<EXCHANGE>	<LINEFEED>	<REPLACE>	<VWINDOW>
<CHLINELEN>	<EXECUTE>	<LOWER>	<RESTOR>	<WAIT>
<CHWINDOW>	<EXIT>	<MACRO>	<RETURN>	<WINL>
<CLI>	<FILL>	<MARGIN>	<RIGHTARROW>	<WINR>
<CODE>	<FILTER>	<MINUSFIND>	<RVLINEFEED>	<WORDLEFT>
<COMPARE>	<FORMAT>	<MINUSLINE>	<SAVE>	<WORDRIGHT>
<COMPRESS>	<GOTO>	<MINUSPAGE>	<SETATTRIB>	<WRAP>
<DEFDIR>	<HELP>	<PICK>	<SPACE>	
<DELCHAR>	<HOME>	<PLUSFIND>	<SRTAB>	
<DOWNARROW>	<HWINDOW>	<PLUSLINE>	<STARTLINE>	

**Beispiel**

- Legen Sie die Datei *makrodef1* an und geben Sie die nachstehenden Makros ein. Achten Sie darauf, daß Ihre Eingabe am linken Fenster- rand beginnt, da sonst alle führenden Leerzeichen als Textbestandteil der Makros betrachtet werden.

```
<MACRO=X_INIT>
<ENTER>2<SPACE>
<ENTER>'Guten Morgen, Egon. Ich wünsche Dir frohes Schaffen'<PUT>
<ENTER>2<SPACE>
<ENDMACRO>
```

```
<MACRO=formabs formatiert Absatz; Eingabe: Formatraender>
<ENTER>@1 @2<MARGIN><ENTER>^;<PLUSFIND>
<ENTER>^<PLUSFIND><FILL>
<ENDMACRO>
```

- Sichern Sie die Datei *makrodef1* und geben Sie folgendes Kommando:  
<ENTER> mdefine makrodef1 makrodef1c <EXECUTE>  
Damit wird die Datei *makrodef1* in die MACRO-Bibliothek *makrodef1c* übersetzt.
- Öffnen Sie die MACRO-Bibliothek *makrodef1c*. Geben Sie dazu das Kommando:  
<ENTER> #makrodef1c <EDIT>

Beim Öffnen der MACRO-Bibliothek wird sofort das Makro *X\_INIT* ausgeführt. In der bearbeiteten Datei erscheint der Satz:  
Guten Morgen, Egon. Ich wünsche Dir frohes Schaffen

- Dann rufen Sie das Makro *formabs* mit den gewünschten Formaträndern auf:  
<ENTER> formabs 10 60 <MACRO>  
Die eingegebenen Zahlen ersetzen dann im Makro die Parameter für die Formatgrenzen *@1* und *@2*. Das Makro *formabs* formatiert den nächstfolgenden Absatz der bearbeiteten Datei im Flattersatz.

### 7.1.2 Textbausteine in einer MACRO-Bibliothek anlegen

Wollen Sie die Textbausteine einer MACRO-Bibliothek anlegen, so müssen Sie zunächst deren Bereichstyp und den PICK-Puffernamen (IDENTIFIER) festlegen. Am Ende eines Textbausteins steht das Schlüsselwort <ENDBUFFER>. Die Textbausteine können als RANGE, RECTANGLE, STREAM, RECORD oder BLOCK definiert sein.

Quelldateien für Textbausteine sehen wie folgt aus:

```
<STREAM=IDENTIFIER1>  
    text  
<ENDBUFFER>
```

```
<RANGE=IDENTIFIER2>  
    text  
<ENDBUFFER>
```

```
<RECTANGLE=IDENTIFIER3>  
    text  
<ENDBUFFER>
```

Der Text für den Textbaustein wird so aufgenommen, wie er eingegeben wird. Leerzeichen, Tabulatoren, Zeilenvorschub und Leerzeilen werden als Text behandelt. Bei einem RECTANGLE-Puffer wird die Anzahl der Spalten durch die Länge der längsten Zeile in der Puffer-Definition bestimmt.

Die Eingabe der Schlüsselwörter muß in Spalte eins beginnen!

**Beispiel**

- Legen Sie die Datei *Brief* an und geben Sie den nachfolgenden Text ein. Beginnen Sie mit der Eingabe der Schlüsselwörter immer in Spalte eins. Der Textanfang der Zeilen des Pufferinhalts soll jedoch in Spalte fünf liegen:

```
<RANGE=adr1>
    SIEMENS AG
    Abtlg. DI ST QM 111
    Otto-Hahn-Ring 6
```

```
    8000 München 83
<ENDBUFFER>
```

```
<RANGE=adr2>
    SIEMENS AG
    DI ST QM 232
    - Manualredaktion -
    Otto-Hahn-Ring 6
```

```
    8000 München 83
<ENDBUFFER>
```

```
<STREAM=anr1>
    Sehr geehrte Damen und Herren,
<ENDBUFFER>
```

- Die Datei *Brief* wird abschließend mit <SAVE> gesichert. Sie rufen anschließend das Dienstprogramm *mdefine* auf:  
 <ENTER> mdefine Brief Briefc <EXECUTE>  
 Die Übersetzung von *Brief* in die MACRO-Bibliothek *Briefc* wird asynchron im Hintergrund ausgeführt.
- Nachdem Sie die MACRO-Bibliothek *Briefc* während einer MAXed-Sitzung durch  
 <ENTER> #Briefc <EDIT>  
 oder schon beim Aufruf mit  
 maxed -d=Briefc datei  
 geöffnet haben, können Sie beispielsweise mit dem Kommando  
 <ENTER> adr1 <PUT>  
 die erste Adresse in die bearbeitete Datei *datei* schreiben.

### 7.1.3 Beispiel für eine MACRO-Bibliothek

Als Beispiel 2 des Kommandos LEARN finden Sie die Definition des Makros *brief* in der Lernphase (siehe Kapitel 6: LEARN). Das gleiche Makro können Sie natürlich auch in der Quelldatei einer MACRO-Bibliothek definieren. Es bekommt hier den Namen *siegel*. Die Quelldatei soll *Weihnacht* heißen, im aktuellen Dateiverzeichnis stehen und außerdem noch die Definitionen der benötigten Textbausteine enthalten:

```
<MACRO=siegel      Makro zum Komplettieren vorgefertigter Briefe>
<EDIT>                / obiger Kommentar erscheint in Datei MACROS;
<GOTO><HOME>          / ab dem Schrägstrich folgt auch Kommentar;
<ENTER>4<righta>     / RIGHTARROW ist eindeutig abgekürzt;
<MARGIN>
<ENTER>i<INSERTMODE>
<SPACE>
<ENTER>t<RETURN>
<ENTER>abs +<PUT>     / spitze Klammern sind Sonderzeichen in
<ENTER>"sehr gee>>"<PLUSFIND> / Quelldateien und müssen verdoppelt werden!
<ENTER>fraulherr<PLUSFIND> / ein führendes Leerzeichen würde bei diesem
<ENTER><PICK>         / Suchmuster die 1. Variante beeinflussen!
<ENTER>.<GOTO>
<ENTER>3<RETURN>
Ich wünsche Ihnen,
<ENTER>' % ein frohes Fest!'<PUT>
<RETURN>
<RETURN>
<ENTER>'@1, den @d'<PUT>
<RETURN>
<RETURN>
<ENTER>40<RIGHTARROW>
<ENTER>@2<PUT>
<EDIT>
<LINEFEED>
<ENDMACRO>

<RECTANGLE=GRU2>
Mit ganz lieben Grüßen
<ENDBUFFER>

<RECTANGLE=GRU1>
Mit freundlichen Grüßen
<ENDBUFFER>

<RECTANGLE=ABS>
Taurus Rexor
Kolbunter 13
888 Munix 24
<ENDBUFFER>
```

Wenn Sie die Quelldatei im aktuellen Fenster abspeichern und übersetzen:

```
<SAVE>  
<ENTER> mdefine Weihnacht weihmakro <EXECUTE>
```

und die MACRO-Bibliothek *weihmakro* laden:

```
<ENTER> #weihmakro <EDIT>
```

steht Ihnen das Makro *siegel* zur Verfügung. Sie belegen noch zwei Variablen:

```
<ENTER> ORT1=München <ENTER>  
<ENTER> ORT2=Berlin <ENTER>
```

Sie müssen dann eine Liste mit den entsprechenden Dateinamen einblenden lassen (siehe Kapitel 6: LEARN, Beispiel 2). Sie können sich die Dateinamen auch in einer eigenen Datei *liste* untereinander aufnotieren und sich diese Datei mit dem Kommando

```
<ENTER> @ liste <EDIT>
```

in die Pseudo-Datei EDITFILES einlesen lassen (siehe Abschnitt 4.2.4 Die Pseudo-Datei EDITFILES). Dann positionieren Sie die Schreibmarke auf dem gewünschten Dateinamen und können danach das Makro starten:

```
<ENTER> siegel $ORT1 gru1 <MACRO>
```

Wenn Sie einem/einer besonders guten Freund/in aus *Berlin* schreiben wollen, geben Sie einfach andere Parameter an:

```
<ENTER> siegel $ORT2 gru2 <MACRO>
```

## 7.2 Den Inhalt einer MACRO-Bibliothek anzeigen

Gelegentlich ist es erforderlich, an Makros und Textbausteinen, die Sie in einer MACRO-Bibliothek abgespeichert haben, Veränderungen vorzunehmen. Der MAXed stellt Ihnen dazu das Dienstprogramm *mlist* zur Verfügung. Sie können damit den Inhalt einer MACRO-Bibliothek in lesbarer Form auflisten lassen.

Der Kommandoaufruf für *mlist* hat folgendes Format:

```
mlist [-schalter] [MACRO-bibliothek1 [MACRO-bibliothek2 ...]]
```

### MACRO-bibliothek

Steht für die MACRO-Bibliothek, deren Inhalt Sie in einer lesbaren Form ausgeben lassen wollen. Wenn Sie keinen Dateinamen einer MACRO-Bibliothek angeben, wird der Inhalt der Standard-MACRO-Bibliothek aufgelistet.

Folgende Schalter stehen zur Auswahl:

**o**[utput] = Datei

Der übersetzte Inhalt der MACRO-Bibliothek(en) wird in die angegebene Datei *datei* geschrieben. Standardmäßig erscheint die Ausgabe auf Standard-Ausgabe (Dateikennzahl 1).

**h**[eaders]

Der Name der MACRO-Bibliothek(en) wird mit ausgegeben.

Das Dienstprogramm *mlist* können Sie sowohl von der Shell-Ebene aus als auch mit Hilfe des Kommandos EXECUTE aus dem MAXed heraus aufrufen. Die letztgenannte Möglichkeit bietet Ihnen den Vorteil, daß die Quelldatei über die Standard-Ausgabe in der Pseudo-Datei SYSOUTPUT aufgelistet wird, sodaß Sie sie gleich bearbeiten können (siehe Abschnitt 4.2.11 Die Pseudo-Datei SYSOUTPUT).

**Beispiel**

```
<ENTER> mlist -o=macedit1 mac1 mac2 <EXECUTE>  
<ENTER> macedit1 <EDIT>
```

Das erste Kommando listet die MACRO-Bibliotheken *mac1* und *mac2* in lesbarer Form in der Datei *macedit1* auf. Diese wird dann mit dem nachfolgenden Kommando zur Bearbeitung in das aktuelle Fenster geholt.

### 7.3 Eine neue Konfigurationsdatei erstellen

Eine MAXed-Konfigurationsdatei wird u.a. benötigt, um die Funktionen von MAXed auf den verwendeten Terminal-Typ abzustimmen. Normalerweise wird die Standard-Konfiguration benutzt. Sie wird mit dem MAXed-Paket mitgeliefert. Der MAXed sucht sie beim Aufruf im Verzeichnis *usr/lib/MAXed* unter dem Namen *mcfg.\$TERM* oder ersatzweise unter *max-config.std*, falls benutzer- und terminalspezifische Konfigurationsdateien nicht vorhanden sind (siehe unten).

Sie können die Konfiguration des MAXed an Ihre Erfordernisse anpassen. So können Sie z.B. die Kommandotasten entsprechend Ihren Bedürfnissen neu belegen, d.h. Tasten-Makros festlegen, Tabulatoren setzen, für verschiedene Programmiersprachen BLOCK-Grenzen definieren oder RECORD-Grenzen festlegen. Dazu gehen Sie am besten folgendermaßen vor:

- Kopieren Sie den Inhalt der Quelldatei der Standard-Konfigurationsdatei *usr/lib/MAXed/\$TERM.cfl* in eine eigene Datei.
- Bearbeiten Sie diese Datei, indem Sie die gewünschten Änderungen der Konfiguration vornehmen.
- Sichern Sie die Datei.
- Übersetzen Sie die Datei mit dem Dienstprogramm *mconfig* in eine ausführbare Konfigurationsdatei.
- Rufen Sie den MAXed mit dieser Konfigurationsdatei auf (siehe Abschnitt 5.1, maxed – Den MAXed aufrufen).

Das Dienstprogramm *mconfig* generiert aus dem Quellcode einer Konfigurationsdatei eine neue MAXed-Konfigurationsdatei.

Das Kommando für *mconfig* hat folgendes Format:

```
mconfig [quelldatei1 [quelldatei2 ...]] zieldatei
```

#### quelldatei

Die Quelldatei enthält eine MAXed-Konfigurationsbeschreibung in lesbarer Form. Geben Sie keine Quelldatei an, wird die Datei *maxed.cfg* herangezogen. Kann diese Datei nicht gefunden werden, wird eine Fehlermeldung ausgegeben. Wenn Sie mehrere Quelldateien verwenden, müssen Sie sicherstellen, daß keine Doppelvereinbarungen vorkommen.

#### zieldatei

Das ist der Name der Datei, in die die übersetzte Form der MAXed Konfiguration geschrieben wird. Mit dieser Datei wird gearbeitet, wenn Sie beim Aufruf von MAXed den Schalter *-c = zieldatei* angeben (siehe Abschnitt 5.1, *maxed* – Den MAXed aufrufen). Anstelle der Zieldatei können Sie das Minus-Zeichen *-* angeben. In diesem Fall wird die Information über die Konfiguration in die benutzer- und terminalspezifische Datei */usr/lib/MAXed/tmp/X\$USERttyf* geschrieben. Geben Sie beim Aufruf des MAXed mit dem Schalter *-c* keine Konfigurationsdatei an, so wird automatisch diese benutzer- und terminalspezifische Konfigurationsdatei benutzt.

### Beispiel

Um eine Konfigurationsdatei zu erstellen, die andere Schrittweiten für die Tabulatorsprünge und außerdem die BLOCK-Grenzen *BEGIN* und *END* enthält, gehen Sie wie folgt vor:

- Sie erstellen eine Kopie der Standard-Konfigurationsbeschreibung, z.B. mit dem Namen *\$TERM.cop1*. Die lesbare Datei *\$TERM.cfi* mit dem Quellcode der Standard-Konfigurationsdatei finden Sie im Dateiverzeichnis */usr/lib/MAXed*.
- Sie löschen die Zeile mit den alten Tabulatoren und schreiben dafür neu:  

```
TABS 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130
```

Damit haben Sie die Tabulatoren auf die Spalten 10, 20, 30, ... gesetzt.

- Sie fügen an die bereits vorhandenen BLOCK-Definitionen eine weitere Zeile mit der BLOCK-Definition *BEGIN* und *END* an:  
BLOCK 'BEGIN' ,=, 'END'
- Die veränderte Konfigurationsbeschreibung wird gesichert und dann mit dem Kommando  
mconfig \$TERM.cop1 myconf  
übersetzt. Sollten beim Übersetzen Fehler in der Quelldatei bemerkt werden, so werden entsprechende Fehlermeldungen angezeigt und die neue Konfigurationsdatei wird nicht erzeugt.
- Beim Aufruf des MAXed kann die neue benutzerspezifische Konfigurationsdatei mit  
maxed -c=myconf  
geladen werden.

## 7.4 Eine MAXed-Datei für den Ausdruck aufbereiten

Für die Druckaufbereitung von MAXed-Dateien wurde das Dienstprogramm *maxlpr* entwickelt. Mit diesem Programm werden alle in MAXed-Dateien gesetzten Zeichenattribute für die Ausgabe auf dem angeschlossenen Drucker aufbereitet (siehe Abschnitt 4.4 Die Zeichenattribute). Die Zeichenattribute werden – entsprechend den Möglichkeiten der Drucker – in spezielle Zeichendarstellungen umgesetzt (siehe Abschnitt 4.5.4 Ausdrucken von MAXed-Dateien). Außerdem werden nationale Sonderzeichen aus der "oberen Hälfte" des 8859/1-Zeichensatzes dem jeweiligen Druckertyp angepaßt. Diese Sonderzeichen geben Sie – entsprechend den Definitionen in der Konfigurationsdatei – mit Hilfe des MAXed-Kommandos FONT ein (siehe Kapitel 6: FONT).

Das Dienstprogramm *maxlpr* ist ein Filter-Programm. Es verarbeitet Texte aus MAXed-Dateien, deren Dateinamen Sie als Argumente beim Kommandoaufruf angeben, oder Eingaben von Standard-Eingabe (Dateikennzahl 0) entsprechend den angegebenen Schaltern. Standardmäßig ruft *maxlpr* automatisch die Druckroutine *lpr* auf, sodaß die Texte auf einem angeschlossenen Drucker ausgedruckt werden. Auf Wunsch können Sie sich die von *maxlpr* verarbeiteten Texte stattdessen auf Standard-Ausgabe (Dateikennzahl 1) ausgeben lassen.

Das Kommando für *maxlpr* hat folgende Syntaxdarstellung:

```
maxlpr [[-schalter1] datei11 [datei12 ...] ]
        [ -schalter2 datei21 [datei22 ...] ]
        ...
        [ -schalterN ]
```

Die Texte der angegebenen Dateien werden entsprechend der vorausgehenden Schalterangabe *schalterx* aufbereitet. Wenn der letzten Schalterangabe keine Dateinamen folgen, liest *maxlpr* von der Standard-Eingabe (Dateikennzahl 1). Sie beenden diese Eingabe durch das Zeichen für EOF (CTRL-D).

Für *schalterx* geben Sie einen oder mehrere der folgenden Schalter ein:

```
[-druckertyp] [-zeichenbreite] [-format] [-schacht]
[-zeichensatz] [-schriftart] [-ladbarer zeichensatz]
[-stdout]
```

Außer diesen *maxlpr*-spezifischen Schaltern können Sie Schalter für die Druckroutine *lpr* angeben, die *maxlpr* an diese weiterleitet. Im einzelnen stehen Ihnen folgende Werte für die *maxlpr*-spezifischen Schalter zur Auswahl:

#### -druckertyp

Der Parameter für den Druckertyp kann einen der folgenden Werte haben:

-1	für den Druckertyp 9001
-2	für den Druckertyp 9002
-4	für den Druckertyp 9003
-11	für den Druckertyp 9011
-12	für den Druckertyp 9012
-13	für den Druckertyp 9013
-22	für den Druckertyp 9022
-25	für den Druckertyp 9025
-47	für den Druckertyp 9047

Wenn Sie keinen Druckertyp angeben wählt ihn *maxlpr* selbst aus gemäß den Angaben in der *Spooler-CONFIG*-Datei im Verzeichnis */usr/spool/spooler* bzw. */usr/spool*: Der zuerst aufgeführte Druckertyp der ersten Druckergruppe wird verwendet.

#### Anmerkung

- Sie können beim Aufruf von *maxlpr* den Schalter `-dru=druckergruppe` angeben, der an *lpr* weitergereicht wird. *maxlpr* wählt dann den passenden Druckertyp gemäß den Angaben der *Spooler-CONFIG*-Datei.
- Sie listen sich die Angaben der *Spooler-CONFIG*-Datei auf mit `lpr -qdru`

**-zeichenbreite**

Für die Einstellung der Zeichenbreite steht Ihnen folgendes zur Auswahl:

- pb1** für Zeichenbreite mit 80 Zeichen  
(entspricht 10 Zeichen/Zoll beim Druckertyp 9001)
- pb2** für Zeichenbreite mit 96 Zeichen  
(entspricht 12 Zeichen/Zoll beim Druckertyp 9001)
- pb3** für Zeichenbreite mit 136 Zeichen  
(entspricht 17 Zeichen/Zoll beim Druckertyp 9001)

Standard-Einstellung ist **-pb1**

**-format**

Damit wählen Sie Hoch- oder Querformat für die Ausgabe aus (nur möglich beim Druckertyp 9022):

- p** steht für Hochformat ([p]portrait)
- l** steht für Querformat ([l]landscape)

Standard-Einstellung ist **-p**

**-schacht**

Mit diesem Parameter stellen Sie den Papiereinzugsschacht ein (nur möglich bei den Druckertypen 9022 und 9025):

- hop1** für Schacht eins
- hop2** für Schacht zwei

Standard-Einstellung ist **-hop1**

**-zeichensatz**

Die Angaben für die Einstellung des Zeichensatzes sind identisch mit jenen bei der Standard-Druckroutine *lpr* (nur möglich bei den Druckertypen 9022 und 9025).

**-schriftart**

Sie wählen aus zwischen Normal- und Schönschrift (nur möglich bei den Druckertypen 9001 bis 9031):

**-q** steht für Normalschrift ([d]raft [q]uality)

**-q** steht für Schönschrift ([l]etter [q]uality)

Standard-Einstellung ist **-dq**

**Anmerkung**

Der Schalter **-lq** ist nicht in Kombination mit den Schaltern **-pb2** oder **-pb3** erlaubt!

**-ladbarer Zeichensatz**

Für den Druckertyp 9022 muß vor jedem Ausdruck der verwendete Zeichensatz für die Ausgabe von MAXed-Grafikzeichen (Zeichenattribut *graphic*) geladen werden (siehe Abschnitt 4.4 Die Zeichenattribute). Dies geschieht automatisch. Diesen Vorgang können Sie durch Angabe von Schaltern beeinflussen:

**-gra=eu** Es werden zusätzlich 36 Zeichen aus der oberen Hälfte des 8859/1-Zeichensatzes geladen. Dies sind jene Zeichen des 8859/1-Zeichensatzes, die nicht standardmäßig in den nationalen Zeichensätzen des Druckertyps 9022 enthalten sind.

**-gra=no** Damit verhindern Sie ein Laden des speziellen Klammern-Zeichensatzes, das die Ausgabe von MAXed-Grafikzeichen, also insbesondere gezeichnete Linien, erlauben würde.

**-stdout** Bei Angabe dieses Schalters unterdrücken Sie den automatischen Aufruf der Druckroutine *lpr*. Die aufbereiteten Texte werden stattdessen auf Standard-Ausgabe (Dateikennzahl 1) ausgegeben.

### Anmerkungen zu den Druckertypen

#### Druckertyp 9004:

Dieser Druckertyp erlaubt nicht die Ausgabe von MAXed-Grafik (Zeichenattribut *graphic*). Von der "oberen Hälfte" des 8859/1-Zeichensatzes sind lediglich die deutschen Umlaute darstellbar.

#### Druckertyp 9022:

- Für die Ausgabe von Zeichen mit der Parameterangabe *-pb3* müssen Sie die entsprechende Schriftarten-Kassette einlegen.
- Zu Beginn eines jeden Druckvorgangs werden entsprechend der Angabe des Parameters *-gra* Zeichensätze geladen. Damit diese Daten von der Standard-Druckroutine *lpr* nicht abgeschnitten werden, sollten Sie *lpr* keine Schalter durchreichen lassen, die das Formatieren oder Positionieren betreffen,!
- Achten Sie darauf, daß der Drucker mit genügend Papier versorgt ist, da er bei einer Unterbrechung die geladenen Zeichensätze verlieren würde, sodaß der restliche Ausdruck für Sonderzeichen nur Ersatzdarstellungen liefern würde!

#### Druckertyp 9047:

Dieser Druckertyp unterstützt nicht den vollen 8859/1-Zeichensatz, sodaß er Zeichenattribute und nationalen Sonderzeichen nicht darstellen kann.

## 7.5 MAXed-Steuerprogramme

Zur Kommunikation eines MAXed-Steuerprogramms mit einem laufenden MAXed-Prozeß bietet der MAXed eine C-Schnittstelle mit den folgenden Funktionen:

**m\_control** (format, arg1, arg2, ...)

schickt Steuerzeichen und darstellbare Zeichen an den MAXed.

**m\_getkey** ()

liest eine Eingabe von der Tastatur und stellt sie durch Steuerzeichen bzw. ASCII-Zeichen dar.

**m\_getline** (buf, n)

fordert vom MAXed den Inhalt der aktuellen Zeile, höchstens *n* Zeichen.

**m\_getstatus** ()

fragt den MAXed-Status ab.

**m\_getparam** (buf1, buf2, n)

ersetzt Zeichenketten-Argumente des MAXed (vgl. Kapitel 6: ENTER).

### 7.5.1 Die C-Funktionen

In diesem Abschnitt finden Sie eine ausführlichere Beschreibung dieser Funktionen, welche die C-Schnittstelle des MAXed darstellen.

## Den MAXed steuern

```
m_control (format, arg1, arg2, ...)
char *format;
int arg1, arg2, ...;
```

Die Funktion *m\_control* wandelt die Argumente gemäß den Formatanweisungen um und speichert sie in einem internen Puffer. Diese Folge von Steuerzeichen und darstellbaren Zeichen wird an den MAXed geschickt und steuert so die MAXed-Sitzung.

### Parameter

**char \*format**

Formatzeichenfolge, die außer den darstellbaren Zeichen nachfolgende Formatangaben enthalten kann, die in der Reihenfolge ihres Auftretens durch die angegebenen Argumente ersetzt werden. Die Funktionsweise ist ähnlich der von Ein-/Ausgabe-Routinen der C-Standard-Bibliothek:

**%c** für die Ausgabe eines Zeichens

**%s** für die Ausgabe einer Zeichenfolge (abgeschlossen durch Nullzeichen)

**%C** für die Ausgabe eines Steuerzeichens für den MAXed

Die Steuerzeichen eines MAXed-Kommandos geben Sie folgendermaßen ein: Das Auslösen eines MAXed-Kommandos <KOMMANDO> wird durch die Kommandobezeichnung KOMMANDO repräsentiert. Diese Kommandobezeichnungen sind in der Datei *maxcontrol.h* als symbolische Konstanten definiert. *maxcontrol.h* befindet sich in dem Verzeichnis */usr/lib/MAXed*.

### Beispiel

**BACKSPACE** für die laut Referenzkarte für das Kommando <BACKSPACE> vorgesehene Taste.

Neben den bekannten MAXed-Kommandos aus Kapitel 6, sind für die Funktion *m\_control* zusätzlich die folgenden Kommandos definiert:

**QENTER**

Dient der Eingabe von Kommandoattributen wie das Kommando ENTER. Die Kommandoattribute werden jedoch nicht am Bildschirm ausgegeben.

**APPSTRING**

Dient dem Anfügen einer Zeichenkette ans Ende der Pseudo-Datei SYSOUTPUT (siehe Abschnitt 4.2.11 Die Pseudo-Datei SYSOUTPUT).

*Beispiel*

```
QENTER zeichenkette APPSTRING
```

**APPFILE**

Datei ans Ende der Datei SYSOUTPUT anfügen.

*Beispiel*

```
QENTER datei APPFILE
```

**DISPMES**

Dient der Ausgabe einer Meldung in der MAXed-Steuerzeile.

*Beispiel*

```
QENTER meldung DISPMES
```

**RINGBELL**

Betätigt die Klingel Ihrer Datensichtstation.

**Ergebnis**

int Anzahl der an den MAXed geschickten Zeichen (im Fehlerfall -1).

## Eingabe von der Tastatur lesen

### m\_getkey (0)

Die Funktion *m\_getkey* liest eine Eingabe von der Tastatur und stellt sie durch Steuerzeichen bzw. ASCII-Zeichen dar.

### Ergebnis

int Die Kommandonamen sind in der Datei *maxcontrol.h* als symbolische Steuerzeichen definiert. Diese Datei befindet sich entweder in dem Verzeichnis, dessen Name der Variablen *MAXlib* zugewiesen ist, oder ersatzweise in */usr/lib/MAXed*. Im Fehlerfall, d.h. im Fall einer nicht interpretierbaren Eingabe, liefert *m\_getkey* den Wert -1.

**Inhalt der aktuellen Zeile anfordern**

⌋

```
m_getline (buf, n)
char *buf;
int n;
```

Die Funktion *m\_getline* fordert vom MAXed den Inhalt der aktuellen Zeile an, höchstens aber *n* Zeichen. Die Zeichen stehen nach dem Aufruf des Kommandos im Speicher ab der Adresse, auf welche der Parameter *buf* zeigt. Sie müssen diesen Puffer selbst in ausreichender Größe definieren.

**Parameter**

⌋

```
char *buf
    Puffer, in den der Zeilertext eingelesen wird
```

```
int n
    Länge des Puffers
```

**Ergebnis**

```
int
    Länge der eingelesenen Zeile
```

**Den MAXed-Status abfragen.****m\_getstatus ()**

Die Funktion *m\_getstatus* fragt den MAXed-Status ab.

**Ergebnis**

int 0 bedeutet, daß das zuletzt aufgerufene MAXed-Kommando erfolgreich ausgeführt wurde.

int 1 bedeutet Fehler

## MAXed-Zeichenketten-Argumente ersetzen

```
m_getparam (buf1, buf2, n)
char *buf1, *buf2;
int n;
```

Die Funktion *m\_getparam* schickt den Inhalt von *buf1* an den MAXed, wo dieser entsprechend den Regeln für Textersetzung expandiert wird (siehe Abschnitt 3.6.4 Textersetzungen im Eingabefeld). Die expandierte Zeichenkette, höchstens aber  $n$  Zeichen werden in *buf2* geschrieben. Sie müssen diese Puffer selbst in ausreichender Größe definieren.

### Parameter

**char \*buf1**

Puffer, der eine Zeichenkette enthält, die auch Sonderzeichen für die Textersetzung im Eingabefeld enthalten darf.

**char \*buf2**

Puffer, in den die expandierte Zeichenkette aus *buf1* geschrieben wird.

**int n**

Maximale Länge der ausgewerteten Zeichenkette.

### Anmerkung

Sinnvoll erscheint, für "int n" den Wert der Pufferlänge von *char \*buf2* zu verwenden. Ein zu kleiner Wert kann zum Abschneiden des Endes der Zeichenkette führen!

### Ergebnis

int Länge der zurückerhaltenen expandierten Zeichenkette

**Beispiel**

Der Ergebnisbuffer sei folgendermaßen definiert:

```
#define PUFFERLAENGE 10
...
char buf2 [PUFFERLAENGE]
```

Der Funktionsaufruf

```
m_getparam ("@c", buf2, PUFFERLAENGE)
```

schreibt Zeile und Spalte der aktuellen Schreibmarkenposition, höchstens jedoch 10 Zeichen in den Puffer *buf2*.

## 7.5.2 MAXed-Steuerprogramme übersetzen und aufrufen

MAXed-Steuerprogramme müssen mit *maxcontrol.o* gebunden werden. *maxcontrol.h* befindet sich in dem Verzeichnis */usr/lib/MAXed*. Diese Objektdatei enthält die in Abschnitt 7.5.1 beschriebenen C-Funktionen.

Es gibt zwei Möglichkeiten, ein Steuerprogramm aufzurufen:

- während der MAXed-Sitzung wie ein Makro (siehe Kapitel 6: MACRO):

```
<ENTER> steuerprogramm [programm-parameter] <MACRO>
```

- beim Aufruf des MAXed (siehe Abschnitt 5.1 maxed – Den MAXed aufrufen):

```
m -pr[ogram]=steuerprogramm [datei1 [, datei2 ...]]
```

Wenn Sie nur einen relativen Pfadnamen für *steuerprogramm* angeben, erwartet der MAXed, daß dieses Steuerprogramm in einem bestimmten Dateiverzeichnis steht. Der Reihe nach durchsucht der MAXed folgende Dateiverzeichnisse:

1. Falls die Umgebungsvariable *MAXctpath* definiert ist, werden die darin angegebenen Dateiverzeichnisse in der Reihenfolge ihrer Angabe durchsucht.
2. Falls die Umgebungsvariable *MAXctpath* nicht definiert ist, wird zunächst unter dem aktuellen Dateiverzeichnis gesucht.
3. Falls die Umgebungsvariable *MAXlib* definiert ist, sucht MAXed in dem Dateiverzeichnis, dessen Name dieser Variablen zugewiesen ist.
4. Findet der MAXed auch hier das Programm nicht, sucht er zuletzt im Dateiverzeichnis */usr/lib/MAXed*.

Steuerprogramme laufen immer als Sohn-Prozesse des MAXed-Prozesses ab. Der MAXed ordnet dabei die Standard-Eingabe (Dateikennzahl 0), die Standard-Ausgabe (Dateikennzahl 1) und die Standard-Fehlerausgabe (Dateikennzahl 2) der Datei */dev/null* zu.

Wenn Sie mit Hilfe eines Steuerprogramms den Bildschirm modifizieren wollen, verwenden Sie die dafür vorgesehenen MAXed-Kommandos. Ein Steuerprogramm, das ohne Verwendung von MAXed-Kommandos direkt auf den Bildschirm ausgeben soll, muß vorher die Datei `/dev/tty` neu öffnen. Da der MAXed in diesem Fall von den Veränderungen am Bildschirm nichts weiß, müssen Sie anschließend mit der Funktion `m_control` ein `REFRESH`-Kommando an den MAXed schicken. Dann wird der Bildschirm wieder auf den Stand gebracht, den der MAXed intern gespeichert hat.

Wenn sich ein Steuerprogramm beendet hat, arbeitet der MAXed wieder wie gewohnt und erwartet Eingaben von der Tastatur. Steuerprogramme, die sich noch nicht selbst beendet haben, werden automatisch abgebrochen, wenn der MAXed das `EXIT`-Kommando erhält und damit die Sitzung beendet wird.

**Achtung**

Während ein Steuerprogramm aktiv ist, können Sie nicht mit dem Kommando `CLI` vorübergehend in die Sub-Shell wechseln (siehe Abschnitt 4.5.1 Wechseln in die CLI-Sub-Shell). Dieses Kommando wird dann ignoriert.

### 7.5.3 Beispiele zu Steuerprogrammen

#### Beispiel 1

Mit einem Programm soll verhindert werden, daß während der MAXed-Sitzung eine Subshell aufgerufen wird. Dazu muß das Kommando CLI abgefangen werden.

- Zunächst erstellen Sie das Quellprogramm in der Datei *inhbcli.c*:

```
#include "/usr/lib/MAXed/maxcontrol.h"

void
main ()
{
    int code;

    for (;;)
        if ((code = m_getkey()) == CLI)
            /* Meldung in die MAXed-Steuerzeile ausgeben */
            m_control ("%CKommando CLI ist gesperrt.%C", QENTER, DISPMES);
        else
            /* Steuerzeichen bzw. ASCII-Zeichen an MAXed schicken */
            m_control ("%C", (short int) code);
}
```

- Am besten legen Sie ein Dateiverzeichnis an, in dem Sie alle Steuerprogramme sammeln. Damit der MAXed in diesem Verzeichnis nach Steuerprogrammen sucht, müssen Sie der Variablen *MAXctpath* den Namen dieses Dateiverzeichnisses zuweisen, also z.B.

```
MAXctpath=/usr/harald/maxlib
export MAXctpath
```

- Das obige Quellprogramm übersetzen Sie mit dem folgenden SINIX-Kommando

```
cc -o inhbcli inhbcli.c /usr/lib/MAXed/maxcontrol.o
```

- Das übersetzte Programm führen Sie dann beim Aufruf des MAXed aus, wobei Sie nun nicht den absoluten Pfadnamen angeben müssen, wenn sich *inhbcli* im Verzeichnis */usr/harald/maxlib* befindet:

```
m -pr=inhbcli test.txt
```

Sie können jetzt die Datei *test.txt* wie gewohnt bearbeiten. Allerdings erhalten Sie während der MAXed-Sitzung eine Fehlermeldung, sobald sie das Kommando <CLI> eingeben.

### Anmerkung

Der MAXed würde einen Wechsel in die Sub-Shell sowieso nicht zulassen, wenn ein MAXed-Steuerprogramm läuft. Mit obigem Beispiel nehmen Sie Sie dem MAXed lediglich die Arbeit ab, eine eigene Fehlermeldung auszugeben.

### Beispiel 2

Anhand dieses Beispiels soll demonstriert werden, wie leicht Sie die MAXed-Kommandos mit Hilfe von Steuerprogrammen für eigene Zwecke, hier z.B. für eine einfache Dokumentenaufbereitung, einsetzen können. Da Kontrollstrukturen wie *for* und *if* benötigt werden, wäre das gleiche Programm als MACRO nicht realisierbar.

Im Beispiel wird ein absatzweise formatiertes Dokument mit einem Kopf versehen, der aus einer Überschrift, hier nur ein Wort, das als Parameter beim Aufruf des Steuerprogrammes übergeben werden muß, und dem aktuellen Datum besteht (siehe Abschnitt 4.3.3 Flatter- und Blocksatz). Außerdem werden Seitenvorschubzeichen eingefügt. Eine Erweiterung des Programms z.B. auf die Übergabe der Randinformation als Parameter mit nachfolgender Formatierung, die Ausgabe mehrerer Kopf- und Fußzeilen, etwa mit Hilfe von Textbausteinen ist leicht auszuführen (siehe Abschnitt 7.2). Auch die Ausgabe der Seitennummer durch Verwendung des Kommandos REPLACE mit variabler Ersetzungszeichenkette läßt sich leicht realisieren (vgl. Kapitel 6: REPLACE, Beispiel 3)

- Erstellen Sie folgendes Quellprogramm in der Datei *dokpr.c*:

```

#include "/usr/lib/MAXed/maxcontrol.h"
#define LAENGE "60"

void
main (argc, argv)
int argc;
char **argv;

    /* argv[1] enthaelt ein Wort fuer den Titel */
{
    char send_buf [80], recv_buf [80];

    /* Datum holen */
    sprintf (send_buf, "@d");
    m_getparam (send_buf, recv_buf, 80);

    /* Eingabe-Modus = INSERT-Modus */
    m_control ("%Ci%C", QENTER, INSERTMODE);

    /* Zeilenschaltungs-Modus = NORMAL */
    m_control ("%Cn%C", QENTER, RETURN);

    /* Gehe zur ersten Zeile der Datei */
    m_control ("%C%C%C", GOTO, SPACE, RETURN);

    for (;;)
    {
        /* Titel und Datum einfuegen */
        m_control ("%C%C%C%C%C%C%C%C%C%C%C%C%C%C",
            SPACE, SPACE, SPACE, RETURN, RETURN, argv[1],
            QENTER, "10", RIGHTARROW, recv_buf, WRAP, SPACE, SPACE);

        /* Auf Seitenende positionieren */
        m_control ("%C%C", QENTER, LAENGE, LINEFEED);

        /* Dateieneende erreicht ? */
        m_control ("%C^%C", QENTER, PLUSFIND);
        if (m_getstatus () != 0)
        {
            /* Ende */
            m_control ("%CEnde der Seitenaufbereitung%C", QENTER, DISPMES);
            exit (0);
        }
        else
        {
            /* Seitenvorschubzeichen ausgeben */
            m_control ("%C%C%CL%C", RETURN, SPACE, CODE, RETURN);
        }
    }
}

```

## MAXed-Steuerprogramme

---

- Übersetzen Sie das Programm mit dem Kommando  
cc -o dokpr dokpr.c /usr/lib/MAXed/maxcontrol.o
- Das übersetzte Programm rufen Sie auf mit  
<ENTER> dokpr überschrift <MACRO>

### Beispiel 3

Im folgenden Beispiel sehen Sie, wie das Kommando  
<ENTER> + datei <EDIT>  
als Steuerprogramm implementiert werden kann. Dieses Kommando  
bewirkt, daß eine Datei nur dann zur aktuellen Datei wird, wenn sie beim  
Aufruf des Kommandos noch nicht existiert. Andernfalls wird eine Fehler-  
meldung ausgegeben.

- Erstellen Sie folgendes Quellprogramm in der Datei *plusedit.c*:

```
#include "/usr/lib/MAXed/maxcontrol.h"

void
main (argc, argv)
int argc;
char **argv;

{
    int stat;
                                /* Prüfen, ob Datei existiert */
    stat = access (argv[1], 0);
    if (stat == 0)
    {
                                /* Meldung in die MAXed-Steuerzeile ausgeben */
        m_control ("%C%CDatei existiert bereits%C",
                   RINGBELL, QENTER, DISPMES);
    }
    else
    {
                                /* Neue Datei erstellen */
        m_control ("%C%s%C", QENTER, argv[1], EDIT, EDIT);
    }
    exit (0);
}
```

- Übersetzen Sie das Programm mit dem Kommando  
cc -o plusedit plusedit.c /usr/lib/MAXed/maxcontrol.o
- Das übersetzte Programm rufen Sie auf mit  
<ENTER> plusedit datei <MACRO>

## A Anhang

### Maximalwerte bei der Dateigröße

Maximale Zeilenlänge: 9999 Spalten

Maximale Zeilenzahl:

bei Systemen mit `sizeof(int) <= 2` 32 000 Zeilen  
sonst 99 999 Zeilen

### Erstellen einer Tag-Datei

Der MAXed stellt bei seiner Installation eine *public-domain*-Version des Kommandos *ctags* bereit, falls auf Ihrem System dieses Kommando nicht bereits verfügbar ist. Die Quelldatei dieses Programms finden Sie im Verzeichnis */usr/lib/MAXed*. Für dieses Programms wird keine Gewährleistung übernommen. Der MAXed stellt also sicher, daß das Kommando *ctags* vorhanden ist.

Eine Tag-Datei erstellen Sie auf der Shell-Ebene mit folgendem Kommando:

```
ctags [-BFaetuwx] dateiname(n)
```

Erzeugt zu den angegebenen Dateien eine Tag-Datei, in der die Definitionsstellen von Prozeduren in den jeweiligen Quelldateien aufgeführt sind. Standardmäßig hat diese Tag-Datei den Namen *tags*.

Das MAXed-Kommando

```
<ENTER> '[proc] <GOTO>
```

(siehe Abschnitt 4.2.15 Die Pseudo-Datei TAGLIST, Kaptiel 6: GOTO) greift auf eine Datei *tags* zu. Wenn Sie einer Tag-Datei jedoch einen anderen Namen gegeben haben, müssen Sie die Umgebungsvariable *MAXtags* mit dem Namen besetzen, damit der MAXed auf diese Datei zugreift:

```
<ENTER> MAXtags=dateiname <ENTER>
```

oder vor Aufruf des MAXed auf der *Bourne*-Shell:

```
MAXtags=dateiname  
export MAXtags
```

Dies können Sie ggf. in der benutzerspezifischen Datei *.profile* aufnehmen.

### convert - Dateikonvertierung

Der MAXed wurde von V2.0 auf V3.0 auf den Standard-8-Bit-Zeichensatz *ISO-8859/1* umgestellt. Dies betrifft die Kodierung von Sonderzeichen aus der "oberen Hälfte" des Zeichensatzes, die Sie mit dem Kommando `FONT` eingeben können (siehe Kapitel 6: `FONT`). MAXed-Dateien sind jedoch aufwärtskompatibel, d.h. Sie können mit der Version 3 des MAXed alte, mit der Version 2 erstellte Dateien weiterbearbeiten. Neu eingegebene Sonderzeichen werden stets im *8859/1*-Format gespeichert. Zur Konvertierung von Dateien zwischen den beiden Versions-Formaten wird im Verzeichnis `/usr/lib/MAXed` die Prozedur `convert` mitgeliefert.

`convert` [-r] dateiname(n)

Wandelt alle Sonderzeichen aus der "oberen Hälfte" des *ISO-8859/1*-Zeichensatzes in den angegebenen Dateien um. Wenn Sie keinen Schalter angeben, findet eine Konvertierung von V2.0 nach V3.0 statt: Sonderzeichen wie Umlaute, ß oder § werden im *8859*-Format abgespeichert.

- r Bei Angabe dieses Schalters findet eine Rückkonvertierung von V3.0 nach V2.0 statt.

### Makros und Steuerprogramme aus der Version 2

Makros und MAXed-Steuerprogramme, die Sie mit MAXed V2.0 erstellt haben, müssen Sie zur Wiederverwendung mit MAXed V3.0 neu übersetzen.

Quelldateien für MAXed-Steuerprogramme:

Passen Sie vor der Neuübersetzung den Aufruf der Funktion *m\_control* der veränderten Syntax an (siehe Abschnitt 7.5.1 Die C-Funktionen).

Quelldateien für MACRO-Bibliotheken:

Beachten Sie, daß `<FONT>` in der Version 3 kein Schlüsselwort mehr ist und deshalb nicht in Quelldateien für MACRO-Bibliotheken verwendet werden kann (siehe Abschnitt 7.1.1 Makros in einer MACRO-Bibliothek anlegen). Tauschen Sie in alten Quelldateien ggf. Zeichenketten, die Sonderzeichen darstellen, wie z.B.

`<FONT>a` bzw. `<FONT><331>`

aus durch die direkte Eingabe des jeweiligen Sonderzeichens mit dem Kommando `FONT`.

MACRO-Bibliotheken:

Wenn Sie zu MACRO-Bibliotheken keine Quelldateien besitzen, müssen Sie die Bibliotheken zuerst mit dem Dienstprogramm *mlist* aus der Version 2 in lesbare Form übersetzen, bevor Sie sie wie – oben beschrieben – verändern und mit dem Dienstprogramm *mdefine* der Version 3 neu übersetzen können!

## Fachwörter

### Absatz

Als Absatz identifiziert der MAXed einen Textbereich, der durch mindestens eine vorangestellte und eine nachgestellte Leerzeile gekennzeichnet ist.

### Alternativ-Datei

Bezeichnet die Datei, die unmittelbar vor der aktuell bearbeiteten Datei in demselben Fenster bearbeitet wurde.

### Arbeitsmodus

Der MAXed ermöglicht bei der Texteingabe die Wahl zwischen unterschiedlichen Eingabe-Modi. Eingegebene Zeichen überschreiben im Überschreibe-Modus bereits vorhandenen Text oder werden im Einfüge-Modus in ihn eingefügt. Die Behandlung von Leerzeichen sowie die Durchführung des Zeilenumbruchs, ggfs. mit automatischer Silbentrennung, hängt von der Einstellung des Einfüge-Modus ab.

### blinking

Das Zeichenattribut *blinking* bewirkt die blinkende Darstellung von Zeichen.

### BLOCK

Bezeichnet einen Textbereich, der vom MAXed als STREAM behandelt wird. Anfang und Ende des STREAM werden durch in der Konfigurationsdatei festgelegte Anfangs- und Endmarken automatisch erkannt. Bei Kommandos, die auf einen BLOCK zugreifen, wird automatisch die Symmetrie der Anfangs- und Endmarken geprüft, d.h. jeder Anfangsmarke muß eine Endmarke zugeordnet sein.

### Blocksatz

Der Text ist an den Formaträndern links- und rechtsbündig formatiert. Ggfs. werden dazu in den einzelnen Zeilen Leerzeichen eingefügt.

### FILTER

Ein FILTER ist ein Programm, das seine Eingabedaten von der Standard-Eingabe liest, sie bearbeitet und auf die Standard-Ausgabe zurückschreibt.

### Flattersatz

Der Text ist linksbündig formatiert. Überflüssige Leerzeichen werden beim Formatieren gelöscht.

### **graphic**

Das Zeichenattribut *graphic* bewirkt die Darstellung von Grafikzeichen.

### **IDENTIFIER**

Ein IDENTIFIER ist eine Zeichenkette aus alphanumerischen Zeichen oder dem Unterstrich. Sie dient zur Identifizierung von Dateien, benannten Puffern, Makros oder Marken. Das erste Zeichen muß ein Buchstabe oder der Unterstrich sein.

### **INSERT-Modus**

Mit INSERT-Modus wird der Standard-Einfüge-Modus zum Einfügen von Text bezeichnet. Der MAXed kennt vier verschiedene Einstellungen des Einfüge-Modus. Prinzipiell wird dabei Text, der sich rechts von der Einfügeposition der Schreibmarke befindet, nach rechts verschoben. Die Behandlung der Leerzeichen sowie die Durchführung des Zeilenumbruchs, ggfs. mit automatischer Silbentrennung, hängt von der jeweiligen Einstellung ab.

### **intensity**

Das Zeichenattribut *intensity* bewirkt die halbhelle Wiedergabe von Zeichen.

### **Kommandoattribut**

Sie können MAXed-Kommandos genauere Angaben in Form von Kommandoattributen begeben. Sie leiten eine Attributierung durch das Kommando <ENTER> ein. Attributierungen unterscheidet man grundsätzlich in Kommandoverstärkung und Kommandospezifizierung. Letzteres umfaßt z.B. Bereichsdefinitionen und Angabe von IDENTIFIER.

### **Konversationsmodus**

Konversationsmodus bezeichnet die eingestellte "Dialogsprache". Zur Auswahl stehen Englisch oder Deutsch. Je nach Wahl erscheinen die Hilfsinformationen in englischer oder deutscher Sprache.

### **LABEL**

Bezeichnet eine unsichtbare Marke im Text, die zum schnellen Auffinden wichtiger Textstellen gesetzt werden kann.

### **Lernphase**

Als Lernphase wird der Zeitraum zwischen Beginn und Ende einer Makro-Definition bezeichnet, die mit dem LEARN-Kommando während des normalen Editierens eingeleitet und abgeschlossen wird.

### **Makro**

Bezeichnet die Zusammenfassung mehrerer MAXed-Kommandos, die durch einen Aufruf nacheinander ausgeführt werden. Innerhalb eines Makros kann auch normaler Text stehen.

### **MAXed-Bibliothek**

Das Dateiverzeichnis */usr/lib/MAXed*, die MAXed-Bibliothek, wird bei der Installation eingerichtet. Das Dateiverzeichnis enthält u.a. die Dateien mit den Hilfsinformationen, die Standard-MACRO-Bibliothek und die Quelldatei für die MAXed-Konfigurationsdatei.

### **PATTERN**

Bezeichnet einen regulären bzw. erweiterten regulären Ausdruck, ein Muster, nach dem bei Such- und Ersetzungsoperationen gesucht wird. Sollen Sonderzeichen (nicht-alphanumerische) Zeichen als solche im Suchmuster erhalten bleiben, so sind sie durch einen vorangestellten Gegerschrägstrich `\` zu entwerfen.

### **Protokolldatei**

Während der Sitzung eingegebene Zeichen und durchgeführte Operationen werden in einer speziellen Datei mitgeschrieben. Soll nach einer nicht ordnungsgemäß beendeten Sitzung diese rekonstruiert werden, so geschieht dies mit Hilfe der in der Protokolldatei gespeicherten Informationen. Wird der MAXed normal geladen, wird die Protokolldatei der vorangegangenen Sitzung gelöscht.

### **Pseudo-Dateien**

Der MAXed verwaltet während der laufenden Sitzung eine Reihe von Dateien, wie z.B. RESTORE, die mit @ angezeigten Datei- und Begriffslisten oder die SYS-Dateien. Diese Dateien werden alle nicht explizit angelegt und dienen im wesentlichen zur raschen Information des MAXed-Benutzers. Sie werden bei Beendigung der Sitzung nicht abgespeichert.

### **Puffer**

Bezeichnet einen Zwischenspeicher, der einem bestimmten Kommando, z.B. PICK, ERASE, zugeordnet ist. Es können Daten aus dem Puffer gelesen oder in den Puffer geschrieben werden. Man unterscheidet zwischen Standard-Puffer und benannten Puffern. Alle Eingaben für einen benannten Puffer durchlaufen auch den Standard-Puffer. Bei Aufruf des entsprechenden Kommandos wird – wenn nicht durch die Kommandoattributierung anders bestimmt – automatisch der Standard-Puffer angesprochen. Die während der Sitzung angelegten Zwischenspeicher werden beim Beenden des MAXed gelöscht.

### **RANGE**

Kennzeichnet einen zusammenhängenden Zeilen-Textbereich. Ein RANGE umfaßt über die angegebenen Zeilen die gesamte Breite der Datei. Anfang und Ende werden durch Bewegungen der Schreibmarke oder durch ein Zeichenkettenargument festgelegt.

### **RECORD**

Bezeichnet einen Textbereich, der vom MAXed als STREAM behandelt wird. Anfang und Ende des STREAM werden durch in der Konfigurationsdatei festgelegte Anfangs- und Endemarken automatisch erkannt.

### **RECTANGLE**

Kennzeichnet einen rechteckigen Textbereich. Anfang und Ende werden durch Bewegungen der Schreibmarke oder durch ein Zeichenkettenargument festgelegt.

### **Regulärer/Erweiterter regulärer Ausdruck**

Ein regulärer Ausdruck ist ein Suchmuster (PATTERN). Statt einer festen Zeichenkette enthält das Suchmuster bestimmte Sonderzeichen.

### **RESTORE-Datei**

Die Pseudo-Datei RESTORE enthält alle Textbereiche, die während der Sitzung verändert oder kopiert wurden. Sie wird bei Beendigung der Sitzung nicht gesichert.

### **reversed**

Das Zeichenattribut *reversed* bewirkt die inverse Darstellung von Zeichen.

### **Schlüsselbegriff**

Mit Schlüsselbegriff werden alle Ausdrücke, insbesondere Kommandonamen, bezeichnet, die in Makros eingesetzt werden.

### **Standard-Ein-/Ausgabe**

Die Shell nimmt an, daß im Normalfall die Ein- und Ausgabe mit der Datensichtstation verbunden sind, von der aus ein Kommando eingegeben wurde, d.h. die Eingabe kommt von der Tastatur, die Ausgabe geht auf den Bildschirm.

### **Status-Datei**

Die benutzer- und terminalspezifische Status-Datei enthält die wichtigsten Informationen der laufenden Sitzung. Diese werden beim Beenden der Sitzung automatisch gesichert und zu Beginn der nächsten Sitzung als Voreinstellung genommen. Die Status-Datei enthält u.a. Informationen zu den während der Sitzung geöffneten Dateien, insbesondere Zugriffsrechte, linker und rechter Formatrand, Position von Fenster und Schreibmarke, zum eingestellten Eingabe-Modus und zur Einstellung von <RETURN>.

### **Steuerprogramm**

MAXed-Steuerprogramme sind C-Programme, die eine MAXed-Sitzung beeinflussen. Durch MAXed-Steuerprogramme können Sie wie mit Makros eine Reihe von MAXed-Kommandos absetzen lassen. Jedoch steht Ihnen hierbei im Gegensatz zu Makros der volle Sprachumfang der Programmiersprache C zur Verfügung. Dies verleiht MAXed-Steuerprogrammen größere Flexibilität. Solche Programme können die volle Kontrolle über den MAXed übernehmen, indem sie Ihre Eingabe von der Tastatur während einer MAXed-Sitzung erhalten und in modifizierter Form an den MAXed weiterreichen.

### **STREAM**

Ein STREAM kennzeichnet eine zusammenhängende Zeichenkette. Anfang und Ende werden durch Bewegen der Schreibmarke oder durch Argumenteingabe festgelegt.

### **underscore**

Das Zeichenattribut *underscore* bewirkt die unterstrichene Darstellung von Zeichen.

### **Wort**

Als Wort wird im MAXed jede Zeichenkette betrachtet, der ein nicht alphanumerisches Zeichen voran- und nachgestellt ist.

### **Zeichenattribut**

Zeichenattribute sind Darstellungsmerkmale, wie Unterstreichung, mit denen Sie Textbereiche versehen können. Sie haben auch beim Ausdruck des Textes einen Einfluß auf die Darstellung.



## Literatur

Die mit \* gekennzeichneten Titel sind nicht von der Siemens AG herausgegeben.

- [1] Betriebssystem SINIX V5.2  
**CES**  
C-Entwicklungssystem  
Beschreibung Teil 1 und 2
- [2] Betriebssystem SINIX V5.2  
**Einführung**  
Benutzerhandbuch
- [3] Betriebssystem SINIX V5.2  
**Systemverwaltung**  
Benutzerhandbuch
- [4] Betriebssystem SINIX V5.2  
**Kommandos**  
Beschreibung Teil 1 und 2
- [5]\* Kernighan, Brian W./Ritchie, Dennis M.  
**Programmieren in C**  
München, Wien 1983

### Bestellen von Handbüchern

Die aufgeführten Handbücher finden Sie mit ihren Bestellnummern im *Druckschriftenverzeichnis Datentechnik*. Dort ist auch der Bestellvorgang erklärt. Neu erschienene Titel finden Sie in den *Druckschriften-Neuerscheinungen Datentechnik*.

Beide Veröffentlichungen erhalten Sie regelmäßig, wenn Sie in den entsprechenden Verteiler aufgenommen sind. Wenden Sie sich bitte hierfür an Ihre zuständige SIEMENS-Zweigniederlassung; außerhalb der Bundesrepublik Deutschland hilft Ihnen die zuständige SIEMENS-Vertretung.



---

## Stichwörter

.bak-Datei 3-3, 3-8, 5-8, 6-48, 6-122

.sav-Datei 3-6, 5-8

ISO-8859/1-Zeichensatz A-3

keystroke-Datei 5-8

tag-Datei 6-62

% Prozent-Zeichen 3-46, 3-56

<-Such-Modus 4-23

>-Such-Modus 4-23

<>-Such-Modus 4-24

\-Such-Modus 4-22

\_ -ERASE-Puffer 3-76

\_ -PICK-Puffer 3-85, 3-87

Absatz, aktueller 3-151

absoluter Pfadname 6-21

abspeichern (siehe sichern)

ändern, Eingabe-Modus 3-41

aktives Fenster 3-25, 6-12

aktivieren, Zeichenattribute 4-81

aktuelle Datei 3-9f, 3-24, 3-32, 6-28, 6-66

aktuelles Fenster 3-25, 3-30, 3-32, 3-152, 6-28

– teilen 3-26, 6-66, 6-143

aktuelles Makro 3-95, 3-157, 6-74, 6-83

aktuelles PATTERN 3-161, 6-96

ALIGN - Text ausrichten 4-73, 6-2

Alternativ-Datei 3-22, 3-24, 3-27, 3-153

Angaben, optionale 1-7

anlegen

– MACRO-Bibliothek - Beispiel 7-9

– Makro- LEARN 6-74, 7-4

– neue Datei 3-2

– Textbaustein in MACRO-Bibliothek 7-7

anspringen, bestimmte Textstelle 6-60

anzeigen

– Datei - EDIT 6-28

– Dateianfang 6-60

– Dateiverzeichnis - DEFDIR 6-21

– MACRO-Bibliothek 7-11

– Zeile 6-60

APPEND - Ausgabe-Modus 4-28, 4-96, 6-45

Argument 3-37

– optionales 5-2

ARROW-Kommandos 3-13

– DOWNARROW 3-13

– LEFTARROW 3-13

– RIGHTARROW 3-13

– UPARROW 3-13

ASCII-Code 6-15f

asynchrone SINIX-Kommandos 4-94

asynchrones Programm 6-146

attributieren, Kommando 3-36

ATTRIBUTE 4-87, 6-3f

ATTRIBUTE-Feld 3-35, 4-81, 6-125f

attributieren, Bereich 3-102

aufrufen

– FILTER-Programm - FILTER 6-53

– MAXed 5-1f

– MAXed - mit Schalter 5-3

– MAXed-Steuerprogramm 7-29

– Shell-Kommandos - EXECUTE 6-44

– SINIX-Kommandos 4-90

– Subshell 6-13

aufsetzen auf gespeicherter Dateiver-  
sion 3-5

aufspalten, Makros 4-30

Aufwärtskompatibilität A-3

Ausdruck

– Datei aufbereiten für 7-16

– erweiterter regulärer 3-135

– regulärer 3-68, 3-86, 3-127, 3-132

- 
- ausdrucken 7-1
    - MAXed-Datei 4-101
    - Text - PRINT 6-102
    - vollständige Datei 6-103
  - Ausgabe, Standard 4-28, 4-95
  - Ausgabe-Modus
    - APPEND 4-28, 4-96
    - ECHO 4-29, 4-96
    - NOECHO 4-29
    - OVERWRITE 4-28
  - ausrichten, Text links- und rechtsbündig - JUSTIFY 6-71
  - austauschen, Textbereiche 3-82
  - auswerten, Zeichenkette 3-58, 3-60
  - automatischer Zeilenumbruch 4-68
  
  - BACKSPACE** 6-6
  - BACKTAB** 6-7
  - bearbeiten, Dateiausschnitte in mehreren Fenstern 3-25
  - beenden
    - Makro-Lernphase - LEARN 6-74
    - MAXed 3-5, 3-8
    - MAXed - QUIT 6-108
  - benannte Marke 3-91
  - benannter ERASE-Puffer 4-46
  - benannter PICK-Puffer 3-83, 3-87, 4-47
  - benanntes Makro 3-95, 6-83
  - Bereich 3-37f
    - attributieren 3-102
    - BLOCK 3-101
    - definieren 3-99
    - definieren durch Bewegen der Schreibmarke 3-105
    - definieren durch Zeichenketten-Attribut 3-112
    - RANGE 3-100, 3-104
    - RECORD 3-101
    - RECTANGLE 3-101, 3-104
    - STREAM 3-101, 3-104
  - Bereichsdefinition 3-42
    - voreingestellte 3-149
  - Bereichsgrenzen 3-102
  
  - Betriebs-Modus 3-51
  - Betriebssystem 6-14
    - Kommando lpr 6-102
  - bewegen
    - Schreibmarke 3-12f, 6-33, 6-64
    - Schreibmarke an Zeilenanfang - STARTLINE 6-133
    - Schreibmarke ans Ende der vorigen Zeile - UPEND 6-141
    - Schreibmarke in nächste Zeile - RETURN 6-116
    - Schreibmarke nach links - LEFTARROW 6-79
    - Schreibmarke nach oben - UPARROW 6-139
    - Schreibmarke nach rechts - RIGHTARROW 6-119
    - Schreibmarke nach rechts - TAB 6-134
    - Schreibmarke nach unten - DOWNARROW 6-23, 6-81
  - Bildschirm 3-29, 3-34
    - geteilter 6-13
    - neu schreiben - REFRESH 6-109
  - Bildschirmeditor 1-1
  - Bildschirmrand
    - linker 3-33
    - oberer 3-33
    - rechter 3-33
    - unterer 3-33
  - BLOCK-Bereich 3-101, 6-88, 6-97
    - definieren 3-119
  - Blocksatz - FORMAT 4-69, 6-57
  - Bullet 6-15
  
  - C-Funktion 7-21
  - C-Programm 4-107
  - C-Quelldatei 1-8
  - C-Quellprogramm 6-83
  - C-Schnittstelle 7-21
  - CANCEL 6-9
  - CENTER 6-10

---

CHANGEWINDOW Fenster-Modus  
4-20

CHLINELEN 3-32, 6-11

CHWINDOW Fenster-Modus 3-26,  
6-12, 6-66, 6-143

CLI 6-13f

– Sub-Shell 4-92

CNTRL Steuerzeichen 6-15f

CODE 1-8, 6-15f

COMPARE 6-18

COMPRESS - Text vereinheitlichen  
4-73, 6-20

CONTINUE Abarbeitungs-Modus  
4-25f, 6-45

convert A-3

#### Datei

– .bak 3-3, 3-5, 3-8, 5-8, 6-48, 6-122

– .sav 3-6, 5-8

– *keystroke* 5-8

– *tag* 4-58, 6-62

– aktuelle 3-9f, 3-24, 3-32, 6-28, 6-66

– alternative 3-22, 3-24, 3-27, 3-153,  
6-28

– auf früheren Versionsstand zurück-  
setzen 3-11

– ausdrucken 4-101

– einblenden 3-6, 3-9, 3-26

– erstellen - EDIT 6-28

– für Ausdruck aufbereiten 7-16

– Hilfe- 3-7

– Konfiguration 5-5, 6-11, 6-49, 6-55

– laden 3-2, 3-21

– neu anlegen 3-9

– neu einblenden 3-26

– Original- 3-2

– Quell 6-49

– sichern - SAVE 6-122

– Status 3-6, 5-5, 6-48, 6-72, 6-131

– temporäre 3-2, 3-5, 3-10

– überschreiben 3-10

– wechseln 3-21

Dateiausschnitt verschieben 3-20

– spaltenweise - WIN 3-20

– absolut - HOME 3-20

– zeilenweise nach unten -  
LINEFEED 6-81

– zeilenweise nach oben -  
MINUSLINE 6-91

– seitenweise nach oben -  
MINUSPAGE 6-92

– zeilenweise nach unten -  
PLUSLINE 6-100

– seitenweise nach unten -  
PLUSPAGE 6-101

– zeilenweise nach oben -  
RVLINFEED 6-121

– nach links - WINL 6-147

– nach rechts - WINR 6-148

Dateiausschnitte in mehreren Fenstern  
bearbeiten 3-25

Dateien

– nach Begriff durchsuchen 6-30

Dateihaltung 3-2

Dateiinhalte kopieren 3-72

Dateikonvertierung A-3

Dateiname 3-67

– als IDENTIFIER 3-67

– Konventionen 3-68, 3-98

Dateiverzeichnis 6-21

– anzeigen - DEFDIR 6-21

– wechseln - DEFDIR 6-21

DEFDIR 6-21

definieren, Bereiche 3-99

definierter PICK-Puffer 3-88

DELCHAR 6-22

Dienstprogramm 7-1

– maxlpr 6-4, 6-27, 6-56, 6-102, 7-16

– mconfig 6-49, 7-13

– mdefine 6-76, 7-2

– mlist 7-11

Dokumentation, On-Line 4-40, 6-63

DOWNARROW 6-23ff

DRAW - Linien zeichnen 4-12, 4-88,  
6-25, 6-27

---

Drucker 7-16  
 Druckertypen 7-20

**ECHO** Ausgabe-Modus 4-29, 4-96, 6-45  
**EDIT** 3-10, 6-28ff  
**EDITFILES**, Pseudo-Datei 3-69, 4-43, 6-29f  
**EDITMODE** 4-11  
 einblenden
 

- Datei 3-6, 3-9, 3-26
- Datei im neuen Fenster 3-26
- Kommandozeichnung 1-9

**Einfüge-Modus** 4-2
 

- **WORD** 4-8
- **INSERT** 3-35, 4-2, 4-9f, 6-6, 6-69
- **LEXICAL** 4-3, 4-6, 4-10, 4-34
- **STREAM** 4-3, 4-11, 6-18
- **WORD** 4-3

 einfügen, Leerzeichen - **SPACE** 6-129  
 Eingabe von der Tastatur lesen - **m\_**-**getkey** 7-24  
 Eingabe, Standard- 4-95  
 Eingabe-Modus 3-35, 4-2
 

- im Eingabefeld ändern 3-41
- wählen - **INSERTMODE** 6-69

 Eingabefeld
 

- Steuerzeile 3-41f
- Textersetzung in 3-46, 3-58

 Eingaben in eine Datei annullieren 6-31  
 Eingabepuffer 3-42, 6-7
 

- Kommandoattribute speichern in 3-43
- Steuerzeile 3-41

 eingeben
 

- Kommandoattribute 3-37, 6-34, 6-111
- Kommandoname 3-39
- Steuerzeichen - **CODE** 6-15
- Zeichenketten-Attribute 3-38

 einstellen, Zeichenattribute 4-81, 6-18, 6-99

**EMPTY** Zeilenschaltungs-Modus 4-15, 6-117  
**ENDLINE** 6-33  
**ENTER** 6-34  
 entfernen, Zeichenattribute 4-87  
 entwerten, Sonderzeichen 3-58  
**ERASE** 6-38, 6-41  
**ERASE-Puffer** 3-78, 6-38, 6-114
 

- benannter 3-75, 4-46, 6-38
- Inhalt kopieren - **RESTOR** 6-114
- kopieren 3-81
- mehrmals kopieren 6-115
- Standard- 3-75, 3-154, 6-38
- unbenannter 3-75

**ERASE-Puffername** als **IDENTIFIER** 3-74  
**ERASEBUFFERS**, Pseudo-Datei 3-78, 4-46, 6-115  
 ersetzen
 

- **MAXed**-Variablen 3-53
- **PICK**-Puffernamen 3-56
- **SINIX**-Kommandoaufrufe 3-57
- Textstellen - **REPLACE** 6-110
- Zeichenkette 3-136, 3-145
- Zeichenketten-Argumente - **m\_**-**getparam** 7-27

 Ersetzungszeichenkette 3-138, 3-162  
 erstellen
 

- Datei - **EDIT** 6-28
- Konfigurationsdatei 7-13
- **MACRO**-Bibliothek 7-2
- Makros 4-35

 erweiterter regulärer Ausdruck 3-127, 3-135  
**EXAPPEND** 4-29  
**EXCHANGE** 6-41  
**EXCONTINUE** 4-27  
**EXECUTE** 4-26, 4-52, 4-54, 4-94, 6-44, 6-146  
**EXIT** 3-8, 6-48

**Fenster**

- aktives 6-12

- aktuelles 3-25, 3-30, 3-32, 3-152, 6-28
- Dateiauschnitte bearbeiten in 3-25
- löschen 6-66, 6-143
- passives 3-28, 3-33
- teilen 3-26, 6-66, 6-143
- wechseln - CHWINDOW 3-26, 6-12, 6-66, 6-143
- Fenster-Modus 6-23, 6-79, 6-119
  - CHANGEWINDOW 4-20
  - HORIZONTAL 4-18
  - MOVE 4-19
  - NORMAL 4-18
  - VERTICAL 4-19
- Festplatte 3-2
- FILE-SCAN 3-162
- FILL, Flattersatz 4-69, 6-50
- FILTER-Programm aufrufen - FILTER 3-160, 4-99, 6-53
- FINDERHISTORY, Pseudo-Datei 4-49, 6-89, 6-98
- FINDERMODE 4-24
- FINDERPATTERN 3-124
- FONT 1-8, 6-55
- FORMAT, Blocksatz 6-57
- formatieren 4-63, 4-76
  - Blocksatz - FORMAT 6-57
  - Flattersatz - FILL 6-50
  - Rand - MARGIN 3-33, 6-85
  - Rand setzen - MARGIN 4-66, 6-85
  - rückgängig machen 6-136
  - WRAP 4-76
- Gelöschte Textbereiche sammeln 3-79, 3-88
- gespeicherte Dateiversion, neu aufsetzen auf 3-5
- Gleichheitszeichen 3-60
- GOTO 3-15, 6-60
- Grafikzeichen 4-79
- GRAPHIC 4-12
  - Hilfsdatei 4-79
- graphischer Zeichensatz 6-26
- Großschreibung - UPPER 6-142
- HELP, Pseudo-Datei 1-9, 4-40, 6-63
  - Hilfsdatei 3-7
  - GRAPHIC 4-79
- Hilfsinformationen - HELP 1-9, 4-40, 6-63
- Hilfsprogramm
  - maxlpr 4-101
  - mrecovery 5-7
- HOME 3-14, 3-20, 6-64
- HORIZONTAL, Fenster-Modus 4-18
- HWINDOW 3-26, 6-66
- IDENTIFIER 3-64
  - neuen definieren 3-65
  - vordefinierter 3-65
- INSERT 3-35
  - Einfüge-Modus 4-2, 4-4, 4-9f, 6-6
- INSERT-Modus 6-69
- INSERTMODE 4-2, 6-69
- JUSTIFY - Text vereinheitlichen 4-73, 6-71
- Kleinschreibung - LOWER 6-82
- Kommando
  - abbrechen - CANCEL 6-9
  - attributieren 3-36
  - eingeben 3-36
  - in Quelldateien 1-8
  - korrigieren 3-41
  - mit Standard-Puffern 3-153
  - SINIX- 4-26, 4-28, 4-53f, 5-2, 6-44
  - verstärken 3-63
- Kommando-Abarbeitungs-Modus 4-26
- Kommandoattribut 1-6
  - Eingabe annullieren 6-35
  - Eingabe wiederholen 6-35
  - eingeben - ENTER 3-37, 3-39, 6-34, 6-111
  - speichern 3-43
  - voreingestelltes 3-148

- wiederholen 3-159
- Kommandoattributierung 3-67
  - Sonderzeichen 3-45
- Kommandotaste 1-6
- Konfiguration
  - MAXed 6-49
  - Standard- 6-55, 7-13
- Konfigurationsdatei 5-5, 6-11, 6-49, 6-55
  - erstellen 7-13
  - neu erstellen 6-49
- konstante Bereichsdefinition 3-149
- Konversationsmodus 4-32
- kopieren
  - benannter ERASE-Puffer 6-115
  - benannter PICK-Puffer 6-104
  - Dateiinhalt 3-72
  - ERASE-Puffer 3-81
  - gesamte Datei 6-104
  - mehrfach 6-103, 6-114
  - PICK-Puffer 3-90
  - PICK-Puffer-Inhalt - PUT 6-103
  - Text - PICK 6-93
  - Textbereich 3-83, 3-103
- korrigieren, MAXed-Kommandos 3-41
- Kurzbeschreibung, MAXed 1-5
- LABEL** 6-72
- LABELS**, Pseudo-Datei 3-92f, 4-48, 6-60
- laden, Datei 3-2, 3-6, 3-21
- LANG** 4-32
- LEARN**, Lern-Modus 3-94, 4-35, 4-105, 6-74, 6-83
- Leerzeichen 4-71
  - einfügen - SPACE 6-129
  - löschen - COMPRESS 6-20
- Leerzeilen einfügen - SPACE 6-129
- LEFTARROW** 6-25, 6-79
- Lern-Modus **LEARN** 3-94, 4-35, 4-105, 6-74, 6-83
- letzte Sitzung rekonstruieren 3-5
- LEXICAL** Einfüge-Modus 4-3, 4-6, 4-10, 4-34, 6-6, 6-22, 6-50, 6-57, 6-69f
- LINE-Kommandos** 3-19
  - **LINEFEED** 3-19
  - **MINUSLINE** 3-19
  - **PLUSLINE** 3-19
  - **RVLINFEED** 3-19
- LINEFEED** 6-81
- Linien zeichnen - **DRAW** 4-88, 6-25
- Liste
  - aktuell definierte Makros 6-84
  - benannte ERASE-Puffer 6-115
  - benannte PICK-Puffer 6-105
  - geöffnete Dateien 6-30, 6-124
  - gesetzte Marken anzeigen 6-60
  - Hilfsinformationen 6-64
  - unter Verwendung von Sonderzeichen aufgerufene Dateien 6-30
  - verwendete Suchbegriffe 6-89, 6-98
- löschen
  - **.bak**-Dateien 6-48
  - benannte Marken 6-72
  - Bereich 6-39
  - Einzelzeichen 6-6
  - Fenstereinstellung 6-66, 6-143
  - Makro 4-36
  - rückgängig machen 6-136
  - Tabulatoren - **SRTAB** 6-131
  - Textbereich - **ERASE** 3-75, 6-38
  - überflüssige Leerzeichen - **COMPRESS** 6-20
  - Zeichen - **BACKSPACE** 6-6
  - Zeichen - **DELCHAR** 6-22
  - Zeichenattribut - **SETATTRIB** 6-125
  - Zeilenvorschub 6-38
- LOWER** 6-82
- lpr**, Betriebssystem-Kommando 6-102
- m** 3-6
- MACRO** 4-25, 6-83
- MACRO-Bibliothek** 3-97, 4-105, 6-15, 6-83, 6-123, 7-1
  - anzeigen 7-11

- 
- erstellen 7-2
  - öffnen 6-31
  - Standard- 5-5, 7-4
  - Textbaustein anlegen in 7-7
- MACROS, Pseudo-Datei 3-96, 4-36, 4-50, 6-84
- Makro 3-94
- aktuelles 3-95, 3-157, 6-74, 6-83
  - anlegen - LEARN 4-35, 6-74, 7-4
  - aufspalten 4-30
  - ausführen - MACRO 6-83
  - benannt 3-95, 6-74, 6-83
  - Parameter 3-46, 3-94
  - sichern 4-105, 7-2
  - Standard- 3-95, 3-158, 6-74, 6-83
  - Tasten 6-16
  - unbenannt 6-74, 6-83
  - verändern 7-11
- Makro-Abarbeitungs-Modus 4-25
- Makroname als IDEENTIFIER 3-94
- Makros der Version 2 A-4
- MARGIN 6-85
- Marke 6-72
- benannt 3-91, 6-72
  - setzen 3-91
  - Standard- 3-91, 3-156
  - unbenannt 3-91
- Markenname als IDENTIFIER 3-91
- markieren, Textstelle - LABEL 6-72
- maxed 5-1
- MAXed
- aufrufen 3-1, 5-1f
  - aufrufen mit Schalter 5-3
  - beenden - QUIT 3-5, 3-8, 6-108
  - Bildschirm 3-29, 3-34
  - Bildschirm neu schreiben - REFRESH 6-109
  - Dienstprogramme 7-1
  - Kommandos in Quelldateien 1-8
  - Konfiguration 6-49
  - Kurzbeschreibung 1-5
  - Sitzung rekonstruieren 5-7
  - Sitzung schrittweise zurückspielen 3-12
- Status abfragen - m\_getstatus 7-26
  - Steuerprogramm 3-98, 6-83f, 7-1, 7-21
  - Steuerprogramm aufrufen 7-29
  - Steuerprogramm Beispiel 7-31
  - Steuerprogramm übersetzen 7-29
  - Steuerzeichen 7-22
  - Sub-Shells 4-91
  - Variablen ersetzen 3-53
  - verlassen - EXIT 6-48
  - verlassen ohne Sichern 6-48
  - verlassen vorübergehend - CLI 6-13
- MAXed-Steuerprogramm 4-107
- maxlpr 4-101
- mconfig 7-14
- mdefine 7-2
- Metasyntax 1-6
- MINUSBLOCK 6-88
- MINUSFIND 6-87
- MINUSLINE 6-91
- MINUSPAGE 6-92
- mllist 7-11
- MOVE Fenster-Modus 4-19
- m\_control 7-22
- m\_getkey 7-24
- m\_getline 7-25
- m\_getparam 7-27
- m\_getstatus 7-26
- n** Verstärkungsfaktor 3-38, 3-63
- nationale Sonderzeichen 1-8
- NOECHO Ausgabe-Modus 4-29, 6-45
- NORMAL
- Fenster-Modus 4-18
  - Zeilenschaltungs-Modus 4-14, 6-117
- On-Line-Dokumentation** 1-9, 4-40, 6-63
- OPENFILES, Pseudo-Datei 3-72, 4-42, 6-30, 6-124

- 
- Operatoren 3-132
  - optionale Angaben 1-7
  - optionales Argument 5-2
  - Original-Datei 3-2
  - OVERWRITE 3-35
    - Ausgabe-Modus 4-28
    - Überschreibe-Modus 4-2, 4-6, 6-69
  - PAGE-Kommandos 3-19
    - MINUSPAGE 3-19
    - PLUSPAGE 3-19
  - Parameter
    - für Textersetzung 3-46
    - Makro- 3-46
  - Parametereingabe bei MACRO 6-75, 6-84
  - parametrisierte Makros 3-94
  - passives Fenster 3-28, 3-33
  - PATTERN 3-115, 3-124, 3-127, 3-131, 3-138, 3-162, 4-49, 4-55, 6-96ff
    - als Bereichsargument 3-124
  - Pfadname
    - absoluter 6-21
    - relativer 6-21
  - PICK 6-93, 6-103
  - PICK-Puffer 3-56, 4-103, 6-18, 6-93
    - benannt 3-83, 3-87, 4-47, 6-93
    - definiert 3-88
    - kopieren 3-90
    - mehrmals in den Text kopieren 6-104
    - Standard- 3-84, 3-155, 6-18, 6-93, 6-103, 6-106
    - unbenannt 3-84
  - PICK-Puffername
    - als IDENTIFIER 3-83
    - ersetzen 3-56
  - PICKBUFFERS, Pseudo-Datei 3-86, 4-47, 6-104
  - Platzhalter 1-6, 3-131
  - PLUSBLOCK 6-97
  - PLUSFIND 6-96
  - PLUSLINE 6-100
  - PLUSPAGE 6-101
  - Position, Schreibmarke 3-35
  - PRINT 6-102
  - Programm, asynchron 6-146
  - Protokolldatei 5-7
  - Pseudo-Datei 4-37
    - EDITFILES 3-69, 4-43, 6-29f
    - ERASEBUFFERS 3-78, 4-46, 6-115
    - FINDHISTORY 4-49, 6-89, 6-98
    - HELP 4-40, 6-63
    - LABELS 3-92f, 4-48, 6-60
    - MACROS 3-96, 4-36, 4-50, 6-84
    - OPENFILES 3-72, 4-42, 6-30, 6-124
    - PICKBUFFERS 3-86, 4-47, 6-104
    - PUTFILES 3-73, 3-87, 4-45, 6-104
    - RESTORE 3-79, 3-88, 4-41, 6-2f, 6-30
    - SCANLIST 4-55, 6-61
    - SETUP 3-124, 3-138, 4-11, 4-61
    - SYSCOMMAND 4-54, 4-96, 6-45
    - SYSHISTORY 4-53, 4-96, 6-31
    - SYSOUTPUT 4-52, 4-95, 6-31, 6-44f
    - TAGLIST 4-58, 6-62
    - VARIABLES 4-60, 6-36
  - Puffer 3-64
    - ERASE- 3-78, 6-38, 6-114
    - PICK- 3-56, 6-18, 6-93, 6-105
  - PUT 6-93, 6-103
  - PUTFILES, Pseudo-Datei 3-73, 3-87, 4-45, 6-104
  - Quelldatei 1-8, 6-15, 6-31, 6-49, 7-4f, 7-9, 7-13
    - MAXed-Kommandos in 1-8
  - Quellprogramm, C- 6-83
  - QUIT 3-8, 6-108
  - RANGE-Bereich 3-100, 3-104
    - definieren durch Bewegen der Schreibmarke 3-105

---

- definieren durch Zeichenketten-Attribut 3-114

Rechteck, zeichnen 6-26

RECORD-Bereich 3-101, 6-88, 6-97

- definieren 3-122

RECTANGLE-Bereich 3-101, 3-104

- definieren durch Bewegungen der Schreibmarke 3-106, 3-109

- definieren durch Zeichenkettenattribut 3-116

Referenzkarten 1-4

REFRESH 6-109

regulärer Ausdruck 3-68, 3-86, 3-127, 3-132, 3-135

rekonstruieren, letzte Sitzung 3-5, 5-7

relativer Pfadname 6-21

REPLACE 3-38, 3-138, 6-110

REPLACESTRING 3-138

RESTOR 6-114

RESTORE, Pseudo-Datei 3-79, 3-88, 4-41, 6-2f, 6-30

RETMODE 4-16

RETURN 3-13, 6-116

RIGHTARROW 6-25, 6-119

RVLINEFEED 6-121

Sammeln, gelöschte Textbereiche 3-79, 3-88

SAVE 3-10, 6-122

SCANLIST, Pseudo-Datei 4-55, 6-61

Schlüsselwort 1-8, 7-4

Schnittstelle, C 7-21

Schreibmarke 3-25, 3-30, 6-2, 6-6, 6-8, 6-12

Schreibmarke bewegen 3-12f

- absolut - GOTO 3-16, 6-60

- an Fensterrand - HOME 3-14

- an Zeilenanfang - RETURN 3-13

- an Zeilenende - UPEND 3-13

- an Zeilenanfang - STARTLINE 6-133

- ans Ende der vorigen Zeile - UPEND 6-141

- ans Zeilenende ENDLINE 6-33

- in die linke obere Fensterecke 6-64

- in nächste Zeile - RETURN 6-116

- nach links - LEFTARROW 6-79

- nach oben - UPARROW 6-139

- nach rechts - RIGHTARROW 6-119

- nach rechts - TAB 6-134

- nach unten - DOWNARROW 6-23

- spaltenweise - TAB 3-13

- um eine Zeile nach unten 6-81

- wortweise - WORD 3-13

- wortweise nach links 6-149

- wortweise nach rechts 6-150

- zeilenweise - ARROW 3-13

Seitenvorschub vorgeben 6-18

SETATTRIB 6-125

SETUP, Pseudo-Datei 3-124, 3-138, 4-11, 4-61

setzen

- Formatrand - MARGIN 4-66, 6-85

- Marke 3-91

- Tabulator - SRTAB 4-64, 6-131

- Zeichenattribut 4-84

- Zeichenattribut - ATTRIBUTE 6-3

- Zeichenattribut - SETATTRIB 6-125

Shell 5-2

Shell-Kommando

- abbrechen 6-9

- anzeigen 6-45

- aufrufen - EXECUTE 6-44

- Echo ausschalten 6-45

sichern

- Datei 3-3

- Datei - SAVE 3-10, 6-122

- Makro 6-123, 7-2

- Textbaustein 7-2

Signal SIGTERM 3-8

Silbentrenner 4-71f, 6-51

Silbentrennung 6-50, 6-57

SINIX-Kommando 3-57, 4-26, 4-28, 4-53f, 5-2, 6-44

---

- aktuelles 3-159
- asynchron 4-94
- Aufruf ersetzen 3-57
- aufrufen 4-90
- Echo einschalten 6-45
- für MAXed-Aufruf 5-1

SINIX-Programm 6-146

Sonderzeichen 3-68, 3-70, 3-78, 3-86, 3-92, 3-95, 3-112, 3-128, 3-130, 3-132, 3-136, 3-140

- entwerten 3-128, 3-140
- Kommandoattributierung 3-45
- nationale 1-8

SPACE, Zeilenschaltungs-Modus 4-15, 6-117, 6-129

Sprach-Modi 4-32

SRTAB 6-7, 6-131, 6-134

Standard-Ausgabe 4-28, 4-95

Standard-Einfüge-Modus INSERT 4-4

Standard-Eingabe 4-95

Standard-ERASE-Puffer 3-75, 3-154, 6-38, 6-41

Standard-Konfiguration 6-55, 7-13

Standard-MACRO 6-31

- Bibliothek 5-5, 6-123, 7-4
- Bibliothek erzeugen 6-31
- Bibliothek öffnen 6-31

Standard-Makro 3-95, 3-158, 6-74, 6-83

Standard-Marke 3-91, 3-156, 6-72

Standard-PICK-Puffer 3-84, 3-155, 6-18, 6-93, 6-103, 6-106

Standard-Puffer 3-153

Standard-Such-Modus 3-137, 4-21

Standard-Zeilenzlänge 6-11

STARTLINE 6-133

Status-Datei 3-6, 4-85, 5-5, 6-48, 6-72, 6-131

Steuerprogramm 7-1, 7-31

- als IDENTIFIER 3-98
- MAXed 3-98, 4-107, 6-83f, 7-21
- MAXed aufrufen 7-29
- MAXed übersetzen 7-29

Steuerprogramme der Version 2 A-4

Steuerzeichen

- CNTRL 6-15f
- eingeben - CODE 6-15
- MAXed 7-22

Steuerzeile 3-29, 3-34, 3-42, 6-34, 6-120

- ATTRIBUTE-Feld 4-81
- Eingabefeld 3-41f
- Eingabepuffer 3-41

STREAM Einfüge-Modus 4-3, 4-11

STREAM-Bereich 3-101, 3-104

- definieren durch Bewegen der Schreibmarke 3-107, 3-109
- definieren durch Zeichenketten-Attribut 3-118

STREAM-Modus 6-18, 6-69f

Sub-Shell 6-14

- CLI 4-92
- MAXed 4-91

Such-Modus 3-137

- < 4-23
- < > 4-24
- > 4-23
- \ 4-22
- Standard- 3-137, 4-21

suchen und ersetzen

- mit Abfrage 6-111
- von Zeichenfolgen 6-110

suchen

- Texteinheiten 3-125
- Textstelle - MINUSFIND 6-87
- Textstelle - PLUSFIND 6-96

Suchmuster 3-124, 6-87, 6-90, 6-96

SYSCOMMAND, Pseudo-Datei 4-54, 4-96, 6-45

SYSHISTORY, Pseudo-Datei 4-53, 4-96, 6-31

SYSOUTPUT, Pseudo-Datei 4-52, 4-95, 6-31, 6-44f

TAB 6-134

TAB-Kommando 3-13

- BACKTAB 3-13
- TAB 3-13

- 
- Tabulator 3-33, 4-64, 6-7
    - löschen - SRTAB 6-131
    - setzen - SRTAB 4-64, 6-131
  - Tag-Datei 4-58, A-2
  - TAGLIST, Pseudo-Datei 4-58, 6-62
  - Tastatur blockiert 3-8
  - Tastaturschablone 1-4
  - Tasten-Makro 6-16
  - Teilketten durch Parameter ersetzen 3-46
  - temporäre Datei 3-2, 3-5, 3-10
  - TERMINATE, Abarbeitungs-Modus 4-25f, 6-46
  - Text ausdrucken - PRINT 6-102
  - Text ausrichten
    - ALIGN 4-73, 6-2
    - links- und rechtsbündig - JUSTIFY 6-71
  - Text einrücken 6-2
  - Text in die nächste Zeile bringen 6-151
  - Text kopieren - PICK 6-93
  - Text vereinheitlichen - COMPRESS 4-73
  - TEXT Zeilenschaltungs-Modus 4-16, 6-117
  - Text zentrieren - CENTER 6-10
  - Text zwischenspeichern - PICK 6-93
  - Textanfang verschieben - ALIGN 6-2
  - Textbaustein
    - in MACRO-Bibliothek anlegen 7-7
    - sichern 7-2
  - Textbereich
    - austauschen 3-82
    - kopieren 3-83, 3-103
    - löschen - ERASE 3-75, 6-38
    - verschieben 3-74
    - verschieben - RESTOR 6-114
    - vertauschen - EXCHANGE 6-41
  - Textbereiche, gelöschte sammeln 3-79, 3-88
  - Textersetzung
    - im Eingabefeld 3-46, 3-58, 6-34
    - Parameter für 3-46
  - Textveränderungen rückgängig machen - UNDO 4-30, 6-136
  - Trennmodus 4-34
  - Trennvorschlag eingeben 6-50, 6-57
  - Überlagerung durch das Betriebssystem 6-13
  - Überschreibe-Modus OVERWRITE 3-35, 4-2, 4-6, 6-6, 6-69
  - überschreiben, Datei 3-10
  - übersetzen, MAXed-Steuerprogramm 7-29
  - Umgebungsvariablen 3-53
  - Umlaute darstellen 6-55
  - umschalten, Zeichensatz - FONT 6-55
  - Umwandlung in
    - Großbuchstaben - UPPER 6-142
    - Kleinbuchstaben - LOWER 6-82
  - unbenannte Marke 3-91
  - unbenannter PICK-Puffer 3-84
  - unbenanntes Makro 6-83
  - UNDO Textveränderungen rückgängig machen 4-30
  - unterstreichen 6-3, 6-125
  - UPARROW 6-25, 6-139
  - UPEND 3-13, 6-141
  - UPPER 6-142
  - Variablen 3-53ff
  - VARIABLES, Pseudo-Datei 4-60, 6-36
  - vergleichen, Textstellen - COMPARE 6-18
  - verlassen
    - MAXed - EXIT 6-48
    - MAXed vorübergehend - CLI 6-13
  - verschieben
    - Dateiausschnitt zeilenweise nach unten - LINEFEED 6-81
    - Dateiausschnitt zeilenweise nach oben - MINUSLINE 6-91
    - Dateiausschnitt seitenweise nach oben - MINUSPAGE 6-92
    - Dateiausschnitt zeilenweise nach

---

unten - PLUSLINE 6-100

- Dateiausschnitt seitenweise nach unten - PLUSPAGE 6-101
- Dateiausschnitt zeilenweise nach oben - RVLINFEED 6-121
- Dateiausschnitt nach links - WINL 6-147
- Dateiausschnitt nach rechts - WINLR 6-148
- Textanfang - ALIGN 6-2
- Textbereich - RESTOR 6-114

verstärken, MAXed-Kommando 3-63

Verstärkungsfaktor n 3-38, 3-63

vertauschen, Textbereiche - EXCHANGE 6-41

VERTICAL, Fenster-Modus 4-19

vordefinierter IDENTIFIER 3-65

Voreinstellung

- für Zeichenattribute 6-3
- für Zeichenattribute ändern 6-126

VWINDOW 3-26, 6-143

**Wählen, Eingabemodus - INSERT-MODE 6-69**

WAIT 6-146

wechseln

- Dateiverzeichnis - DEFDIR 6-21
- Fenster - CHWINDOW 3-26, 6-12, 6-66, 6-143
- zwischen Dateien 3-21

wiederholen, Kommandoattribute 3-159

WIN-Kommando 3-20

- WINL 3-20, 6-147
- WINR 3-20, 6-148

WINMODE 4-20

WORD, Einfüge-Modus 4-3, 4-8, 6-20, 6-69

WORD-Kommando 3-13

- WORDLEFT 3-13, 6-149
- WORDRIGHT 3-13, 6-150

WRAP - formatieren 4-76, 6-151

**Zähler 3-142**

Zeichen löschen

- BACKSPACE 6-6
- DELCHAR 6-22

Zeichenattribut 4-79, 6-3f, 6-18, 6-125

- aktivieren 4-81
- aktuelle Einstellung aktivieren 6-125
- einstellen 4-81, 6-3f, 6-18
- entfernen 4-87
- hinzufügen 4-87
- löschen - SETATTRIB 6-125
- setzen 4-84
- setzen - ATTRIBUTE 6-3
- setzen - SETATTRIB 6-125
- setzen/rücksetzen 6-3
- Voreinstellung 6-3
- Voreinstellung ändern 6-126

Zeichenkette

- auswerten 3-58
- ersetzen 3-136, 3-145
- kopieren 6-106

Zeichenketten-Argument 3-64

- ersetzen - m\_getparam 7-27

Zeichenkettenattribut 6-34

- Bereich definieren durch 3-112
- eingeben 3-38
- korrigieren 6-34

Zeichensatz 6-26

- 8859/1 6-55
- graphisch 6-26
- umschalten - FONT 6-55

zeichnen

- Linien - DRAW 6-25
- Rechteck 6-26

Zeichnen-Modus 4-12, 6-25f

Zeile

- aktuelle anfordern - m\_getline 7-25
- anzeigen 6-60

Zeilenlänge 6-11

- Standard 6-11
- umschalten 6-11

---

Zeilenschaltung

- mit Leerzeile 4-15
- RETURN 6-116

Zeilenschaltungs-Modus

- EMPTY 4-15, 6-117
- NORMAL 4-14, 6-116f
- SPACE 4-15, 6-117
- TEXT 4-16, 6-117

Zeilenumbruch, automatisch 4-68

zentrieren 4-73, 6-10

Zugriffsrechte 3-2

zurücksetzen, Datei auf früheren Versionsstand 3-11

Zurücksetzen-Modus 6-137

zurückspielen, MAXed-Sitzung 3-12

zwischenspeichern, Text - PICK 6-93















# Kritik – Anregungen – Korrekturen

Schreiben Sie uns bitte, was Ihnen an diesem Buch gefällt und was Ihnen nicht gefällt:

- vor allem, wo Sie Fehler entdeckt haben,
- wo etwas unklar beschrieben ist.

Absender:

\_\_\_\_\_  
Name

\_\_\_\_\_  
Firma/Dienststelle

\_\_\_\_\_  
Straße, Postfach

\_\_\_\_\_  
PLZ            Ort

\_\_\_\_\_  
Telefon        (            )

\_\_\_\_\_  
betreut von ZN

\_\_\_\_\_  
von Hrn./Fr.

Antwort

Bitte mit  
60 Pf.  
frankieren,  
falls Marke  
zur Hand.

Siemens AG  
DI ST QM2  
Manualredaktion  
Otto-Hahn-Ring 6  
Postfach 830951

D-8000 München 83

Absender:

\_\_\_\_\_  
Name

\_\_\_\_\_  
Firma/Dienststelle

\_\_\_\_\_  
Straße, Postfach

\_\_\_\_\_  
PLZ            Ort

\_\_\_\_\_  
Telefon        (            )

\_\_\_\_\_  
betreut von ZN

\_\_\_\_\_  
von Hrn./Fr.

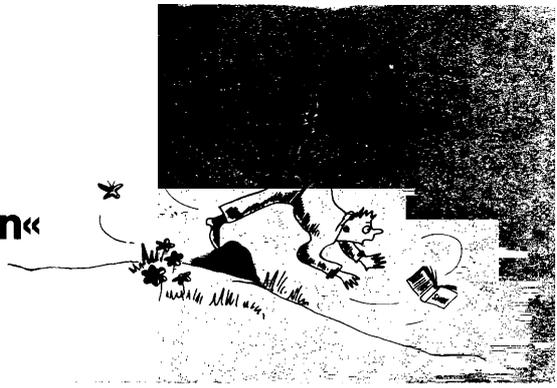
Antwort

Bitte mit  
60 Pf.  
frankieren,  
falls Marke  
zur Hand.

Siemens AG  
DI ST QM2  
Manualredaktion  
Otto-Hahn-Ring 6  
Postfach 830951

D-8000 München 83

# Nennen Sie uns Ihren »Stolperstein«



Buch:

Buch: SINIX MAXed V3.0, U2635-J-Z95-4

Seite

Mein »Stolperstein«:

Datum:

Das Handbuch benutze ich:  sehr häufig  gelegentlich zum Nachschlagen  
 als Programmierer  Systemverwalter  
 Sachbearbeiter  \_\_\_\_\_

Buch:

Buch: SINIX MAXed V3.0, U2635-J-Z95-4

Seite:

Mein »Stolperstein«:

Datum:

Das Handbuch benutze ich:  sehr häufig  gelegentlich zum Nachschlagen  
 als Programmierer  Systemverwalter  
 Sachbearbeiter  \_\_\_\_\_